

LERNEN EINFACH GEMACHT



2. Auflage

Scrum

für
dummies[®]



Die Vorteile
von Scrum kennen

Scrum als Projektmanagement-
methode einsetzen

Mit Scrum auch im Alltag
greifbare Ergebnisse
erzielen

Mark C. Layton
David Morrow

Scrum für Dummies

Schummelseite

Der Fokus von Scrum liegt auf kontinuierlicher Verbesserung, Flexibilität beim Umfang, Input durch das Team und der Lieferung von Qualitätsprodukten. Scrum befolgt das Agile Manifest und die 12 Agilen Prinzipien, die sich auf die Personen, die Kommunikation, das Produkt und Flexibilität fokussieren.

SCRUM: EIN MANIFEST FÜR AGILE SOFTWAREENTWICKLUNG

Scrum ist ein Teamansatz für das Projektmanagement, der am Agilen Manifest ausgerichtet ist. Das Agile Manifest ist eine absichtlich knapp gehaltene Beschreibung der Grundwerte des agilen Projektmanagements.

»Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir die folgenden Werte zu schätzen gelernt:

Individuen und ihre Interaktionen stehen über Prozessen und Werkzeugen.

Funktionierende Software steht über einer umfassenden Dokumentation.

Zusammenarbeit mit dem Kunden steht über der Vertragsverhandlung.

Reagieren auf Veränderung steht über dem Befolgen eines Plans.

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.«

Agiles Manifest Copyright © 2001: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas. This declaration may be freely copied in any form, but only in its entirety through this notice.

Prinzipien, die Scrum und dem Agilen Manifest zugrunde liegen

Scrum ist ein Projektmanagementansatz, der an den Werten des Agilen Manifests und den 12 Agilen Prinzipien ausgerichtet ist. Die 12 Agilen Prinzipien bilden einen Satz von Richtlinien, die Projektteams bei der Implementierung agiler Projekte unterstützen.

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufriedenzustellen.
2. Heiße Anforderungsänderungen sind selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.

5. Baue Projekte rund um motivierte Individuen auf. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen, und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an ein und innerhalb eines Entwicklungsteams zu übermitteln, ist das persönliche Gespräch.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein konstantes Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf hohe technische Qualität und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt daraufhin sein Verhalten entsprechend an.

Scrum und der agile Wertschöpfungsplan

Scrum ist eine Implementierung des agilen Projektmanagements. Die Roadmap to Value, der Wertschöpfungsplan, ist eine allgemeine Darstellung eines agilen Projekts, der für Ihr Projekt als Leitfaden dienen kann. Die Roadmap to Value besteht aus den folgenden Phasen:

- ✓ In Phase 1 identifiziert der Product Owner die *Produktvision*.

Die Produktvision definiert, was Ihr Produkt ist, wie das Produkt Ihre Firmen- oder Organisationsstrategie

unterstützt und wer das Produkt verwendet. Überprüfen Sie bei länger dauernden Projekten die Produktvision mindestens einmal jährlich.

- ✓ In Phase 2 erstellt der Product Owner zusammen mit den Business-Stakeholdern eine *Produkt-Roadmap*.

Bei der Produkt-Roadmap handelt es sich um eine übergeordnete Darstellung der Produktanforderungen und einen allgemeinen Zeitrahmen dafür, wann Sie diese Anforderungen entwickeln werden. Das Ermitteln der Anforderungen, deren Priorisierung und die grobe Aufwandsschätzung für diese Anforderungen macht einen großen Teil der Arbeiten bei der Erstellung der Roadmap aus. Überprüfen Sie bei länger dauernden Projekten die Produkt-Roadmap mindestens halbjährlich.

- ✓ In Phase 3 erstellt der Product Owner einen Release-Plan. Der *Release-Plan* ist ein allgemeiner Zeitplan für die Veröffentlichung nutzbarer Funktionalität. Ein agiles Projekt besitzt viele Releases, wobei die Features mit der höchsten Priorität zuerst eingeführt werden. Üblicherweise besteht ein Release aus drei bis fünf Sprints. Erstellen Sie einen Release-Plan am Anfang eines Release.

- ✓ In Phase 4 planen der Product Owner, das Entwicklungsteam und der Scrum Master die Sprints, auch Iterationen genannt, und beginnen in diesen Sprints mit dem Erstellen der Produktfunktionalität.

Zu Beginn jedes Sprints finden Sprint-Planungs-Meetings statt. Dort legt das Scrum-Team fest, welche Anforderungen in der nächsten Iteration enthalten sein werden. Außerdem untergliedert das Entwicklungsteam die Anforderungen in die Aufgaben, die für die Erstellung der Funktionalität erforderlich sind,

- ✓ In Phase 5 trifft sich das Entwicklungsteam während jedes Sprints zu den *Daily-Scrum-Meetings*.

Im Daily Scrum treffen Sie sich für maximal 15 Minuten und diskutieren, was Sie gestern erledigt haben, woran Sie

heute arbeiten werden und auf welche Hindernisse Sie gestoßen sind.

- ✓ In Phase 6 führt das Team ein *Sprint-Review-Meeting* durch.

Im Sprint Review, das am Ende jedes Sprints stattfindet, demonstrieren Sie den Produkt-Stakeholdern die einsatzfähige Produktfunktionalität.

- ✓ In Phase 7 hält das Team eine *Sprint-Retrospektive* ab.

Die Sprint-Retrospektive ist ein Meeting, in dem das Team den beendeten Sprint im Hinblick auf die Prozesse und die Arbeitsumgebung diskutiert und Pläne für Verbesserungen im nächsten Sprint macht. Wie das Sprint Review-Meeting, in dem das Produkt überprüft und angepasst wird, findet die Sprint-Retroperspektive am Ende jedes Sprints statt.

Scrum-Rollen

Scrum mobilisiert das Projektteam für einen bestimmten Satz an Funktionalität, die die Organisation im Markt veröffentlichen will. Scrum-Teams bestehen aus Personen, die diese drei Projektmanagement-Rollen innehaben:

- ✓ **Entwicklungsteam:** Die Leute, die das Produkt entwickeln. Programmierer, Tester, Designer, Autoren und alle anderen, die bei der Produktentwicklung eine praktische Rolle innehaben, sind Mitglied des Entwicklungsteams.
- ✓ **Product Owner:** Die Person, die dafür verantwortlich ist, Brücken zwischen dem Kunden, den Business-Stakeholdern und dem Entwicklungsteam zu schlagen. Der Product Owner ist ein Experte für das Produkt und für die Bedürfnisse und Prioritäten des Kunden. Der Product Owner arbeitet täglich mit dem Entwicklungsteam, um Anforderungen zu verdeutlichen.

Manchmal wird der Product Owner auch *Kundenvertreter* genannt.

- ✓ **Scrum Master:** Die Person, die für die Unterstützung des Entwicklungsteams verantwortlich ist. Er beseitigt organisatorische Hindernisse und sorgt dafür, dass die Prozesse die agilen Prinzipien befolgen.

Ein Scrum Master wird manchmal auch *Projektbetreuer* genannt.

Für das Gesamtteam des Projekts sollten Sie erwägen, auch die beiden folgenden Rollen zu besetzen:

- ✓ **Stakeholder:** All diejenigen, die ein Interesse an dem Projekt haben. Sie sind letztendlich nicht für die Durchführung des Projekts verantwortlich, jedoch liefern sie Input und sind vom Ergebnis des Projekts betroffen. Die Gruppe der Stakeholder ist divers und kann Personen aus verschiedenen Abteilungen und sogar aus verschiedenen Firmen umfassen.
- ✓ **Agiler Mentor:** Jemand, der Erfahrungen bei der Implementierung agiler Projekte besitzt und der diese Erfahrung mit einem Projektteam teilen kann. Der agile Mentor kann neuen Projektteams und Projektteams, die auf einem höheren Niveau arbeiten wollen, wertvolles Feedback und Ratschläge geben.

Scrum-Artefakte

Scrum-Teams verwenden drei Scrum-Artefakte und Liefergegenstände sowie drei weitere gängige agile Praktiken für das Projektmanagement der Produktentwicklung. Wenn Ihr Team seinen Plan implementiert, sollten Sie auf diese Artefakte und Praktiken achten:

- ✓ **Product Vision Statement:** Das Product Vision Statement ist ein Elevator Pitch oder eine kurze Zusammenfassung,

um zu kommunizieren, wie Ihr Produkt die Strategien Ihrer Firma oder Organisation unterstützt. Das Vision Statement muss die Ziele des Produkts verdeutlichen. Das Product Vision Statement ist eine gebräuchliche agile Vorgehensweise, jedoch kein Scrum-Artefakt.

- ✓ **Produkt-Roadmap:** Die Produkt-Roadmap ist ein allgemeiner Überblick der Produktanforderungen, die zum Erreichen der Produktvision erforderlich sind. Sie erlaubt es einem Projektteam, einen allgemeinen Zeitplan für die Entwicklung und die Bereitstellung dieser Anforderungen zu erstellen. Die Produkt-Roadmap ist ebenfalls eine gebräuchliche agile Vorgehensweise, jedoch kein Scrum-Artefakt.
- ✓ **Produkt-Backlog:** Die vollständige Liste des Projektumfangs, nach Prioritäten sortiert. Nachdem Sie die erste Anforderung definiert haben, besitzen Sie ein Produkt-Backlog.
- ✓ **Release-Plan:** Ein allgemeiner Zeitplan für das Release funktionierender Software. Der Release-Plan ist eine gebräuchliche agile Praktik; die Release-Planung ist außerdem ein inhärenter Bestandteil von Scrum.
- ✓ **Sprint-Backlog:** Das Ziel, die User Stories sowie die Aufgaben, die zum aktuellen Sprint gehören.
- ✓ **Inkrement:** Die einsetzbare Produktfunktionalität, die den Stakeholdern am Ende des Sprints demonstriert wird und die potenziell an die Kunden ausgeliefert werden kann.

Scrum-Aktivitäten

Scrum-Projekte enthalten für die Produktentwicklung fünf sich wiederholende Aktivitäten plus zwei gebräuchliche agile Praktiken. Diese Prozesse verbessern ab dem ersten Tag Ihres Projektes bis zu seinem letzten die Effizienz und Performance:

- ✓ **Projektplanung:** Die anfängliche Planung Ihres Projekts. Zur Projektplanung gehören das Erstellen eines Product Vision Statements und einer Produkt-Roadmap. Sie kann an einem Tag durchgeführt werden. Die Projektplanung ist eine gebräuchliche agile Praktik, jedoch keine Scrum-Aktivität.
- ✓ **Release-Planung:** Planen der nächsten Gruppe von Produktfeatures, die veröffentlicht werden sollen, und Bestimmen eines Produkteinführungsdatums, für das das Team mobilisiert werden kann. Bei agilen Projekten planen Sie immer nur ein Release gleichzeitig. Obwohl die Release-Planung in Scrum verwendet wird, handelt es sich hierbei um eine gebräuchliche agile Praktik, jedoch nicht um eine Scrum-Aktivität.
- ✓ **Sprint:** Ein kurzer Entwicklungszyklus, in dem das Team potenziell auslieferfähige Produktfunktionalität erstellt. Sprints, die auch *Iterationen* genannt werden, dauern in der Regel zwischen einer und vier Wochen. Sprints können auch eintägig sein, aber nicht länger als vier Wochen dauern. Die Sprintdauer kann sich in Verlauf des Projekts ändern, jedoch ändert sich hierdurch auch die Velocity.
- ✓ **Sprint-Planung:** Ein Meeting zu Beginn jedes Sprints, in dem sich das Scrum-Team auf ein Sprintziel verpflichtet. Außerdem werden die Anforderungen ermittelt, die dieses Ziel unterstützen und die Teil des Sprints werden, sowie die einzelnen Tasks, die für die Fertigstellung der jeweiligen Anforderungen erledigt werden müssen.
- ✓ **Daily Scrum:** Ein 15 Minuten dauerndes Koordinations- und Synchronisierungsmeeting, das täglich in einem Sprint abgehalten wird. Die Mitglieder des Entwicklungsteams geben an, was sie am Vortag fertiggestellt haben, was sie am aktuellen Tag fertigstellen werden und ob ihnen irgendwelche Hindernisse im Weg stehen.
- ✓ **Sprint Review:** Ein Meeting am Ende jedes Sprints, das durch den Product Owner eingeleitet wird und in dem das

Entwicklungsteam den Stakeholdern die im Sprint fertiggestellte, einsatzfähige Produktfunktionalität demonstriert.

- ✓ **Sprint-Retrospektive:** Ein Meeting am Ende jedes Sprints, in dem das Team seine Prozesse überprüft und anpasst, erörtert, was gut lief, was geändert werden muss, und in dem ein Plan für die Implementierung von Veränderungen im nächsten Sprint erstellt wird.

Scrum-Organisationen, Zertifizierungen und Ressourcen

Die Scrum-Community stellt mächtige und hilfreiche Onlinedienste bereit, mit denen Sie Ihre Fähigkeiten entdecken und weiterentwickeln können. Hier sind ein paar nützliche Links zu Mitgliedern der Scrum-Community, die beim Management Ihrer Projekte nützlich sein können:

- ✓ **Scrum Alliance** (www.scrumalliance.org): The Scrum Alliance ist ein gemeinnütziger Berufsverband, der das Verständnis und den Einsatz von Scrum fördert. The Scrum Alliance bietet eine Vielzahl professioneller Zertifizierungen an:
 - Certified Scrum Master (CSM)
 - Advanced Certified ScrumMaster (A-CSM)
 - Certified Scrum Product Owner (CSPO)
 - Advanced Certified Scrum Product Owner (A-CSPO)
 - Certified Scrum Developer (CSD)
 - Advanced Certified Scrum Developer (A-CSD)
 - Certified Scrum Professional (CSP) for ScrumMasters (CSP-SM), Product Owners (CSP-PO) and Developers (CSP-D)

- Certified Team Coach (CTC)
- Certified Enterprise Coach (CEC)
- Certified Agile Leadership (CAL)
- Certified Scrum Trainer (CST)

✓ [Scrumguides.org](http://www.scrumguides.org) (www.scrumguides.org): Dort finden Sie den Leitfaden *Der Scrum Guide: Der gültige Leitfaden für Scrum: Die Spielregeln* in mehr als 30 Sprachen. Der Scrum Guide ist sowohl online als auch im PDF-Format verfügbar und kann kostenlos verwendet werden.

✓ [Scrum.org](http://www.scrum.org) (www.scrum.org): [Scrum.org](http://www.scrum.org) bietet Tools und Ressourcen für Scrum-Praktiker an sowie Prüfungen und Zertifizierungen, wie beispielsweise

- Professional Scrum Master (PSM I, II, III)
- Professional Scrum Product Owner (PSPO I, II)
- Professional Scrum Developer (PSD)

✓ **ScrumPloP** (www.scrumplop.org): Pattern Languages of Programs (PloP) sind Methoden zur Beschreibung von Designpraktiken innerhalb von Fachgebieten. Zu diesen Methoden werden häufig Konferenzen mit dem Ziel organisiert, gemeinsam zu lernen. ScrumPloP veröffentlicht Muster von Scrum-Profis, die von verschiedenen Organisationen erfolgreich eingesetzt wurden, um mit Scrum zu beginnen.

✓ **Scaled Agile Framework (SAFe)**

(www.scaledagileframework.com): Das Scaled Agile Framework (SAFe) ist eine Wissensbasis für die Implementierung agiler Praktiken und ein Framework für die Implementierung von Scrum at Scale. SAFe ist ein eingetragenes Warenzeichen von Scaled Agile Inc.

✓ **Large-Scale Scrum (LeSS)** (<https://less.works>): Large-Scale Scrum (LeSS) ist eine Scrum-Scaling-Methode, die zwei Frameworks zur Verfügung stellt, bekannt als LeSS und LeSS Huge.

✓ **Platinum Edge** (platinumedge.com): Seit dem Jahre 2001 unterstützt Platinum Edge Firmen dabei, den organisatorischen Return on Investment (ROI) zu maximieren. Im Blog können Sie sich über den aktuellen Stand bei Praktiken, Werkzeugen und innovativen Lösungen informieren. Außerdem bietet Platinum Edge die folgenden Dienstleistungen an:

- *Agiles Auditing*: Untersuchung Ihrer aktuellen organisatorischen Strukturen und Prozesse, um eine agile Implementierungsstrategie zu entwickeln, die zu besseren Ergebnissen führt.
- *Personalbeschaffung*: Aufgrund des Kontakts zu den besten agilen und Scrum-Begabungen – weil sie bei Platinum Edge ausgebildet wurden – hilft Platinum Edge Ihnen dabei, die richtigen Leute zu finden, einschließlich Scrum Master, Scrum Product Owner und Scrum-Entwickler.
- *Ausbildung*: Maßgeschneiderte agile und Scrum-Schulungen und Zertifizierung:
Ausbildung zum Certified Scrum Master (CSM)
Ausbildung zum Certified Scrum Product Owner (CSPO)
Ausbildung zum Certified Scrum Developer (CSD)
SAFe Scaled Agile, Ausbildung und Implementierungen
- *Transition*: Nachdem Sie die talentierten Mitarbeiter eingestellt und ausgebildet haben, bietet Platinum Edge eingebettetes agiles Coaching und Mentoring an. So ist gewährleistet, dass im richtigen Leben die richtigen Praktiken eingesetzt werden.

✓ **Scrum-Development-Yahoo!-Gruppe**

(groups.yahoo.com/neo/groups/scrumdevelopment/info):

Die Scrum-Development-Yahoo!-Gruppe ist nach wie vor

eines der besten Scrum Message Boards im Internet, um mit der globalen Scrum-Community in Kontakt zu bleiben.

- ✓ **InfoQ** (www.infoq.com/scrum) : InfoQ ist eine unabhängige Online- Community mit einem herausragenden Bereich zu agilem Projektmanagement. Sie finden dort Nachrichten, Artikel, Video-Interviews, Videopräsentationen sowie Minibooks, die alle von Scrum-Experten verfasst sind.



Mark C. Layton

Scrum

für
dummies[®]

Übersetzung aus dem Amerikanischen von
Rainer G. Haselier
2. Auflage

WILEY

WILEY-VCH Verlag GmbH & Co. KGaA

Scrum für Dummies

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

2. Auflage 2019

© 2019 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

Original English language edition Scrum for Dummies © 2018 by John Wiley & Sons, Inc. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with John Wiley and Sons, Inc.

Copyright der englischsprachigen Originalausgabe Scrum for Dummies © 2018 by John Wiley & Sons, Inc. Alle Rechte vorbehalten inklusive des Rechtes auf Reproduktion im Ganzen oder in Teilen und in jeglicher Form. Diese Übersetzung wird mit Genehmigung von John Wiley & Sons, Inc. publiziert.

Wiley, the Wiley logo, Für Dummies, the Dummies Man logo, and related trademarks and trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries. Used by permission.

Wiley, die Bezeichnung »Für Dummies«, das Dummies-Mann-Logo und darauf bezogene Gestaltungen sind Marken oder eingetragene Marken von John Wiley & Sons, Inc., USA, Deutschland und in anderen Ländern.

Das vorliegende Werk wurde sorgfältig erarbeitet. Dennoch übernehmen Autoren und Verlag für die Richtigkeit von Angaben, Hinweisen und Ratschlägen sowie eventuelle Druckfehler keine Haftung.

Coverfoto: © MicroOne - stock.adobe.com

Korrektur: Isolde Kommer

Print ISBN: 978-3-527-71598-5

ePub ISBN: 978-3-527-82186-0

Über den Autor

Mark C. Layton, weltweit als Mister Agile bekannt, ist Organisationsstrategie, Zertifizierungstrainer der Scrum Alliance und hat mehr als 20 Jahre Erfahrung im Bereich Projektmanagement und Programmmanagement.

Er ist Vorsitzender der Abteilung Los Angeles des *Agile Leadership Network*, Autor der Bücher *Scrum for Dummies* und *Agiles Projektmanagement für Dummies* (beide bei Wiley erschienen), Urheber des Videokurses *Agile Foundations Complete Video Course* (Pearson Education) und Gründer von Platinum Edge, LLC – ein Unternehmen, das sich auf organisatorische Verbesserung spezialisiert hat und das Firmen bei der Transformation von Wasserfall- zu agilen Methoden unterstützt.

Vor der Gründung von Platinum Edge im Jahre 2001 erwarb Mark seine Expertise als Unternehmensberater, Programmmanagement-Coach und Projektleiter. Außerdem war er elf Jahre lang Kryptographie-Experte der US-Luftwaffe. Dort wurde er für seine Dienste mit der Commendation Medal und Achievement Medal ausgezeichnet.

Mark hat an der University of California, Los Angeles und der National University of Singapore ein MBA-Studium abgeschlossen. Sein Bachelor-Studium am Pitzer College/University of La Verne im Fachgebiet Verhaltenswissenschaften hat er mit summa cum laude beendet. Außerdem hat er die US Air Force Leadership School mit Auszeichnung abgeschlossen. Mark ist Certified Scrum Trainer (CST), zertifizierter Project Management Professional (PMP), Absolvent der Advanced Project Management Certification (SCPM) an der Stanford University sowie zertifizierter Scaled Agile Framework Program Consultant (SAFe SPC).

Neben der Veröffentlichung seiner Bücher und Videos ist Mark häufig als Redner bei wichtigen Konferenzen zu Lean, Scrum, XP

und anderen agilen Lösungen anzutreffen.

Weitere Informationen finden Sie unter platinumedge.com.

David Morrow ist Executive Agile Coach, der Leitungs- und Scrum-Teams dabei unterstützt, ihre Rollen bei der Sicherstellung des Erfolgs einer agilen Transformation zu erfüllen. David ist Certified Scrum Professional (CSP) und Certified Agile Coach (ICP-ACC) und verfügt über mehr als zwei Jahrzehnte Erfahrung im Aufbau und Coaching von Teams, damit diese leistungsfähiger und erfolgreicher werden. Er ist ein alter Hase im Bereich der Softwareentwicklung. Seine ökonomieorientierte Denkweise verleiht ihm eine einzigartige Perspektive auf die Teamdynamik.

David war 15 Jahre lang CEO von Devnext, bevor er Vollzeit-Coach bei Platinum Edge wurde. Jetzt spricht er mit jedem, der zuhören will, über Scrum und besucht lokale agile Benutzergruppen und Pubs.

Widmung

Für die Liebe meines Lebens. – Mark C. Layton

Für meine geliebte Frau und Familie. Ihr liebt mich so sehr. –
David Morrow

Danksagungen

Wir möchten noch einmal den zahlreichen Personen danken, die zur ersten Ausgabe dieses Buches beigetragen und geholfen haben, dass es Realität wird.

Wir sind auch denjenigen sehr dankbar, die geholfen haben, dass diese zweite Ausgabe ein noch wertvollerer Praxisleitfaden wird. Steve Ostermiller: Seine Erfahrung und Ratschläge waren für unseren Erfolg entscheidend. Dean Kynaston für seine Expertise und technischen Verbesserungen. Den Beratern aus verschiedenen Industriebereichen, deren Input die breiten Einsatzmöglichkeiten von Scrum illustriert: Amber Allen (Los

Angeles Unified School District) und Renee Jumper für den Bereich Bildung, Anna Kennedy (Autorin von *Business Development for Dummies*, erschienen bei Wiley & Sons, Inc.) für die Bereiche Geschäftsfeldentwicklung und Vertrieb, Brian Dreyer für den Themenbereich Videospiele und Geschäftsfeldentwicklung, Adi Ekowibowo, Farid Kazimi, Charles Park und Scot Kramarich für den Bereich Videospieldentwicklung, Kelly Anderson zum Thema Talentmanagement (Human Resources), Melani Chacon zum Gebiet Kundendienst, Hiren Vashi, Lisa White, Doc Dochtermann und Sunil Bhandari für die Einsatzmöglichkeiten im Gesundheitswesen; Lowell Feil, Rob Carstons und Steffanie Ducher für Enterprise Resource Planning (ERP), Brady Mortensen, Mogenns Gilmour, Joe Justice und J.J. Sutherland für den Bereich Hardwareentwicklung, Elana Glazer über Fertigung, Dean Leffingll, Alex Yakyma, Patrick Roach, Bas Vodde und Craig Larman zum Thema Enterprise-Skalierungsmodelle.

Unserer besonderer Dank geht an Jeff Sutherland und Ken Schwaber, die Mitbegründer von Scrum. Dank euch arbeiten wir alle besser.

Außerdem möchten wir dem fantastischen Team von Wiley & Sons danken: Charlotte Kughen, ohne deren Geduld und Weisheit dieses Buch nicht zustande gekommen wäre, sowie den vielen anderen, die ihre Zeit und Fachkenntnisse eingebracht haben, damit dieses Buch der Ratgeber wird, den wir uns erhofft hatten.

Inhaltsverzeichnis

[Cover](#)

[Titelseite](#)

[Impressum](#)

[Über den Autor](#)

[Widmung](#)

[Danksagungen](#)

[Einleitung](#)

[Über dieses Buch](#)

[Törichte Annahmen über die Leser](#)

[In diesem Buch verwendete Symbole](#)

[Über dieses Buch hinaus](#)

[Wie es weitergeht](#)

[Teil I: Erste Schritte mit Scrum](#)

[Kapitel 1: Die Grundlagen von Scrum](#)

[Die Grundlagen aus der Vogelperspektive](#)

[Das Feedback-Fest](#)

[Agile Wurzeln](#)

[Die fünf Scrum-Werte](#)

[Teil II: Ein Scrum-Projekt durchführen](#)

[Kapitel 2: Die ersten Schritte](#)

[Ihr Scrum auf die Beine stellen](#)

[Die Macht des Product Owners](#)

[Warum Product Owner Scrum lieben](#)

[Das Unternehmensziel und die Strategie: Phase 1](#)

Der Scrum Master

Gängige Rollen außerhalb von Scrum

Kapitel 3: Planen Sie Ihr Projekt

Die Produkt-Roadmap: Phase 2

Anforderungen unterteilen

Ihr Produkt-Backlog

Häufig verwendete Produkt-Backlog-Praktiken

Kapitel 4: Talent und Timing

Das Entwicklungsteam

Vorsprung durch Schätzung des Produkt-Backlogs

Ihre Definition of Done

Gebräuchliche Schätztechniken

Kapitel 5: Release- und Sprint-Planung

Release-Plan-Grundlagen: Phase 3

Sprint zu Ihren Zielen

Planung Ihrer Sprints: Phase 4

Ihr Sprint-Backlog

Kapitel 6: Das Beste aus Sprints herausholen

Kapitel 7: Überprüfen und anpassen: So korrigieren Sie Ihren Kurs

Bedürfnis nach Sicherheit

Die Feedbackschleife

Transparenz

Antimuster

Externe Kräfte

Kurskorrektur während des Fluges

Testen in der Feedbackschleife

Innovationskultur

Teil III: Scrum in der Industrie

Kapitel 8: Software-Entwicklung

Scrum und Softwareentwicklung: Füreinander geschaffen

Software-Flexibilität und Refactoring

Veränderung positiv sehen

Scrum-Anwendungen in der Software

Kapitel 9: Produktion materieller Güter

Der Sturz des Wasserfalls

Baugewerbe

Scrum im Wohnungsbau

Fertigung

Hardware-Entwicklung

Kapitel 10: Dienstleistungssektor

Gesundheitswesen und Scrum

Bildung und Scrum

Militär und Strafverfolgung

Kapitel 11: Medienlandschaft im Umbruch

Eine sich verändernde Landschaft im Verlagswesen

Prüfen, Anpassen und Refactoring

Scrum anwenden

Nachrichtenmedien und Scrum

Definieren, wann Inhalte als »done« gelten

Das Nachrichtenmedien-Scrum-Team

Sprint-Flexibilität

Teil IV: Scrum für betriebliche Funktionen

Kapitel 12: IT-Management und Operations

Big Data und Migration im großen Maßstab

Data-Warehouse-Projektmanagement

Unternehmensressourcenplanung

Das Dilemma zwischen Dienstleistung und Kontrolle

Die Babyboomer-Ruhestands-Lücke

Gewinn- und Verlustpotenzial

Innovation versus Stabilität

Kapitel 13: Portfoliomanagement

Herausforderungen im Portfoliomanagement

Lean Startup

Skalierung von Scrum für große Portfolios

Eine Übersicht über vertikalen Schnitt

Scrum of Scrums

Scrum at Scale

Scaled Agile Framework (SAFe)

Vorteile des SAFe-Modells

Large-Scale Scrum (LeSS)

Kapitel 14: Personal und Finanzen

Personal und Scrum

Die richtige Kultur schaffen

Finanzen

Kapitel 15: Business Development – Geschäftsfeldentwicklung

Scrum und Marketing

Scrum in Aktion für das Marketing

Scrum für den Vertrieb

Kapitel 16: Kundendienst

Kunden: Die wichtigsten Stakeholder

Scrum und Kundenservice

Scrum im Einsatz im Kundenservice

Teil V: Scrum im Alltag

Kapitel 17: Partnersuche und Familienleben

Partnersuche mit Scrum

Familien und Scrum

Kapitel 18: Scrum für Lebensziele

Der Weg in den Ruhestand

Fitness- und Gewichtsziele erreichen

Das Leben im Gleichgewicht halten

Reiseplanung

Schule und Studium

Teil VI: Der Top-Ten-Teil

Kapitel 19: Zehn Schritte für die Einführung von Scrum

Durchführung eines Audits

Talente identifizieren und rekrutieren

Sicherstellung der richtigen Ausbildung

Mobilisierung eines Übergangsteams

Pilotprojekt identifizieren

Maximieren der Effizienz des Umfelds

Single Points of Failure reduzieren

Definition of Done festlegen

Mit dem Pilotprojekt loslegen

Überprüfen, anpassen, sich entwickeln und skalieren

Kapitel 20: Zehn Klippen, die Sie umschiffen sollten

Unechtes Scrum

Mangelnde Ausbildung

Ineffektiver Product Owner

Fehlende automatisierte Tests

Fehlende Unterstützung in der Transitionsphase

Unangemessene Umgebung

Schlechte Teamauswahl

Laxe Disziplin

Fehlende Unterstützung für das Lernen

Verwässerte Prozesse

Kapitel 21: Zehn Hauptvorteile von Scrum

Bessere Qualität

Verkürzte Markteinführungszeit

Erhöhte Kapitalrendite

Größere Kundenzufriedenheit

Bessere Teammoral

Bessere Zusammenarbeit und mehr

Eigenverantwortung

Weitere relevante Metriken

Verbesserte Sichtbarkeit und Transparenz des Fortschritts

Verbesserte Projektsteuerung

Reduziertes Risiko

Kapitel 22: Zehn wichtige Kennzahlen für Scrum

Sprintziel-Erfolgsquoten

Defekte

Produkteinführungszeit – Time to Market

Return on Investment

[Kapitalumschichtung](#)

[Umfragen zur Zufriedenheit](#)

[Fluktuation im Team](#)

[Projektverschleiß](#)

[Vielseitigkeit der Qualifikationen](#)

[Verhältnis Manager-Produktersteller](#)

[Kapitel 23: Zehn wertvolle Ressourcen für Scrum](#)

[The Scrum Alliance](#)

[The Agile Alliance](#)

[Scrumguides.org](#)

[Scrum.org](#)

[Scruminc.com \(Scrum at Scale\)](#)

[ScrumPLoP](#)

[Scaled Agile Framework \(SAFe\)](#)

[LeSS](#)

[InfoQ](#)

[Platinum Edge](#)

[Stichwortverzeichnis](#)

[End User License Agreement](#)

Illustrationsverzeichnis

Kapitel 1

[Abbildung 1.1: Die sieben Phasen der Roadmap to Value](#)

[Abbildung 1.2: Eine vereinfachte Übersicht der Ereignisse und Zyklen von Scrum](#)

[Abbildung 1.3: Im Zentrum eines Scrum-Teams befindet sich das Entwicklungsteam](#)

Kapitel 2

[Abbildung 2.1: Die Produktvision ist Phase 1 der Roadmap to Value](#)

Kapitel 3

[Abbildung 3.1: Die Produkt-Roadmap ist Phase 2 der Roadmap to Value.](#)

[Abbildung 3.2: Ein Beispiel für eine Produkt-Roadmap nach Quartalen](#)

[Abbildung 3.3: Unterteilungsstufen](#)

[Abbildung 3.5: Ein Format zum Erstellen von User Stories](#)

[Abbildung 3.6: Eine fertige Karteikarte mit einer User Story](#)

Kapitel 4

[Abbildung 4.1: Schätzpokerkarten zur Aufwandsschätzung einer Anforderung](#)

[Abbildung 4.2: Fist of Five ist ein effizienter Weg, um in vielen Situationen einen Konsens zu finden.](#)

Kapitel 5

[Abbildung 5.1: Phase 3 der Roadmap to Value ist der Release-Plan.](#)

[Abbildung 5.2: Ein typischer Release-Plan mit Releaseziel und -datum sowie einem optionalen Release-Sprint](#)

[Abbildung 5.4: Backlog-Prioritätsmatrix](#)

[Abbildung 5.5: Der einwöchige Sprintlebenszyklus](#)

[Abbildung 5.6: Sprint Burndown-Diagramm](#)

[Abbildung 5.7: Das Sprint Backlog ist ein wichtiges Scrum-Artefakt.](#)

[Abbildung 5.8: Priorisierung innerhalb eines Sprints](#)

Kapitel 6

Abbildung 6.1: Das Daily Scrum ist ein integraler Bestandteil des Sprints und Phase 5 der Roadmap to Value.

Abbildung 6.2: Ein Team-Taskboard

Abbildung 6.3: Der Sprint-Review ist ein Scrum-Event und Phase 6 der Roadmap to Value.

Abbildung 6.4: In einem typischen Scrum-Projekt gibt es mehrere Feedback-Ebenen.

Abbildung 6.5: Die Sprint-Retrospektive, die siebte und letzte Phase der Roadmap to Value

Kapitel 7

Abbildung 7.1: Kegel der Unsicherheit

Abbildung 7.2: Feedback-Prozess

Abbildung 7.3: Flugbahn

Kapitel 8

Abbildung 8.1: Der klassische Ablauf der Entwicklung von Videospielen

Abbildung 8.2: Der Spielentwicklungsprozess mit Scrum

Kapitel 12

Abbildung 12.1: DevOps wägt eine Geschäftsanforderung gegen Stabilität ab.

Abbildung 12.2: Eine gängige Praxis für die Trennung der Instandhaltungs- und Entwicklungsaufgaben in einem Scrum-Team

Abbildung 12.3: Die Teammitglieder werden in angemessenen Zeitabständen zwischen den Teams gewechselt, um die gegenseitige Befruchtung durch Wissen und Fähigkeiten zu gewährleisten.

Kapitel 13

Abbildung 13.1: Vergleich der finanziellen Kosten für Verspätungen als Folge von Thrashing statt des Einsatzes von Teams

Abbildung 13.2: Ein Enterprise-Scrum-Team mit einem Integrationsteam

Abbildung 13.3: Scrum of Scrums zur Koordination von Scrum-Teams

Abbildung 13.4: Scrum-of-Scrums-Modell in Scrum at Scale für fünf Teams

Abbildung 13.5: Scrum-of-Scrums-of-Scrums-Modell in Scrum at Scale für 25 Teams

Abbildung 13.6: Dritte Ebene des Scrum-of-Scrums-of-Scrums-Modells aus Agile at Scale mit einem Executive Action Team (EAT)

Abbildung 13.7: Meta-Scrum für Product Owner in Scrum at Scale

Abbildung 13.8: Dritte Ebene des Meta-Scrum-Modells aus Agile at Scale mit einem Executive Meta Scrum (EMS)

Abbildung 13.9: Ausrichtung der dritten Ebene des Scrum-of-Scrums-of-Scrums- und Meta-Scrum-Modells aus Agile at Scale

Abbildung 13.10: SAFe im Überblick

Abbildung 13.11: Das Basisframework von Less

Abbildung 13.12: Das LeSS-Huge-Framework

Kapitel 14

Abbildung 14.1: Das 360-Grad-Review zur Leistungsbeurteilung

Kapitel 15

Abbildung 15.1: Ein klassischer Verkaufstrichter, der den Verkaufsprozess von Anfang bis Ende darstellt.

Abbildung 15.2: Verkaufs-Taskboard mit Spalten, die die Verkaufstrichterstufen darstellen

Kapitel 16

Abbildung 16.1: Alistair Cockburns Kommunikationsmodi. Copyright © Humans and Technology, Inc.

Kapitel 19

Abbildung 19.1: Aufeinander abgestimmte Abläufe des agilen Transition-Teams und des Scrum-Teams für das Pilotprojekt

Abbildung 19.2: Modell für Veränderungen nach Satir

Einleitung

Willkommen bei *Scrum für Dummies*. Scrum ist ein agiles Projektmanagement-Framework, das nachweislich die Markteinführungszeit um 30 bis 40 Prozent verkürzt, die Produktqualität verbessert und zu höherer Kundenzufriedenheit führt, während gleichzeitig die Kosten um 30 bis 70 Prozent gesenkt werden. Scrum erzielt diese Ergebnisse durch die Integration von Talenten aus dem Business- und Entwicklungsbereich, verbesserte Kommunikationsmodelle, größere Sichtbarkeit der Performance, regelmäßiges Feedback von Kunden und Stakeholdern und eine Mentalität des Überprüfens und Anpassens, die auf Empirie basiert. Sie können auch sehr komplexe Projekt effizient mit Scrum verwalten, um Ihr Geschäftsergebnis zu verbessern.

Über dieses Buch

Das Ziel ist es, explizit zu demonstrieren, wie man Scrum für beliebige Projekte nutzen kann, nicht nur im Bereich der Softwareentwicklung. Dieses Buch ist als Handbuch für die Anwendung von Scrum in der Praxis gedacht. Obwohl es die Scrum-Grundlagen im Detail behandelt, sollen Sie mithilfe dieses Buchs auch die erstaunlichen Vorteile von Scrum erkennen und für sich nutzen lernen.

Das Konzept von Scrum ist einfach zu erklären; die Anwendung und Beherrschung von Scrum sind oft schwierig. Alte Gewohnheiten und organisatorische Denkweisen müssen verändert und neue Wege beschritten werden. Aus diesem Grund schildern wir auch Erfolgsgeschichten, damit Sie sehen können, wie Scrum in Ihre Situation passt.

Die drei Rollen, drei Artefakte und fünf Ereignisse, die die Grundlage von Scrum bilden und die wir gründlich erörtern, sind zum Verständnis von zentraler Bedeutung. Wir berücksichtigen auch gängige Praktiken, die wir und andere anwenden, sodass

Sie selbst entscheiden können, was für Ihr Projekt am besten geeignet ist.

Scrum ist keine technische Vorgehensweise. Tatsächlich ist sein Grundprinzip der gesunde Menschenverstand. In vielen Fällen haben wir diese Informationen in die Technikwelt verpackt und mit Hilfe von Fachbegriffen erklärt. Wo es sinnvoll ist, haben wir diese Begriffe definiert.

Außerdem beschreiben wir gängige Praktiken von Scrum-Experten aus der ganzen Welt. Man kann so viel von anderen lernen, die dieses Framework in einem scheinbar grenzenlosen Spektrum von Projekten nutzen.

Scrum fällt in die Projektmanagementkategorie, die *agiles Projektmanagement* genannt wird. Weder Scrum noch agile Praktiken sind ein Eigenname. Scrum ist ein Framework für die Organisation Ihrer Arbeit. Während agil ein Adjektiv ist, das eine Vielzahl von Praktiken beschreibt, die den Werten des Agilen Manifests und den 12 Agilen Prinzipien entsprechen. Scrum und agil sind nicht identisch oder austauschbar; dennoch werden sie in vielen Quellen, vor allem online, synonym verwendet. In diesem Buch werden Sie Terminologie aus beiden Bereichen finden, da es sich bei Scrum um eine häufig verwendete Untermenge agiler Praktiken handelt.

Törichte Annahmen über die Leser

Es gibt zahlreiche Bücher über Scrum, jedoch unterscheidet sich dieses im Hinblick auf seine praktische Anwendbarkeit. Jeder der Autoren hat mehr als ein Jahrzehnt lang Erfahrung mit agilen Methoden und Scrum, die wir Ihnen in einem praktischen Leitfaden vermitteln. Wir stellen keine Vermutungen darüber an, was Sie bereits wissen: Sie müssen kein Wissenschaftler oder Programmierer sein; alles, was Sie brauchen, sind ein Projekt und die Leidenschaft, es auf die bestmögliche Weise zu verwirklichen.

Wir zeigen Ihnen Beispiele, angefangen vom Bau von Düsenjägern bis hin zu einer Familie, die einen Urlaub organisiert. Wir konzentrieren uns auf die Schritte, die notwendig sind, damit die Magie von Scrum für Sie funktioniert.

Unsere Zielgruppe sind Programmierer, Vertriebsprofis, Produktentwickler, Führungskräfte und mittelständische Manager sowie Pädagogen, die nach einer Möglichkeit suchen, ihre Schüler und Studenten zu motivieren.

Wenn Sie in einer Technologiebranche tätig sind, haben Sie wahrscheinlich die Begriffe agil und/oder Scrum gehört. Vielleicht haben Sie sogar in einer Scrum-Umgebung gearbeitet, möchten aber Ihre Fähigkeiten und Ihren Wortschatz in diesem Bereich verbessern und andere in Ihrer Firma mitnehmen. Wenn Sie nicht in der Technologiebranche arbeiten, haben Sie vielleicht gehört, dass Scrum eine großartige Möglichkeit ist, Projekte durchzuführen (was wahr ist). Vielleicht ist Scrum neu für Sie und Sie suchen nach einem Weg, Ihr Projekt zugänglicher zu machen, oder Sie haben eine großartige Idee, die in Ihnen brennt, und wissen nicht, wie Sie sie verwirklichen können. Wer auch immer Sie sind, es gibt einen einfachen Weg, Ihr Projekt durchzuführen, und dieser Weg wird Scrum genannt. Auf den folgenden Seiten zeigen wir Ihnen, wie Sie Scrum nutzen können.

In diesem Buch verwendete Symbole

In diesem Buch werden die folgenden Symbole verwendet, um Absätze hervorzuheben, die für Sie von besonderem Interesse sein könnten:



Bei den Tipps handelt es sich um Ideen und Konzepte, auf die wir Sie besonders hinweisen wollen. Es handelt sich in den meisten Fällen um praktische Tipps, die Sie im Zusammenhang mit dem jeweiligen Thema anwenden können.



Das Warnsymbol wird weniger oft verwendet als die anderen Symbole in diesem Buch. Es soll Ihnen helfen, Zeit zu sparen; wir weisen Sie auf Stolperfallen hin, die Sie besser vermeiden sollten.



Wenn Sie kein großes Interesse an technischen Details haben, können Sie diese Absätze überspringen, ohne dass Ihnen dadurch viel entgeht. Falls Sie sich für technische Dinge interessieren, finden Sie in diesen Absätzen faszinierende Informationen.



Dieses Symbol markiert Absätze, auf die wir Sie besonders hinweisen wollen, etwa ein Konzept, eine Idee oder eine erprobte Vorgehensweise, die wir für besonders erwähnenswert halten.

Über dieses Buch hinaus

Obwohl dieses Buch das Spektrum des agilen Projektmanagements mit Scrum im Großen und Ganzen abdeckt, können wir nur so viel behandeln, wie auf eine bestimmte Seitenanzahl passt. Wenn Sie am Ende dieses Buches angekommen sind und denken: »Das war ein tolles Buch! Wo kann ich mehr darüber erfahren, wie ich meine Projekte mit Scrum voranbringen kann?«, dann sollten Sie [Kapitel 23](#) aufschlagen oder einen Blick auf die englischsprachige Dummies-Website unter www.dummies.com werfen.

Wie es weitergeht

Um die ersten Schritte mit Scrum zu gehen, können Sie es zunächst auf kleinere Projekte anwenden, um ein Gefühl dafür zu bekommen. Bald werden Sie Ihre wichtigsten Projekte auf die gleiche Weise abwickeln. Dieses Buch wendet sich an eine Vielzahl von Lesern und ist so organisiert, dass Sie spezifische Interessengebiete finden können, die für Sie relevant sind. Sie können jedes Kapitel zum Nachschlagen verwenden, wenn Sie eine technische Frage haben, oder ein Beispiel dafür suchen, wie Scrum im wirklichen Leben eingesetzt wird.

Wenn Scrum neu für Sie ist, beginnen Sie mit [Kapitel 1](#), um die grundlegenden Konzepte und Terminologie zu verstehen; arbeiten Sie sich dann durch bis [Kapitel 7](#), um das gesamte Framework kennenzulernen. Ab [Kapitel 8](#) sehen Sie, wie Sie Scrum in unterschiedlichsten Situation und Bereichen anwenden können.

Wenn Sie bereits mit Scrum vertraut sind und mehr darüber erfahren möchten, wie Scrum in einer Vielzahl von Branchen eingesetzt wird, lesen Sie die [Kapitel 8](#) bis [11](#).

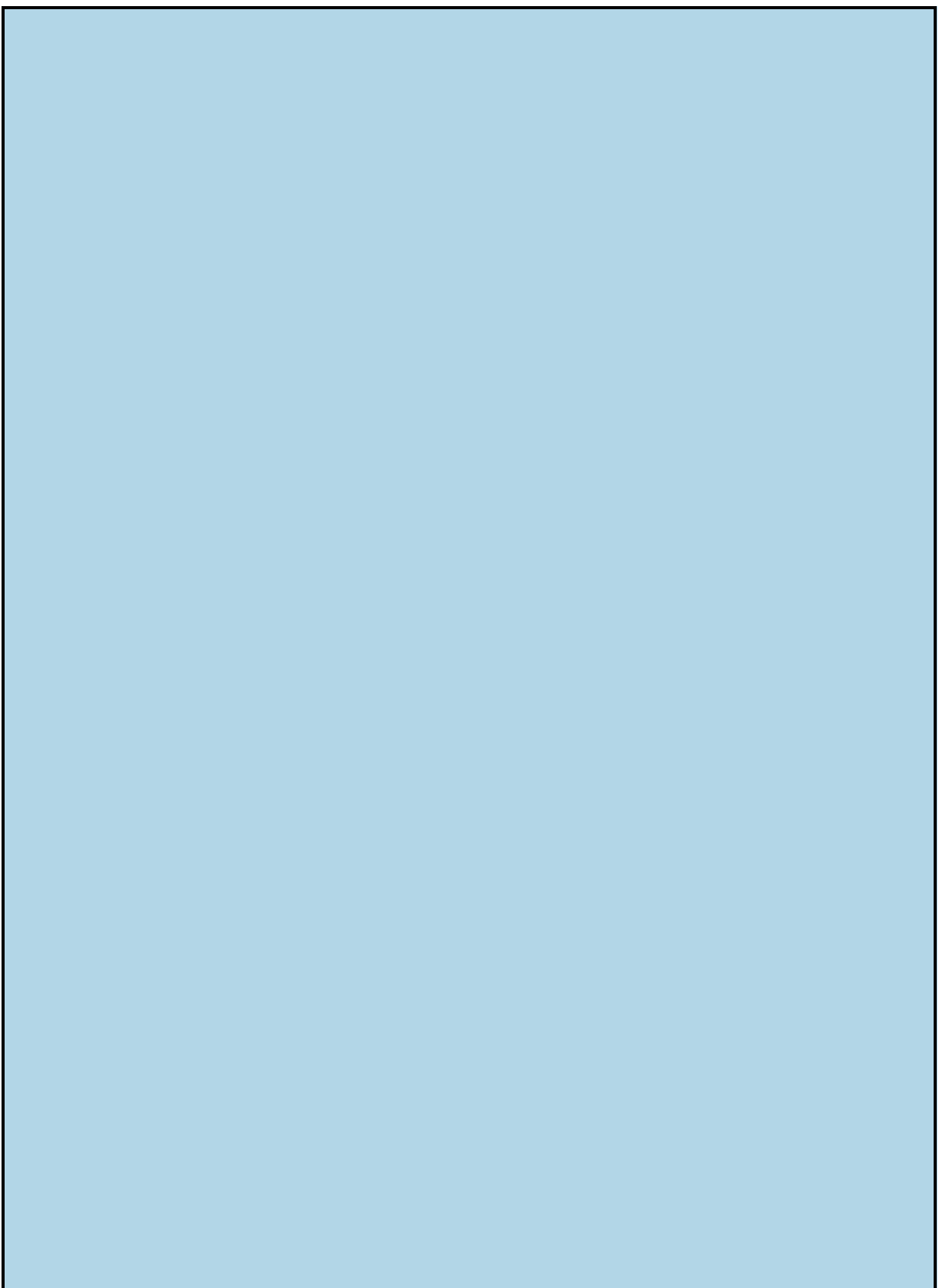
Falls Sie Product Owner, Scrum Master oder Business Leader sind und mehr über die Skalierung von Scrum erfahren möchten, lesen Sie zunächst [Kapitel 13](#) und den gesamten [Teil 6](#), in dem Sie wertvolle Ressourcen finden.

Wenn Sie mit Scrum vertraut sind und wissen wollen, wie es Ihnen helfen kann, Dinge des Alltags zu meistern, lesen Sie die [Kapitel 17](#) und [18](#); dort finden Sie Inspiration und Beispiele.

Teil I

Erste Schritte mit Scrum





IN DIESEM TEIL ...

Verknüpfen Sie Scrum und die Prinzipien des agilen Projektmanagements.

Nutzen Sie konstantes Feedback durch Transparenz und Quantifizierung, um die Erfolgsquote von Projekten zu erhöhen.

Werden Sie taktisch flexibel und erzeugen Sie so strategische Stabilität.

Kapitel 1

Die Grundlagen von Scrum

IN DIESEM KAPITEL

Wesentliche Scrum-Grundsätze erkennen

Nützliche Scrum-Werte und Strukturen identifizieren

Scrum ist ein empirisches Expositionsmodell, das bedeutet, dass Menschen, die das Scrum-Modell anwenden, Wissen aus der Praxis gewonnen haben und auf der Grundlage dieser Erfahrungen Entscheidungen treffen. Es bietet die Möglichkeit, Ihr Projekt zu organisieren – sei es die Veröffentlichung eines neuen Smartphones oder die Koordination der Geburtstagsparty Ihrer Tochter in der fünften Klasse. So können Sie herauszufinden, ob Ihr Ansatz zu den gewünschten Ergebnissen führt. Wenn Sie etwas erledigen müssen, stellt Scrum eine Struktur bereit, die zu größerer Effizienz und schnelleren Ergebnissen führt.

Bei Scrum regiert der gesunde Menschenverstand. Sie konzentrieren sich auf das, was heute getan werden kann, mit dem Ziel, die zukünftige Arbeit in überschaubare Teile aufzuteilen. Sie sehen sofort, wie gut Ihre Entwicklungsmethodik funktioniert. Wenn Sie in Ihrem Ansatz Ineffizienzen erkennen, können Sie mit Scrum darauf reagieren und zielgerichtet und schnell Anpassungen vornehmen.



Obwohl das empirische Modell der Betrachtung und Weiterentwicklung auf den Anbeginn der Zeit in der Kunst zurückgeht – in der Bildhauerei zum Beispiel meißelt man weg, überprüft die Ergebnisse, nimmt erforderliche Anpassungen vor und meißelt weiter –, stammt ihre heutige Verwendung aus der Computermodellierung. Das empirische Modell der Betrachtung und Weiterentwicklung bedeutet, die tatsächlichen Ergebnisse zu beobachten oder in Erfahrung zu bringen, anstatt sie mit Hilfe von Forschung oder mathematischer Formeln zu simulieren und dann auf der Grundlage dieser Erfahrungen Entscheidungen zu treffen. Bei Scrum zerlegen Sie Ihr Projekt in machbare Teile; anschließend beobachten Sie bei jedem Schritt Ihre Ergebnisse. So können Sie sofort die notwendigen Änderungen vornehmen, um Ihr Projekt auf dem bestmöglichen Weg zu halten.

Die Grundlagen aus der Vogelperspektive

Scrum ist keine Methodik, sondern eine neue Denkweise. Es handelt sich dabei nicht um einen »Malen nach Zahlen«-Ansatz, bei dem Sie am Ende ein Produkt erhalten. Es ist ein einfaches Rahmenwerk für die klare Definition von Rollen und die Organisation der zu erledigenden Arbeiten, damit Sie diese effektiver priorisieren und die ausgewählte Arbeit effizienter abschließen können. Frameworks sind weniger präskriptiv als Methoden und bieten ein angemessenes Maß an Flexibilität für Prozesse, Strukturen und Werkzeuge, die diese ergänzen. Bei der Verwendung dieses Ansatzes können Sie komplementäre Methoden und Praktiken anwenden und schnell feststellen, ob Sie echte, greifbare Fortschritte erzielen. Sie erstellen innerhalb von Wochen, Tagen oder (in einigen Fällen) Stunden getestete, verwertbare Ergebnisse.

Wie der Prozess, ein Haus Stein für Stein zu bauen, ist Scrum ein iterativer, inkrementeller Ansatz. Es gibt Ihnen einen frühen empirischen Nachweis von Leistung und Qualität. Die Rollen sind klar und selbstbestimmt, wobei Einzelpersonen und Teams die benötigte Freiheit und Werkzeuge erhalten, um ihre Arbeit zu erledigen. Umständliche Fortschrittsberichte, redundante Meetings und aufgeblähte Verwaltungsebenen gibt es nicht. Wenn Sie einfach nur die Arbeit erledigen wollen, ist Scrum der richtige Ansatz.



Scrum ist ein Begriff, der aus dem rauen Rugby-Spiel stammt. Huddles oder Scrums werden gebildet, indem die Stürmer des einen Teams ihre Arme ineinander verhaken, die Köpfe nach unten beugen. Ihnen gegenüber stehen die Stürmer der gegnerischen Mannschaft, welche die gleiche Körperhaltung einnehmen. Der Ball wird dann in die Mitte dieses geordneten Spielergedränges geworfen. Obwohl jedes Teammitglied eine bestimmte Position hat, spielen alle Teammitglieder sowohl angreifende als auch verteidigende Rollen und arbeiten zusammen, um den Ball über das Spielfeld zu bewegen. Wie beim Rugby ist Scrum darauf angewiesen, dass talentierte Menschen mit unterschiedlichen Verantwortlichkeiten und Aufgabenbereichen in Teams eng zusammenarbeiten, um ein gemeinsames Ziel zu erreichen.

Wir möchten ein wenig beachtetes Konzept von Scrum betonen, mit dem sich zwei Drittel dieses Buches befassen: die erstaunliche Vielseitigkeit von Scrum. Leute, die Scrum kennen, denken, dass es für Software, Informationstechnologie (IT) oder technische Anwendungen maßgeschneidert wurde, aber das ist nur die Spitze des Scrum-Bergs. Absolut jedes Projekt – egal ob groß, klein, technisch, künstlerisch, sozial oder persönlich – kann das Scrum-Framework produktiv nutzen. In den [Kapiteln 8](#) bis [18](#) zeigen wir Ihnen, wie. Seien Sie vorgewarnt! Scrum ist so ein süchtig machendes Framework, dass Sie es benutzen werden, um die Fußballmannschaft Ihres Kindes zu trainieren, Ihre

Nachbarschaftswache zu planen und sogar Ihre Übungsroutine zu verbessern.

Roadmap to Value – Der Wertschöpfungsplan

In diesem Buch erörtern wir Techniken, die von einigen erfahrenen Scrum-Praktikern als übliche Praxiserweiterungen für Scrum verwendet werden. Diese Techniken ergänzen das Scrum-Framework, sie ersetzen es aber nicht. Wir weisen auf Unterschiede hin, wenn sie auftreten. Alle gängigen Praktiken, die wir einbeziehen und empfehlen, sind erprobt und getestet – gleichzeitig sind wir uns darüber bewusst, dass diese Praktiken außerhalb des grundlegenden Scrum-Frameworks liegen. Es handelt sich dabei um Vorschläge, die Sie in Ihrer eigenen Situation in Erwägung ziehen und eventuell einsetzen können.

Wir nennen diese Ansammlung von Scrum und geprüften gängigen Praktiken die »Roadmap to value«, den Wertschöpfungsplan. Diese Roadmap besteht aus sieben Phasen, die Sie – angefangen bei der Visionsphase Ihres Projekts bis zur Aufgabenebene und wieder zurück – in einem kontinuierlichen Prozess der Überprüfung und Anpassung führen. Mit anderen Worten, die Phasen helfen Ihnen, zu erkennen, was Sie erreichen wollen. Sie müssen dann diese Vision in einen effizienten Zyklus unterteilen, mit welchem Sie jeden Tag, jede Woche und jeden Monat echte Ergebnisse erzielen.

Welche Idee, die Millionen von Euro wert ist, schlummert seit Jahren schon in Ihrem Hinterkopf? Folgen Sie den sieben Phasen. Sie zeigen Ihnen die Machbarkeit und Abwegigkeit Ihrer Idee. Aber auch wo Sie Verbesserungen vornehmen können – Schritt für Schritt, Stück für Stück.

[Abbildung 1.1](#) zeigt eine ganzheitliche Betrachtung der Roadmap to Value. In der Abbildung sehen Sie, dass Sie mit der Produktvision beginnen, die Planungsphase durchlaufen und dann in die zyklische Welt der Sprints, Reviews und Retrospektiven eintreten.

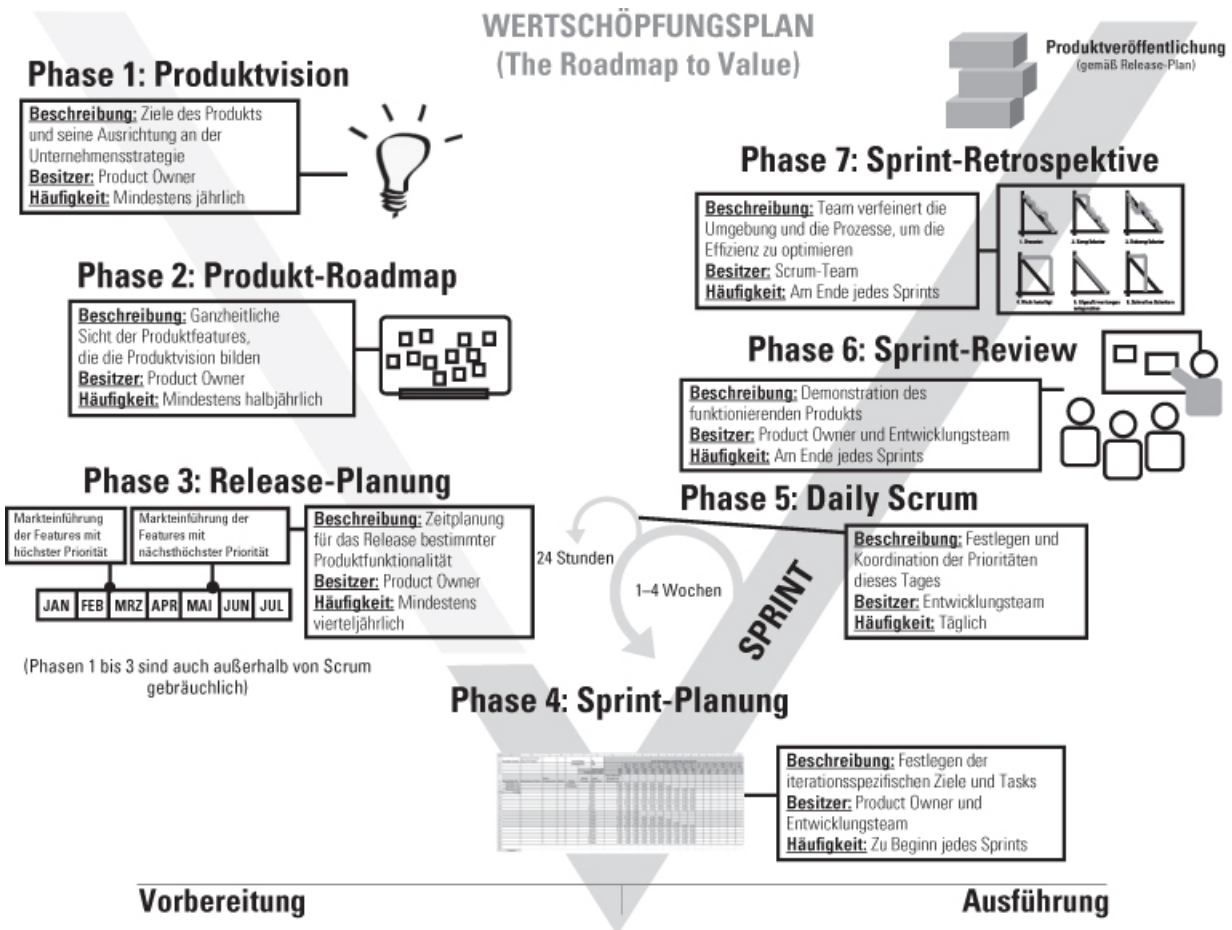


Abbildung 1.1: Die sieben Phasen der Roadmap to Value

Scrum im Überblick

Der Prozess von Scrum ist einfach und kreisförmig, wobei die Überprüfung und Anpassung konstante Elemente darstellen. Zunächst wird eine grob geordnete To-do-Liste – das so genannte Produkt-Backlog – erstellt und gepflegt. Dann werden die Elemente mit den höchsten Prioritäten für einen festen, regelmäßigen Zeitraum – den sogenannten Sprint – ausgewählt, innerhalb dessen das Scrum-Team einem vorgegebenen und gemeinsam vereinbarten Ziel nachstrebt.

Abbildung 1.2 zeigt eine Übersicht über den Scrum-Ansatz.

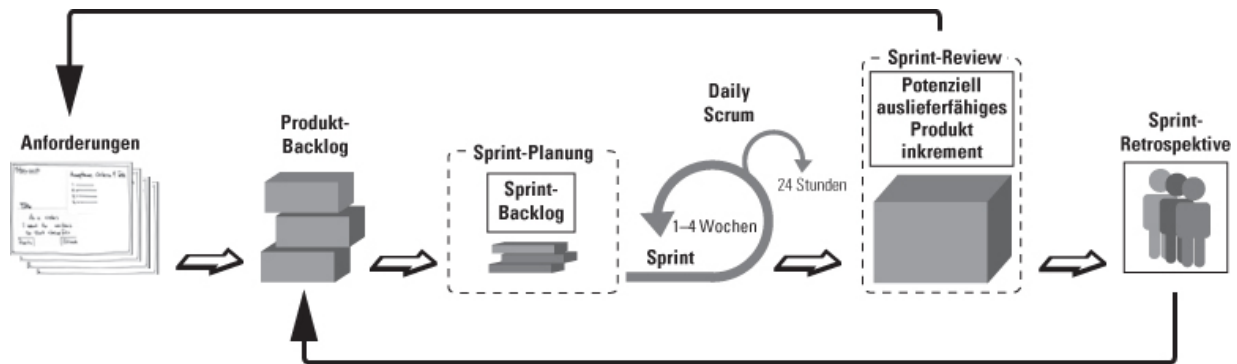


Abbildung 1.2: Eine vereinfachte Übersicht der Ereignisse und Zyklen von Scrum

Der Scrum-Prozess erlaubt es Ihnen, sich schnell an verändernde Marktkräfte, technologische Zwänge, Vorschriften, neue Innovationen und fast alles andere, das Sie sich vorstellen können, anzupassen. Der Schlüssel hierbei ist, dass fortlaufend die Elemente mit den höchsten Priorität bis zu deren Fertigstellung abgearbeitet werden. Alle Elemente mit der höchsten Priorität werden mittels der folgenden Schritte vollständig entwickelt und getestet:

- ✓ Ausarbeitung der Anforderung
- ✓ Entwurf
- ✓ Entwicklung
- ✓ Umfassende Tests
- ✓ Integration
- ✓ Dokumentation
- ✓ Abnahme



Um die einzelnen Anforderungen vollständig zu erstellen, werden für jedes Element die oben aufgeführten sieben Schritte ausgeführt. Jede Anforderung, die für einen Sprint übernommen wird, egal wie klein oder groß, wird vollständig entwickelt, getestet und dann genehmigt oder abgelehnt.

Wenn eine Anforderung akzeptiert und daher als »auslieferfähig« angesehen wird, wissen Sie, dass sie funktioniert. Hoffnung und

Rätselraten werden so aus der Gleichung herausgenommen und durch die Realität ersetzt. Sie erstellen Ihr Produkt Schritt für Schritt und präsentieren diese greifbaren Inkremente den Stakeholdern, um Rückmeldung zu erhalten. Dieses Feedback generiert neue Anforderungen, die in das Produkt-Backlog aufgenommen und die gegen bereits bekannte Anforderungen priorisiert werden.



Was ist wichtiger: Effizienz oder Effektivität? Effektivität ist zweifellos wichtiger. Machen Sie sich keine Sorgen um Effizienz, bis Sie herausgefunden haben, wie Sie effektiv sein können. Ein sehr effizientes Entwicklungsteam, das an den falschen Dingen arbeitet, vergeudet seine Zeit. Ein supereffektives Entwicklungsteam kann jedoch leicht Effizienz lernen. Arbeiten Sie immer zuerst an den *richtigen* Dingen. Wie der Wirtschaftswissenschaftler und Managementautor Peter F. Drucker sagte: »Es gibt nichts Unnützeres, als effizient das zu tun, was überhaupt nicht getan werden sollte«.

Der Scrum-Zyklus wird immer wieder durchlaufen. Neben dem konstanten Feedback-Fluss hilft Ihnen der Umstand, dass nur die Produkt-Backlog-Einträge mit der höchsten Priorität entwickelt werden, zu erkennen, was Ihre Kunden wirklich wollen und ihnen genau das schneller und mit höherer Qualität zu liefern.

Scrum-Teams

Unabhängig vom Umfang Ihres Scrum-Projekts hat Ihr Scrum-Team ähnliche Eigenschaften. Die Größe der Entwicklungsteams variiert etwas, aber die Rollen bleiben die gleichen. Die einzelnen Rollen werden im Laufe des Buchs ausführlich besprochen.

[Abbildung 1.3](#) stellt ein Scrum-Team grafisch dar.

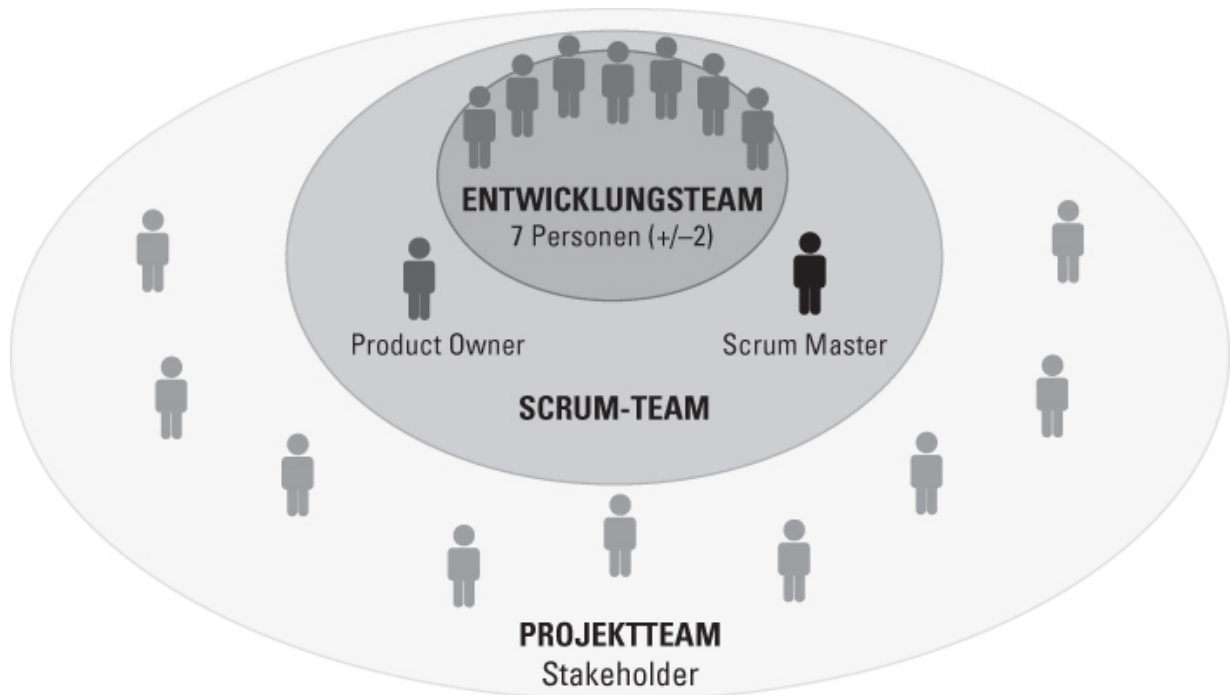


Abbildung 1.3: Im Zentrum eines Scrum-Teams befindet sich das Entwicklungsteam

Das Herz des Scrum-Teams bildet das Entwicklungsteam – die Leute, die zusammenarbeiten, um das Produkt selbst zu entwickeln. Sie arbeiten direkt mit einem Product Owner und Scrum Master zusammen. Diese stimmen die Geschäfts- und Entwicklungsprioritäten der Organisation aufeinander ab und beseitigen Ablenkungen. So kann sich das Entwicklungsteam auf die Entwicklung eines qualitativ hochwertigen Produkts konzentrieren.

Stakeholder sind zwar keine Scrum-Rollen, jedoch nehmen wir sie in [Abbildung 1.3](#) auf, weil sie Ihr Projekt beeinflussen. Es kann sich dabei um interne oder externe Stakeholder handeln.

Teammitglieder des Marketings, der Rechts- oder Complianceabteilung, aber auch insbesondere Kunden sind Beispiele für Stakeholder.

Das Scrum-Team selbst hat die komplette Verantwortung. Die Teammitglieder finden heraus, wie sie ihre Ziele in dem Umfeld, in dem sie sich befinden, erreichen können.

Gewaltenteilung

Scrum kennt drei verschiedene Rollen, die zwar gleichberechtigt sind, deren Funktionen aber unabhängig und getrennt voneinander sind:

- ✓ Product Owner: Das *Was* und *Wann* (und nicht das *Wieviel*)
- ✓ Entwicklungsteam: Das *Wie* und das *Wieviel*
- ✓ Scrum Master: Der *Prozess*

Jede Rolle hat einen definierten Zweck, um die Produktivität des Teams zu steigern.

Die Macher von Scrum haben sich diese Rollen nicht zufällig ausgedacht, sondern diese durch jahrelange Erfahrung in der Arbeit mit Projektteams aller Art entwickelt. Sie sahen gute und schlechte Kombinationen und fanden heraus, dass sich die besten Ergebnisse mit diesen drei Rollen erzielen lassen.



Wir bevorzugen, dass jede Person in einer Scrum-Rolle eine Vollzeitkraft ist, die sich ausschließlich dem Projekt des Scrum-Teams widmet. Lassen Sie Ihre Teammitglieder nicht gleichzeitig an mehreren Projekten arbeiten und setzen Sie keine Teilzeitkräfte ein. Wie viele Fußballmannschaften der Bundesliga haben Teilzeitspieler oder solche, die für mehrere Teams spielen? Keine, die erfolgreich sind.



Im Scrum steht keine einzelne Person oder Rolle über der anderen. Alle stehen auf der gleichen Stufe, niemand ist ein Chef oder Untergebener. Die Betonung liegt auf dem »Wir« und nicht auf dem »Ich«.

Scrum-Framework

Scrum ist eher ein Framework als eine Methodik. Es bietet Klarheit über Verantwortlichkeiten durch Rollen, Sichtbarkeit durch Artefakte und Möglichkeiten zur Überprüfung und Anpassung durch Ereignisse. Innerhalb dieser Struktur ist Scrum

ein Container für andere Prozesse und Werkzeuge, die für die spezifischen Bedürfnisse eines Teams, einer Organisation oder eines Produkts geeignet sind.



Ein Scrum-Projekt verwendet ein 3-3-5-Framework:

- ✓ Drei Rollen
- ✓ Drei Artefakte
- ✓ Fünf Ereignisse

All diese Framework-Elemente passen in den Scrum-Prozess, der iterativ und inkrementell ist. Sie erstellen und verbessern Ihr Produkt schrittweise. Auch verbessern Sie Ihren Prozess zunehmend mit dem folgenden einfachen Framework:

- ✓ Rollen
 - Product Owner
 - Entwicklungsteam
 - Scrum Master
- ✓ Artefakte
 - Produkt-Backlog
 - Sprint-Backlog
 - Produktinkrement
- ✓ Ereignisse
 - Sprint
 - Sprint-Planung
 - Daily Scrum
 - Sprint-Review
 - Sprint-Retrospektive



In der Scrum-Welt sind Artefakte Listen von zu erledigenden Arbeiten oder Arbeitsprodukten, die als auslieferfähig angesehen werden. Im Gegensatz zu archäologischen Artefakten sind Scrum-Artefakte allerdings nicht in Stein gemeißelt. Der Scrum-Prozess erfordert eine kontinuierliche Überprüfung und Bewertung der Artefakte, um sicherzustellen, dass Sie in die richtige Richtung graben.

Jede Rolle, jedes Artefakt und jedes Ereignis im Scrum hat einen bestimmten Zweck. Sie stellen Ihr Projekt in den Scrum-Rahmen, indem Sie sich durch die sieben Phasen der Roadmap to Value bewegen (siehe weiter vorn in diesem Kapitel), aber die eigentlichen Werkzeuge und Techniken zur Erreichung Ihrer Ziele sind Ihre eigenen. Scrum sagt Ihnen nicht, wie Sie Ihr Ziel erreichen sollen, sondern bietet lediglich einen Rahmen, in dem Sie klar sehen können, was Sie tun.

Das Konzept von Scrum ist einfach, aber die Implementierung kann kompliziert sein. Mit Scrum es ist so ähnlich, wie körperlich in Form zu kommen. Das Konzept sagt, dass Sie mehr trainieren und weniger Kalorien zu sich nehmen müssen; in der Praxis kann der Prozess jedoch weit komplexer sein.

Es folgen einige gängige Praktiken, die Scrum (zusätzliche Elemente sind in Kursivschrift dargestellt) ergänzen und zu unglaublichen Erfolgen geführt haben. Wir haben hier auf ein 5-6-7-Framework umgestellt:

✓ Rollen

- Product Owner
- Entwicklungsteam
- Scrum Master
- *Stakeholder*
- *Scrum Mentor*

✓ Artefakte

- Vision

- *Produkt-Roadmap*
- Produkt-Backlog
- Release-Plan
- Sprint-Backlog
- Produktinkrement

✓ Ereignisse

- *Projektplanung*
- *Release-Planung*
- Sprint
- Sprint-Planung
- Daily Scrum
- Sprint-Review
- Sprint-Retrospektive

Das Framework ist immer noch einfach, jedoch enthält es zusätzliche Rollen, Artefakte und Ereignisse, die den Prozess glätten sollen. Wir besprechen diese Rollen, Artefakte und Ereignisse ausführlich im weiteren Verlauf dieses Buches.

Die Ursprünge von Scrum

Obwohl wir seit fast 100 Jahren an den agilen Frameworks arbeiten, die wir heute verwenden, wurde das erste Scrum-Team 1993 von Jeff Sutherland gegründet, nachdem er die Konzepte des Harvard-Business-Review-Artikels aus dem Jahre 1986 mit dem Titel »The New New New Product Development Game« angewandt hatte. Mit Scrum-Co-Creator Ken Schwaber formalisierte Jeff das Scrum-Framework auf der OOPSLA 95 (International Conference on Object-Oriented Programming, Systems, Languages, and Applications).

Das Feedback-Fest

Ein klarer Vorteil von Scrum gegenüber anderen Projektmanagement-Frameworks ist die Feedbackschleife, die Ihnen frühzeitig und kontinuierlich zeigt, was funktioniert, was nicht funktioniert, was fehlt und was merkwürdig ist.

Feedback wird regelmäßig von Scrum-Teammitgliedern, Stakeholdern und Kunden generiert. Der Prozess läuft in etwa so ab:

1. Tägliches Feedback findet unter den Mitgliedern des Entwicklungsteams statt, während sie die Projektanforderungen entwickeln.
2. Täglich findet eine direkte Interaktion zwischen dem Product Owner und dem Entwicklungsteam statt, um vor Ort Antworten und Feedback zu erhalten.
3. Der Product Owner gibt direktes Feedback, da er jede fertiggestellte Anforderung akzeptiert oder ablehnt.
4. Am Ende eines jeden Sprints geben interne Stakeholder Feedback.
5. Am Ende jedes Release erhält das Team Feedback vom externen Markt.

Sie erzielen mit dem Scrum-Modell bessere Ergebnisse als mit traditionellen Projektmanagementmodellen, da Scrum mehr Wert auf die Produktentwicklung als auf die Entwicklung von Artefakten legt. Daher ist es bei Scrum wichtiger, greifbare, getestete Produkte anzuliefern anstatt Berichte über das, was möglich ist. Sie erhalten regelmäßig Feedback, was es Ihnen ermöglicht, Ihr Produkt inkrementell so schnell wie möglich auf den Markt bringen zu können.

Am Ende des Projekts, das genau genommen aus einer Reihe von Sprints innerhalb einer Reihe von Marktreleases besteht, müssen Sie sich nicht fragen, ob Sie das produziert haben, was Ihre Kunden wollen. Da Sie die ganze Zeit über mit den Kunden

kommuniziert und von ihnen Feedback erhalten haben. Der Überprüfungs- und Anpassungsprozess dient Ihren eigenen Interessen, während Sie das liefern, was Ihre Kunden tatsächlich wollten.

Agile Wurzeln

Um Scrum zu verstehen, hilft es, in die weite Welt der agilen Techniken einzutauchen, denn Agilität ist der Oberbegriff, unter dem Scrum steht.

Agilität ist eine Beschreibung von Ansätzen, die sich an den Werten des Agilen Manifests und den zwölf Agilen Prinzipien orientieren, die wir in diesem Abschnitt skizzieren. Scrum ist einer dieser agilen Ansätze.



Einen umfassenden Einblick in agile Techniken finden Sie im Buch *Agiles Projektmanagement für Dummies* von Mark Layton und Steven Ostermiller (Wiley-VCH Verlag GmbH & Co, KGaA).

Drei Säulen der Verbesserung

Das empirische Prozesssteuerungsmodell steht auf drei Säulen, die für agile Ansätze und damit auch für Scrum typisch sind:

- ✓ Transparenz
- ✓ Überprüfung
- ✓ Anpassung

Transparenz

Besonders zeichnen sich Agile Techniken im Allgemeinen und bei Scrum im Besonderen durch Transparenz aus. Wenn die Kommunikationskanäle klar und zugänglich sind, werden Informationen im gesamten Unternehmen verbreitet. Die gesamte Organisation weiß, was getan wurde, woran gerade gearbeitet wird, was noch zu tun ist und welche Hindernisse den Weg

blockieren. Von Anfang an liefern Sie echte Ergebnisse, die getestet und dann genehmigt oder zur Anpassung zurückgeschickt werden. Die Verzögerungszeit zwischen dem Startdatum und den nutzbaren Ergebnissen beträgt nun Tage statt Monate.

Bei der Transparenz geht es nicht nur darum, schnell Ergebnisse zu sehen. Jeder sollte den Prozess aus dem gleichen Blickwinkel betrachten. Ein Framework (beispielsweise Scrum) wird gemeinsam genutzt, zusammen mit einer vereinbarten Definition of Done. Sowohl Beobachter als auch Teilnehmer sehen die Ergebnisse und können sie in der gleichen Weise interpretieren.

Überprüfung

Wie Sie in den folgenden Kapiteln erfahren, werden agile Projekte in die kleinstmöglichen verwertbaren Teile zerlegt (häufig als User Stories erfasst; siehe [Kapitel 3](#)). Ziele werden innerhalb eines festgelegten Zeitrahmens gesetzt: der Sprint, das Release und das Projekt. Nachdem ein bestimmtes Teilelement fertig ist, wird es überprüft. So wird sichergestellt, dass es funktioniert und das tut, was der Kunde wünscht.

Diese Überprüfungen werden von Personen durchgeführt, die mit dem Auftrag vertraut sind – diejenigen, die die Arbeit ausführen, und diejenigen, die den Kunden vertreten. Dieser Prozess eliminiert die Zeitverzögerung, die eine externe Person benötigt, um diese Aufgabe zu erledigen. Es bedeutet auch, dass alle Anpassungen schnell vorgenommen werden können, da das erforderliche Wissen vorhanden ist.

Anpassung

Ergeben sich im Rahmen der Überprüfung Ungenauigkeiten und/oder Ineffizienzen, das heißt, wenn das Feature nicht richtig funktioniert, muss eine Anpassung vorgenommen werden. Die Anpassung sollte so schnell wie möglich und vor dem Wechsel zum nächsten zu erledigenden Punkt auf der To-Do-Liste erfolgen. Mit anderen Worten, wenn Sie weitermachen, wissen Sie, dass alles, was Sie bereits erledigt haben, richtig funktioniert.

Scrum ermöglicht die sofortige Durchführung von Überprüfung und Anpassungen auf Team- und Projektebene in Form von Reviews, Retrospektiven und dem Daily Scrum (siehe [Kapitel 6](#) und [7](#)).

Ein agiles Manifest

Scrum ist ein Framework und keine Methodik mit vorgegebenen Schrittfolgen. Sie müssen immer noch selbst denken und Entscheidungen treffen. Einer der Vorteile des Scrum-Frameworks besteht darin, dass es Ihnen ermöglicht auf Basis der Realität, in der Sie sich befinden, die für Sie besten Entscheidungen zu treffen.

Im Jahr 2001 einigten sich 17 Software- und Projektextperten auf vier Hauptwerte für ihre Programmiermethoden. Diese Werte werden als Agiles Manifest bezeichnet:

Wir erschließen bessere Wege, um Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- ✓ **Individuen und ihre Interaktionen** stehen über Prozessen und Werkzeugen.
- ✓ **Funktionierende Software** steht über einer umfangreichen Dokumentation.
- ✓ **Zusammenarbeit mit dem Kunden** steht über Vertragsverhandlungen.
- ✓ **Reagieren auf Veränderungen** steht über dem Befolgen eines Plans.

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.

Agiles Manifest © 2001: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Auch wenn das Agile Manifest und die Prinzipien von und für Software-Experten geschrieben wurden, bleiben die Werte für jedes Scrum-Projekt gültig. Das Global Positioning System (GPS) zum Beispiel wurde vom und für das Militär entwickelt, aber es bedeutet nicht, dass man davon nicht profitieren kann, wenn man in einen unbekannten Stadtteil fährt.



Weitere Informationen über das Agile Manifest und seine Urheber finden Sie unter <http://agilemanifesto.org>.

Zwölf Agile Prinzipien

Die Urheber des Agilen Manifestes haben sich ebenfalls auf zwölf Agile Prinzipien geeinigt. Innerhalb Ihres Scrum-Projekts können Sie anhand dieser Prinzipien überprüfen, ob Ihr Framework den agilen Zielen entspricht:

1. Unsere oberste Priorität ist es, den Kunden durch frühzeitige und kontinuierliche Lieferung wertvoller Software zufriedenzustellen.
2. Begrüßen Sie Anforderungsveränderungen, auch spät in der Entwicklung. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefern Sie häufig funktionierende Software, von ein paar Wochen bis zu ein paar Monaten, und bevorzugen Sie kürzere Zeiträume.
4. Geschäftsleute und Entwickler müssen während des gesamten Projekts täglich zusammenarbeiten.
5. Bauen Sie Projekte um motivierte Menschen herum. Geben Sie ihnen die Umgebung und Unterstützung, die sie brauchen, und vertrauen Sie darauf, dass sie die Arbeit erledigen.
6. Die effizienteste und effektivste Methode, Informationen an ein und innerhalb eines Entwicklungsteams zu übermitteln, ist das persönliche Gespräch.
7. Funktionierende Software ist das wichtigste Maß für den Fortschritt.

8. Agile Prozesse fördern eine nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Nutzer sollten in der Lage sein, ein konstantes Tempo auf unbestimmte Zeit beizubehalten.
9. Ständiges Augenmerk auf hohe technische Qualität und gutes Design fördern die Agilität.
10. Einfachheit – die Kunst, den Umfang der nicht geleisteten Arbeit zu maximieren – ist unerlässlich.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen aus selbstorganisierenden Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann, und passt sein Verhalten entsprechend an.



Die Prinzipien ändern sich nicht, aber die Werkzeuge und Techniken, um sie zu erreichen, können sich ändern.

Einige der Prinzipien sind leichter umzusetzen als andere. Nehmen Sie als Beispiel Prinzip 2. Vielleicht sind einige Bereiche Ihres Unternehmens (oder Ihrer Gruppe oder Familie) offen für Veränderungen und neue Ideen. Für diese ist Scrum natürlich und sie sind bereit loszulegen. Es kann aber auch andere geben, die widerstandsfähiger gegenüber Veränderungen sind.

Oder nehmen Sie Prinzip 6. Das Arbeiten von Angesicht zu Angesicht ist in Ihrem Projekt möglicherweise nicht möglich. Mit dem Internet und der Globalisierung der Belegschaften haben Sie vielleicht Teammitglieder von Mumbai über Moskau bis Miami. Sie könnten technische Lösungen in Betracht ziehen – wie Skype, Google Hangouts und Telefonkonferenzen – aber keine dieser Lösungen entspricht der Absicht von Prinzip 6 und keine ist so gut wie die persönliche Kommunikation.

Ihr Projekt steht vor besonderen Herausforderungen. Lassen Sie sich von einem Schluckauf oder einem weniger perfekten Szenario nicht davon abhalten, Ihr Projekt weiterzuführen. Ein Teil des Spaßes am Umgang mit Scrum ist die Lösung von Problemen, um Ergebnisse zu erzielen. Halten Sie sich an die

zwölf Prinzipien und Ihre Projekte werden schnell zu qualitativ hochwertigen Ergebnissen führen.

Der Marshmallow-Test

Im Jahr 2010 hielt der Autor und Redner Tom Wujec einen bemerkenswerten TED-Vortrag mit dem Titel »The Marshmallow Challenge«, in dem er eine Designübung des Designers Peter Skillman erörtern. In dieser Übung erhalten kleine Gruppen von Teilnehmern 18 Minuten Zeit, um eine freistehende Skulptur mit seltsamen, minimalen Werkzeugen zu bauen: 20 Spaghetti, 1 Stück Klebeband, 1 Stück Schnur und 1 Marshmallow. Wujec begann, diesen Test durchzuführen und die Ergebnisse zu studieren. Die meisten Gruppen bemühten sich darum, etwas Hochwertiges oder Verlässliches zu schaffen. Die Gruppenmitglieder diskutierten Optionen, planten ein endgültiges Design und bauten es zusammen, nur um herauszufinden, dass die Struktur nicht standhalten würde.

Die Gruppen, die am schlechtesten abgeschnitten haben, waren kürzliche Absolventen von Business Schools. Die Gruppen mit den besten Leistungen waren Kindergartenkinder, die immer höhere und kreativere Strukturen hervorbrachten.

Wujec sagte, wenn Wirtschaftsstudenten an einer Idee arbeiten, glauben sie, dass es nur eine »richtige« Lösung gibt, also verbringen sie einen Großteil ihrer Zeit damit, über diese nachzudenken und sie zu planen. Die Kinder hingegen begannen einfach, mit den Werkzeugen zu spielen. Sie lernten, was nicht funktionierte, und änderten es; dann fanden sie heraus, was funktionierte, und behielten es. Mit anderen Worten, sie haben die ganze Zeit Prototypen gebaut.

Im Hinblick auf den Kontext von Scrum lässt sich erkennen, dass die Prinzipien der Überprüfung und Anpassung der Natur des Menschen entsprechen. Wir wollen überprüfen und anpassen, aber im Laufe der Zeit verlernen wir es. Stattdessen wird uns beigebracht, dass die Planung und die vorherige vollständige Erarbeitung einer Lösung der richtige Weg ist.

Aber Kinder erinnern uns daran, dass dieser Denkansatz falsch sein kann.

Drei Platinum-Prinzipien

Wir arbeiten seit mehr als einem Jahrzehnt mit agilen und Scrum-Projekten und haben Dutzende von kleinen Firmen, großen Unternehmen und gemeinnützigen Organisationen beraten. Wir wissen, wie gut diese Prinzipien funktionieren, weil wir ihren Wert bei der Umsetzung gesehen haben.

Hier sind drei weitere Prinzipien, die die Leistung der Teams, die wir berieten, kontinuierlich verbessert haben:

- ✓ Widerstehen Sie Formalitäten.
- ✓ Denken und handeln Sie als Team.
- ✓ Visualisieren Sie, statt zu schreiben.

Diese Prinzipien lassen sich auf jedes Projekt anwenden, nicht nur auf die Softwareentwicklung. Ein Teil der Schönheit der agilen Techniken ist, dass man sie für alles verwenden kann.

Widerstehen Sie Formalitäten

Haben Sie jemals eine umwerfende Präsentation gesehen und sich gefragt, wie viel Zeit jemand damit verbracht hat, diese zusammenzustellen? Denken Sie nicht einmal daran, dies für ein Scrum-Projekt zu tun. Sie können es auf ein Flipchart schreiben, es an eine Wand kleben, wo die Leute es sehen können, und sich dann wieder der Wertschöpfung zuwenden. Wenn eine Diskussion erforderlich ist, gehen Sie zu den betroffenen Parteien und führen die Diskussion. Die Visualisierung jeder Iteration des Entwurfsprozesses nimmt nur sehr wenig Zeit in Anspruch. Verwenden Sie Ihre wertvolle Zeit und Mühe auf die Entwicklung des Produkts statt auf ausgefeilte Präsentationen.



Atos Origin hat eine unabhängige Studie erstellt, die zeigt, dass der durchschnittliche Mitarbeiter im Unternehmen 40 Prozent seines Arbeitstages mit internen E-Mails verbringt, die keinerlei geschäftlichen Nutzen bringen. Das bedeutet, dass die eigentliche Arbeitswoche erst am Mittwoch beginnt.

Prunk wird zu oft mit Professionalität und Fortschritt verwechselt. In Scrum-Projekten werden Sie bei Fragen ermutigt, sofort, direkt und informell zu kommunizieren. Auch sollen Sie eng mit Ihren Teammitgliedern zusammenzuarbeiten, um die Effizienz zu steigern und Zeit zu sparen.

Vermeiden Sie diese unproduktiven Fallen:

- ✓ Ausgefallene, zeitraubende Präsentationen
- ✓ Lange und/oder unkonzentrierte Meetings
- ✓ Riesige Mengen an Dokumentation
- ✓ Übermäßiger Aufwand, der den Fortschritt rechtfertigt

Betonen Sie diese Produktivitätssteigerer:

- ✓ Machen Sie nur so viel, dass es gerade ausreicht. Bei allen Dingen sollte nur so viel gearbeitet werden, um gerade so das Ziel zu erreichen. (Verwechseln Sie Angemessenheit nicht mit Mittelmäßigkeit. Ausreichend ist ausreichend; mehr ist verschwenderisch, und dadurch, dass Dinge unnötig aufgebläht werden, entstehen oft Probleme. Siehe Agiles Prinzip 10 weiter vorn in diesem Kapitel).
- ✓ Kommunizieren Sie häufig mit allen Beteiligten, um den Bedarf an umfangreichen Updates zu reduzieren.
- ✓ Kommunizieren Sie einfach und direkt. Wenn Sie mit jemandem persönlich sprechen können, tun Sie das.

Finden Sie den einfachsten Weg, um das zu bekommen, was Sie brauchen, immer mit dem Ziel, das Produkt mit der besten Qualität zu liefern.



Über kurz oder lang wird sich in Ihren Projekten eine Scrum-Kultur entwickeln. Wenn sich die Menschen in diesem Prozess weiterentwickeln und weil sie die Ergebnisse sehen, wird sich im Laufe der Zeit ihre anfängliche Definition für »eben ausreichend« entsprechend anpassen. Ertragen Sie also jeden anfänglichen Rückschlag mit Weitsicht, Geduld und Konsequenz.

Im Team denken und handeln

Das Herzstück von Scrum ist die Teamarbeit. Das Umfeld des Scrum-Teams kann jedoch zunächst beunruhigend sein, da die deutsche Unternehmenskultur das Individuum betont: »Wie kann ich in diesem Umfeld erfolgreich sein, um mich von anderen abzuheben und befördert zu werden?«

Bei Scrum überlebt oder stirbt das Projekt auf Teamebene. Indem Sie das Talent jedes Einzelnen in einem Team einsetzen, bringen Sie das Projekt von durchschnittlich zu hyperproduktiv. Wie Aristoteles sagte: »Das Ganze ist mehr als die Summe seiner Teile.«

Wie entsteht diese Teamkultur? Das Scrum-Framework selbst betont die Teamarbeit. Arbeiten am gleichen Ort, gemeinsame Ziele und kollektives Eigentum schreien nach »Team«. Fügen Sie die folgenden Praktiken zu Ihrem Scrum-Framework hinzu:

- ✓ Verabschieden Sie sich von Titeln. Niemand »besitzt« Entwicklungsbereiche; der Status wird durch Fähigkeiten und Beiträge bestimmt.
- ✓ Lassen Sie Teammitglieder paarweise arbeiten, um die Cross-Funktionalität und die Qualitätssicherung zu verbessern. Wechseln Sie dann häufig die Paarungen.
- ✓ Berichten Sie immer in Team-Metriken, nicht Einzel- oder Paarkennzahlen.

Visualisieren statt schreiben

Im Großen und Ganzen sind Menschen visuell, sie denken bildlich und erinnern sich bildlich. Die meisten Kinder mögen Bilder – visuelle Illustrationen von Texten. Erwachsene sind da nicht anders. Vermutlich blättern wir zuerst ein Magazin durch, um uns die Bilder anzusehen, und kehren dann zu den Artikeln zurück, um diejenigen zu lesen, die unser Interesse geweckt haben.

Bilder, Diagramme und Grafiken geben Informationen sofort weiter. Schriftliche Berichte erfordern die Bereitschaft des Lesers, sich damit zu beschäftigen; die Lesebereitschaft nimmt ab, je umfangreicher die Berichte werden.



Twitter war daran interessiert, die Effektivität von Tweets mit Fotos gegenüber denen, die nur Text enthielten, zu untersuchen. Das Unternehmen führte eine Studie mit dem SHIFT Media Manager durch und kam zu einigen interessanten Ergebnissen. Bei Tweets mit Fotos waren die Nutzer fünfmal häufiger im Einsatz als bei reinen Text-Tweets. Und die Rate der Retweets und Antworten mit Fotos wurde verdoppelt. Die Kosten pro Einsatz von Foto-Tweets waren jedoch halb so hoch wie bei reinen Text-Tweets (Shift Newsroom, 17. Januar 2014).

Wenn möglich, ermutigen Sie Ihr Team, Informationen visuell darzustellen, auch wenn dies lediglich bedeutet, dass Sie ein Diagramm auf einer Tafel skizzieren. Wenn jemand das Diagramm nicht versteht, kann er danach fragen. Änderungen können direkt an Ort und Stelle vorgenommen werden. Außerdem stehen Ihnen mit der aktuellen Technologie jederzeit Grafiken, Diagramme und Modelle zur Verfügung.

Die fünf Scrum-Werte

Scrum basiert auf fünf Werten, an denen sich jedes Mitglied des Teams orientiert:

✓ Engagement

- ✓ Fokus
- ✓ Offenheit
- ✓ Respekt
- ✓ Mut

Diese Werte sind keine Zauberei. Stattdessen fallen sie in die vertraute Kategorie des gesunden Menschenverstands. Dennoch sind sie entscheidend für die erfolgreiche Umsetzung von Scrum, weshalb sie es verdienen, hier besprochen zu werden.

In den folgenden Abschnitten betrachten wir jeden dieser Werte genauer, um zu zeigen, wie wichtig sie innerhalb Scrum-Projekten sind.

Engagement

Mitglieder von Scrum-Teams müssen sich dem Erfolg verpflichtet fühlen und bereit sein, realistische Ziele festzulegen und sich daran zu halten. Jeder muss mitmachen. Ein Scrum-Projekt ist eine »Alle-in-einem-Boot-Situation«. Jeder ist Teil eines Teams und alle arbeiten zusammen, um die Verpflichtungen des Teams zu erfüllen. Glücklicherweise stellt das Scrum-Modell sicher, dass Sie die Autorität und Freiheit haben, genau das zu tun.

Der Kern von Scrum ist ein Ereignis namens Sprint, das wir in [Kapitel 5](#) behandeln. Ein Sprint erfordert klare Ziele, die innerhalb festgelegter Zeitrahmen erfüllt werden. In diesem Modell teilen Sie diese Ziele in die kleinstmöglichen Einheiten, damit Sie wissen, worauf Sie sich einlassen. Weil Sie wissen, was realistisch ist, können Sie sich die entsprechenden Ziele setzen und Ihre Verpflichtungen erfüllen.

Fokus

Ein Teil der Magie von Scrum besteht darin, dass es um das Konzept des Fokus herum aufgebaut ist. Konzentrieren Sie sich auf ein paar Dinge gleichzeitig. Sie haben eine klare Rolle und klare Ziele innerhalb dieser Rolle. Ihre Aufgabe ist es, mit Ihrer Rolle dazu beizutragen, das Ziel zu erreichen. Jeden Tag wissen

die Teammitglieder, worauf sie sich konzentrieren müssen, um an diesem Tag erfolgreich zu sein. Das wirkt befreiend.

Sie haben die Ziele bereits festgelegt und sich dazu verpflichtet, sie zu erreichen. Konzentrieren Sie sich auf diese Ziele und nichts anderes.

Machen Sie sich keine Sorgen; geben Sie Ihr Bestes; seien Sie zufrieden.

Offenheit

Alles in Ihrem Projekt und in allen anderen Projekten ist transparent und kann überprüft und verbessert werden. Die Ziele und Fortschritte aller am Projekt Beteiligten – Sie, Ihr Chef, Ihre Mitarbeiter, Ihre Schwiegereltern – sind offen und sichtbar. Vorbei sind die Zeiten der (unangenehmen) Überraschungen, die erst nach sechs Monaten zu Tage treten.

Die Basis von Scrum sind glücklicherweise die agilen Säulen der Empirie: Transparenz, Überprüfung und Anpassung. Informationsverteiler (große, sichtbare Diagramme) und Echtzeitinformationen ermöglichen ungehindertes Handeln. Die meisten Menschen sind dieses Maß an Offenheit nicht gewohnt. Aber nachdem sich Scrum in Ihrer Organisation durchgesetzt und Anklang gefunden hat, wird es ohne diese Offenheit nicht mehr gehen.

Respekt

Jedes Teammitglied wird nach seinen Stärken ausgewählt. Zu diesen Stärken kommen Schwächen und Möglichkeiten zu lernen und zu wachsen. Die goldene Regel innerhalb von Scrum ist, dass jeder Teilnehmer alle anderen respektieren muss.

Harmonie entsteht durch die Synchronisation von Rollen, die im Laufe des Projekts einen Entwicklungsrhythmus erzeugt. Wenn eine Person ein wenig verstimmt ist, ist es in Ihrem besten Interesse, dieser Person zu helfen, da alle Teammitglieder als Team zur Verantwortung gezogen werden.

Menschen wollen gute Arbeit leisten, so sind wir nun einmal angelegt. Wenn Sie das Positive suchen, werden Sie das Positive finden. Genauso gilt: Wenn Sie das Negative suchen, werden Sie das Negative finden. Respekt ist die brennende Glut der Positivität.

Mut

Bei Scrum geht es um Veränderung und Ehrlichkeit und jede Idee, die Sie haben, wird in einem Scrum-Modell herausgefordert. Kein Verfahren ist gerechtfertigt, weil Sie die Dinge immer auf eine bestimmte Weise gemacht haben. Verabschieden Sie sich von Verfahren, die Sie aus Gewohnheit angewandt haben, und begrüßen Sie einen Prozess, der auf dem basiert, was das Team für erfolgreich hält. Jacob Bronowski könnte über das Scrum-Modell gesprochen haben, als er sagte:

Es ist wichtig, dass die Schülerinnen und Schüler einen gewissen Lumpensammler mitbringen, der barfuß respektlos studiert; sie sind nicht hier, um das Bekannte zu verehren, sondern um es in Frage zu stellen.

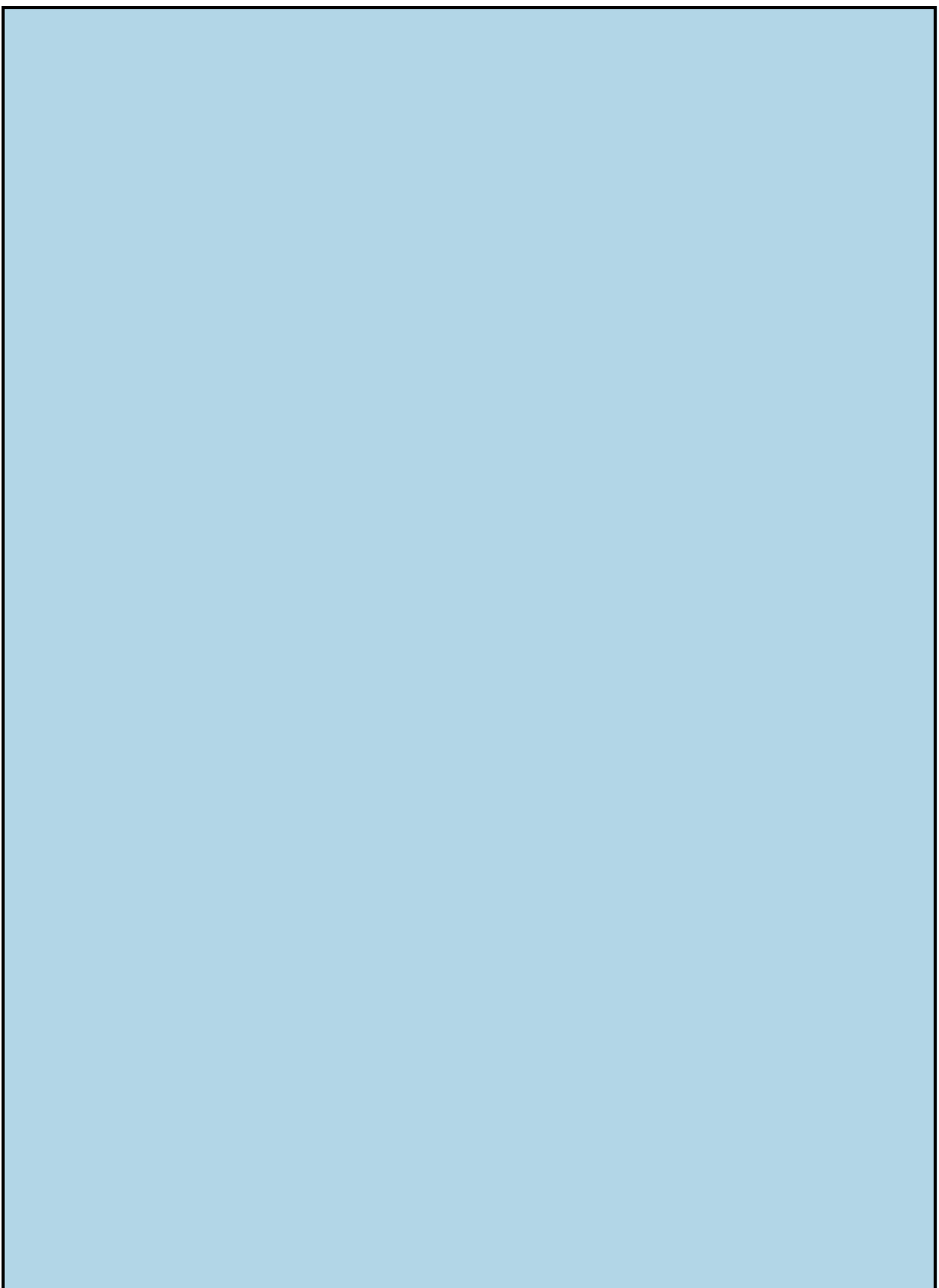
Lehren werden herausgefordert. Die Regeln werden getestet. Routinen werden gebrochen. Es wird Verbesserungen geben. Veränderung kann hart sein. Veränderung braucht Mut.

Scrum braucht Mut.

Teil II

Ein Scrum-Projekt durchführen





IN DIESEM TEIL ...

Ergänzen Sie Scrum durch weit verbreitete agile Praktiken.

Transformieren Sie die allgemeine Vision über den Projektumfang in ausführbare Aufgaben.

Planen Sie Releases und Sprints.

Optimieren Sie die Sprintergebnisse.

Korrigieren Sie den Kurs des Projekts durch Überprüfung und Anpassung.

Kapitel 2

Die ersten Schritte

IN DIESEM KAPITEL

Die Vorteile von Scrum quantifizieren
Die Rolle des Product Owners strukturieren
Ihre Produktvision erstellen
Die Scrum-Master-Rolle implementieren
Gängiger Praxis folgen

Die Arbeit dehnt sich genau in dem Maße aus, wie Zeit für ihre Erledigung zur Verfügung steht.

Parkinsonsches Gesetz

Scrum ist einfach im Konzept, aber oft schwierig in der Anwendung. Das Paradigma der Produktentwicklung der letzten 70 Jahre zu ändern, ist eine Herausforderung. Dennoch sind 30 bis 40 Prozent schnellere Markteinführung von Produkten und 30 bis 70 Prozent Kosteneinsparung realistisch. Jeff Sutherland, einer der Begründer von Scrum, hat 1.000 Prozent Leistungssteigerung durch den Einsatz von Scrum dokumentiert. Angesichts dieses Potenzials lohnt es sich, aus seiner Komfortzone herauszukommen und sich mit den organisatorischen Dysfunktionen zu befassen, die Sie zurückhalten.

Wenn der wichtigste Trend in der IT die Umstellung auf Scrum und die damit verbundenen agilen Engineering-Ansätze (wie eXtreme Programming) sind, dann ist es auch für Ihr Unternehmen möglich, den Übergang zu vollziehen. Viele Unternehmen trauten sich und machten ihre Sache gut. Der

Prozess erfordert Aufgeschlossenheit und die Bereitschaft, sich auf Neues einzulassen – etwas, das für uns alle gut ist. Am Ende dieses Kapitels haben Sie Ihr Projekt auf die Beine gestellt und sind in der Lage, die nächsten Scrum-Schritte zu gehen.

Ihr Scrum auf die Beine stellen

Bei der Umstellung auf Scrum spielen zwei Faktoren eine Rolle:

- ✓ **Die Art des Projekts:** Es ist leicht herauszufinden, ob die Art eines Projekts zu Scrum passt, denn Scrum ist für alle da. Jedes Projekt, für das Sie einen frühen, empirischen Leistungs- und Qualitätsnachweis wünschen, kann und sollte mit einem Scrum-Framework durchgeführt werden.
- ✓ **Die sozialen Rahmenbedingungen, in denen sich das Projekt befindet:** Soziale Rahmenbedingungen sind komplex, weil Menschen komplex sind. Veränderungsprozesse können einfach sein, Menschen sind es nicht. Jeder Mensch und jede Gruppe von Menschen weist bestimmte Eigenheiten auf. So unterhaltsam diese Eigenheiten auch beim Grillen sind, sie können beim Erlernen neuer Projektmanagement-Techniken eine Hürde sein, die es zu überwinden gilt.

Es ist natürlich, dass sich Menschen dem Wandel widersetzen. Dabei widersetzen sich Menschen in unterschiedlichem Maße und auf unterschiedliche Weise. Wenn Menschen jedoch verstehen, wie sie von den Veränderungen direkt profitieren, verläuft die Umstellung schneller und einfacher.

Wie Sie in den [Teilen 3 bis 5](#) dieses Buches erfahren, profitieren milliardenschwere Unternehmen ebenso von Scrum wie Leute von nebenan. Ein Kollege benutzte Scrum, um die einzelnen Tage seines Urlaubs zu planen. Er und seine Frau waren sich einig, dass dies der beste Urlaub war, den sie je hatten, weil Scrum die richtige Kombination aus Struktur und Freiheit erlaubte. ([Kapitel 17](#) enthält weitere Beispiele für die Verwendung von Scrum im familiären Bereich.)

Zeig mir, wo das Geld steckt!

Nehmen Sie das Einmaleins des Nettobarwerts: Ein Dollar ist heute mehr wert als es ein Dollar in sechs Monaten sein wird. Das größte Problem in Organisationen ist nicht die Effizienz der taktischen Ausführungsteams, sondern das schlechte Portfoliomanagement. Führungskräfte zeigen nicht die notwendigen Führungsqualitäten, um harte Prioritäten zu setzen. Dies führt dazu, dass viele Projekte auf eine niedrigere Managementebene gedrängt werden, welcher wiederum die Macht fehlt, sich zu wehren. (In [Kapitel 13](#) finden Sie mehr Details zu dieser Dynamik.)

Diese Dysfunktion wird dadurch verdeckt, dass Mitarbeiter gleichzeitig mehreren Projekten zugewiesen werden, sodass jede Geschäftseinheit etwas abbekommt. Es dauert so wesentlich länger, alles zu erledigen. Gleichzeitig werden die Manager jedes Projekts von umfangreichen Aktenordnern beschwichtigt, die ihnen das Gefühl vermitteln, wie großartig ihre Produkte sein werden, wenn sie sie schließlich bekommen. Diese Konflikte zwischen den Projekten haben ihren Preis.

Scrum ist das Gegenteil. Sie konzentrieren sich, Sie liefern greifbare Ergebnisse, und dann machen Sie Schritt für Schritt weiter. Das Produkt-Backlog (im Detail in [Kapitel 3](#) beschrieben) zwingt Sie, effektiv zu sein, bevor Sie sich Gedanken über die Effizienz machen.

Ihre Organisation kann ein Milliarden-Dollar-Unternehmen oder ein Tante-Emma-Laden sein, das beziehungsweise der darum kämpft, eine großartige Idee auf den Markt zu bringen. Vielleicht sind Sie einer von einer Million Mitarbeitern, und Sie haben dieses eine Projekt bekommen, um sich zu beweisen. In jedem Fall können disziplinierte Priorisierung, gesteigerte Effizienz und inkrementell getesteter Fortschritt Ihnen helfen zu überleben.

Durch die Priorisierung innerhalb von Scrum arbeiten Sie ausschließlich an den Features, die den höchsten Wert bringen. Sie perfektionieren kein Third-Tier-Widget statt eines hochwertigen Features. Sie suchen immer nach dem Fleisch auf

dem Teller. Daher ist das, was Sie bei jedem Sprint produzieren, das Wichtigste, Praktischste und sofort Erwünschte. Mit jedem Release erhalten Sie etwas, das der Markt schätzt. Das ist Scrum. Genau dort liegt das Geld.



Wer mit dem Rücken an der Wand steht, greift auf agile Techniken zurück, ob er es nun realisiert oder nicht. Hätte Ihr Unternehmen nur noch für 60 Tage Barreserven zur Verfügung, würde sich niemand Gedanken darüber machen, ob ein Statusbericht das richtige Deckblatt hat. Bürokratie ist der Luxus derjenigen, die finanziell gut ausgestattet sind, und es ist ein Luxus, der sich wegen der Schnelllebigkeit der heutigen Wirtschaft über Nacht ändern kann.



In unseren Seminaren lehren wir, dass es besser ist, eine Sache richtig zu tun als viele Dinge gleichzeitig, aber nur halbherzig. Wenn Sie warten, bis alles fertig ist, besteht die Gefahr, dass nichts passiert. Nehmen Sie stattdessen die konkreten Schritte des Fortschritts, die Sie durch Scrum erreichen, bringen Sie sie auf den Markt, erhalten Sie Feedback und lassen Sie die Einnahmen fließen.

Glücklicherweise ist Scrum darauf ausgelegt, getestete und brauchbare Ergebnisse, frühzeitig und oft zu liefern. Sie warten nicht darauf, Ergebnisse zu sehen. Sie sehen sie nach jedem Sprint.

Probieren Sie diesen Ansatz selbst aus. Fragen Sie sich selbst oder Ihren Kunden: »Wenn wir nur einen Monat Zeit hätten, um dem Markt einen Mehrwert zu liefern, was würden wir bauen und wie?« Sehen Sie, wie dieser inkrementelle Ansatz den Wert in den Vordergrund rückt?

Machen Sie eine Sache richtig, bevor Sie vieles nur halbherzig tun

Dem Ansatz folgend, eine Sache richtig zu tun anstatt vieles nur halbherzig, stimmte ein Klient zu, ein Produkt auf den Markt zu bringen, bei welchem der Kunde nur eine Möglichkeit hatte, um das Produkt zu bezahlen. Anstatt sicherzustellen, dass jede erdenkliche Kreditkarte unterstützt wird, PayPal funktioniert und Schecks schnell verarbeitet werden können, entschied sich der Klient, mit nur einer Zahlungsoption, per Kreditkarte, schnell Umsatz zu generieren. Das Ergebnis? Zwischen Oktober und Januar wurden Umsatzerlöse im Wert von mehr als einer Million Dollar erzeugt. Inzwischen kann die Website mehrere Kreditkarten, PayPal und verschiedene andere Zahlungsoptionen verarbeiten – jede davon wurde einzeln eingeführt, *nachdem* das Produkt aktiv Einnahmen generierte.



Scrum mag für viele noch ein neues Konzept sein, aber seine Nutzung nimmt sprunghaft zu. Im Jahr 2008 erfuhren wir mit Begeisterung, dass mehr Unternehmen agile Techniken als Wasserfalltechniken für die Softwareentwicklung einsetzten. Forrester schätzte, dass bis 2012 in mindestens 80 Prozent der Unternehmen agile Techniken für die Softwareentwicklung eingesetzt wurden. Im Mai 2012 fanden sich auf SimplyHired.com 20.000 Jobs, die Scrum-Kenntnisse erforderten, im Mai 2013 waren es bereits 670.000. Wir haben die Kluft überwunden und das Wachstum der agilen Projekte ist laut der jährlichen Umfrage von VersionOne.com während 2016 weiter gestiegen.

Ich will es sofort!

Schon mal gehört? Nicht nur dreijährige Vorschulkinder sagen es, auch Chefs und Kollegen fordern »Ich will es sofort«, oder man hört sogar die innere Stimme im eigenen Kopf. Mit Scrum ist es eher möglich, Ergebnisse sofort zu sehen. Bei Scrum-Projekten ist die für die Markteinführung benötigte Zeit regelmäßig 30 bis 40 Prozent kürzer. Wie geht das?

Die Antwort ist einfach: Indem Sie früh mit der Entwicklung beginnen, beenden Sie die Entwicklung schneller. Sie erstellen von Anfang an Produkte, die Sie potenziell ausliefern können. Sie warten nicht Monate oder Jahre auf Ergebnisse, die ihr technologisches Verfallsdatum überschritten haben könnten. Sie planen, erstellen, prüfen, passen an, liefern aus und profitieren. Dabei schaffen Sie frühzeitig und kontinuierlich Werte.



In der Wissenschaft wie in der Wirtschaft gilt die Regel, dass nicht die Stärksten überleben, sondern die Schnellsten. Wer sich am schnellsten in Sicherheit bringen kann, verringert die Gefahr, zur Beute zu werden. In der Wirtschaft werden Innovationen mit exponentiell steigender Geschwindigkeit auf den Markt gebracht. Neue Marken werden über Nacht geschaffen und verschwinden auch wieder über Nacht. Sie können es sich einfach nicht leisten, zu spät zu kommen.

Das ist nicht der einzige Grund, warum Sie eine schnellere Markteinführung wünschen. Während Sie Ihren Produkt-Backlog (die Projekt-To-do-Liste) erstellen, ordnen und priorisieren Sie die einzelnen Elemente. Bei der Priorisierung berücksichtigen Sie zwei Dinge:

- ✓ Einträge mit dem höchsten Wert
- ✓ Einträge mit dem größten Risiko

Beide Faktoren stehen ganz oben auf der Liste.

Ich bin mir nicht sicher, was ich will

Die meisten Menschen wissen nicht, was sie wollen, zumindest nicht, bis sie damit interagieren. Die meisten Menschen, Unternehmen und Organisationen wissen erst dann, was sie wollen, wenn sie direkt mit dem Produkt oder der Dienstleistung arbeiten. Die Lücke ist der Unterschied zwischen Wasserfall (es in der Dokumentation zu sehen) und Scrum (es zu verwenden).

In Ihrer Roadmap to Value (siehe [Kapitel 1](#)) beginnen Sie mit einer Vision davon, was Ihr Produkt sein soll. Diese Vision

fungiert als Leuchtfeuer für Ihr Team. Die Roadmap ermöglicht den natürlichen Verlauf der Entscheidungsfindung, von großen schwammigen Allgemeinheiten bis hin zur kleinen spezifischen Operationalisierung dieses Ziels. Die Vision bildet die äußere Grenze dessen, was sich ändern kann. Wenn das resultierende Produkt von der Vision abweicht, handelt es sich um ein anderes Projekt.

Scrum ermöglicht es Ihnen, ein erfolgreiches Unternehmen aufzubauen, egal, was es ist. Zum Beispiel

- ✓ Entwicklung einer Website, auf der Menschen bei Restaurants Allergiker-freundliches Bio-Essen bestellen können, dass dann nach Hause geliefert wird.
- ✓ Aufbau eines Alzheimer-Patientenwohnheims mit individueller Vor-Ort-Betreuung, Alarmierung und Sicherheit.
- ✓ Verkauf von Großmutter's Donut-Rezept, von dem Sie überzeugt sind, dass es zur Etablierung einer neuen Schnellrestaurantkette führen wird.

Wie sich diese Ideen in der Realität auswirken würden, steht noch nicht fest. Die gute Nachricht ist, dass Sie dank der Nutzung des Scrum-Frameworks den effektivsten Weg zur Erzielung von Fortschritten verwenden. Der Prozess der Entwicklung von etwas Greifbarem, die Überprüfung und Anpassung geben Ihnen die Werkzeuge an die Hand, um das benötigte Produkt zu entwickeln.

Ist das ein Problem?

Jedes Element, das das Entwicklungsteam vervollständigt, wird getestet und integriert, um sicherzustellen, dass es funktioniert. Der Product Owner ist dafür verantwortlich, jede Anforderung nach ihrer Fertigstellung zu akzeptieren oder abzulehnen. Mit anderen Worten, wenn etwas nicht funktioniert, verlässt es nicht den Sprint.

Natürlich können Probleme bei der Enterprise-Integration, bei Belastungsgrenzen auftreten, nachdem ein Produkt in die Produktion gelangt ist. Jedoch ist der Feedbackzyklus während

der Entwicklung so intensiv, dass sobald ein Defekt oder eine Prozessineffizienz festgestellt wird, sofort Korrekturen vorgenommen werden können. Das Problem wird in diesem Sprint behoben oder es wird wieder in das Produkt-Backlog eingefügt und wird dann gegenüber zukünftigen Arbeiten priorisiert.

Ihre Unternehmenskultur

Wenn die Leute den Erfolg und den Wert von Scrum erkennen, wird die Nutzung von Scrum einfacher. Die Mitarbeiter entdecken, dass Scrum die Kommunikation und Zusammenarbeit verbessert, Teamgeist entstehen lässt, einen natürlichen Lebenszyklus hat, zu einem ehrlichen und transparenten Umfeld führt und die Eigenverantwortung und das Maß an Entscheidungsfreiheit erhöht. All das beeinflusst die Unternehmenskultur enorm positiv beeinflusst.

Die Widerstandsfähigkeit gegen Veränderungen ist von Unternehmenskultur zu Unternehmenskultur unterschiedlich. Die Lösung ist, wie bei so vielen Dingen, ein greifbarer Erfolg. (Für nachgewiesenen Erfolg müssen Sie sich nicht verteidigen!) Finden Sie heraus, was die Schlüsselpersonen brauchen – höhere Gewinne, bessere Produktqualität, schnellere Lieferbarkeit oder bessere Talentbindung –, und zeigen Sie ihnen, wie das Scrum-Modell dies ermöglicht.

In jeder Gruppe gibt es die Personen, die viel Einfluss haben – diejenigen mit Schlagkraft, die Veränderungen ins Rollen bringen können. Vielleicht sind Sie jemand, der viel Einfluss hat, oder vielleicht ist es jemand anders. Holen Sie diese Person an Bord (was bedeuten kann, dass Sie zu einer höheren Managementebene gehen), und Ihre Arbeit wird einfacher.

Die Macht des Product Owners

Involvierung erzeugt Engagement. Sie wollen ein Team bilden, das den Motor der Veränderung am Laufen hält. Hierbei spielt der Product Owner eine Schlüsselrolle.

Die Hauptaufgabe des Product Owners besteht darin, sich um die geschäftliche Seite des Projekts zu kümmern. Diese Person ist für die Maximierung des Produktwerts verantwortlich, indem sie dem Unternehmen einen Return on Investment (ROI) liefert. Der Product Owner ist eine Person, kein Komitee, und ein vollzeitliches, engagiertes Mitglied des Scrum-Teams. Diese Person besitzt nicht das Produkt, sondern übernimmt die geschäftlichen Aufgaben und vertritt die Stakeholder und Kunden.

Zu den Hauptverantwortlichkeiten des Product Owners gehören:

- ✓ Ziele und Visionen für das Produkt festlegen, einschließlich der Erstellung des Vision Statements
- ✓ Produkt-Roadmap erstellen und pflegen, welche einen umfassenden Überblick über den Umfang des Produkts und das anfängliche Produkt-Backlog bietet
- ✓ Sofortige Prioritätensetzung und Entscheidungen für Kompromisslösungen
- ✓ Sichtbarkeit des Produkt-Backlogs sicherstellen
- ✓ Die Arbeit des Entwicklungsteams optimieren
- ✓ Die volle Eigenverantwortung für das Produkt-Backlog übernehmen
- ✓ Die vorgeschlagenen Anforderungen akzeptieren und nach Priorität im Produkt-Backlog sortieren
- ✓ Freigabe- und Sprintziele definieren
- ✓ Festlegen, welche Einträge des Produkt-Backlogs in den nächsten Sprint gehen
- ✓ Mit Geschäftsaspekten des Projekts umgehen, einschließlich ROI und Geschäftsrisiko sowie Schnittstellen zu Geschäftsinteressenten und Kunden
- ✓ Vision und Roadmap bekannt machen
- ✓ Den ganzen Tag über zur Verfügung stehen, um direkt mit den Mitgliedern des Entwicklungsteams zu arbeiten, wodurch die

Effizienz durch klare und unmittelbare Kommunikation erhöht wird

- ✓ Arbeitsergebnisse während des gesamten Sprints annehmen oder ablehnen, idealerweise an dem Tag, an dem sie abgeschlossen sind

Wir sind oft schockiert, dass Unternehmen, die Millionen Dollar in ein Projekt investieren wollen, behauptet, nicht die Ressourcen für einen engagierten Product Owner, der sicherstellt, dass die geschäftlichen und technischen Prioritäten aufeinander ausgerichtet sind und, dass das erstellte Produkt auch das benötigte Produkt ist. Dennoch haben viele dieser Organisationen einen Projektmanager, der das Projekt leitet. Da die Rolle des Projektmanagers in Scrum nicht existiert (die betreffenden Aufgaben sind Bestandteil der drei Scrum-Rollen; siehe [Kapitel 1](#)), kann das Geld für den Product Owner von dort abgezogen werden.

Product Owner klären, priorisieren und kümmern sich um ein Umfeld, das fokussiertes Arbeiten erlaubt. Sie stellen sicher, dass das Scrum-Team effektiv ist. Der Product Owner bestimmt, welche Anforderungen verfolgt werden und wann sich die Arbeit auf diese Anforderungen verlagert – also das Was und Wann, nicht aber das Wie oder das Wieviel. Das Wie und das Wieviel liegt in der Verantwortung des Entwicklungsteams.

Stellen Sie sich vor, Ihre Leidenschaft besteht darin, etwas zu erschaffen. Bei Scrum wären Sie ein Mitglied des Entwicklungsteams. Für Sie ist der Product Owner ein Geschenk. Diese Person zeichnet sich im Portfoliomanagement dadurch aus, dass sie befähigt ist, Entscheidungen zu treffen, zu klären, zu priorisieren und zu kämpfen, um sicherzustellen, dass sich die Teammitglieder jeweils auf ein Projekt konzentrieren. Durch effektive Product Owner werden die Mitglieder des Entwicklungsteams von Ablenkungen und externen Störungen befreit und haben mehr Zeit für ihre eigentliche Arbeit aufwenden.

Sowohl der Product Owner als auch der Scrum Master arbeiten daran, das bestmögliche Umfeld für das Entwicklungsteam zu

schaffen, damit es die bestmögliche Arbeit leisten kann. Der Product Owner kümmert sich um geschäftliche Belange und Interessen und der Scrum Master sorgt dafür, dass andere organisatorische Unterbrechungen das Entwicklungsteam nicht beeinträchtigen.

Die Abstraktionsschicht, die sich durch die Rollen von Product Owner und Scrum Master ergibt, führt nicht zu weniger Störfaktoren seitens der geschäftlichen Ebene. Sie führt jedoch meistens dazu, dass sich das Entwicklungsteam nicht mit diesen Störfaktoren auseinandersetzen muss.

Auf der anderen Seite können die Mitglieder des Entwicklungsteams direkt mit Stakeholdern, anderen Teammitgliedern oder Nicht-Teammitgliedern in Kontakt treten, wenn sie Klärungsbedarf bei etwas haben, an dem sie gerade arbeiten. Die Filterung bei der Priorisierung, nicht aber bei der Klärung ist wie die Membran einer Zelle, die bestimmte Flüssigkeiten in die eine Richtung, nicht aber in die andere fließen lassen lässt.

Das Ergebnis ist, dass das Entwicklungsteam vor äußeren Einflüssen geschützt, aber nicht in seinem Streben nach Wissen behindert wird. Diese Grenzen sind wichtig und integraler Bestandteil des erfolgreichen Funktionierens des Entwicklungsteams.

Warum Product Owner Scrum lieben

Product Owner lieben Scrum aus den folgenden Gründen:

- ✓ Entwicklungs- und Business-Aspekte sind aufeinander ausgerichtet und werden als Ganzes zur Rechenschaft gezogen und stehen nicht wie bei traditionellen Ansätzen zueinander im Widerspruch.

- ✓ Zeitplan und Kosten werden empirisch prognostiziert, und die Product Owner haben täglich Klarheit über den Fortschritt.
- ✓ Nach jedem Sprint wissen die Product Owner, dass die Elemente mit der höchsten Priorität voll funktionsfähig und potenziell auslieferfähig sind.
- ✓ Das Kundenfeedback ist früh und kontinuierlich.
- ✓ Die frühestmögliche greifbare Messung des ROI ist verfügbar – das heißt nach jedem Sprint.
- ✓ Auf sich ändernde Geschäftsanforderungen kann strukturell reagiert werden. Hierdurch hat der Product Owner kontinuierlich alle Flexibilität, um Anpassungen an die Marktgegebenheiten vorzunehmen.
- ✓ Es wird nicht durch die Entwicklung von Prozessartefakten (in der Regel Dokumentation), sondern durch den Schwerpunkt auf einer priorisierten Produktentwicklung die Verschwendung bei Produkt und Prozess reduziert.



Das wichtigste Merkmal des Product Owners sollte die Entschlossenheit sein. Diese Person trifft jede Stunde und jeden Tag harte, pragmatische und unbequeme Entscheidungen. Sie muss in der Lage sein, ein vertrauensvolles Umfeld zu schaffen und Anpassungen vorzunehmen, wenn Veränderungen erforderlich sind. Der Product Owner muss zunächst tun, was er für richtig hält, und sich dann auf der Grundlage empirischer Erkenntnisse ändern. Um effektiv zu sein, muss ein Product Owner taktische Entscheidungen treffen können, ohne hierfür ein Problem auf eine höhere Managementebene eskalieren zu müssen.

Die Rolle eines Scrum Product Owners unterscheidet sich deutlich von der eines traditionellen Projektmanagers. Stellen Sie sich vor, Sie sagen einem Golfer, er soll den Ball 400 Meter weit und direkt ins Loch schlagen, und dass er mit seinem eigenen Golfschläger geschlagen wird, wenn er es nicht schafft. Die

traditionelle IT-Welt funktioniert so. Mit Scrum schlägt der Golfer den Ball, bewertet die Ergebnisse und passt sich an, um das Ziel in der bestmöglichen Weise zu erreichen, und zwar basierend darauf, wo er gerade steht und nicht, wo er sich befinden sollte.

Product Owner Agent als Option

Heutzutage ist es nicht immer möglich, dass der Product Owner vor Ort arbeitet. Möglicherweise haben Sie Ihren Hauptsitz in Kalifornien, jedoch findet die Entwicklung der Projekte in einer Zweigstelle in Hyderabad in Indien statt. In unserer Firma haben wir eine umstrittene, aber praktikable Lösung in Form eines Product Owner Agent gefunden. Diese Rolle ist eine Person vor Ort, die für die tägliche Kommunikation und Entscheidungsfindung zuständig ist. Der Agent ist die physische Vertretung des Product Owners, welcher im Namen des Product Owners, der nicht vor Ort ist, spricht und handelt.

Product Owner Agents sind kein Element von Scrum. Obwohl wir Product Owner Agents nicht als Selbstverständlichkeit empfehlen, haben wir sie in bestimmten Situationen mit Erfolg eingesetzt, während die Organisation reift und sich entwickelt. Das Ziel ist es, die Rolle des Agenten nur temporär einzusetzen. Ähnlich wie bei einem Ausbildungsplatz, kann der Mitarbeiter so seine Fähigkeit, ein echter Product Owner zu sein, unter Beweis stellen.

Mit dem Product Owner Agent sind Sie in der Lage, schnell Klarstellungen zu erzielen und Entscheidungen zu treffen, so wie es Scrum erfordert. So wie ein Immobilienmakler die letztendliche Verantwortung für die Entscheidungen und Handlungen eines seiner Mitarbeiter übernimmt, ist der Product Owner am Hauptsitz für die Entscheidungen des Agenten verantwortlich. Die Verantwortung liegt weiterhin bei einer Person.

Anstatt an jedem Ort nur einen Fuß in der Tür zu haben und marginale Ergebnisse zu erzielen, ist es besser, den Product

Owner vor Ort einzusetzen. Die Vorabkosten mögen hoch erscheinen, aber wenn Sie die verbesserte Geschwindigkeit und höhere Qualität des Ergebnisses berücksichtigen, werden Ihre tatsächlichen Projektkosten gesenkt.

Das Unternehmensziel und die Strategie: Phase 1

Vision Statements sind kein Bestandteil des Scrum-Modells, aber nützlich und weit verbreitet. Unternehmen, gemeinnützige Organisationen und Einzelpersonen verwenden oft Produktvisionen.



Wenn wir Kunden coachen, lassen wir sie immer eine Vision erstellen, sodass sie ihr Ziel klar vor Augen haben. Wir suchen nach einer knackigen, prägnanten und klaren Aussage, einem Elevator Pitch, der während einer Aufzugfahrt vom Erdgeschoss in den vierten Stock kommuniziert werden könnte. Wir bitten sie, ihre Vision zu formulieren, bevor wir uns zum ersten Mal mit ihnen treffen. Dann verbringen wir eine Stunde mit den Verantwortlichen und erarbeiten mit ihnen etwas, womit wir arbeiten können. Es dauert nicht lange, dieses unschätzbare Artefakt zu erschaffen.

Vision Statements sind so nützlich, dass wir sie zur Phase 1 unserer Roadmap gemacht haben (siehe [Abbildung 2.1](#)). Stellen Sie sich Ihre Vision als Ziel mit einem Leuchtfeuer vor. Sie haben vielleicht 100 Möglichkeiten, zum Ziel zu gelangen, es spielt aber keine Rolle, welchen Weg Sie nehmen; entscheidend ist, dass Sie dort ankommen. Mit diesem Leuchtfeuer einer Aussage wissen Sie immer, wohin Sie unterwegs sind, denn Sie haben das Ziel im Blick. Von diesem stabilen strategischen Ziel aus haben Sie grenzenlose taktische Flexibilität.

Phase 1: Produktvision

Beschreibung: Ziele des Produkts und seine Ausrichtung an der Unternehmensstrategie
Besitzer: Product Owner
Häufigkeit: Mindestens jährlich



Abbildung 2.1: Die Produktvision ist Phase 1 der Roadmap to Value

Eine Produktvision ist

- ✓ intern ausgerichtet, ohne Marketing-Schnörkel,
- ✓ auf die Ziele des Marktes und die Kundenbedürfnisse abgestimmt,
- ✓ strategischer Natur; sie beschreibt das Was und nicht das Wie,
- ✓ jährlich überprüft (bei mehrjährigen Projekten),
- ✓ Eigentum des Product Owners.



Ihre Vision muss in der gesamten Organisation oder in der Personengruppe, mit der Sie zusammenarbeiten, kommuniziert werden. Ob das Team ein neues Sportwagenmodell entwirft oder eine Hochzeit plant, jeder muss das Ziel klar verstehen, da hierdurch die Erwartungen und der Akzent des Projekts bestimmt werden.

Strukturierung Ihrer Vision

In seinem ausgezeichneten Buch *Crossing the Chasm* (Collins Business Essentials) empfiehlt Geoffrey Moore eine effektive Methode zur Erstellung Ihrer Vision. Wir verwenden dieses Modell oft und mit erstklassigen Ergebnissen.

Die gesamte Aussage sollte nicht länger als zwei oder drei Sätze sein. Moore empfiehlt dieses Modell:

- ✓ Für <Zielkunden>
- ✓ die <Bedürfnisse>
- ✓ ist <Produktname >
- ✓ ein/eine <Produktkategorie>
- ✓ mit dem/der <Produktnutzen, zwingender Grund für die Anschaffung>
- ✓ im Unterschied zu <primären Alternativen von Wettbewerbern>
- ✓ ist unser Produkt <Wettbewerbsvorteil, Nutzenversprechen>

Wir empfehlen, diese Schlussfolgerung hinzuzufügen:

- ✓ was unsere Strategie zur <Unternehmensstrategie> unterstützt.

Hier sind Beispiele, wie dieses Format im wirklichen Leben aussieht:

- ✓ Durchlauferhitzer
 - *Für* Hausbesitzer, *die* einen kontinuierlichen Warmwasserfluss und eine optimale Energienutzung wünschen, *ist* der Acme Tankless Water Heater ein bei Bedarf arbeitender und an der Verbrauchsstelle installierter Durchlauferhitzer, *der* das Wasser effizient erwärmt, während Sie es nutzen. *Im Unterschied zu* Boilern, die einen Tank aufweisen, *bietet unser Produkt* eine kontinuierliche Warmwasserversorgung bei konstanten Temperaturen und niedrigeren Betriebskosten bei einem Wirkungsgrad von 94 Prozent, *was unsere Strategie unterstützt*, für Nachhaltigkeit zu sorgen, indem wir heute die Vergeudung natürlicher Ressourcen reduzieren.

✓ Urlaub auf Hawaii

- *Für* meinen Ehepartner und mich, *die* wir total gestresst sind, *ist* der Hawaii-Urlaub ein spontaner, kurzfristiger Ausflug, *der* uns lange genug von unserem hektischen Leben befreien wird, um neue Erfahrungen zu machen. *Im Unterschied zu* Familienferien oder vorgeplanten Rundreisen bietet *unser Urlaub* völlige Flexibilität ohne Erwartungen, *was unser Streben unterstützt*, das Beste aus jedem gemeinsamen Moment zu machen.



Die Produktvision selbst ist funktional, aber die Ergänzung der Geschäftsstrategie ist emotional. Geben Sie Ihrem Projekt einen Sinn in Form einer Unternehmensstrategie, die das Leben der Menschen verbessert. Es ist nie Ihre Unternehmensstrategie, Geld zu verdienen; die Strategie besteht darin, etwas Wertvolles zu schaffen, das dann monetarisiert werden kann.

Das Fadenkreuz finden

Das Vision Statement wird vom Product Owner erstellt und ist integraler Bestandteil des Projekts. Eine Meinung ist jedoch genau das: eine Meinung. Zwar ist der Product Owner Eigentümer des Statements, jedoch wird er sicherlich erfolgreicher sein, wenn er es mit Hilfe von kollektiver Intelligenz erstellt und verfeinert. Zu diesem Zweck kann der Product Owner wählen, ob er von den Mitgliedern des Entwicklungsteams, dem Scrum Master, externen oder internen Stakeholdern und sogar den Anwendern selbst Input erhalten möchte. Was mit dem Input zu tun ist, ist die Entscheidung des Product Owners und auch dessen Verantwortung.

Wenn Product Owner offen für Input von anderen sind, können sie Nuancen, Features und Marktaspekte erkennen, an die eine Person allein nicht denken würde. Es ist eine kluge Strategie, wenn der Product Owner das Feedback entgegennimmt und es

dann anschließend sorgfältig durch sein eigenes Verständnis des Projekts filtert.

Der Scrum Master

Im Buch *Der neue Minuten Manager* (erschienen im Rowohlt Taschenbuch Verlag) beschreiben Kenneth Blanchard und Spencer Johnson, dass einige der effektivsten Manager, mit denen sie sich beschäftigten, genau die technischen Fähigkeiten fehlten, die ihre Mitarbeiter auszeichneten. Seltsamerweise hatten genau diese Manager auch sehr viel Zeit. Wenn sie den Job nicht selbst erledigen konnten, worin waren diese Manager dann gut?

Diese Manager konnten den Weg frei machen, damit ihre Mitarbeiter die Arbeit erledigen konnten. Genau das ist die Rolle des Scrum Masters. Während der Product Owner eine leitende Rolle spielt, hat der Scrum Master eine unterstützende Rolle. Der Scrum Master ist für das Umfeld, das zum Erfolg führt, verantwortlich.

Eigenschaften des Scrum Masters

Die wichtigste Eigenschaft des Scrum Masters ist neben tiefgehender Expertise in Scrum, seine Durchsetzungsfähigkeit. Diplomatie, Kommunikationsfähigkeit und Führungsqualitäten sind gute Eigenschaften, aber der Scrum Master muss auch Respekt und Durchsetzungsfähigkeit besitzen, um schwierige Situationen zu lösen. Woher die Durchsetzungsfähigkeit kommt – Kompetenz, Ausdauer, Charisma, Verbundenheit – spielt keine Rolle, denn sie wirkt in der Scrum-Master-Rolle.

Als Servant Leader lehrt und ermutigt der Scrum Master. Außerdem beseitigt er taktische Hindernisse. Aber noch wichtiger, er beseitigt strategische Hindernisse, damit die taktischen nicht wieder auftauchen. So wie bei den anderen Rollen sollte auch der Scrum Master am besten in Vollzeit arbeiten und sich ausschließlich seinen Aufgaben als Scrum Master widmen. Das ist besonders bei neuen Teams, Projekten und Organisationen wichtig.



Wenn der Product Owner der der Spielmacher ist, der den Spielzug plant, und das Entwicklungsteam der Mittelstürmer, der das Tor erzielt, ist der Scrum Master der Außenstürmer, welcher die Räume schafft. Dennoch sind sie alle gleichberechtigt und haben ein gemeinsames Ziel.

Unsere Erfahrung zeigt, dass Teams ihre zukünftigen Leistungen schlechter anhand der in der Vergangenheit ermittelten Performance vorhersagen können, wenn die Entwickler gleichzeitig auch Scrum Master sind oder wenn Scrum Master in mehreren Teams arbeiten. Diese Situation führt zu Schwankungen bei der Verfügbarkeit und bietet dem Entwicklungsteam einen schlechteren Schutz. Dies macht quantitativ selten Sinn, da eine geringfügige Verbesserung der Geschwindigkeit eines Scrum-Teams (siehe [Kapitel 4](#)) oft einen großen Einfluss auf das Endergebnis hat. Wenn Ihr Team seine organisatorischen Unterbrechungen planen kann und so ausgereift ist, dass es sich nicht weiter verbessern kann, kontaktieren Sie uns; wir schreiben dann ein Buch über Sie.

Wie jede andere Rolle sollte der Scrum Master eine Vollzeitrolle sein und die Person, die sie innehat, sollte sich ausschließlich dieser Aufgabe widmen. Auch das ist insbesondere für neue Teams, Projekte und Organisationen wichtig.



Ein ideales Entwicklungsteam besteht aus drei bis neun Personen. Der Scrum Master verbessert damit die Leistung von bis zu neun Personen. Schon eine geringfügige Leistungsreduzierung hat einen neunfachen Effekt.

Zusätzlich zum How-to-Scrum-Coaching des Teams, unterstützt der Scrum Master die Events: Sprint-Planungsmeetings, Daily Scrums, Sprint-Reviews und Sprint-Retrospektiven.

Ein Scrum-Team ist ein Haufen intelligenter, engagierter Menschen mit einem hohen Maß an Eigenverantwortung für ihre Arbeit. Setzen Sie diese Leute in einem Meeting zusammen, und die kreative Energie kann dazu führen, dass sie explodieren –

oder dass das Team von Thema abschweift. Der Scrum Master hat die Aufgabe, diese Energie zu bündeln.

Der Einfluss des Scrum Masters erstreckt sich auf alle Beteiligten, einschließlich Stakeholder und Product Owner. Der Scrum Master ist ein Coach für alle, da alle am Scrum-Prozess Beteiligten ständige Weiterbildung und kontinuierliche Prozessbegleitung und -unterstützung brauchen.



Wenn Sie als Scrum Master Entscheidungen treffen, dann sind Sie auf der falschen Fährte. Ein Entwicklungsteam wird sich nie selbst organisieren, wenn es die Entscheidungen nicht selbst trifft. Scrum Master ziehen sich aus den täglichen Entscheidungen zurück, erzeugen eine förderliche Kollaborationsumgebung und schützen das Team vor Beeinträchtigungen.

Da der Scrum Master das Entwicklungsteam vor äußeren Einflüssen schützt, nimmt die Geschwindigkeit des Teams drastisch zu. Denken Sie daran, wie gut Sie arbeiten, wenn die Tür geschlossen, das Telefon ausgeschaltet ist und alle weg sind oder schlafen; anders als wenn Sie ständig von Kollegen, Familienmitgliedern und sogar von Ihrem Hund unterbrochen werden.

Selbst wenn die Einmischung von außen so gering wie möglich gehalten wird, ist es nicht ungewöhnlich, dass dennoch mehr Konflikte auftreten, weil die soziale Dichte im Scrum höher ist. Der Druck, Produkte in kurzen Sprint-Zeitfenstern zum Laufen zu bringen, kann sehr hoch sein, sodass die Aufgabe des Scrum Masters darin besteht, Konflikte auf die richtige Ebene zu bringen. Aufgabenkonflikte (bereit sein, für das zu kämpfen, was Sie für richtig halten) sind gesund und hilfreich. Persönliche Konflikte (jemanden persönlich herausfordern) sind es nicht.

Scrum Master als Servant Leader

Das Konzept des Servant Leader stammt aus der Zeit um 500 v. Chr. und wurde von Laotse entwickelt, der vermutlich der Autor

des *Daodejing* ist. Dieses Konzept wird jedoch in allen großen Religionen erwähnt und ist in modernen Unternehmensführungsmodellen beliebt. Die Stichwörter lauten Ausdauer und Durchhaltevermögen.

Der Servant Leader stellt andere an erster Stelle, damit sie ihre Arbeit tun können. Diese Führungskraft ermöglicht anderen, selbst ihre Aufgabe zu bewältigen, statt die Lösung auf einem Silbertablett zu präsentieren. Wenn jemand sagt: »Ich habe Hunger«, gibt ihm der Servant Leader keinen Fisch. Vielmehr fragt er: »Wie kann ich dir helfen, dass du heute, morgen oder nächstes Jahr keinen Hunger hast?« Der Scrum Master hilft jeder Person, Fähigkeiten zu entwickeln und die Lösung zu finden, die für diese Person am besten geeignet ist, unabhängig davon, ob die Antwort innerhalb oder außerhalb des Projekts liegt.

Als dienender Leiter lehrt, ermutigt und beseitigt ein Scrum Master taktische Hindernisse und beseitigt strategische Hindernisse, sodass taktische Hindernisse nicht wieder auftauchen.

Warum Scrum Master Scrum lieben

Scrum Master lieben Scrum aus vielen Gründen, unter anderem den folgenden:

- ✓ Sie können sich darauf konzentrieren, quantifizierbaren Einfluss zu haben, statt sich mit administrativen Aufgaben zu beschäftigen. Zunehmende Geschwindigkeit hat einen direkten Bezug zum Mehrwert, den das Scrum-Team liefern kann.
- ✓ Sie coachen Menschen, anstatt als Manager einfach nur Checklisten abzuarbeiten.
- ✓ Sie können Menschen befähigen, anstatt sie zu leiten.
- ✓ Sie stellen Fragen, statt Antworten zu geben.
- ✓ Sie nehmen an weniger und kürzeren Meetings teil.

- ✓ Sie ermöglichen den Aufbau eigenverantwortlicher, motivierter Teams, die selbstständig denken und mit Autorität handeln.
- ✓ Sie sind nicht für die Performance der Teammitglieder verantwortlich (»Hey, Joe, was ist der Status der Aufgaben, an denen Nancy arbeitet?«). Stattdessen wird die Verantwortung direkt der Person übertragen, die die Arbeit erledigt.



Ein gutes Scrum-Master-Motto lautet: »Iss niemals alleine zu Mittag.« Bauen Sie immer Beziehungen auf und pflegen Sie diese. Einfluss ist Ihre Währung, also stellen Sie sicher, dass Sie ein Umfeld erzeugen, in dem Sie einfach jemand anrufen oder zum Schreibtisch einer Person gehen können, um Ergebnisse zu erzielen.

Das Problem der Unterbrechungen

Studien haben gezeigt, dass es 15 Minuten dauert, um die richtige Konzentration für die Spitzenproduktivität zu erreichen – ein Zustand, der oft als »sich im Flow befinden« (mentaler Zustand der völligen Vertiefung) bezeichnet wird. Dennoch dauert es nur 2,3 Sekunden, bis die Blase platzt, sodass weitere 15 Minuten benötigt werden, um den Fokus wiederherzustellen. Eine 4,4-Sekunden-Unterbrechung verdreifacht die Anzahl der Fehler, die in der anschließend erledigten Aufgabe gemacht werden.

Am Arbeitsplatz herrschen drei Arten von Störungen vor:

- ✓ **Persönliche Unterbrechungen:** Persönliche Unterbrechungen sind E-Mails, Anrufe von Tante Martha und Textnachrichten mit Links zu Katzenvideos. Disziplin ist die Lösung. Schalten Sie die Technik ab und konzentrieren Sie sich auf Ihre Aufgabe. Lesen Sie auch den Kasten »Die Pomodoro-Technik« etwas weiter hinten.
- ✓ **Teamunterbrechungen:** Teamunterbrechungen sind solche, die von Arbeitskollegen verursacht werden. Das

Team muss die Zeit für die Zusammenarbeit und die Zeit zum konzentrierten Arbeiten ermitteln. Dann müssen diese Zeiten auf der Basis der Teamdynamik und der Projektanforderungen in eine gute Balance zueinander gebracht werden.

- Während der *Zeit zur Zusammenarbeit* ist es in Ordnung, zu unterbrechen, zu reden und Ideen auszutauschen. Diese Zeit ist eine gesunde und produktive Zeit, und sie sollte den Großteil des Tages eines jeden Teammitglieds ausmachen.
- Die *Konzentrationszeit* kann durch physikalische Indikatoren wie zum Beispiel geräuschunterdrückende Headsets und »Nicht stören«-Schilder angezeigt werden. Diese Zeit hilft den Teammitgliedern, ein hohes Maß an Konzentration zu erreichen und das Produkt zu entwickeln.

✓ **Externe Unterbrechungen:** Externe Unterbrechungen sind die Domäne des Scrum Masters. Diese Unterbrechungen können jederzeit, den ganzen Tag und jeden Tag auftreten. Der Scrum Master hat die Aufgabe, die Teammitglieder vor diesen Unterbrechungen zu schützen.

Die meisten Störungen werden durch Vorgesetzte verursacht, die taktische Notfälle an das Team weiterleiten, welches für die Entwicklung von Qualitätsprodukten verantwortlich ist. Hierdurch werden die Teammitglieder davon abgehalten, die für die nächste Zielerreichung erforderliche Arbeit zu leisten. Dies ist mit ein Grund dafür, warum der Scrum Master seine Aufgabe in Vollzeit ausüben und große organisatorische Durchsetzungsfähigkeit besitzen sollte. Manchmal muss der Scrum Master Beeinträchtigungen mit Vorgesetzten besprechen, um zu verhindern, dass der aktuelle Fortschritt entgleist.

Die Pomodoro-Technik

Die Pomodoro-Technik des Zeitmanagements, die Ende der 80er Jahre von Francesco Cirillo entwickelt wurde, besteht darin, persönliche Arbeitssprints in Einheiten von 25 Minuten (die Dauer kann angepasst werden) zu unterteilen, um seine eigenen Angelegenheiten zu erledigen. Die Technik umfasst die folgenden Schritte:

1. Erstellen Sie eine Aufgabenliste und legen Sie fest, welche Elemente den größten Wert besitzen.
2. Arbeiten Sie 25 Minuten lang an einem Element mit höchster Priorität.
3. Machen Sie fünf Minuten Pause.
4. Kehren Sie zum Element mit der höchsten Priorität, mit dem Sie begonnen haben, zurück und stellen Sie es vollständig fertig. Machen Sie dann mit dem Element mit der nächsthöheren Priorität weiter.
5. Nachdem Sie vier Pomodoros (in diesem Fall vier 25-Minuten-Arbeitszeiten) absolviert haben, machen Sie eine längere Pause von 15 bis 30 Minuten, um Ihre Konzentrationsfähigkeit wiederherzustellen.

Wiederholen Sie diese Schritte den ganzen Tag über, um sicherzustellen, dass Sie Ihre wichtigsten Aufgaben erfüllen.

Gängige Rollen außerhalb von Scrum

Product Owner und Scrum Master sind Scrum-Rollen, die von den Urhebern von Scrum entwickelt wurden und in jedem Scrum-Projekt vorhanden sind. Eine weitere Rolle ist das Mitglied des Entwicklungsteams (siehe [Kapitel 4](#)). Aber wie alle guten Dinge

entwickelt sich auch das Projektmanagement weiter und wächst. Scrum bleibt ein solides Gerüst und Fundament. Einige gängige und bewährte Praktiken können einen Mehrwert bringen.

Die folgenden beiden Rollen sind kein Scrum, aber wir haben festgestellt, dass sie einen enormen Mehrwert und Klarheit bringen, wenn sie richtig angegangen werden. Erwägen Sie, diese Rollen zu Ihrem Projekt hinzuzufügen; sie können den Wert Ihrer Scrum-Bemühungen erhöhen.

Stakeholder

Stakeholder sind Menschen, die Ihr Projekt beeinflussen oder von ihm betroffen sind. Interne Stakeholder sind innerhalb Ihres Unternehmens oder Ihrer Organisation; sie können aus den Bereichen Recht, Vertrieb, Marketing, Management, Beschaffung oder aus jeder anderen Abteilung Ihres Unternehmens stammen. Externe Stakeholder können Investoren oder Nutzer sein.



Obwohl Scrum nur Scrum-Teammitglieder vorschreibt, ist die Rolle des Stakeholders explizit mit Scrum-Teams verknüpft, und wir ziehen es vor, diese Rolle explizit anzuerkennen, damit wir sie explizit verwalten können. Der Product Owner ist die Geschäftsschnittstelle für das Scrum-Team, daher sollten die Stakeholder über den Product Owner mit dem Entwicklungsteam zusammenarbeiten. Stakeholder können direkt während der Sprint-Reviews, oder bei Kontaktierung durch ein Mitglied des Teams zur Klärung, mit dem Entwicklungsteam kommunizieren. Jedoch verwenden die Stakeholder in der Regel den Product Owner als Schnittstelle.

Zwei Rollen befassen sich mit den Stakeholdern:

- ✓ Wenn sich die Stakeholder auf der geschäftlichen Seite befinden (zum Beispiel Kunden, Verkaufsteams oder Marketing), ist der Product Owner verantwortlich.

- ✓ Wenn sich die Stakeholder auf der nicht geschäftlichen Seite befinden (zum Beispiel Lieferanten oder Auftragnehmer), ist der Scrum Master verantwortlich.

Der Schlüssel zur erfolgreichen Interaktion mit den Stakeholdern liegt darin, den Einfluss der Stakeholder zu erkennen und zu nutzen und gleichzeitig das Entwicklungsteam vor jeglicher Beeinträchtigung und Störung zu schützen.

Scrum-Mentoren

Wie gesagt, Scrum ist vom Konzept her ein einfaches Rahmenwerk, aber in der Praxis komplex. Dasselbe gilt für den Golfsport. Theoretisch benutzt man einen Golfschläger, um eine bewegungslose weiße Kugel mit möglichst wenigen Schlägen in ein Loch zu schlagen. Doch in der Praxis ist Golf nicht einfach. Wie bei Scrum ist es ein Spiel der Nuancen. Kleine Faktoren machen einen enormen Leistungsunterschied aus.

Der Mentor, manchmal auch Scrum-Coach genannt, arbeitet mit dem Team zusammen, um die Scrum-Praxis weiterzuentwickeln und reifen zu lassen. Der Vorteil beim Einsatz eines Mentors ist, dass die Mentoren außerhalb der normalen Firmenpolitik stehen und sich darauf konzentrieren, das Produkt aus der Tür zu bekommen. Sie können zurücktreten und objektiv beobachten, wie das Team arbeitet. Sie können nicht nur alte Gewohnheiten erkennen, sondern auch hausgemachte Änderungen am Scrum bremsen, mit denen genau diese alten Gewohnheiten eingeschleust werden sollen.

Sie werden die größte Mühelosigkeit und den größten Erfolg mit Scrum haben, wenn Sie sich in seiner wahrsten Form daran halten. Die Aufgabe eines Scrum-Mentors ist es, den Teammitgliedern zu helfen, in Form zu bleiben. Wie Sporttrainer stehen sie zur Seite, sehen alte, unproduktive Gewohnheiten und helfen Ihnen, neue Gewohnheiten zu entwickeln, die Sie erfolgreich machen.

Schwungvoll zum Erfolg

Es ist ein paar Jahre her, dass Mark mit dem Golfspielen begann und sich hierbei mit dem Golfvirus infizierte. Er liebte es, trainierte und verbesserte sein Spiel. Jedoch war das Problem seine Konsistenz. Also heuerte er einen Golftrainer an.

In der ersten Sitzung legte der Trainer einen Ball auf das Tee und Mark schlug ihn. Der Trainer beobachtete diesen einen Golfschwung: »Ich weiß, was dein Problem ist.« Mark war ungläubig. Wie kann der Trainer wissen, was mit Marks Schwung nicht stimmt, wenn er ihn nur einmal beobachtet?

Dann sagte der Trainer: »Du hast Baseball gespielt, nicht wahr?« Tatsächlich hatte Mark jahrelang Baseball gespielt und der Trainer konnte sofort erklären, dass sein Golfschwung einem Baseballschwung ähnelte. Mark hatte eine alte Gewohnheit in die neue Sportart übernommen.

Wenn Sie auf Scrum umsteigen, werden sich auch bei Ihnen alte Gewohnheiten einschleichen (auch solche, von denen Sie nicht wissen, dass Sie sie haben), die aber vermutlich in den meisten Fällen aus Wasserfallansätzen stammen.

Kapitel 3

Planen Sie Ihr Projekt

IN DIESEM KAPITEL

- Eine umsetzbare Produkt-Roadmap erstellen
- Ihr Projekt in umsetzbare Abschnitte aufteilen
- Ihre Master-To-do-Liste verwalten
- Gängige Techniken zur Erweiterung des Produkt-Backlogs einsetzen

Wilhelm von Ockham, ein Logiker und Franziskanermönch aus dem 14. Jahrhundert, wurde mit den Worten zitiert: »Entitäten sollten nicht unnötig vermehrt werden«. In einfacheren, modernen Begriffen ist diese Aussage als Ockhams Rasiermesser und als Sparsamkeitsprinzip bekannt. (Wenn Sie zwei konkurrierende Theorien haben, die die gleichen Vorhersagen machen, ist die einfachere besser.)

Anders ausgedrückt: Keep It Simple, Smarty. Sie können dieses Mantra immer wieder anwenden, wenn Sie Ihr Projekt mit Scrum verwalten. Wenn sich etwas nicht richtig anfühlt, ist es wahrscheinlich auch nicht richtig.

In diesem Kapitel sehen Sie, dass die Einfachheit der Dinge sowohl auf eine Technik zutrifft, mit der wir Scrum-Projekte verbessern – die so genannte Produkt-Roadmap –, als auch auf die Verfahren, um die Features Ihres Produkts in die kleinstmöglichen Anforderungen zu zerlegen.

In diesem Buch weisen wir auf gängige Praktiken hin, die Scrum-Trainer und Coaches erfolgreich anwenden. Obwohl sie optional sind, empfehlen wir Ihnen, sie auszuprobieren. Sie können dazu

beitragen, dass die Umstellung und der Abschluss Ihres Projekts erfolgreicher werden.

Die Produkt-Roadmap: Phase 2

In den sieben Phasen der Roadmap to Value, die wir in [Kapitel 1](#) skizzieren, beginnen Sie mit dem Ziel, Ihre Vision zu formulieren (eine gängige, agile Praxis, die gut funktioniert). Der nächste Schritt ist die Erstellung einer Roadmap, die Ihnen hilft, diese Vision zu erreichen (siehe [Abbildung 3.1](#)).

Eine gebräuchliche agile Praxis

Phase 2: Produkt-Roadmap



[Abbildung 3.1](#): Die Produkt-Roadmap ist Phase 2 der Roadmap to Value.

Obwohl weder die Produktvision noch die Produkt-Roadmap formale Bestandteile von Scrum sind, ist die Verwendung beider Elemente in der agilen Branche üblich.

Die Erstellung einer Produkt-Roadmap ist eine gebräuchliche Praxis, um sich auf den Weg zu machen und eine ganze Gruppe am gleichen Ziel auszurichten. Wenn Sie sich mit einer Crew und Passagieren auf eine Reise begeben, würden Sie Ihr Ziel und höchstwahrscheinlich Ihre Route kennen wollen, bevor Sie die Sicherheit des Hafens verließen. Dasselbe gilt für fast jedes Projekt. Im Folgenden werden die allgemeinen Schritte zur Erstellung einer Roadmap beschrieben:

1. Entscheiden Sie, wohin Sie gehen wollen (entwickeln Sie die Vision).
2. Finden Sie heraus, wie Sie dorthin gelangen können (entwerfen Sie die Produkt-Roadmap).
3. Bringen Sie Ihre Vision unter die Leute (kommunizieren Sie den Plan).

Die Produkt-Roadmap kann sich ändern, aber sie gibt Ihnen etwas Greifbares, um die Effizienz zu steigern. Indem Sie die Roadmap als Artefakt der Planung oder Neuplanung bekannt machen, können Sie sicherstellen, dass Einflussnehmer und Stakeholder die Richtung und den Weg verstehen. Anpassungen des Plans sind weniger kostspielig und eine gemeinsame Vision trägt zur Vereinheitlichung des Teams bei.

Wählen Sie die langfristige Perspektive

Die Produkt-Roadmap ist eine ganzheitliche, übergeordnete Sicht auf die Funktionen, die notwendig sind, um das Produktziel zu erreichen, das Sie in Ihrer Vision formuliert haben. Natürliche Projektaffinitäten werden hergestellt (»Wenn wir dies tun, sollten wir logischerweise jenes tun ...«), und Lücken in den Features werden leicht sichtbar gemacht (»Hey, wo ist ...?«).

Die Produkt-Roadmap ist das anfängliche Produkt-Backlog (Ihre Master-To-do-Liste). Wenn die Wahrscheinlichkeit der Entwicklung von Produkt-Backlog-Elementen steigt, werden diese nach und nach untergliedert (progressive Ausarbeitung). Das Produkt-Backlog erweitert sich um Einträge, die

- ✓ klein (unmittelbar entwickelbar; oft als User Stories bezeichnet)
- ✓ mittelgroß (mittelfristig entwickelbar; oft Epics genannt)
- ✓ konzeptionell (klar, aber ohne Details; oft als Features bezeichnet)

sind.



Im Kasten »Roadmap- und Backlog-Terminologie« weiter hinten in diesem Kapitel finden Sie weitere Informationen zu Features, Epics und User Stories.



Die Produkt-Roadmap ist nicht in Stein gemeißelt und vollständig gepflastert, sondern ein lebendiges, dynamisches Artefakt. Wir überprüfen und aktualisieren diese Roadmaps mindestens zweimal jährlich, wobei die Häufigkeit je nach konkretem Projekt variiert.

Obwohl sich die Vision vollständig im Besitz des Product Owners befindet, muss das Entwicklungsteam an der Erstellung der Roadmap beteiligt sein. Die Mitglieder des Entwicklungsteams sind die technologischen Experten, die auch die Arbeit erledigen werden. Deswegen müssen sie technische Einschränkungen und Aufwandsabschätzungen vorlegen. Sollte noch kein Entwicklungsteam ausgewählt worden sein, binden Sie die Funktionsmanager ein, nachdem Sie die erste Produkt-Roadmap erstellt haben. Diese Manager können dabei helfen, die erforderlichen Fähigkeiten zu identifizieren und das Entwicklungsteam so schnell wie möglich zusammenzustellen, damit das Team die allgemeinen Aufwandsabschätzungen vornehmen kann.

Wenn wir mit einem Kunden arbeiten, setzen wir uns mit den Stakeholdern, dem Product Owner, dem Entwicklungsteam und dem Scrum Master zusammen und erstellen am ersten Tag das Vision Statement und die Produkt-Roadmap. Wir stellen dies am ersten Tag fertig, damit die eigentliche Entwicklung am zweiten Tag beginnen kann.

Einfache Werkzeuge verwenden

Verwenden Sie bei der Erstellung Ihrer Produkt-Roadmap die einfachsten Werkzeuge. Wir bevorzugen ein Whiteboard und

Haftnotizen. Jedes Produktfeature passt auf eine Haftnotiz. Diese Methode ist in der Praxis gelebte Einfachheit.

Das menschliche Gehirn wurde nicht im digitalen Zeitalter erschaffen. Studien haben bewiesen, dass Elektronik eine dämpfende Wirkung hat. Tatsächlich braucht es weniger Gehirnwellenfunktion, um fernzusehen, als um Farbe trocknen zu sehen. Doch die Verwendung eines einfachen Systems wie Haftnotizen ist körperlich und geistig einnehmend und fördert ein Umfeld der Veränderung und Kreativität.

Wir haben vielen Kunden geholfen, eine Produkt-Roadmap zu erstellen, indem wir einfach Haftnotizen mit den wichtigsten Anforderungen an die Wand gehängt und Ideen und Themen besprochen haben. Für ein Großprojekt (im Wert von fast einer halben Milliarde Dollar) haben wir in weniger als drei Stunden eine Roadmap erstellt. Wir beginnen mit der höchsten Priorität. Während wir über die Vision sprechen, sind Haftnotizen einfach zu verschieben, während die Roadmap Gestalt annimmt.

Erstellen Sie Ihre Produkt-Roadmap

Sie können sieben einfache, gängige Schritte verwenden, um Ihre Produkt-Roadmap zu erstellen. Führen Sie diese Schritte am ersten Tag Ihres Projekts durch; sie sollten nicht länger als ein paar Stunden dauern. Bei kleineren Projekten erledigen Sie die Arbeit beim Morgenkaffee. Der Product Owner schließt zusammen mit dem Rest des Projektteams die Schritte (dem gesamten Scrum-Team und den Stakeholdern des Unternehmens) ab.

Mit Hilfe von Haftnotizen, verschiedenen Flaggenfarben und einem Whiteboard können Sie Ihre Produkt-Roadmap erstellen:

1. Schreiben Sie pro Haftnotiz eine Produkthanforderung auf.
Denken Sie an so viele Produkthanforderungen wie möglich.
2. Falls angemessen und synergistisch, ordnen Sie einige Anforderungen in verwandte Kategorien oder Gruppen ein.
3. Priorisieren Sie die Anforderungen auf der Roadmap.

Auf der Makroebene befinden sich die Elemente mit der höchsten Priorität auf der linken Seite und die Elemente mit der niedrigsten Priorität auf der rechten Seite. Auf der Mikroebene befinden sich die Anforderungen mit der höchsten Priorität oben und die mit der niedrigsten Priorität unten. Die Elemente mit der höchsten Priorität sind oben links und die mit der niedrigsten Priorität unten rechts.

4. Identifizieren Sie geschäftliche Abhängigkeiten (gekennzeichnet durch farbige Flaggen auf der Haftnotiz).
5. Lassen Sie das Entwicklungsteam technische Abhängigkeiten erkennen (gekennzeichnet durch Flaggen in einer anderen Farbe).
6. Lassen Sie das Entwicklungsteam die Größenordnung des Aufwands der einzelnen Haftnotizen schätzen.
7. Passen Sie die Reihenfolge entsprechend den Abhängigkeiten und Schätzungen an.



Achten Sie insbesondere auf die Elemente, die die meisten Diskussionen oder Bedenken im Teams auslösen; diese Punkte können die riskantesten sein oder eine weitere Aufschlüsselung erfordern. Schätzmethode, die Ihr Entwicklungsteam verwenden kann, finden Sie in [Kapitel 4](#).

Legen Sie Ihren Zeitrahmen fest

Die Art Ihres Produktes bestimmt die Häufigkeit und den Zeitpunkt Ihrer Produktreleases. Wir empfehlen mindestens einmal pro Quartal, aber das Innovationstempo nimmt zu. Wir haben Kunden, die jeden Tag in die Produktion gehen. Die erfolgreichsten Social-Media- und Internetfirmen releasen zweimal täglich oder öfter.

Im Idealfall sollten Sie nach jedem Sprint ein geprüftes, abgenommenes Produkt freigeben oder besser noch mehrmals während eines Sprints (kontinuierliche Lieferung). Gelegentlich erstellen wir bei der Zusammenarbeit mit Kunden Produkt-

Releases auf einer allgemeinen Ebene (für die Festlegung der Anforderungen, nicht für die zeitliche Planung). So können wir erkennen, wie diese Releases mit anderen Produkt-Releases, Budgetzyklen, Urlaubszyklen übereinstimmen.

[Abbildung 3.2](#) zeigt ein Beispiel für vierteljährliche Produktveröffentlichungen (eine gängige Praxis bei börsennotierten Unternehmen). Das erste Release spiegelt die initiale Releaseplanung wieder und ist mit einem Commitment versehen. Alles danach zeigt, wie sich äußere Einflüsse auf das Projekt auswirken können.

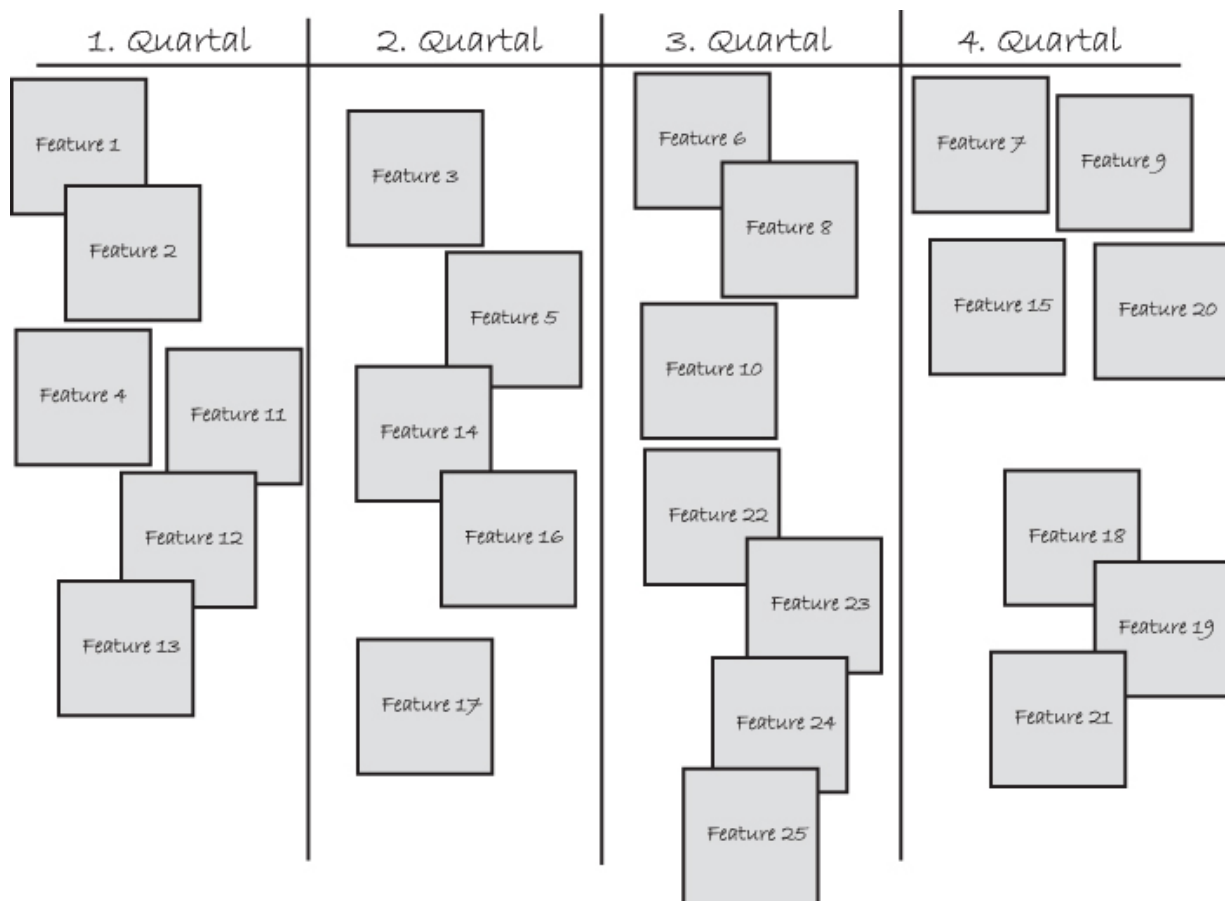


Abbildung 3.2: Ein Beispiel für eine Produkt-Roadmap nach Quartalen

Wenn Ihr Zeitrahmen kürzer oder für Ihr Projekt nicht relevant ist, überlegen Sie sich den Zeitrahmen Ihres Projekts. Teilen Sie ihn zunächst in logische Gruppen ein, wobei Sie nicht weiter als nötig

vorrausplanen (normalerweise nicht länger als ein Jahr). Folgen Sie diesen Schritten:

1. **Erstellen Sie den Wert der Funktionen der ersten Gruppe, beginnend mit den Anforderungen der höchsten Priorität, und machen Sie jeweils mit der nächstniedrigen Priorität weiter.**
2. **Gehen Sie bei jedem Produktrelease davon aus, dass Sie geprüfte, genehmigte und auslieferfähige Ergebnisse erhalten werden.**
3. **Schreiben Sie über jedes Release das jeweilige konzeptionelle Thema.**
4. **Nehmen Sie, falls nötig, Anpassungen vor.**



Die Produkt-Roadmap und das Produkt-Backlog eignen sich hervorragend für die Umfangssteuerung. In traditionellen Projektmanagement-Modellen muss jede neue Idee begründet werden. Bei Scrum fügen Sie die Idee der Liste hinzu, wenn sie Sie der Projektvision näher bringt. Die Priorität der Idee entscheidet später darüber, ob sie umgesetzt und fertiggestellt wird. Manchmal werden nur die Dinge mit der höchsten Priorität erledigt; in anderen Fällen wird jede Anforderung erledigt, je nach Budget und organisatorischem Aufwand für das Scrum-Team.

Anforderungen unterteilen

Es wird nicht lange dauern, bis Sie merken, dass eine Produkthanforderung in mehrere Teile zerlegt werden kann. Die Chancen stehen gut, dass diese Teile weiter zerlegt werden können, und diese Teile – wir sind bereit, hierauf zu wetten – können noch weiter zerteilt werden. In diesem Abschnitt wird erklärt, wie man diesen Prozess der Zerlegung, Unterteilung, Decomposition steuert.

Priorisierung der Anforderungen

Beim Erstellen der Produkt-Roadmap beginnen Sie Sie mit den größten Teilkomponenten Ihres Projekts. Diese Roadmap bietet eine gute Vogelperspektive und basiert auf dem, was im Hinblick auf den Geschäftswert und/oder die Risikovermeidung sinnvoll ist. Diese Teile (Anforderungen) werden vom Product Owner priorisiert; er entscheidet, was am wichtigsten ist, was den Kunden zuerst erreichen soll und welche Anforderungen logisch zusammengehören.

Da die Anforderungen priorisiert werden, werden sie Teil des Produkt-Backlogs. In Scrum arbeiten Sie an der kleinsten Menge von Elementen mit der höchsten Priorität, die für die Wertschöpfung notwendig sind, und nicht an irgendetwas, das zum Projektumfang gehört. Dies ist der Grund dafür, dass Scrum-Projekte schneller Funktionalität releasen können.

Berücksichtigen Sie bei der Priorisierung zwei Dinge: den Geschäftswert und das Risiko. Geschäftswert ist einfach. Wenn eine Sache einen hohen Wert hat, hat sie hohe Priorität. Sie sollten aus vier Gründen zuerst die risikoreichsten Anforderungen angehen:

- ✓ Am Anfang haben Sie das einfachste System, mit dem Sie arbeiten können. Sie sollten Ihre risikoreichsten Entwicklungen im einfachsten System angehen.
- ✓ Am Anfang steht Ihnen die größte Geldmenge zur Verfügung. Wenn Sie etwas mit hohem Risiko angehen wollen, sollten Sie dies mit den größten Ressourcen tun, die Ihnen zur Verfügung stehen.
- ✓ Weil Sie die größten Menge an Geld haben, können Sie das Entwicklungsteam am längsten finanzieren. Wenn Sie ein risikoreiches Element angehen, dann tun Sie dies mit dem Vorteil einer möglichst langen Start- und Landebahn.
- ✓ Wenn Sie scheitern, dann scheitern Sie früh und billig. Wenn ein grundsätzlicher Fehler vorliegt, sollten Sie das so früh wie

möglich wissen. Die risikoreichsten Anforderungen befinden sich dort, wo die Landminen lauern.



Die Arbeit an den Anforderungen mit der höchsten Priorität bis zur deren Fertigstellung ist nicht nur bequem, sondern spart auch Geld. Das ist der Grund, warum Scrum-Projekte mit 30 bis 70 Prozent Kostenersparnis realisiert werden können. Wenn Ihr Projekt zu 80 Prozent fertig ist und Ihnen dann das Geld ausgeht, können Sie ohne Zweifel sagen, dass 80 Prozent der Elemente mit der höchsten Priorität fertiggestellt, funktionsfähig und abgenommen sind. Die restlichen 20 Prozent? Wenn sie so eine niedrige Priorität hatten, kann man oft auch ohne sie leben.

Die Produkt-Roadmap erstellt Ihren ersten Schnitt von Abhängigkeiten. Zum Beispiel müssen Sie sich keine Gedanken darüber machen, ob Sie eine Funktion zur Übermittlung von Anwendungsfunktionen haben, noch bevor Sie eine Website haben, auf der diese Funktionalität untergebracht ist. Kritische Abhängigkeiten werden frühzeitig aufgedeckt und helfen dem Product Owner, die Einträge im Produkt-Backlog zu priorisieren. Die Teams beginnen an Tag 2, nicht an Tag 120, mit der Arbeit an den Elementen mit der höchsten Priorität.

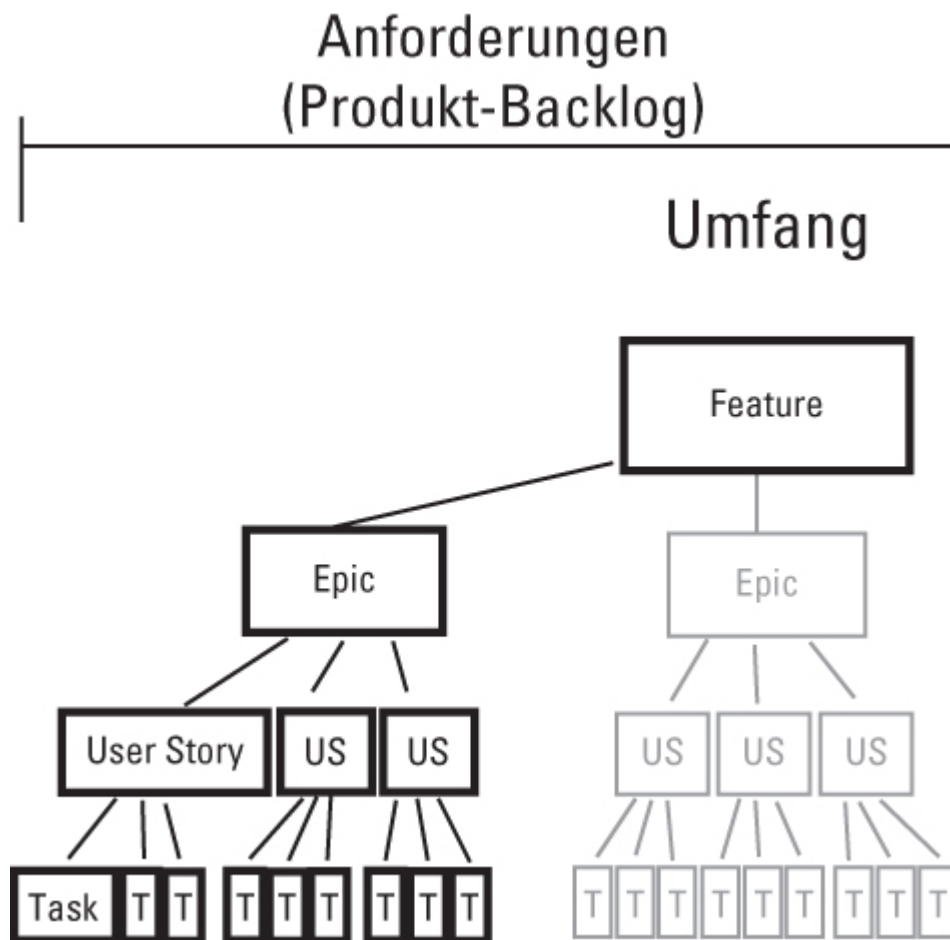


Eine Anforderung muss sich das Recht verdienen, dass in sie investiert wird. Nur die Elemente mit der höchsten Priorität verdienen Ihre Mühe, sie in verdauliche Anforderungen zu zerlegen. Alles andere kann warten. So arbeiten Sie immer nur an den wichtigsten Dingen. Verschwenden Sie nicht Ihre Zeit damit, sich zu viel zuzumuten. Etwas dazu zu lernen und Anpassung sind auch während der Erstellung der Anforderung wertvoll.

Unterteilungsstufen

Wenn Sie Ihre Anforderungen in kleinere Teile zerlegen, sollten Sie so wenig Details wie möglich erfassen. In der Tat möchten wir,

dass Sie ein Experte darin werden, nur das Nötigste zu tun. Sie sollen die Anforderungen nach und nach auszuarbeiten, und zwar dann, wenn die Wahrscheinlichkeit, sie in einen Sprint aufzunehmen, zunimmt. Vorbei sind die Zeiten, in denen endlose Stunden damit verbracht wurden, Anforderungen zu definieren und zu verfeinern, die nie das Licht der Welt erblicken. [Abbildung 3.3](#) (nächste Seite) zeigt die verschiedenen Ebenen für die Unterteilung der Anforderungen.



[Abbildung 3.3](#): Unterteilungsstufen

Sieben Schritte bei der Erstellung von Anforderungen

Sieben Schritte sind erforderlich, um den Umfang jeder Anforderung zu bestimmen. Nach diesen sieben Schritten wissen

Sie, dass jede Anforderung funktioniert, integriert werden kann und vom Product Owner genehmigt wurde. Sie entwickeln echte Produkte, um sie den Stakeholdern vorzustellen, von denen Sie Feedback erhalten können. Dieses Feedback wird dann verwendet, um neue Anforderungen zu verfeinern und zu erstellen, die das Produkt besser auf die Bedürfnisse der Stakeholder abstimmen. Danach wird der Zyklus erneut durchlaufen. Sie verbessern das Produkt schrittweise, basierend auf der Realität.

Die sieben Schritte zur Erstellung jeder Anforderung sind

1. Anforderungsausarbeitung
2. Entwurf
3. Entwicklung
4. Umfassende Tests
5. Integration
6. Dokumentation
7. Abnahme

Roadmap- und Backlog-Terminologie

Die folgenden Begriffe zu Produkt-Roadmaps und Produkt-Backlogs wurden im Laufe der Jahre von Scrum-Experten entwickelt. Sie sind kein Scrum, sondern Teil einer Sammlung gängiger Praktiken, die viele von uns in diesem Bereich anwenden:

- ✓ **Themen/Themes:** Ein Thema ist eine logische Gruppe von Features, das Sie in Ihrer Produkt-Roadmap anlegen. Wenn es sich bei einem Merkmal um ein neues Feature handelt, das ein Kunde erhalten wird, ist ein Thema die logische Gruppierung dieser Merkmale. Die Funktionalität, die es einem Kunden ermöglicht, Artikel von einer Website zu kaufen, ist ein Beispiel für ein Thema. Ihre Produkt-

Roadmap identifiziert und/oder gruppiert Features als Teil eines Themas.

✓ **Features:** Features sind Funktionen, die Ihre Kunden erhalten werden und die sie vorher nicht hatten. Die Funktionalität, die es einem Kunden ermöglicht, Artikel online über ein Mobiltelefon zu kaufen, ist ein Feature. Ihre Produkt-Roadmap besteht in der Regel aus Anforderungen auf der Ebene der Features.

✓ **Epics:** Epics sind, eine Reihe von Aktionen, die sich auf ein Feature beziehen, und der nächste Schritt, um Features in umsetzbare Anforderungen zu zerlegen. Die Funktionalität, die es einem Kunden ermöglicht, einen Artikel mit seinem Handy aus dem Warenkorb per Kreditkarte zu kaufen, ist ein Epic. Ein Epic ist kleiner als ein Feature (einen Artikel online kaufen), aber größer als die Kreditkartenintegrationen, die es ermöglichen, einen Artikel zu kaufen.

Hinweis: Wir erlauben in einem Release-Plan keine Anforderungen, die größer als Epics sind, da sie nicht spezifisch genug sind, um mit ihnen genau zu planen.

✓ **User Stories:** User Stories sind die kleinsten Formen von Anforderungen, die für sich allein stehen können. Eine User Story besteht aus einer Aktion, die einen bestimmten Wert schöpft, oder der Integration eines Wertes. Der Kauf eines Artikels mit einem Handy aus einem Einkaufswagen mit einer Visa Card ist eine User Story. Einen Artikel mit einer Mastercard zu kaufen ist eine andere Integration und eine andere User Story. User Stories sind so klein, dass sie in Sprints aufgenommen werden können und dass mit ihrer Entwicklung begonnen werden kann. In den folgenden Abschnitten gehen wir detailliert auf die User Stories ein.

✓ **Aufgaben/Tasks:** Aufgaben sind die Schritte, die notwendig sind, um eine User Story zu implementieren. Bei der Sprint-Planung wird eine User Story in Aufgaben zerlegt.

Zur Erinnerung: Anforderungen sind Dinge, die Benutzer tun; Aufgaben sind das, was das Entwicklungsteam tut, damit die Anforderung funktioniert.

Ihr Produkt-Backlog

Das Produkt-Backlog ist ein echtes Scrum-Artefakt und die Master-To-do-Liste für das gesamte Projekt. Alle Scrum-Projekte verwenden Produkt-Backlogs, die sich im Besitz des Product Owners befinden und von diesem gepflegt werden.

Die Anforderungen aus der Produkt-Roadmap bilden das anfängliche Produkt-Backlog; die Anforderungen mit der höchsten Priorität werden an Tag 1 in User Stories untergliedert. (Wie das geht, beschreiben wir im Abschnitt »Ausarbeitung des Produkt-Backlogs« weiter hinten in diesem Kapitel.) [Abbildung 3.4](#) zeigt ein Beispiel eines Produkt-Backlogs.

PRODUKT-BACKLOG

Nr.	ID	Element	Typ	Status	Schätzung
1	121	Als Administrator möchte ich Konten mit Profilen verknüpfen, damit Kunden auf neue Konten zugreifen können.	Anforderung	Nicht begonnen	5
2	113	Traceability Matrix der Anforderungen aktualisieren	Overhead	Nicht begonnen	2
3	403	Schulung Testautomatisierung für Michael	Verbesserung	Nicht begonnen	3
4	97	Refactoring der Login-Klasse	Wartung	Nicht begonnen	8
5	68	Als Besucher der Website will ich Filialen suchen können, damit ich Bankdienste nutzen kann.	Anforderung	Nicht begonnen	8

Abbildung 3.4: Das Produkt-Backlog ist die sortierte To-do-Liste Ihres Projekts.

Jeder Eintrag im Produkt-Backlog besitzt die folgenden Elemente:

- ✓ Nummer, die die Priorität des Eintrags im Produkt-Backlog angibt



Auch wenn die Einträge im Produkt-Backlog eine ähnliche Priorität aufweisen, können sie nicht gleichzeitig bearbeitet werden.

- ✓ Beschreibung
- ✓ Aufwandsschätzung für die Fertigstellung des Elements
- ✓ ID-Nummer (optional)
- ✓ Status (optional)
- ✓ Wert des Elements für das Unternehmen oder für das Produkt
- ✓ Typ des Eintrags (optional), bei dem es sich um eine Anforderung, Overhead, eine Wartung oder eine Verbesserung handeln kann (siehe »Die dynamische Aufgabenliste« weiter hinten in diesem Kapitel).



Die Anforderungen mit der höchsten Priorität werden in die kleinstmöglichen umsetzbaren Anforderungen, die das Produkt-Backlog erlaubt, untergliedert. Eine kleine Anforderung ist jedoch nicht automatisch ein Element mit einer hohen Priorität. Viele kleine Anforderungen besitzen nur eine geringe Priorität und sehen nie die Produktion.



Jeder kann ein Produktmerkmal oder eine Anforderung schreiben. Eine Idee oder ein Anliegen kann von einem Business-Stakeholder, einem Mitglied des Entwicklungsteams, dem Scrum Master, dem Product Owner und sogar dem Barista der Firma initiiert werden. Ideen werden von allen erwartet. Nur der Product Owner hat die

Möglichkeit, eine Anforderung in das Produkt-Backlog aufzunehmen oder abzulehnen, je nachdem, ob sie die Vision für das Produkt unterstützt oder nicht. Der Product Owner ist der alleinige Eigentümer dieses Artefakts. Wenn die neue Anforderung akzeptiert wird, wird sie vom Product Owner nummeriert, verfeinert, priorisiert und sortiert.

Die dynamische Aufgabenliste

Das Produkt-Backlog existiert, solange das Projekt aktiv ist. Wenn Sie ein Projekt haben, dann haben Sie ein Produkt-Backlog, das sich ständig ändert. Da größere Anforderungen in kleinere Anforderungen aufgeteilt werden, ändert sich auch das Backlog. Wenn Sie Feedback von Kunden erhalten, ändert sich das Backlog. Wenn Ihre Wettbewerber neue Angebote auf den Markt bringen, ändert sich das Produkt-Backlog.



Der Product Owner priorisiert nicht nur die Einträge im Produkt-Backlog; er ordnet sie auch und bringt sie in eine logische Reihenfolge. So kann der nächste Sprint anhand dieser sortierten Einträge schnell und effizient organisiert werden.

Alle Änderungen im Produkt-Backlog werden vom Product Owner vorgenommen. Wenn jemand (Product Owner, Entwickler, Stakeholder) eine neue Anforderung oder Design-Idee einbringt, erhält diese der Product Owner, der sie dann gemeinsamen mit allem anderen priorisiert.

In traditionellen Projektmanagement-Frameworks wird Veränderung als Zeichen für schlechte Planung gesehen. Wenn etwas geändert werden muss, dann nur, weil jemand Mist gebaut hat. In Scrum sehen wir Veränderung als Zeichen von Wachstum. Wenn Sie Ihr Produkt genauer kennenlernen, merken Sie, welche Änderungen vorgenommen werden müssen. In Scrum gilt: Wenn Sie nichts verändern und anpassen, dann lernen Sie auch nicht, und das ist ein Problem. Es ist ein Mangel an Veränderung, der als Zeichen des Scheiterns gilt. Man muss jeden Tag etwas lernen und das führt zu Veränderungen. Mit Scrum ist Veränderung nicht

länger etwas, vor der man unter den Schreibtisch kriecht und sich versteckt. Veränderung ist gut. Veränderung ist Leben.
Veränderung ist Scrum.

Ausarbeitung des Produkt-Backlogs

Durch die Ausarbeitung und Präzisierung des Produkt-Backlogs, das auch Refinement genannt wird, nimmt das Verständnis des Scrum-Teams für die Einträge im Produkt-Backlog zu. Das Team wird auch auf die bevorstehenden Sprints vorbereitet. Die Ausarbeitung des Produkt-Backlogs ist ein kontinuierlicher Prozess der Präzisierung, bei dem Rückmeldungen und Reaktionen auf Veränderungen für die zukünftige Entwicklung aufbereitet werden. Dieser Prozess befindet sich im Besitz des Product Owners und wird mithilfe der Mitglieder des Entwicklungsteams durchgeführt. Diese stellen klärende Fragen und geben auf Grundlage der zu diesem Zeitpunkt bestmöglichen verfügbaren Informationen Schätzungen ab. Für diesen Prozess wendet das gesamte Scrum-Team bis zu 10 Prozent seiner aktiven Sprintzeit auf. Der Scrum Master unterstützt in der Regel die Verfeinerung des Produkt-Backlogs, damit die Gruppe fokussiert und aufgabenorientiert bleibt.

Die angestrebten Ergebnisse der Verfeinerung des Produkt-Backlogs sind:

- ✓ **Klarheit:** Alle Entwickler und der Product Owner erzielen einen deutlichen Konsens über den Umfang der diskutierten Produkt-Backlog-Einträge
- ✓ **Akzeptanzkriterien:** Alle Anforderungen (Backlog-Einträge) beinhalten hinreichende Akzeptanzkriterien.
- ✓ **Risiken sind identifiziert und so weit wie möglich gemindert:** Andere bekannte Risiken sollten dokumentiert und bei Bedarf vom Team akzeptiert werden.
- ✓ **Umfangsbestimmung:** Die Anforderungen werden geschätzt und so aufgeschlüsselt, dass sie innerhalb eines Sprints erfüllt werden können.



Der Begriff Akzeptanzkriterien wird in der Scrum-Praxis häufig verwendet. Er bezieht sich auf einen Bereich auf der Rückseite der Story Card oder in der Projektsoftware mit einer Definition, was für jede Story »done« (fertig) genau bedeutet. Akzeptanzkriterien können Testkriterien und konkrete Beispiele dafür enthalten, wie das Feature nach Abschluss der Arbeit funktionieren soll.

Die weitere Ausarbeitung des Produkt-Backlogs sollte regelmäßig stattfinden, jedoch schreibt Scrum weder vor, wie formell oder wie häufig diese Ausarbeitungsdiskussionen stattfinden sollten. Auch eine bestimmte Tagesordnung ist nicht vorgeschrieben. Hier sind Vorschläge, die für uns und unsere Kunden gut funktioniert haben:

- ✓ **Format:** Der Product Owner stellt dem Team jeweils eine Anforderung vor. Für jede Anforderung stellen die Teammitglieder Fragen, stellen Annahmen in Frage, erforschen Umsetzungsstrategien und verwenden ein Whiteboard zum Zeichnen und zur Klärung, bis sie sich über die Details und den Umfang der Anforderung einig sind. Dann verwendet das Entwicklungsteam Schätzpoker oder Affinitätsschätzung (siehe [Kapitel 4](#)), um eine relative Schätzung zu erstellen.

Für eine Anforderung, die sich in der Priorität nach oben bewegt, gibt die Schätzung an, wann die Anforderung klein genug ist, um in einen Sprint zu passen.

- ✓ **Zeitaufwand:** Scrum-Teams verbringen durchschnittlich etwa 10 Prozent eines Sprints damit, den Backlog für den nächsten Sprint zu verfeinern.
- ✓ **Häufigkeit:** Bei der Weiterentwicklung des Produkt-Backlogs handelt es sich um eine schrittweise durchgeführte Tätigkeit. Die Anforderungen werden allmählich, just in time, weiterentwickelt. Diejenigen, die näher an der Auslieferung sind, werden für den Sprint vorbereitet. Teams können für das

Produkt-Backlog-Refinement einen dieser Zeitpläne verwenden:

- Täglich, am Ende eines jeden Tages
Dieser Zeitplan dient auch als Demobilisierungsübung für die Mitglieder des Entwicklungsteams, um für diesen Tag nach Hause zu gehen.
- Täglich, zu einer beliebig fest vereinbarten Zeit
- Einmal pro Sprint, während eines regelmäßigen, festgelegten Zeitraums
- Einmal pro Woche, während eines regelmäßigen, festgelegten Zeitraums
- Mehrmals pro Woche, während eines regelmäßigen, festgelegten Zeitraums
- Je nach Bedarf, vom Team in jedem Sprint festlegt und geplant

✓ **Aktivitäten:** Zum Refinement des Produkt-Backlogs gehören unter anderem die folgenden Aktivitäten:

- Diskussionen mit dem gesamten Team, einschließlich Whiteboarding, Modellbildung, Frage-und-Antwort-Sitzungen und Design Diskussionen
- Untersuchung der während der Teambesprechungen identifizierten Backlog-Einträge
- Interviews mit Fachexperten oder Stakeholdern, um weitere Erkenntnisse zur Ermittlung des Umfangs und zu Lösungsvorschläge für die Anforderungen zu erhalten
- Sitzungen für Schätzpoker oder Affinitätsschätzung

Während das Team die Einträge mit der nächsthöheren Priorität diskutiert und verfeinert, sollten Sie so viele der folgenden Fragen und Richtlinien wie nötig verwenden, um das Team durch den Ausarbeitungsprozess zu führen. Nicht alle Punkte gelten für jedes Team und/oder jede Anforderung.

✓ **Untergliederung großer Anforderungen:**

- Erfüllt die User Story die INVEST-Kriterien (siehe den Kasten »INVEST« weiter hinten in diesem Kapitel)?
- Kann die Anforderung innerhalb eines Sprints oder als Teil eines Sprints fertiggestellt werden?
- Ist die User Story eine einzelne Aktion von Wert oder eine Reihe von Aktionen?
- Ist der Umfang gerade ausreichend?
- Hat der Product Owner technische Aufgaben oder andere technische Details hinzugefügt, die dem Entwicklungsteam selbst überlassen werden sollten?
- Wurde die Definition of Done des Scrum-Teams (siehe [Kapitel 4](#)) berücksichtigt, bevor entschieden wurde, dass die User Story ausreichend untergliedert ist?
- Ist weitere Forschung nötig, bevor das Team eine Aufwandsschätzung abgeben kann?

✓ **Klärung und Refinement, wo nötig**

- Versteht das Entwicklungsteam die Geschäftsabsicht und/oder den Wert der Anforderung?
- Haben die Mitglieder des Entwicklungsteams versucht, die Anforderung zu paraphrasieren, um sicherzustellen, dass alle auf dem gleichen Stand sind?
- Ist das gewünschte Lieferprodukt deutlich?
- Ist ein Implementierungsansatz sowohl aus technischer als auch aus betriebswirtschaftlicher Sicht sinnvoll?
- Berücksichtigt das Team die gesamte Arbeit, die nötig ist, um die Story vollständig fertigzustellen und zu abzuliefern?
- Weiß das Team, welche Aufgaben erforderlich sind?
- Wird das Team in der Lage sein, am Ende dieses Sprints ein vollwertiges Produktinkrement zu liefern?

✓ **Sicherstellung angemessener Akzeptanzkriterien**

- Falls erforderlich, wurden Personas identifiziert (siehe »Häufig verwendete Produkt-Backlog-Praktiken« weiter hinten in diesem Kapitel)?
 - Sind die Akzeptanzkriterien vollständig und ausreichend?
 - Haben die testorientierten Mitglieder des Entwicklungsteams erklärt, dass die Kriterien ausreichend sind?
- ✓ Umgang mit potenziellen Problemen und Risiken
- Kann die User Story für sich alleine stehen oder gibt es Abhängigkeiten?
 - Welche Konflikte können bei der Umsetzung auftreten?
 - Welche technischen Verpflichtungen könnte diese Anforderung nach sich ziehen?
- ✓ Aufwandsschätzungen auf hoher Ebene
- Hat sich das Scrum-Team für eine konsistente Schätzmethode entschieden (siehe [Kapitel 4](#))?
 - Hat sich das Team darauf geeinigt, wie ein Konsens erreicht werden kann?
 - Haben Schätzungen ergeben, dass die Anforderungen immer noch zu groß sind?

Wenn Schätzungen zusätzliche Punkte aufwerfen, die geklärt werden müssen, verwenden Sie diese Richtlinien, um sie vor einer erneuten Schätzung weiter auszuarbeiten und zu präzisieren.



Während dieses Prozesses müssen Sie einige Faktoren beachten:

- ✓ Nur die Mitglieder des Entwicklungsteams schätzen den für die Anforderung benötigten Aufwand, weil sie es sind, die die Arbeit erledigen.

- ✓ Der Product Owner klärt die Fragen der Entwickler sofort, indem er sich aktiv an den Ausarbeitungsdiskussionen beteiligt.
- ✓ Die Häufigkeit und Dauer der Refinement-Aktivitäten variiert von Team zu Team, von Projekt zu Projekt und sogar von Sprint zu Sprint.

Weitere mögliche Backlog-Einträge

Wenn wir Kunden beim Erstellen ihres Produkt-Backlogs coachen, ermutigen wir sie, die Art des Backlog-Eintrags zu identifizieren (siehe [Abbildung 3.4](#) weiter vorne in diesem Kapitel).

Wir verwenden vier Kategorien, um den Typ des fertigzustellenden Eintrags anzugeben. Alle diese Elemente können nach dem Ermessen des Produkteigentümers verwendet werden (oder nicht):

- ✓ **Anforderung:** Grundlegende Geschäftsanforderungen machen den größten Teil des Produkt-Backlogs aus. Eine Anforderung erklärt, warum ein Feature entwickelt wird und was zu tun ist.
- ✓ **Wartung:** Designverbesserungen für den Code, wie beispielsweise das grundlegende Überdenken einer Login-Klasse, die taktisch nicht mehr weiter verbessert werden kann. Wartungspositionen reduzieren die technischen Altlasten des Projekts.
- ✓ **Verbesserungen:** Basierend auf den Ergebnissen der Sprint-Retrospektive, was im Prozess anders gemacht werden kann, wie zum Beispiel die Erweiterung der automatisierten Testfähigkeiten.
- ✓ **Overhead:** Unternehmen haben oft Overhead-Positionen, die sie gerne sehen möchten, wie etwa eine Abhängigkeitenmatrix für die Anforderungen, gleichzeitig wird solch eine Anforderung häufig nicht berücksichtigt. Die Unternehmen gehen dann davon aus, dass diese Matrix nicht kostet, aber

auch sie kostet Zeit und Geld. Wir wollen diese Kosten gerne offenkundig machen.

Häufig verwendete Produkt-Backlog-Praktiken

Agile Techniken werden von Tausenden von Unternehmen und Organisationen auf der ganzen Welt praktiziert. Als natürliche Folge gibt es viele gängige agile Praktiken. Einige dieser Praktiken verwenden wir im Rahmen unserer Beratungstätigkeiten, andere nicht. Wir empfehlen Ihnen, das Gleiche zu tun. Es gibt keine universellen optimalen Verfahren.

User Stories

User Stories werden verwendet, um Kundenanforderungen zu erfassen. User Story ist ein gängiger Begriff für Anforderungen, die klein genug sind, um in einen Sprint aufgenommen und in Aufgaben zerlegt zu werden. User Stories sind eine Aktion von Wert, die ein Benutzer erreichen wird, wie zum Beispiel »Als Käufer möchte ich in der Lage sein, einen Produkt-Barcode mit meinem Handy zu scannen, damit ich den niedrigsten Preis des gleichen Produkts in mehreren Geschäften vergleichen und finden kann.«

In einen Sprint können auch mehrere User Stories aufgenommen werden; jede besitzt zu diesem Zeitpunkt höchste Priorität. Im Durchschnitt haben Sie sechs bis zehn User Stories pro Sprint. Daher haben Sie am Ende jedes Sprints, wenn sich das Entwicklungsteam konzentriert und eine Anforderung nach der anderen erfüllt (ein Prozess, der *Swarming* genannt und in [Kapitel 6](#) beschrieben wird), immer etwas vorzuweisen.

Wir verwenden Karteikarten im Format DIN A6, um User Stories zu erstellen. Da wir das Modell *Card – Conversation – Confirmation* verwenden, sind die Anforderungen sehr klar und deutlich, bevor die Entwicklung beginnt. User Stories sind nicht der einzige Weg, um zu beschreiben, was getan werden muss,

jedoch sind sie unser Erfahrung nach unglaublich effektiv.
[Abbildung 3.5](#) zeigt beispielhaft User Stories als Karteikarten.

Das Diagramm zeigt eine User Story Karteikarte in zwei Ansichten: Vorderseite und Rückseite.

Vorderseite:

- ID # _____
- PLATINUM | EDGE
- Titel _____
- Als <Benutzer> _____
- Will ich <Aktion> _____
- damit <Nutzen> _____
- Wert _____ Autor _____ Schätzung _____

Rückseite:

- Card
- Conversation
- Confirmation
- Wenn ich dies tue: _____
- Passiert das: _____

Abbildung 3.5: Ein Format zum Erstellen von User Stories



Verwenden Sie in Ihren User Stories keinen Fachjargon. Schreiben Sie sie mit Blick auf den Benutzer. Sorgen Sie dafür, dass sie direkt, kundenorientiert und einfach sind.



Im Modell *Card – Conversation – Confirmation* verwendet der Product Owner die Karteikarte als Erinnerung an die Anforderung. Eine Erinnerung daran, dass ein Gespräch stattfinden muss, um die Anforderung auszuarbeiten, zu präzisieren und zu klären. Dieses Modell ermöglicht die ungemein wertvolle Aktivität, sich mit den Mitgliedern des Entwicklungsteams zu unterhalten und ihre Fragen zu beantworten. Dieser Vorgang wird durch die explizite Beschreibung dessen, was als Erfolg gewertet wird, unterstützt. Diese Beschreibung befindet sich auf der Rückseite der Karte. Früher erstellten wir, schon bevor wir ein Entwicklungsteam hatten, umfangreiche Dokumente und versuchten dabei, alle Fragen zu beantworten, die dieses Team möglicherweise stellen könnte – ein Unterfangen, bei dem wir immer scheiterten.

Manchmal ist es am einfachsten, mit Personas zu arbeiten: fiktive Charaktere, die die Zielqualitäten Ihrer Kunden vereinen. Eine Persona könnte ein 35-jähriger alleinstehender Mann sein, der in Portland, Oregon, lebt und nach seiner zukünftigen Lebensgefährtin sucht. Verwenden Sie eine Persona, um sicherzustellen, dass Ihr Projekt die Bedürfnisse Ihres Zielkunden erfüllt.

Eine Beschreibung in Form einer User Story ist einfach, aber fokussiert und beschreibt klar den Benutzer, die Aktion und den Nutzen. Geben Sie auf jeder Karte die folgenden Zeilen ein:

- ✓ Eine ID-Nummer, die die gleiche wie im Produkt-Backlog ist und die vom Product Owner vergeben wird, wenn er einen Eintrag in das Produkt-Backlog aufnimmt.



Halten Sie Ihre Produkt-ID-Nummern einfach. Sie müssen nicht in der Lage sein, jeden Eintrag bis zu seiner genauen Herkunft zurückzuverfolgen; verwenden Sie einfache numerische Bezeichnungen, etwa 123 und nicht 10.8.A.14. Sie minimieren den Arbeitsaufwand, indem Sie Ihr Projekt von allem befreien, was Zeit vergeudet, sodass Sie sich auf das Wesentliche konzentrieren können.

- ✓ Einen kurzen Titel
- ✓ Eine Beschreibung des Benutzers
- ✓ Was der Anwender erreichen will
- ✓ Der Grund, warum der Benutzer die Aufgabe erfüllen will

Entscheidend für die User Story ist, was auf der Rückseite der Karte steht – die Akzeptanzkriterien. So wissen Sie, dass die User Story erfolgreich umgesetzt wurde. Verwenden Sie diese Formulierung: »Wenn ich dies tue ... passiert das.«

[Abbildung 3.6](#) zeigt ein Beispiel für eine fertige Karte.

ID #	123	PLATINUM EDGE
Titel Als Einkäufer will ich ein Bestellkennzeichen ergänzen können, damit ich nach Gruppen von Bestellungen mit dem gleichen Kennzeichen suchen kann.		
Hoch		5 Punkte
Wert	Autor	Schätzung

Vorderseite

Wenn ich dies tue:	passiert das:
auf Kennzeichen hinzufügen klicke	sehe ich das Dialogfeld Kennzeichen hinzufügen

Rückseite

Abbildung 3.6: Eine fertige Karteikarte mit einer User Story



Jede User Story wird aus der Ich-Perspektive formuliert («Wenn *ich* das tue, bekomme *ich* das»). Die Verwendung der ersten Person versetzt den Autor in die Lage des Benutzers oder Kunden.

Die Person, die die User Story schreibt, gibt sie dem Product Owner, der diese mit dem Entwicklungsteam teilt. Häufig nimmt auch der Stakeholder an der Gesprächsphase teil und setzt sich mit dem Product Owner und dem Entwicklungsteam zusammen, um jede Karte zu besprechen. Dieser Akt der direkten Kommunikation ist entscheidend für ein gründliches Verständnis der Aufgaben, die notwendig sind, um das gewünschte Ergebnis zu erreichen. Dadurch wird die Kommunikation klarer und die Fehlerquote geringer, die Projektqualität und die Liefergeschwindigkeit steigen.

INVEST

Bill Wake, ein früher Influencer in der eXtreme Programming(XP)-Bewegung, beschreibt in seinem Blog das INVEST-Modell für User Stories. INVEST ist ein Akronym für die Merkmale, die gute User Stories erfüllen sollten:

- ✓ **Independent/unabhängig** (so gut wie möglich): Die Story benötigt für ihre Implementierung und dafür, dass der Benutzer mit ihr interagiert, keine anderen Stories.
- ✓ **Negotiable/verhandelbar**: Der Product Owner und das Entwicklungsteam müssen die Nuancen und Details der Geschichte diskutieren und erweitern. Der Nutzen einer User Story liegt im Gespräch über sie, nicht in der Karte.
- ✓ **Valueable/wertvoll**: Die Story zeigt dem Kunden den Produktwert, nicht die technischen Schritte (Aufgaben), die der Entwickler verwendet, um die Story zu ermöglichen.
- ✓ **Estimable/schätzbar**: Die Story ist so ausreichend präzisiert, dass die Entwickler den Aufwand für die Erstellung der Funktionalität schätzen können.
- ✓ **Small/klein**: Kleinere Stories lassen sich einfacher schätzen. Eine Faustregel lautet, sechs bis zehn User Stories in jeden Sprint aufzunehmen, sodass eine Story während des Sprints von einer oder zwei Personen fertiggestellt werden kann.
- ✓ **Testable/prüfbar**: Die Story muss testbar sein, damit das Entwicklungsteam weiß, dass es die Story korrekt implementiert hat. Kriterien wie »Es muss schnell sein« oder »Es muss intuitiv sein« sind nicht testbar. Kriterien wie »Ich brauche von der Anwendung in weniger als einer Sekunde eine Antwort« oder »Ich muss die Aktion mit drei oder weniger Klicks abschließen« sind hingegen testbar.

Weitere Verfeinerung

Der Arbeitsaufwand für das Entwicklungsteam bestimmt, ob eine Anforderung weiter aufgeschlüsselt werden muss, um in die Produkt-Roadmap, den Release-Plan oder den Sprint zu gelangen (weitere Details finden Sie in [Kapitel 5](#)). Besonders am Anfang, wenn Sie lernen, wie Sie in Ihrem Scrum-Framework User Stories anwenden, werden Sie vielleicht feststellen, dass die von Ihnen erstellten User Stories zu groß für einen Sprint sind.

Obwohl die meisten User Stories aus mehreren Schritten bestehen, sind einige größer, als es praktisch ist; diese können (und sollten) in detailliertere Anforderungen unterteilt werden. Dieser Prozess ist Teil der Entdeckungsreise und einer der Vorteile beim Einsatz von User Stories.

Es kann sein, dass Sie während der Diskussion, in der die User Stories ausgearbeitet werden, feststellen, dass zusätzliche Anforderungen in das Produkt-Backlog aufgenommen werden müssen. Ausgezeichnet! Diese Erkenntnisse helfen Ihnen, die Bedürfnisse des Kunden und/oder Anwenders besser zu verstehen.

Kapitel 4

Talent und Timing

IN DIESEM KAPITEL

Das Scrum-Entwicklungsteam strukturieren

Das Produkt-Backlog schätzen

Die Definition of Done

Vorteile von Schätzverfahren nutzen

Talent gewinnt Spiele, aber Teamwork und Intelligenz gewinnen Meisterschaften.

Michael Jordan

Das Entwicklungsteam steht im Mittelpunkt von Scrum-Projekten. Das Hauptaugenmerk von Product Owner und Scrum Master liegt darauf, dass das Umfeld für das Entwicklungsteam so optimal wie möglich ist, um maximale Produktivität zu erreichen. In diesem Kapitel erfahren Sie, wie ein Entwicklungsteam organisiert ist und somit am besten zum Scrum-Team beitragen kann.

Nachdem Sie Ihre Teamrollen, Ihre Vision und Ihre Roadmap festgelegt haben, besteht der nächste Schritt darin, den Arbeitsaufwand für jede Anforderung abzuschätzen. Das Entwicklungsteam beginnt mit der Schätzung des Aufwands, der für die übergeordneten Anforderungen erforderlich ist. Hierbei legen Sie einen Startpunkt fest, auf den Sie sich in der Zukunft beziehen. Sie legen die Beziehung zwischen Zähler (Gesamtzahl des Produkt-Backlogs) und Nenner (Durchschnittsgeschwindigkeit) fest, mit der Sie die geschätzte Anzahl der Sprints ermitteln, die zur Fertigstellung aller Einträge (oder eines Teils der Einträge) im Produkt-Backlog erforderlich

sind. Wir zeigen Ihnen, wie Sie schnell und präzise diese Schätzungen auf einer höheren Ebene erstellen.

Das Entwicklungsteam

In Hollywood werden Schauspieler und Sänger als »Talente« bezeichnet. Es sind diejenigen, die auf die Bühne gehen und den Job machen. Alle anderen unterstützen diesen Prozess, denn wenn das Talent erfolgreich ist, sind alle erfolgreich. Betrachten Sie das Entwicklungsteam als Talent. Dieses Talent zu erhalten, ist eines der Nebenprodukte einer guten Scrum-Implementierung.

Das Entwicklungsteam ist das Talent, das für die Erstellung und Entwicklung des eigentlichen Produkts verantwortlich ist. Das Team steuert das Wie und das Wie viel. Mit anderen Worten, es bestimmt, wie es die Anforderungen entwickelt und wie viele Anforderungen in einem Sprint fertiggestellt werden können. Es widmet sich jeweils einem Projekt, funktionsübergreifend, selbstorganisierend und selbstverwaltend.

Das Entwicklungsteam ist bewusst auf etwa sechs Mitglieder, plus oder minus drei, begrenzt. Die optimale Größe des Entwicklungsteams ist klein genug, um beweglich zu bleiben, und groß genug, um wichtige Arbeiten innerhalb eines Sprints zu erledigen. Diese Größe ermöglicht eine autarke und selbstorganisierende Gruppe mit unterschiedlichen Fähigkeiten, ist aber nicht zu groß und unhandlich. Denken Sie daran, dass für jedes neue Mitglied die Kommunikationslinien zwangsläufig zunehmen. Besteht das Entwicklungsteam aus mehr als neun Mitgliedern, ist Selbstorganisation nicht mehr möglich. Bei einem Wasserfallmodell wird dieses Thema mit einem Projektmanager maskiert, der für die Koordination der gesamten Kommunikation verantwortlich ist. Scrum verzichtet auf die Gemeinkosten von zu vielen Kommunikationswegen und setzt auch keinen Projektmanager ein.

Obwohl sechs Personen, plus oder minus drei, eine überschaubare und effiziente Größe darstellen, sind die

Hauptfaktoren die Autarkie und die Selbstständigkeit. Ist das Team in der Lage, die Anforderungen zu erarbeiten, entwerfen, entwickeln, testen, integrieren, dokumentieren und dann abnehmen zu lassen? Wurden Single Points of Failure eliminiert (dass heißt, sind mindestens zwei Personen in der Lage, eine bestimmte Fertigkeit auszuführen)? Besteht die Gesamtgröße nicht aus mehr als neun Personen? Wenn die Antwort auf alle drei Fragen ja ist, dann haben Sie ein gutes, funktionsübergreifendes Entwicklungsteam. Während ein Product Owner entscheidungsfreudig sein und ein Scrum Master Einfluss haben muss, muss ein Mitglied des Entwicklungsteams vielseitig, intellektuell neugierig und bereit sein zu teilen. Primadonnen sind nicht erlaubt. Bei den Mitgliedern des Entwicklungsteams wird das Feuer der Begeisterung beim Entwickeln und Erschaffen entzündet.

Die Einzigartigkeit von Scrum-Entwicklungsteams

In vielerlei Hinsicht ist das Scrum-Entwicklungsteam das Gegenteil eines traditionellen Teams. In Scrum entwickeln die Mitglieder des Entwicklungsteams funktionsübergreifende Fähigkeiten. Sie nehmen am Zielsetzungsprozess teil und haben als Team die volle Kontrolle darüber, wie sie die Entwicklung gestalten. Zusätzlich erhalten sie Anerkennung als Team und nicht als Einzelperson.

Ein motivierendes Umfeld schaffen

Daniel Pink befasst sich in seinem ausgezeichneten Buch *Drive: Was Sie wirklich motiviert* (Ecowin Verlag) mit Motivation. Hier räumt Pink mit dem Mythos auf, dass der einzige Weg, Menschen zu motivieren, darin besteht, immer mehr Geld zu verdienen. Überraschenderweise stellte Pink fest, dass je höher der monetäre Anreiz ist, desto schlechter die Ergebnisse sind. Geld ist ein Faktor, jedoch nur deswegen, um das Thema Geld vom Tisch zu fegen.

Während er weitergrub, fand Pink drei spezifische Faktoren, die die Umgebung für die besten Ergebnisse produzierten, ob zu Hause oder am Arbeitsplatz: Autonomie, Können und Zweck. Lassen Sie Ihre Mitarbeiter bestimmen, wie sie ihre Arbeit tun, geben Sie ihnen den Raum und die Zeit, aktuelle und neue Fähigkeiten zu meistern, und vermitteln Sie ihnen einen Sinn für das, was sie tun.

Das Scrum-Framework erlaubt es, das von Pink diskutierte Umfeld zu schaffen. Das Entwicklungsteam ist selbstorganisierend und autonom in seiner Arbeitsweise. Crossfunktionalität und die Entwicklung neuer Fähigkeiten sind von grundlegender Bedeutung. Durch die Vision und die direkte Kommunikation mit dem Product Owner und den Stakeholdern können die Mitglieder des Entwicklungsteams verstehen, woran sie arbeiten und warum.

Schauen Sie sich Pinks Buch oder seine TED-Vorträge an, wenn Sie weitere Informationen über seine Forschungsergebnisse sehen wollen. Seine Website finden Sie unter www.danpink.com.

Engagierte Teams und Cross-Funktionalität

Wie bereits erwähnt, widmen sich die Mitglieder des Entwicklungsteams jeweils einem Projekt. Lassen Sie die Entwicklungsteams nicht zwischen verschiedenen Projekten hin- und her wechseln, was als *Thrashing* bezeichnet wird. Sie möchten, dass sich die Mitglieder des Entwicklungsteams jeden Tag auf das Ziel des aktuellen Sprints konzentrieren. Unabhängig davon, ob das funktionale Management oder jemand anderes die Teammitglieder wählt, ist eine Vielfalt von Fähigkeiten gefragt. Sodass das Team über alle Fähigkeiten verfügt, die es für den Erfolg braucht. Damit kommen wir zum nächsten Punkt: der Crossfunktionalität.



Menschen sind nicht unbedingt von Anfang an crossfunktional; sie wachsen in diesen Zustand hinein. Beginnen Sie mit einem Team aus verschiedenen Talenten und bauen Sie dieses Team dann organisch zu einem Team auf, bei dem alle Mitglieder crossfunktional arbeiten. Im Idealfall sollte jeder Entwickler alles können. Das ist nicht immer möglich, aber man möchte zumindest, dass jeder Entwickler mehr als eine Sache tun kann und dass es für jede Fähigkeit mehr als eine Person gibt. Crossfunktional zu werden, ist ein Prozess, und dieser verhindert Engpässe bei der Ablieferung der Arbeitsergebnisse.

Gleichgültig, in welcher Art Unternehmen oder Organisation Sie tätig sind, die Fakten bleiben gleich: Menschen gehen in den Urlaub, sie werden krank, sie übernehmen neue Rollen und Jobs. An einem Tag stehen sie neben Ihnen und am nächsten Tag sind sie vielleicht woanders. Wenn in traditionellen Projekten ein wichtiges Mitglied des Entwicklungsteams in den Urlaub fährt, geht auch das Projekt in den Urlaub. Es ergeben sich zwangsweise Verzögerungen, während Sie auf die Rückkehr dieser Person warten oder (im Falle von Fluktuation) bis Sie eine andere Person einstellen und mobilisieren.

Streben Sie daher an, dass sich Ihr Entwicklungsteam durch Crossfunktionalität auszeichnet. Auf diese Weise eliminieren Sie diesen Single Point of Failure. Wenn ein Mitglied des Entwicklungsteams an der Grippe leidet oder intensiv an einer anderen Aufgabe arbeitet, kann jemand anderes seinen Platz einnehmen und seine Arbeit erledigen. Crossfunktionalität bietet außerdem diese Vorteile:

- ✓ Sie ermöglicht vielfältige Anregungen zur Entwicklung, um so optimale Lösungen zu erhalten.
- ✓ Sie unterstützt die Entwicklung in Paaren (die später in diesem Abschnitt beschrieben wird), um eine höhere Qualität zu gewährleisten.

- ✓ Sie ist einer der besten Wege, um Ihre primären Fähigkeiten zu verbessern. Das Erlernen einer verwandten Fertigkeit erlaubt Ihnen, Ihre primäre Fertigkeit aus anderen Perspektiven zu betrachten.
- ✓ Sie erlaubt Menschen, an unterschiedliche Dingen zu arbeiten, wodurch die Arbeit weiter interessant bleibt.

Es gibt mehrere Möglichkeiten, aus crossfunktionalen Individuen crossfunktionale Teams zu bilden:

- ✓ **Verwenden Sie keine Titel.** Fördern Sie ein gleichberechtigtes Spielfeld. Wir haben festgestellt, dass dies junge Entwickler dazu anregt, sich schneller auf den neuesten Stand zu bringen. Außerdem nehmen die Fähigkeiten auf Senior-Ebene zu, weil ältere Menschen nicht von jungen, hungrigen Talenten übertroffen werden wollen. Ein Mangel an Titeln betont die Fähigkeiten und nicht eine feste Hierarchie, wodurch die Entwicklung von Fähigkeiten gefördert wird. Der informelle Status existiert immer noch, aber er basiert jetzt auf Fähigkeiten.
- ✓ **Verwenden Sie die Paarprogrammierung.** Häufig verwendet in Software-Entwicklungspraktiken wie eXtreme Programming, kann die Paarprogrammierung von einem Entwicklungsteam verwendet werden, um jegliche Art von Produkt zu entwickeln. Zwei Entwickler arbeiten gemeinsam an der gleichen Funktionalität. Entwickler A entwickelt taktisch (zum Beispiel einen Code erstellen), während Entwickler B strategisch über die Funktionalität (Skalierbarkeit, Erweiterbarkeit, Risiken etc.) nachdenken kann. Sie wechseln diese Rollen im Laufe des Tages. Da diese Entwickler so eng zusammenarbeiten, können sie schnell Fehler erkennen. Nach unserer Erfahrung fokussieren sich die Entwickler bei der Paarprogrammierung mehr auf ihre Aufgabe und machen weniger Fehler, was im Ergebnis zu einer insgesamt schnelleren Entwicklung führt.
- ✓ **Verwenden Sie Shadowing.** Wieder arbeiten zwei Entwickler zusammen, aber in diesem Fall macht nur einer die Arbeit,

während der andere zuschaut und lernt.

Auch das Shadowing erhöht die Produktqualität. Denken Sie daran, dass Sichtbarkeit und Leistung korrelieren: Wenn Sie die Sichtbarkeit verbessern, dann steigern Sie gleichzeitig die Performance. Der arbeitende Entwickler will keine faule Abkürzung vor dem Lernenden nehmen, und der Lernende wird diese klugen »dummen Fragen« stellen. Wenn man etwas erklärt, verbessert sich das eigene Wissen darüber. Und wenn man etwas in Worte fasst, wird ein anderer Gehirnbereich genutzt und die Funktionsweise verbessert. Schließlich wird durch Lehren und Erklären die Eigenverantwortung gestärkt.



Der Hawthorne-Effekt (oder Beobachtungseffekt) basiert auf Studien, die zeigen, dass die Produktivität eines Arbeiters steigt, wenn er beobachtet wird. Der Effekt ist nach Hawthorne Works, einer Elektrofirma bei Chicago, benannt, in der die ersten Experimente stattfanden.



Quietscheentchen-Debugging ist ein Begriff, der ein interessantes Phänomen beschreibt, das häufig in der Softwareentwicklung verwendet wird. Programmierer werden aufgefordert, ihr Kodierungsproblem laut und detailliert einem Quietscheentchen zu erklären. In den meisten Fällen fällt dem Programmierer noch bevor er das Problem überhaupt erklärt hat, die Lösung ein. Das gleiche Phänomen tritt auf, wenn ein Freund mit einem Problem zu Ihnen kommt. Allein der Prozess des Aussprechens des Problems stimuliert einen anderen Gehirnbereich, sodass die Antworten sich fast von selbst ergeben und quasi auf der Hand liegen.

Paarprogrammierung

Während die Paarprogrammierung auf den ersten Blick teurer erscheint – zwei Entwickler werden gleichzeitig mit der gleichen Aufgabe beschäftigt – reduziert sie die Kosten, wenn

man sie ganzheitlich betrachtet. Die Entwicklungskosten mögen zwar zunehmen, jedoch nehmen die Kosten für die Qualitätssicherung drastisch ab.

Es ist exponentiell billiger, Defekte zu vermeiden, als diese später aus einem sich bereits im Einsatz befindlichen System zu entfernen, wie es die folgenden Fälle verdeutlichen:

- ✓ Wenn Sie den Fehler ganz schnell erkennen, sind die Kosten für seine Behebung minimal.
- ✓ Wenn der Fehler nach dem Commit des Codes gefunden wird, kostet es 6,5-mal mehr, sich in den Code einzuarbeiten und dann den Fehler zu finden und zu beheben.
- ✓ Wenn der Fehler bei den Tests durchgerutscht ist, in die Produktionsumgebung gelangt und schließlich einige Tage oder Wochen später identifiziert wird, ist die Mobilisierungszeit viel länger, es dauert länger, den Fehler zu identifizieren, und die Kosten liegen etwa 15-mal höher.
- ✓ Wenn derselbe Fehler auf den Markt kommt und von einer Gruppe korrigiert werden muss, die die Anwendung nie entwickelt hat – wie zum Beispiel der Supportabteilung –, können die Kosten bis zu 100-mal höher liegen.

Für jeden Defekt muss das Entwicklungsteam die Ursache des Fehlers finden, den Code entwickeln, um ihn zu korrigieren, und hoffen, dass die Korrektur für diesen speziellen Fehler keinen weiteren Fehler verursacht. Denken Sie daran, dass für jeweils zwei Fehler, die behoben wurden, ein neuer Fehler erzeugt wird.

Selbstorganisierend und selbstverwaltend

Das Schlüsselwort ist Eigentum. Selbstorganisierende und selbstverwaltete Teams entwickeln Eigenverantwortung für das,

was sie tun. Bei einem Scrum-Entwicklungsteam ist die Eigenverantwortung des gesamten Teams Teil dessen, was zu Effizienz und Erfolg führt.

Manchmal werden wir von Kunden gefragt, wie wir sicherstellen können, dass ihr Entwicklungsteam 100 Prozent gibt. Als Antwort fragen wir die Kunden, wie sie uns versichern können, dass eine professionelle Sportmannschaft 100 Prozent geben wird. Die Antwort ist, dass das Team 100 Prozent gibt, weil es sonst verlieren würde. Die Sichtbarkeit und die Anerkennung für ihre harte Arbeit steigert den Antrieb und die Motivation.



Sichtbarkeit und Leistung stehen in direktem Zusammenhang. Steigern Sie die Leistung, indem Sie die Sichtbarkeit erhöhen.

Eine Technik, die manchmal verwendet wird, besteht darin, zwei (oder mehr) Entwicklungsteams einzusetzen. Synchronisieren Sie deren Sprints, sodass die Sprint-Reviews zur gleichen Zeit am selben Tag stattfinden. Dann laden Sie eine Führungskraft ein, für ein paar Minuten zufällig zu beiden Sprint-Reviews zu kommen und lassen Sie sie mindestens eine Frage stellen, bevor Sie gehen. Jedes Entwicklungsteam weiß, dass seine Leistung für Führungskräfte sichtbar sein wird, und alle Teams wollen gut aussehen. Historisch gesehen haben sie vielleicht keine Anerkennung für ihre Arbeit erhalten. Jetzt können sie ein Produkt herstellen, auf das sie stolz sein können. Sie sind auf der Bühne und bekommen den ganzen Ruhm. Das ist enorm motivierend und steigert das Engagement und den Einsatz.

Die Eigenverantwortung und damit die Verantwortlichkeit in einem Scrum-Entwicklungsteam lassen sich wie folgt vergrößern:

- ✓ Das Entwicklungsteam ist direkt für die von ihm erstellten Ergebnisse verantwortlich. Das ist nicht immer einfach für sie, denn Sichtbarkeit bringt intrinsischen Leistungsdruck mit sich, aber diese Sichtbarkeit führt auch zu Verantwortung.

- ✓ Crossfunktionalität schafft gemeinsame Verantwortlichkeit, da keine Konkurrenz zwischen »meine Arbeit« und »deine Arbeit« existiert. Alles ist unsere gemeinsame Arbeit.
- ✓ Da das gesamte Scrum-Team zur Verantwortung gezogen wird, wird die individuelle Leistung gesteigert. Alle gewinnen als Team und alle tragen zum Erfolg eines jeden Sprints bei.
- ✓ Das Entwicklungsteam ist aktiv an der Erstellung der Sprint-Ziele und während des Sprint-Reviews an der Demonstration der Funktionsfähigkeit beteiligt.
- ✓ Das Entwicklungsteam ist täglich für den taktischen Statusbericht verantwortlich. In weniger als einer Minute Administration pro Tag (siehe die Burndown-Diagramme in [Kapitel 5](#)) erhält die Organisation ein Niveau taktischer Statusberichte, das sie noch nie zuvor hatte.



Entwicklungsteams arbeiten am besten, wenn sie stabil sind. Füttern Sie sie mit Projekten und geben Sie ihnen das, was sie brauchen, um bestmöglich arbeiten zu können. Jedes Mal, wenn Sie die Mitglieder Ihres Teams wechseln, benötigt dieses Zeit, um sich wieder zu stabilisieren. Schützen Sie Ihr Entwicklungsteam, um eine gute Dynamik zu fördern.



Die kognitive Konsistenztheorie beschreibt die Tendenz des Menschen, Informationen, Überzeugungen und Stimuli zu suchen, die mit aktuellen Überzeugungen und Einstellungen übereinstimmen. Wenn die Mitglieder des Entwicklungsteams eine Stimme haben und wenn sie sich einbringen und kontrollieren können, bemühen sie sich stärker, ihre arbeitsbezogenen Ziele zu erreichen. Sie versuchen, Konsistenz zwischen dem Eigentum, das sie geschaffen haben, und ihrem zukünftigen Output zu finden.

Kollokation oder das, was mir am nächsten ist

Viele Menschen haben vergessen, wie es ist, in einem Team zu arbeiten, das sich an demselben Ort befindet (Kollokation), und die damit verbundene erhöhte Produktion zu erleben. Die meisten Menschen schätzen das Konzept, verstehen aber die zugrunde liegenden Prinzipien nicht. Hier sind ein paar Vorteile eines Teams, das an demselben Ort arbeitet:

- ✓ Erhöhte Geschwindigkeit und Effektivität der persönlichen Kommunikation, insbesondere durch Körpersprache, Stimmtonalität, Mimik und so weiter.

Der Wert von persönlichen Gesprächen sollte nicht unterschätzt werden. Albert Mehrabian, Doktor der Philosophie und emeritierter Professor für Psychologie an der University of California-Los Angeles, fand folgendes heraus:

- 55 Prozent der Bedeutung wird durch Körpersprache und Mimik vermittelt.
- 38 Prozent der Bedeutung ist *paralinguistisch* (vermittelt durch die Art, wie wir sprechen).
- 7 Prozent der Bedeutung wird in den tatsächlich gesprochenen Worten vermittelt.

Diese Statistiken allein sind schon ein hervorragendes Argument für die Kollokation Ihres Entwicklungsteams.

- ✓ Leichtes Handling von einfachen Tools für die Planung und Kommunikation, etwa Whiteboards und Haftnotizen.
- ✓ Sofortige Klärung von Fragen wird vereinfacht.
- ✓ Verstehen, woran andere Mitglieder arbeiten.
- ✓ Einfache Unterstützung anderer Teammitglieder bei ihren Aufgaben.
- ✓ Kosteneinsparungen durch kürzere Verzögerungszeiten und weniger Missverständnisse, die zu Fehlern oder unnützer Arbeit führen.



An einem gemeinsamen Standort haben Ihr Product Owner und Ihr Entwicklungsteam den ganzen Tag über Zugang zueinander. Alle möglichen Banalitäten werden ausgetauscht und es ist verlockend zu glauben, dass dies verschwenderisch sein könnte. Wir haben jedoch festgestellt, dass während dieses Austausches die wirklich gute Arbeit getan wird. Kleinigkeiten sind nicht wirklich klein; sie sind Unterscheidungsmerkmale und sie sind die Dinge, die wichtig sind. Qualität braucht Input und Input braucht Zugang. Wenn der direkte Zugang zueinander einfach ist, sind großartige Dinge möglich.



Manchmal ist Outsourcing für Ihr Unternehmen oder Ihre Organisation die einzig praktikable Lösung. Sollte dies der Fall ist, dann machen Sie es richtig. Bringen Sie das gesamte Scrum-Team gemeinsam an einem anderen Ort unter. Ihr Entwicklungsteam benötigt einen Product Owner, der verfügbar ist; entsenden Sie ihn daher an den Outsourcing-Standort, um direkt mit dem Remote-Scrum-Team zu arbeiten. Oder sorgen Sie für einen lokalen Product Owner, auch wenn er als Product Owner Agent starten muss (siehe [Kapitel 2](#)). Die Zunahme an Qualität und Effizienz übertrifft bei weitem die Kosten für einen Product Owner.

Warum Entwicklungsteams Scrum lieben

Entwicklungsteams lieben Scrum aus vielen Gründen, nicht zuletzt, weil es ein selbstbestimmtes und motivierendes Umfeld bietet. Wenn Sie den Menschen erlauben, ihren Beitrag zu leisten, werden sie Teil des Prozesses und übernehmen die Verantwortung für die Ergebnisse. Dies führt zu einem Schneeballeffekt:

- ✓ Der Erfolg ist durch das Sprintziel, die Definition of Done und die für jede Anforderung aufgelisteten Erfolgskriterien klar definiert.

- ✓ Die Mitglieder beteiligen sich an der Erstellung des Sprintziels, was zu höherer Eigenverantwortung führt.
- ✓ Die Akzeptanzkriterien der Anforderungen sind deutlich. Das Team weiß genau, was Erfolg bedeutet.
- ✓ Im Arbeitsumfeld zählt nur das Ergebnis. Arbeiten Sie, wie Sie wollen; bringen Sie einfach Ergebnisse.
- ✓ Direkte Kommunikation mit den Product Ownern ist möglich.
- ✓ Das Team bekommt seine Anerkennung in den Sprint-Reviews.
- ✓ Strukturierte Prozessverbesserung wird durch die Sprint-Retrospektiven erreicht.
- ✓ Systematisches Wissen wird durch Sprint-Retrospektiven aufgebaut.

Vorsprung durch Schätzung des Produkt-Backlogs

In der Phase der Produkt-Roadmap (siehe [Kapitel 1](#)) schätzt das Entwicklungsteam den Aufwand für das Projekt auf einer allgemeinen, hohen Ebene. Der praktische Wert dieses Schätzverfahrens kommt erst bei den Sprints zum Tragen (siehe [Kapitel 5](#)). Aber diese erste Schätzung legt eine Messlatte fest, aus der zukünftige Schätzungen berechnet werden können. Das Entwicklungsteam macht die Schätzung, da nur die Leute, die auch die Arbeit erledigen, den Aufwand dieser schätzen sollten.



Diese Backlog-Schätztechniken sind keine Anforderungen von Scrum. Es sind gängige Praktiken, die Scrum-Praktiker im Alltag als nützlich erachtet haben.

Die Produkt-Roadmap ist der Beginn des Produkt-Backlogs. Was auf Ihrer Roadmap steht, ist das, was Sie entwickeln werden. Wie nehmen Sie also all diese Einträge in Ihren Produkt-Backlog auf und wie können Sie den Arbeitsaufwand zumindest ungefähr schätzen? Je nach Situation werden einige gängige Verfahren angewendet. Bevor wir diese Schätzverfahren einzeln betrachten, ist es wichtig zu verstehen, was Sie erreichen wollen.

Ihre Definition of Done

Wenn Sie die Mitglieder eines Scrum-Teams fragen, was sie nach der Fertigstellung einer Anforderung erwarten, erhalten Sie so viele Antworten, wie es Teammitglieder gibt. Daher definieren Scrum-Teams, vor dem Projektstart, in Form einer Definition of Done, was »done«, also fertiggestellt, genau bedeutet. Bis sie sich auf diese Definition geeinigt haben, basieren die Schätzungen auf schlechten Daten.

Wie in [Kapitel 3](#) beschrieben, durchlaufen Sie für jede Anforderung innerhalb eines Sprints die folgenden Entwicklungsphasen:

- ✓ Ausarbeitung der Anforderung
- ✓ Entwurf
- ✓ Entwicklung
- ✓ Umfassende Tests
- ✓ Integration
- ✓ Dokumentation
- ✓ Abnahme

Diese Definition of Done muss spezifisch, ausgearbeitet und darauf fokussiert sein, was Sie tun müssen, um das angestrebte Qualitätsniveau in Ihrem Projekt zu erzielen. Überlegen Sie, in welcher Umgebung das Produkt oder die Dienstleistung funktionieren soll und auf welchem Integrationsniveau.

»Funktioniert in der Entwicklungsumgebung« ist ein Beispiel für eine schlechte, ungenaue Definition of Done.

Berücksichtigen Sie in Ihrer Definition of Done diese vier Faktoren:

- ✓ **Entwickelt:** Das Produkt wurde vollständig vom Entwicklungsteam entwickelt.
- ✓ **Getestet:** Das Entwicklungsteam hat das Produkt vollständig getestet, um sicherzustellen, dass es in der gewünschten Umgebung ohne Probleme funktioniert.
- ✓ **Integriert:** Das Produkt wurde vollständig in das Produkt als Ganzes und alle damit zusammenhängende Systeme integriert.
- ✓ **Dokumentiert:** Das Entwicklungsteam hat alle notwendigen Dokumentationen erstellt. Denken Sie nur daran, dass das Ziel aller agilen Dinge »gerade ausreichend« ist.



Nachdem Sie Ihre Definition of Done erstellt haben, schreiben Sie diese auf einen weißen Karton und hängen ihn an die Wand. Hierdurch haben das Entwicklungsteam und der Product Owner die Definition of Done immer vor Augen. Wir nennen es »permanent sichtbare Dokumentation«. Kein Deckblatt, kein Inhaltsverzeichnis – sondern die einfachste Form der Kommunikation, welche die Informationen am besten sichtbar macht.

Berücksichtigen Sie bei der Definition of Done nicht nur die Entwicklung, sondern auch die Testtiefe und die Art der Dokumentation, die Sie eventuell benötigen. Sie können sich überlegen, welche Tests die folgenden Prüfungen bestehen müssen:

- ✓ Unit-Test
- ✓ Funktionaler Test/Systemtest
- ✓ Leistungs-/Last-Test

- ✓ Sicherheitstest
- ✓ Benutzerakzeptanz

Überlegen Sie auch, welche Dokumentation Sie benötigen:

- ✓ Technische Dokumentation
- ✓ Benutzerdokumentation
- ✓ Wartungsdokumentation

Jeder dieser Test- und Dokumentationspunkte kann sich zwischen dem Sprint- und dem Release-Level unterscheiden, obwohl wir es vorziehen, dass die Definition of Done auf der Sprint-Ebene alles beinhaltet, was für das Release notwendig ist. Möglicherweise haben Sie auch organisatorisch spezifische Elemente, die Sie einbeziehen möchten. Es ist Ihre Entscheidung. Es geht darum, eine klare Definition of Done zu haben, die vom Scrum-Team definiert wird, damit alle daran arbeiten können.



Ein *Release* erfolgt dann, wenn eine Reihe von marktfähigen Features außerhalb des Scrum-Teams veröffentlicht wird. Dies kann mehrmals während eines Sprints, am Ende eines jeden Sprints oder nach einer Reihe von Sprints geschehen. Diese fertigen Anforderungen können im Markt veröffentlicht und so den Anwendern zur Verfügung gestellt werden. Sie können auch an interne oder externe Stakeholder weitergeleitet werden, damit sie praktisch eingesetzt werden und das Team so Feedback erhält.



Ein normaler Sprint beinhaltet die Entwicklung, das Testen, die Dokumentation und die Freigabe von Einträgen im Sprint-Backlog. Vor einem Produktrelease können andere Aktivitäten erforderlich sein (zum Beispiel Performance- und Lasttests), auf die das Entwicklungsteam in einem regulären Sprint keinen Zugriff hätte. Daher verwenden Scrum-Teams manchmal kurz vor dem Release einen Release-Sprint, damit diese zusätzlichen Tests durchgeführt werden können. Entscheidend ist, dass am Ende jedes Sprints die Anforderungen funktionieren und demonstriert werden können. Sie können im Release-Sprint das Produkt im Großen testen und optimieren, aber die Anforderungen müssen nach jedem Sprint funktionieren.

Gebräuchliche Schätztechniken

Die Aufwandsschätzung für die Entwicklung des Produkt-Backlogs ist ein kontinuierlicher Prozess. (In [Kapitel 3](#) besprechen wir das Produkt-Backlog-Refinement.) Zum Beispiel könnten Sie jeden Abend 30 Minuten lang um 17 Uhr Schätzungen durchführen, bevor die Teammitglieder nach Hause gehen. Auf diese Weise haben Sie am Ende der Woche umfangreiches Material erstellt und sind bereit für jeden Sprintstart. Manche Teams entwickeln nicht gerne am Freitagnachmittag und nutzen diese Zeit für die Ausarbeitung des Produkt-Backlogs.

Die Teams werden ihre Schätzungen wahrscheinlich auf drei Ebenen verfeinern, um die Anforderungen für die Ausführung auf Sprintniveau zu untergliedern. Abhängig von Ihrem Produkt können Sie mehr Ebenen verwenden. Die Ausarbeitung und Verfeinerung der Schätzungen erfolgt in der Regel in dieser Reihenfolge:

1. Produkt-Roadmap
2. Release-Planung
3. Sprint-Planung



Das Entwicklungsteam ist für die Abschätzung des Aufwands verantwortlich, der zur vollständigen Erstellung der Anforderungen notwendig ist. Der Scrum Master kann den Prozess erleichtern und der Product Owner kann Klarheit schaffen, aber die Entscheidung wird von den Entwicklern getroffen, die die eigentliche Arbeit machen.



Wir verwenden die relative Schätzung anstelle der präzisen (absoluten) Schätzung, weil sie in vielen Situationen viel praktikabler ist. Wenn Sie gebeten werden, aus dem Fenster zu schauen und zu sagen, wie hoch das benachbarte Gebäude ist, wie präzise wäre Ihre Antwort? Sehr präzise – es sind 16 Meter. Wie korrekt wäre Ihre Antwort? Nicht sehr. Warum? Weil Sie es, ehrlich gesagt, nicht wissen; Sie haben eine wilde Vermutung angestellt. Aber wenn wir Sie bitten, sich zwei benachbarte Gebäude anzusehen und uns mitzuteilen, welches größer ist, können wir garantieren, dass Sie uns die richtige Antwort geben. *Relative* Größenbestimmung ist ein effektiver Weg, um die Schwierigkeiten zu überwinden, die wir Menschen mit absoluten Schätzungen haben.

Fibonacci-Zahlen und Story Points

Die Fibonacci-Folge ist eine hervorragende Methode zur relativen Schätzung.

Mit ihr bekommt man eine Vorstellung davon, wenn etwas größer ist, wie viel größer es ist. Die letzten beiden Zahlen in der Folge werden addiert, um die nächste Zahl zu erzeugen. Fibonacci-Zahlen sehen so aus:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 und so weiter.

Je weiter die Nummerierung fortschreitet, desto größer wird der Abstand zwischen den Zahlen. Wir verwenden diese Technik, um die geringe Genauigkeit bei der Aufwandsschätzung von größeren Arbeitsabschnitten zu berücksichtigen.



Ein Story-Point ist die Fibonacci-Nummer, die einer individuellen Anforderung (das heißt einer User-Story) zugeordnet ist.



Leonardo Pisano Bigollo, auch bekannt als Fibonacci, lebte von etwa 1170 bis zirka 1250 in der Nähe von Pisa, Italien. Weitgehend als brillanter Mathematiker anerkannt, machte er die Fibonacci-Folge berühmt und initiierte die Verbreitung des indisch-arabischen Zahlensystems in Europa – heute die häufigste symbolische Darstellung von Zahlen in der Welt.

Die anfänglichen allgemeinen Anforderungen werden auf der Ebene der Produkt-Roadmap geschätzt:

- ✓ Bei Scrum-Teams, mit denen wir zusammenarbeiten, wissen die Entwicklungsteams, dass Anforderungen mit Fibonacci-Zahlenschätzungen von 1 bis 8 in einem Sprint untergebracht werden können. Diese Verfeinerung führt in der Regel zu einer User Story.
- ✓ Anforderungen mit Schätzungen von 13 bis 34 sind diejenigen, die Sie in ein Release einfließen lassen würden. Diese müssen aber weiter aufgeschlüsselt werden, bevor Sie sie in einen Sprint einfließen lassen würden. Auf dieser Ebene der Verfeinerung nennen wir sie Epics.
- ✓ Anforderungen von 55 bis 144 sind zu groß für ein Release, aber auf der Ebene der Produkt-Roadmap schätzbar. Diese Anforderungen spiegeln typischerweise Features wider.



Anforderungen, die größer als 144 sind, müssen aufgeschlüsselt werden, bevor das Entwicklungsteam den Anschein einer genauen Schätzung erwecken kann, sodass wir nicht über 144 hinaus schätzen. Diese Anforderungen können weiter gefasste Themen darstellen.

Unabhängig von der Fibonacci-Zahl werden nur die Karten mit der höchsten Priorität in Sprint-Level-Größen aufgeteilt (wir

empfehlen nicht mehr als eine 8). Wenn Sie also eine Anforderung hoher Priorität mit der Fibonacci-Nummer 21 haben, muss sie in kleinere Anforderungen zerlegt werden, bevor sie in einen Sprint aufgenommen werden kann.

Mit diesen festgelegten Größen können wir einige Techniken anwenden, um die Anforderungen zu schätzen:

- ✓ Wenn wir kürzere Anforderungslisten haben, beginnen wir mit dem Schätzpoker.
- ✓ Wenn wir Hunderte von Anforderungen haben, beginnen wir mit der Affinitätsschätzung (die weiter hinten in diesem Kapitel besprochen wird).

Im Schätzverfahren bei kleineren Projekten haben wir das Entwicklungsteam wie folgt beginnen lassen: Das Team setzt sich hin und hat den Stapel Karteikarten mit den Anforderungen vor sich. Dann bitten wir das Team, eine Anforderung auszuwählen, bei der alle zustimmen können, dass sie einen Aufwand von 5 hat. Hierdurch entsteht ein Bezugspunkt.

Wir lassen die Teammitglieder dann eine andere Karte auswählen und fragen sie, basierend auf der ersten Karte mit dem Aufwand 5, welchen Aufwand diese Anforderung benötigt. Wenn der Aufwand größer als 5 ist – ist es eine 8, eine 13 oder eine 21? Dieser Prozess setzt sich fort, bis einige repräsentative Größen festgelegt sind.

Schätzpoker

Eine beliebte Möglichkeit, die Anforderungen zu schätzen, ist eine Pokervariante. Sie benötigen ein Kartenspiel mit Schätzpokerkarten, wie in [Abbildung 4.1](#) gezeigt. (Sie finden es auf unserer Website unter

<https://platinumedge.com/store/estimation-poker-cards>). Sie können auch unsere Poker-App für iPhone und/oder Android herunterladen, indem Sie im App Store Ihres Geräts nach *Platinum Edge Estimation Poker* suchen oder mit Karteikarten und einem Filzstift Ihr eigenes Kartenspiel erstellen.

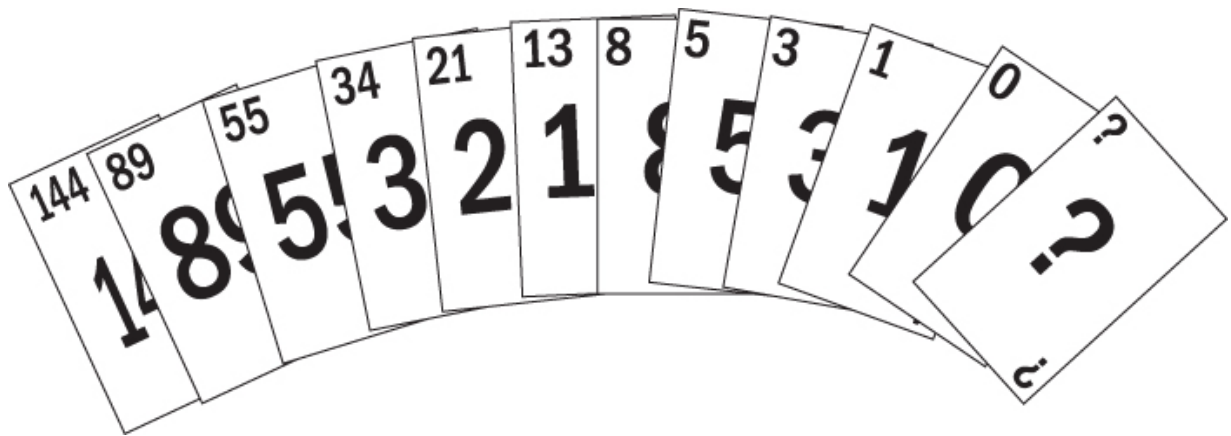


Abbildung 4.1: Schätzpokerkarten zur Aufwandsschätzung einer Anforderung

Da nur das Entwicklungsteam entscheidet, wie groß der Aufwand ist, um eine Anforderung zu entwickeln, spielt nur das Entwicklungsteam Schätzpoker. Der Scrum Master unterstützt das Team und der Product Owner liest die Anforderungen und liefert Anforderungsdetails, aber keiner der beiden gibt Schätzungen ab. Es läuft so ab:

1. Der Product Owner liest dem Entwicklungsteam eine bestimmte Anforderung inklusive Abnahmekriterien vor.
2. Das Entwicklungsteam stellt alle Fragen und erhält alle notwendigen Klarstellungen.
3. Jedes Mitglied des Entwicklungsteams wählt aus seinem Kartenspiel eine Karte aus, die seine Einschätzung des Schwierigkeitsgrades der Anforderung wiedergibt.



Es geht immer darum, den Aufwand für die komplette Fertigstellung der Definition of Done zu schätzen, nicht nur den, um den Code zu schreiben.

Die Teammitglieder zeigen niemandem ihre Karten, damit niemand anderes beeinflusst wird.

4. Nachdem jeder eine Zahl ausgewählt hat, decken die Teammitglieder gleichzeitig ihre Karten auf.

Wenn alle die gleiche Schätzung haben, bleibt nichts mehr zu diskutieren. Weisen Sie der Anforderung diese Schätzung zu und machen Sie mit der nächsten Anforderung weiter.

Wenn es Unterschiede in den Schätzungen gibt, werden diejenigen mit den höchsten und niedrigsten Schätzungen gebeten, dies zu erklären. Weitere Verdeutlichung und Klarstellungen nimmt der Product Owner bei Bedarf vor.

5. Mit dem neuen Wissen ausgestattet wählt jeder eine neue Karte für diese Anforderung aus; die Schritte 3 und 4 werden wiederholt.

Wir spielen normalerweise für jede Anforderung bis zu drei Runden Schätzpoker, um die Kernannahmen auf den Tisch zu bekommen und zu klären. An diesem Punkt haben wir normalerweise die Schätzungen in einem engeren Cluster von Zahlen.

Wenn sich alle Entwickler nach drei Runden auf eine einzige Zahl einigen, sind Sie bereit, mit der nächsten Anforderung weiterzumachen. Aber nicht immer sind sich alle Entwickler nach drei Runden über eine einzige Zahl einig. An dieser Stelle verwenden wir eine Technik der Konsensbildung, die Fist-of-Five-Methode genannt wird.

Fist-of-Five-Methode

Fist of Five ist eine schnelle und effiziente Methode, um einen Konsens zu erzielen. Sie kann sowohl für sich allein als auch als Ergänzung zum Schätzpoker verwendet werden. Das Ziel von Fist of Five besteht darin, schnell eine Schätzung abzugeben, die alle Teammitglieder unterstützen können (siehe [Abbildung 4.2](#)).



5 = Fantastische Idee!

4 = Gute Idee.

3 = Ja, ich kann die Idee unterstützen.

2 = Ich habe Vorbehalte, lasst uns diskutieren.

1 = Ich bin gegen diese Idee. Nicht weitermachen.

Abbildung 4.2: Fist of Five ist ein effizienter Weg, um in vielen Situationen einen Konsens zu finden.

Möglicherweise haben beim Schätzpoker einige Teammitglieder einer Anforderung eine 5 gegeben, andere eine 8.

Es beginnt damit, dass der Scrum Master die betreffende Anforderungskarte hochhält und zum Beispiel sagt: »Wie sehr könnten Sie einer 8 zustimmen?« Jedes Mitglied des Entwicklungsteams hält die Anzahl der Finger hoch, die seinem Grad der Zustimmung entspricht. Wenn alle drei, vier oder fünf Finger hochhalten, ist das Thema erledigt.

Wenn einige Entwickler einen oder zwei Finger hochhalten, werden die Ausreißer wie beim Schätzpoker gebeten, dies zu erläutern, und wenn nötig, werden weitere Informationen gesammelt. Danach folgt eine weitere Runde Fist of Five. Machen Sie so lange weiter, bis alle Teammitglieder der Aufwandszahl mindestens eine 3 geben können (was bedeutet: »Ich finde es zwar nicht toll, aber ich kann es unterstützen«).

Nachdem Sie Fist of Five beendet und die Anforderungen geschätzt haben, sind Sie bereit für die Release- oder Sprint-Planung. Diese werden in [Kapitel 5](#) behandelt.

Affinitätsschätzung

Schätzpoker und Fist of Five sind effektive Methoden zur Konsensfindung in kleinen Projekten. Was aber, wenn sich in Ihrem Produkt-Backlog mehrere hundert Anforderungen befinden? Es könnte Tage dauern, bis Sie mit dem Schätzen fertig sind. Hier kommt die Affinitätsschätzung ins Spiel.

Anstelle von Fibonacci-Zahlen verwenden Sie ein vertrautes Konzept, und zwar das der T-Shirt-Größen (XS, S, M, L und XL).

Bei der Affinitätsschätzung erstellen Sie zunächst mehrere Bereiche, die jeweils mit einer Größe markiert sind, und weisen dann jeder Anforderung eine der Größenkategorien zu. Es läuft so ab:

1. Verwenden Sie kleine Tische, um die Karten zu sortieren.

Beschriften Sie einen Tisch mit *Klärung nötig* und beschriften Sie weitere kleine Tische mit diesen Größenkategorien:

- Extra Small
- Small
- Medium
- Large
- Extra Large
- Epic (zu groß, um in den Sprint zu passen, unter der Voraussetzung, dass je Sprint maximal sechs bis zehn Anforderungen umgesetzt werden können; mehr dazu in [Kapitel 5](#))

2. Geben Sie Ihrem Entwicklungsteam für jede Größenkategorie 60 Sekunden Zeit, um eine Anforderung aus dem gesamten Anforderungsstapel auszuwählen und sie auf den entsprechenden Tisch zu legen.

Dadurch wird der *repräsentative Fixpunkt* für jede Größe festgelegt.

3. Jedes Mitglied des Entwicklungsteams schnappt sich einen Stapel Anforderungen.

4. Jedes Mitglied legt jede Karte seines Stapels auf den Tisch, der seiner Meinung nach die Größe dieser Anforderung, basierend auf der Größe des repräsentativen Fixpunkts dieser Größe, am besten widerspiegelt.

Wie Sie sehen können, sind diese wie T-Shirt-Größen. Jede »Größe« kann schließlich einer Fibonacci-Zahl zugeordnet werden:

- Extra-small entspricht 1.
- Small entspricht 2.
- Medium entspricht 3.
- Large entspricht 5.
- Extra-large entspricht 8.

Alles, was größer als 8 ist, muss untergliedert werden, bevor es in einen Sprint aufgenommen werden kann.

[Abbildung 4.3](#) zeigt, wie das Verhältnis zwischen T-Shirt-Größen und Fibonacci-Zahlen aussieht.

Größe	Story Points
Extra Small (XS)	1
Small (S)	2
Medium (M)	3
Large (L)	5
Extra Large (XL)	8

Abbildung 4.3: Die Affinitätsschätzung verwendet T-Shirt-Größen für Story-Größen und weist jeder T-Shirt-Größe eine entsprechende Fibonacci-Zahl zu.



Lassen Sie die Teammitglieder nicht zu lange bei ihrem Anforderungsstapel verweilen. Legen Sie ein Zeitfenster fest, damit sie innerhalb von zum Beispiel 20 Minuten 20 Karten abarbeiten können.



Zeitfenster oder *Timebox* ist ein Begriff, der sich auf die zugeteilte Zeit für ein Ereignis oder eine Aktivität bezieht. Wenn Ihre Sprints zwei Wochen dauern, beträgt das Zeitfenster zwei Wochen.

5. Lassen Sie die Mitglieder des Entwicklungsteams das sogenannte »Galerie«-Spiel spielen, bis sich alle Mitglieder auf die Größen für jede Anforderung geeinigt haben.
In der Galerie sehen sich die Teammitglieder alle Karten auf allen Tischen an und geben nur zu den Karten Feedback, die nicht auf dem richtigen Tisch zu liegen scheinen.
Wenn ein Teammitglied zum Beispiel eine Story von Small nach Medium verschieben möchte, überprüfen Sie, ob die Person, die sie ursprünglich dort platziert hat, nicht anderer Meinung ist. Wenn die Person nicht einverstanden ist, legen Sie diese Karte auf den Tisch *Klärung nötig*. Führen Sie zu diesem Zeitpunkt keine ausgiebigen Diskussionen.
6. Laden Sie den Product Owner ein, größere Unterschiede zu überprüfen:
 - Wenn der Product Owner auf dem mittleren Tisch eine Anforderung sieht, die er für eine kleine hält, verschwenden Sie keine Zeit mit Diskussionen. Das Entwicklungsteam entscheidet schlussendlich über die Größe der Anforderung und ein Unterschied von nur einer Größe ist es nicht wert, darüber zu diskutieren.
 - Falls das Entwicklungsteam und der Product Owner meinen, dass der Größenunterschied einer Anforderung größer als »ein Tisch« ist, dann legen Sie diese auf den Tisch *Klärung nötig*. Möglicherweise hat das Entwicklungsteam die Erklärung des Product Owners zu dieser Anforderung nicht richtig verstanden.
7. Spielen Sie mit den Karten, die sich auf den Tisch *Klärung nötig* befinden, Schätzpoker (siehe weiter vorn in diesem Kapitel).

In relativ kurzer Zeit konnten Sie den Aufwand für hunderte von Einzelanforderungen zuverlässig abschätzen. Sie sind bereit, Ihr erstes Release und/oder Ihren ersten Sprint zu planen.

Velocity (Geschwindigkeit)

Nachdem Sie die Fibonacci-Zahlen Ihren Anforderungen zugeordnet haben, können Sie mit den Story Points arbeiten.

[Kapitel 5](#) zeigt, wie man sowohl Releases als auch Sprints plant. Um Ihren ersten Sprint zu planen, wählt das Scrum-Team aus den Anforderungen, die zwischen 1 und 8 Story Points liegen, eine geringe Anzahl von Story Points aus, an denen es arbeiten und die es in diesem Sprint fertigstellen will. Am Ende des Sprints addiert das Scrum-Team die Story Points der Anforderungen, die komplett fertiggestellt wurden. Das Ergebnis können 15, 25 oder 35 Story Points sein. Dies ist die Geschwindigkeit des Entwicklungsteams für den ersten Sprint und der Startwert für ein Team, das bestimmt, wie viel es im nächsten Sprint erreichen kann.



Velocity ist die Summe der Story Points der Anforderungen, die Ihr Entwicklungsteam in einem einzelnen Sprint absolviert hat. Sie ist ein Post-Sprint-Fakt, der für die Extrapolation verwendet wird, und kein Prä-Sprint-Ziel.



Um die Velocity genau zu bestimmen, braucht man mehr als einen Sprint, damit man einen Durchschnitt bilden kann. Die Velocity folgt dem Gesetz der großen Zahlen: Je mehr Datenpunkte Sie haben, desto besser. Sie benötigen mindestens drei Datenpunkte, um eine optimistische, pessimistische und höchstwahrscheinliche Extrapolation darüber zu erstellen, wie viele Anforderungen während des Projekts umgesetzt werden können. Nachdem Sie diese haben, können Sie den Stakeholdern optimistische, pessimistische und höchstwahrscheinliche Schätzungen darüber geben, wie viele der wichtigsten Anforderungen aus dem Produkt-Backlog innerhalb eines Projekts fertiggestellt werden können. Diese Art der Schätzung bietet den Stakeholdern den gewünschten Detaillierungsgrad und ermöglicht dem Entwicklungsteam eine gewisse Flexibilität, da es noch damit beschäftigt ist, seinen Entwicklungsrhythmus zu finden.

In den ersten Sprints eines Projekts ist die Velocity des Teams meist sehr unterschiedlich. Sie wird stabiler, wenn das Team seinen Entwicklungsrhythmus gefunden hat. Natürlich führen Änderungen der Teammitglieder oder der Sprintdauer zu einer Variabilität in der Geschwindigkeit des Teams.



Geschwindigkeit ist kein Ziel, sondern eine Tatsache, die für die Extrapolation verwendet wird. Höhere Zahlen sind nicht automatisch besser als niedrigere Zahlen.

Wenn Story Points verwendet werden, werden Sie feststellen, dass manche Teams pessimistisch in ihren Schätzungen sind; ihre Zahlen sind immer hoch. Andere Teams sind optimistisch. Ihre Gesamtzahl der Story Points ist niedrig.

Angenommen, dass die Geschwindigkeit eines optimistischen Teams 15 beträgt und dass die Schätzung für das Release insgesamt 150 Punkte beträgt. Es würde das Team zehn Sprints kosten, das Release fertigzustellen.

Wenn die Geschwindigkeit eines pessimistischen Teams 30 wäre und die Schätzung für das Release 300 Punkte betragen würde, würde es ebenfalls zehn Sprints benötigen.



Optimistisch oder pessimistisch spielt keine Rolle. Die Teams werden in der Regel auf die eine oder andere Weise konsistent sein und sich daher am Ende ausgleichen. Aus diesem Grund bewerten Sie die Geschwindigkeit eines Teams immer im Vergleich zu dem Team selbst, nicht im Vergleich zu anderen Teams. Sagen Sie niemandem, wie schnell ein Team ist. Betrachten Sie es als Hochrechnungssystem, nicht als Leistungssystem.

Story Points und Velocity in Echtzeit

Ein Beispiel aus dem echten Leben, an dem Sie den Wert von Story Points und Velocity ablesen können, könnte so aussehen.

gehen Sie von den folgenden Daten aus:

- ✓ Das verbleibende Produkt-Backlog besteht aus 500 Story Points.
- ✓ Ihre Sprintdauer (Timebox) beträgt eine Woche, Montag bis Freitag.
- ✓ Die Geschwindigkeit des Entwicklungsteams beträgt durchschnittlich 20 Story Points je Sprint.
- ✓ Heute ist Montag, 8. Januar 2018.
- ✓ Die jährlichen Kosten für ein Mitglied des Entwicklungsteams betragen 100.000 € und Sie haben sieben Mitglieder – also 700.000 € pro Jahr oder 13.462 € pro Woche.

Innerhalb weniger Minuten sind Sie in der Lage, eine solide Schätzung der Kosten und des Liefertermins des Projekts vorzunehmen. Angesichts dieser Daten wird Ihr Projekt 25

Sprints umfassen und somit 25 Wochen dauern. Ihre Kosten belaufen sich auf 336.550 € und Ihr Fertigstellungstermin ist Freitag, der 29. Juni 2018.

Was passiert, wenn Ihr Entwicklungsteam seine Geschwindigkeit auf 25 Story Points pro Sprint erhöht? Setzen Sie einfach die neuen Zahlen ein. Jetzt benötigen Sie nur noch 20 Sprints. Ihre Kosten sinken auf 269.240 € (Sie sparen 67.310 €) und Ihr Fertigstellungsdatum ist Freitag, der 25. Mai 2018. Dies ist der Vorteil, wenn man einen engagierten Scrum Master hat, der Hindernisse und organisatorischen Widerstand, die das Team behindern, beseitigen kann, um die Velocity zu erhöhen.

Wenn Sie eine klare Methode zur Aufwandsabschätzung haben, welche zur Fertigstellung jeder Anforderung erforderlich ist, und wenn Sie die durchschnittliche Geschwindigkeit für das Entwicklungsteam ermittelt haben, können Sie den Umfang des Produkts, der innerhalb der festgelegten Kosten- und Zeitvorgaben erreicht werden kann, genau vorherzusagen. Sie nutzen hierzu Ihre Variablen.

Erfolgreiche Scrum-Teams müssen die Verantwortung für die Ergebnisse gemeinsam tragen und glaubwürdige Transparenz zwischen dem Team und seinen Stakeholdern schätzen. Fortschritt ist für alle sichtbar.

Kapitel 5

Release- und Sprint-Planung

IN DIESEM KAPITEL

Roadmap zu Release-Plänen weiterentwickeln
Den Scrum-Sprintzyklus anwenden
Ihre Sprints im Hinblick auf maximale Effektivität planen
Das Sprint-Backlog verwenden

Ein komplexes System, das funktioniert, ist aus der Weiterentwicklung eines einfachen Systems entstanden, das ebenfalls funktioniert hat. Ein komplexes System, das von Grund auf neu entwickelt wurde, funktioniert nie und kann nicht repariert werden, damit es funktioniert. Man muss mit einem funktionierenden, einfachen System neu anfangen.

John Gall

Bis jetzt haben Sie ein Produkt, einen Product Owner und ein Entwicklungsteam. Bei der Release- und Sprint-Planung geht es nun richtig los. Wie bereits erwähnt, werden Sie mit Scrum mehr planen als je zuvor, jedoch ist die Planung fokussiert, kontinuierlich, ergebnisorientiert und so gebündelt, dass Sie sich fragen werden, wie Sie jemals Projekte anders verwalten konnten.

In diesem Kapitel erfahren Sie, wie Sie die Freigabe von Projektfeatures logisch und übersichtlich planen. Die Release-Planung soll, das gesamte Projektteam für eine bestimmte Gruppe von Funktionalitäten mobilisieren, die das Unternehmen auf den Markt bringen möchte. Zu diesem Zeitpunkt werden andere Abteilungen oder Stakeholder wie die Marketingabteilung,

der Außendienst und der Kundendienst mobilisiert und darauf vorbereitet, die Funktionalität zu unterstützen, die Endkunden in der realen Welt nutzen werden.

Wir stellen Ihnen außerdem das Scrum-Event der Sprints vor. Im Kern von Scrum dreht sich alles um den Sprintzyklus. Wir zeigen Ihnen, wie Sie dieses wertvolle Verfahren nutzen und die bestmöglichen Ergebnisse erzielen.



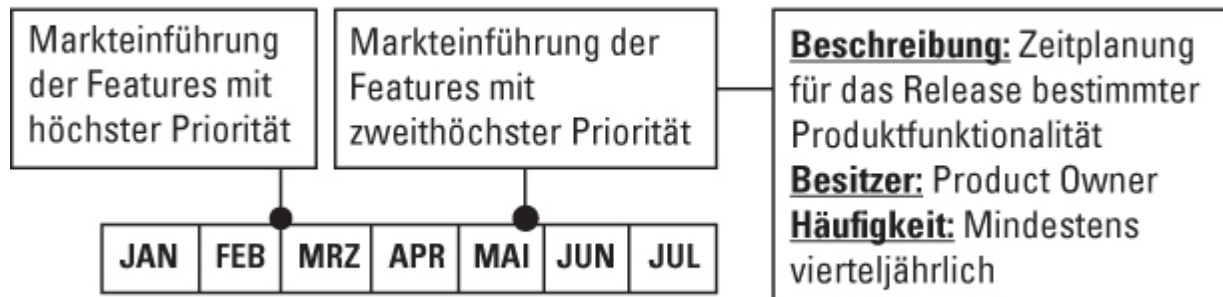
Ein Sprint ist ein festes Zeitfenster, in dem die Entwicklung eines funktionierenden, potenziell auslieferfähigen Produkts erfolgt. Ein typischer Sprint kann eine oder zwei Wochen dauern oder auch nur einen Tag oder einen Monat.

Wir erörtern, wie die Priorisierung von Releases und Sprints die Markteinführung beschleunigt und den Return on Investment (ROI) maximiert. Außerdem erklären wir, wie Sie mit Feedbackschleifen ein Produkt entwickeln, welches sich die Kunden wünschen und welches die tatsächlich gewünschten Funktionen enthält.

Release-Plan-Grundlagen: Phase 3

Wir beginnen mit der Roadmap to Value in [Kapitel 1](#) mit den Grundlagen der Vogelperspektive: Ihrem Vision Statement. So können Sie Ihr Ziel und das Ende des Projekts festlegen. Die zweite Phase, die Produkt-Roadmap, bietet eine ganzheitliche Sicht auf die Features, die die Vision unterstützen. Wenn Sie sich fokussieren und die Produktdetails ausarbeiten, kommen Sie in die Release-Planung, die Phase 3 Ihrer Roadmap (siehe [Abbildung 5.1](#)). Das Konzept der Release-Planung ist Bestandteil von Scrum. Allerdings ist das Release-Plan-Artefakt eine gängige Praxis, die viele Scrum-Anhänger mit Erfolg nutzen.

Phase 3: Release-Planung



(Phase 1 ist auch außerhalb von Scrum gebräuchlich)

Abbildung 5.1: Phase 3 der Roadmap to Value ist der Release-Plan.

Ein *Release-Plan* ist ein allgemeiner Zeitplan für die Freigabe einer Reihe von Produktanforderungen. Sie arbeiten sich also stückweise nach unten vor, angefangen bei der Roadmap, die das große Ganze darstellt, bis zum Release-Plan, der auf einer mittleren Ebene angesiedelt ist. (Später in diesem Kapitel behandeln wir die nächstkleinere Größe, nämlich den eigentlichen Sprint.) Der Release-Plan bietet den Fokus, um den herum sich das Projektteam mobilisieren kann.



Machen Sie das minimal Mögliche, mit dem Sie immer noch ein funktionierendes Produkt erzeugen. Wenn Sie eine Reihe von Funktionen für die Markteinführung planen, fragen Sie sich immer: »Was ist die kleinste Gruppe von Features, die meinem Kunden einen Mehrwert bringt?« Dies wird als Minimum Viable Product (MVP), das kleinstmöglich durchführbare/realisierbare Produkt, bezeichnet.

TK + AK > W

Indem Sie eine einfache Formel auf Ihr Projekt anwenden, können Sie ermitteln, wann Sie die bestmöglichen Ergebnisse erzielt haben. Mit anderen Worten, Sie wissen, dass Sie das Projekt beenden müssen, sobald die Summe aus tatsächlichen Kosten (TK) des Projekts und den

Alternativkosten (AK), wenn Sie nicht an anderen Projekten arbeiten, größer sind als der Wert (W), den Sie von der zukünftigen Funktionalität erwarten.

Als Formel sieht dies so aus:

$$TK + AK > W$$

Diese Formel hilft Ihnen festzustellen, wann es Zeit ist, ein Projekt zu beenden. Sie können eine Zeitersparnis von 30–40 Prozent und die Kostenersparnis von 30–70 Prozent erreichen durch

- ✓ stabile, engagierte Teams,
- ✓ Priorisierung zur Vermeidung von Features, die Kunden nicht nutzen wollen,
- ✓ Iteration durch konsequente Feedbackschleifen mit Ihren Kunden.

Sie sollten das Produkt so oft auf den Markt bringen, wie Sie Ihren Kunden etwas Wertvolles zu bieten haben – das kleinstmögliche Produkt. Ihre Produkt-Releases können auch regelmäßig erfolgen (zum Beispiel viermal im Jahr am Ende eines jeden Quartals). Aber mit den Fortschritten in der Technologie und der Notwendigkeit, schnell auf den Markt zu kommen, kann es heutzutage auch erforderlich sein, monatlich, wöchentlich oder sogar mehrmals täglich zu releasen.



Berücksichtigen Sie Kugelschreiber-und-Bleistift-Regel. Sie können einen Kugelschreiber verwenden, um den Plan für das erste Release zu bestätigen, aber alles, was über diesen Plan hinausgeht, ist mit Bleistift geschrieben. Es ist eine Just-in-Time-Planung für jedes Release.

Ein Projekt kann mehrere Releases haben. Für jedes Release beginnen Sie mit dem Ziel, das von den Funktionen und Anforderungen der nächsthöheren Priorität unterstützt wird. Sie werden merken, dass dies ein Muster ist, dem Sie in jeder Phase

der Roadmap folgen. Jedes Release hat ein Release-Ziel und Sie verpflichten sich zu jeweils nur einem Release.

[Abbildung 5.2](#) zeigt einen typischen Release-Plan mit einem optionalen Release-Sprint.

Release-Ziel: Kunden ermöglichen, auf eigene Konten zuzugreifen, Konteninformationen anzusehen und Umbuchungen vorzunehmen

Release-Datum: 31. März 2021



US = User Story

r = optionaler Release-Sprint

Abbildung 5.2: Ein typischer Release-Plan mit Releaseziel und -datum sowie einem optionalen Release-Sprint



Die Anzahl der Sprints innerhalb der einzelnen Releases kann je nach Release-Ziel und den notwendigen Anforderungen zur Zielerreichung, variieren. Eine Option ist die Verwendung sogenannter Release Trains, bei denen jedes Release einen festen Zeitplan hat. Releases sind festgelegt (wie ein Zugfahrplan), damit jeder weiß, wann er sie erwarten kann.

Bei jedem geplanten Release entscheidet der Product Owner, welche der fertiggestellten Features zu diesem Zeitpunkt freigegeben werden. Auf diese Weise wissen Unternehmen und Kunden, dass sie neue Features nach einem festgelegten Zeitplan erwarten können. Scrum-Teams können ihre Release-Pläne vorhersehbar innerhalb dieses Rhythmus organisieren.

Scrum ist ein empirischer Ansatz. Bei jedem Schritt überprüfen Sie Ihre Ergebnisse und passen sich sofort den sich ändernden

Bedürfnissen Ihrer Kunden an. Dieser Ansatz ermöglicht es Ihnen, erhebliche Kosten- und Zeiteinsparungen zu erzielen und gleichzeitig das zu liefern, was der Kunde wünscht. Sie können vier Releases für Ihr Projekt aus der Vogelperspektive betrachten, aber Sie planen immer nur ein Release auf einmal. Was Sie beim ersten Release lernen, kann Ihre Vorgehensweise beim zweiten Release ändern.



Ihr Unternehmen könnte Angst haben, Software inkrementell freizugeben. Es kann schwierig sein, abgeschlossene Arbeiten freizugeben, bevor der gesamte Produkt-Backlog fertiggestellt ist. Manchmal ergibt sich diese Zögerlichkeit aus dem Gefühl, das Produkt nach dem Release nicht mehr verbessern oder ergänzen zu können. Überwinden Sie diese Antipatterns, um effektivere und häufigere Releases zu ermöglichen.

Priorisieren, priorisieren, priorisieren

Sie haben das schon mal gehört, wenn es hier um Scrum ging, und Sie werden es in diesem Buch noch oft hören. Die Ermittlung der Anforderungen mit der höchsten Priorität, je nach Geschäftswert und Risiko, und das Arbeiten an nur diesen Anforderungen sind wichtige gemeinsame Praktiken innerhalb der agilen Gemeinschaft.

Betrachten Sie jedes Release als das nächste, inkrementelle MVP Ihres Produkts. Stellen Sie bei der Planung jedes Releases die folgenden Fragen, um die Features mit der höchsten Priorität zu identifizieren:

- ✓ Was macht eine Anforderung für dieses Release wichtig?
- ✓ Welche Mindestanzahl von Features, die dem Kunden einen Mehrwert bieten, müssen wir auf den Markt bringen?
- ✓ Welche Anforderungen stellen das größte Risiko dar und sollten eher früher als später angegangen werden?



Eine berühmte Studie der Standish Group aus dem Jahr 2001 zeigte bei einem repräsentativen Softwareentwicklungsprojekt, dass nur 20 Prozent der Features immer oder oft von den Kunden genutzt wurden. Interessanterweise wurden 64 Prozent der Features nie oder nur selten genutzt. Da Scrum den Schwerpunkt auf Priorisierung legt, werden zunächst nur die wichtigsten und nützlichsten Funktionen entwickelt. Was, wenn Ihr Projekt keine Mittel mehr hat und Sie nur zu 80 Prozent fertig sind? Basierend auf den Ergebnissen der Standish-Studie hätten Sie die restlichen 20 Prozent vielleicht sowieso nicht gebraucht. $TK + AK > W$ bedeutet das Ende des Projekts. Schichten Sie dieses Kapital so um, dass es höheren und besseren Nutzen erzielt.

Aus den folgenden vier Hauptgründen sollten Sie den Funktionsumfang pro Release so klein wie möglich halten:

- ✓ Pioniervorteile (First Mover Advantage) durch schnelle Markteinführung und Steigerung des Marktanteils
- ✓ Beschleunigter Kunden-Feedback-Zyklus
- ✓ Maximierter ROI (ein Euro ist heute buchstäblich mehr wert als ein Euro in sechs Monaten)
- ✓ Reduziertes internes Risiko durch Faktoren wie organisatorische Veränderungen und »Budgetdiebstahl«

Der Product Owner ist alleiniger Eigentümer des Release-Plans und derjenige, der die Anforderungen im Produkt-Backlog priorisiert und das Releaseziel festlegt (wie im folgenden Abschnitt gezeigt). Um eine Entscheidung bezüglich der Prioritäten und der Freigabe der auslieferungsfähigen Features zu treffen, berät der Product Owner sich mit den Stakeholdern und dem Entwicklungsteam. [Abbildung 5.3](#) (auf der folgenden Seite) wendet die 80/20-Regel auf das Konzept Wert versus Aufwand an. Es ist keine Zauberformel, aber dabei hilfreich, Wert zu verstehen und zu erklären.

Das Pareto-Prinzip (80/20-Regel) und das Gesetz des sinkenden Grenzertrags

	Stellen Sie sich ein aktuell noch nicht begonnenes Projekt vor und lassen Sie uns die 80/20-Regel mehrmals anwenden.	Gelieferter Geschäftswert	Aufwand	Verhältnis Wert zu Aufwand
Niedrig hängende Früchte	80 % des Geschäftswertes wurden mit 20 % der zur Verfügung stehenden Ressourcen geliefert (ein guter Wert, weil zuerst der höchste Werte und die am besten definierten User Stories geliefert wurden).	80 %	20 %	4:1
Schwierig oder riskant, aber benötigt	Wenn wir nun 80 % des verbleibenden Geschäftswertes (20 %) nehmen, erhalten wir 16 %. Hierfür werden 20 % der noch zur Verfügung stehenden Ressourcen (80 %) benötigt, also 16 %.	16 %	16 %	1:1
Wir haben uns festgefahren	Wir haben bisher 96 % des Geschäftswertes geliefert und hierfür lediglich 36 % der Ressourcen aufgewendet. Lassen Sie uns die Regel ein weiteres Mal anwenden. 80 % von 4 % ergibt 3,2 % Wert und 20 % der verbleibenden 64 % Ressourcen ergibt einen Aufwand von 12,8 %.	3,2 %	12,8 %	1:4
	Uns stehen noch mehr als 50 % des geschätzten Aufwands zur Verfügung, um Geschäftswerte zu liefern, die kaum noch Zusatznutzen bringen.	99,2 %	48,8 %	

Abbildung 5.3: Anwendung des Pareto-Prinzips (80/20-Regel) auf Scrum

Diese Darstellung spiegelt keine exakte Projektschätzung wider, jedoch verdeutlicht sie etwas, das Scrum-Praktiker in unterschiedlichen Projekten erleben. Wenn Sie dem Werteschöpfungsplan, der Roadmap to Value, folgen, dann werden Sie die User Stories, die den größten Wert liefern, zuerst erstellen.

- ✓ Mehr als die Hälfte aller Softwarefeatures wird nicht benutzt (siehe die Standish-Studie)
- ✓ Das Gesetz des sinkenden Grenzertrags bezieht sich auf einen Punkt, ab dem der Nutzen, den man erzielen kann, geringer ist als die Menge des Aufwands (Zeit, Kosten), den man dafür aufbringen muss.

Release-Ziele

Jeder Release-Plan hat ein übergeordnetes Geschäftsziel, das als Release-Ziel bezeichnet wird. Dieses Ziel wird vom Product Owner erstellt und knüpft direkt an das Endziel des Produktes an, das im Product Vision Statement definiert ist. Das Release-Ziel legt die mittelfristigen Grenzen für bestimmte Funktionalitäten fest, die den Kunden zur Nutzung in der realen Welt zur Verfügung gestellt werden.

Ein explizites Release-Ziel beschleunigt den Priorisierungsprozess: Wenn eine Anforderung nicht mit dem Ziel übereinstimmt, müssen Sie sich in diesem Release keine Sorgen um diese Anforderung machen. Jede gegebene Anforderung sollte sich das Recht auf Ihre Investition verdienen. Lassen Sie die Anforderung im Produkt-Backlog stehen, bis sie ein bestimmtes Release unterstützen kann. [Abbildung 5.4](#) zeigt eine Matrix, die Ihnen hilft, die Prioritäten der Storys zu bestimmen.

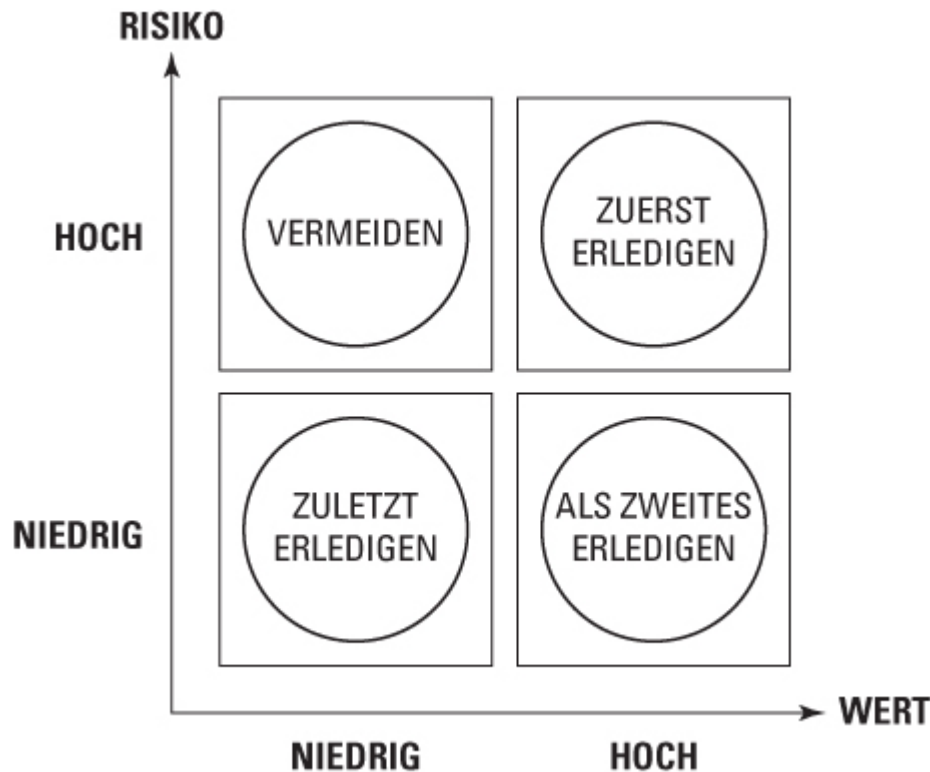


Abbildung 5.4: Backlog-Prioritätsmatrix



Stellen Sie sich die drei Ebenen der Zielsetzung als Filtersystem zur Priorisierung vor. Die Vision ist die oberste Ebene. Wenn ein Feature nicht zur Gesamtvision passt, gehört es nicht in dieses Projekt. Fügen Sie diese Anforderung nicht einmal in das Produkt-Backlog ein. Derjenige, der die Anforderung einbringt, muss sich die Mittel für diese Idee separat besorgen. Lassen Sie nicht zu, dass Funktionen zu blinden Passagieren in Ihrem Projekt werden.

Als Nächstes kommt das Release-Ziel. Das ist die mittelfristige Ebene. Falls eine Anforderung nicht zum Release-Ziel passt, verbleibt sie im Produkt-Backlog und wird zu einem späteren Zeitpunkt erneut bewertet. Schließlich ist die kürzeste Ebene das Sprintziel.



Ziele steuern das Produkt-Backlog und den Release-Plan, nicht umgekehrt. Jedes Feature und jede Anforderung müssen im Hinblick darauf betrachtet werden, ob sie zum Ziel

passen. Das ist *zweckgerichtete Entwicklung*. Die Ziele steuern die Inhalte des Produkt-Backlogs. Die Ziele steuern, was in einem Release enthalten ist. Und Ziele bestimmen, was in den jeweiligen Sprints enthalten ist.

Release-Sprints

Manchmal ist es notwendig, bestimmte Aufgaben zu erledigen, die nicht in einen normalen Entwicklungssprint passen. Im Idealfall werden alle Aktivitäten, die zur Markteinführung eines Produkts erforderlich sind, im Rahmen eines normalen Sprints durchgeführt. Aber wenn die Organisationsstruktur dies nicht zulässt, kann ein Release-Sprint verwendet werden, um folgende Ziele zu erreichen:

- ✓ Überprüfung der Skalierung (zum Beispiel Last- oder Leistungstests)
- ✓ Erweiterte Testaktivitäten (zum Beispiel Fokusgruppen oder Validierung der entwickelten Funktionalität mit Live-Daten)

Obwohl wir es vorziehen, keine Release-Sprints zu verwenden, werden, einfach ausgedrückt, in einen Release-Sprint alle anderen Dinge erledigt, die nötig sind, um das Produkt auf den Markt zu bringen.

Der Release-Sprint wird üblicherweise nach einer Reihe von normalen Sprints innerhalb eines Releases verwendet. Die Dauer eines Release-Sprints kann sich von der Dauer der Entwicklungssprints im Release unterscheiden. Die Länge des Release-Sprints hängt von der Art der Aktivitäten und dem Arbeitsaufwand ab, der erforderlich ist, um die fertigen Produktinkremente der einzelnen Sprints freizugeben. Das Scrum-Team ermittelt alle diese Faktoren während der Release-Planung.



Während eines Release-Sprints erfolgt nicht die eigentliche Entwicklung von Anforderungen. Alle Entwicklungsaufgaben (wie Tests, technische Dokumentation, Qualitätssicherung und Peer-Review) werden innerhalb der Sprints durchgeführt, um die Definition of Done des Teams zu erfüllen, was wiederum ein potenziell auslieferfähiges Produkt am Ende jedes Sprints sicherstellt. Aber bevor man das Produkt auf den Markt bringen kann, müssen möglicherweise andere Dinge (wie Fokusgruppen oder Last- und Leistungstests) durchgeführt werden.

Da sich die Länge und die Aktivitäten des Release-Sprints oft von den Entwicklungs-Sprints unterscheiden, existiert in Release-Sprints das Konzept der Velocity nicht. Entwicklungsteams schätzen nach bestem Wissen den Aufwand und die Komplexität der Aufgaben für den Release-Sprint. Im gesamten Team sollten alle einverstanden sein und alle sollten sich mit der festgelegten Dauer des Release-Sprints wohlfühlen.



Der Release-Sprint ist eine Art Anti-Pattern in Organisationen, die innerhalb des Sprints keine skalierten Test- und Supportaufgaben durchführen können. Falls Sie den Release-Sprint nicht benötigen, dann setzen Sie ihn nicht ein.

Einschließlich der bereits genannten Beispiele können in Release-Sprints die folgenden Aktivitäten durchgeführt werden:

- ✓ Durchführung von Fokusgruppen (denken Sie daran, dass dies nicht dazu dient, neue Features zu identifizieren, sondern um zu überprüfen, was Sie getan haben, und um Release-Probleme zu identifizieren)
- ✓ Skalierungstests
- ✓ Leistungssteigerung durch Skalierung der Testergebnisse
- ✓ Integration des Produkts in unternehmensweite Systeme

- ✓ Erstellung von Dokumentationen wie zum Beispiel Benutzerhandbüchern
- ✓ Finalisierung aller regulatorischen Anforderungen

Der Release-Plan in der Praxis

Um zu sehen, wie ein Release-Plan in der realen Welt funktioniert, folgen Sie diesen Schritten:

1. Entwickeln Sie ein Release-Ziel.

Dieses Ziel ist die Zielvorgabe für alles andere. Der Product Owner stellt sicher, dass das Ziel mit der Produktvision übereinstimmt und arbeitet mit dem Scrum-Team zusammen, um sicherzustellen, dass sich das gesamte Team mit dem Ziel wohlfühlt.

2. Legen Sie das angestrebte Veröffentlichungsdatum fest.

Dieses Datum kann durch Faktoren beeinflusst werden, die außerhalb der Kontrolle des Scrum-Teams liegen.

3. Identifizieren Sie aus dem Produkt-Backlog die Anforderungen mit der höchsten Priorität (MVP), die das Release-Ziel unterstützen.

Die Priorität ist eine Frage von Wert und Risiko. Nehmen Sie zuerst die Positionen mit dem höchsten Wert beziehungsweise dem höchsten Risiko in Angriff. In [Abbildung 5.4](#) finden Sie eine Wertematrix.

4. Verfeinern Sie, wenn nötig, die Aufwandsschätzung der Anforderungen.

Manchmal werden Probleme und/oder Synergien entdeckt, wenn das Entwicklungsteam das kleinere Paket der Anforderungen betrachtet, die in den Release-Plan einfließen. Auch die Anforderungen selbst werden detaillierter und mehr aufgeschlüsselt (sie sollten nicht größer als 34 Story Points sein; siehe [Kapitel 4](#)) als bei der ursprünglichen Schätzung des Entwicklungsteams auf der Ebene der Produkt-Roadmap.

5. Identifizieren Sie die Velocity des Teams.

Wenn das Team stabil ist und schon zusammen am gleichen Projekt gearbeitet hat, ist die in früheren Sprints ermittelte Velocity ein guter Ausgangspunkt. Wenn das Team seine Velocity noch nicht festgestellt hat, starten Sie bescheiden, bis Sie ein paar Sprints durchlaufen haben und die Velocity ermittelt werden kann.

6. Planen Sie einen Release-Sprint (falls erforderlich).

Legen Sie als Scrum-Team fest, ob Sie einen Release-Sprint benötigen und wie lange er dauern soll.

7. Finalisieren Sie den Umfang des Releases.

Basierend auf der Velocity, der Gesamtaufwandsschätzung der Anforderungen und der Anzahl der Sprints innerhalb des Release-Zeitfensters: Wie viel Funktionalität können Sie einbinden? Was ist flexibler – das Datum oder der Umfang der Funktionalität? Welche Anpassungen müssen am Veröffentlichungsdatum oder am Umfang der Anforderungen vorgenommen werden, um innerhalb Ihres Zeitfensters so viel wie möglich releasen zu können?

Angenommen, Ihre Velocity beträgt 20, das Release-Zeitfenster beträgt 5 Sprints und die geschätzte Gesamtpunktzahl für das Release ist 110. Damit liegen Sie zehn Story Points über dem, was im Release erledigt werden kann. Der Product Owner muss eine Entscheidung treffen. Sind die Anforderungen, die die letzten zehn Punkte der Release ausmachen, wertvoll genug, um sie aufzunehmen, oder kann die Veröffentlichung ohne sie weitergehen? Oder muss das Release um einen Sprint verlängert werden und ist das für die Stakeholder und Kunden akzeptabel?



Eine Möglichkeit bei der Release-Planung ist die Verwendung des Release-Train-Modells. In diesem Szenario sind die Releases nicht von unterschiedlicher Dauer, sondern genau gleich lang, zum Beispiel sechs Wochen. Am Ende jedes sechswöchigen Zyklus wird die fertige Funktionalität aus jedem Sprint verpackt und freigegeben. Auf diese Weise wird ein Entwicklungsrhythmus geschaffen, und jeder in der Organisation kann seine Arbeitsbelastung antizipieren und die weitere Entwicklung planen.

Verwenden Sie eine der in [Kapitel 4](#) beschriebenen Schätz- und Konsensbildungstechniken, um die im Release enthaltenen Einträge aus dem Produkt-Backlog zu verfeinern, wenn die Anforderungen noch verfeinert und geschätzt werden müssen (siehe den vorherigen Schritt 4). Wenn Ihre Releases aus vielen Produkten bestehen, verwenden Sie für die Release-Planung die Affinitätsschätzung (die in [Kapitel 4](#) beschriebene Technik, die auf T-Shirt-Größen basiert).



Wie bei allen Scrum-Dokumentationen bevorzugen wir möglichst einfache Werkzeuge. Der gesamte Release-Plan kann auf Ihrem Whiteboard und Haftnotizen dargestellt werden. Dies ermöglicht einfache Änderungen und sofortigen Zugriff. Außerdem spart es einfach Zeit.

Sprint zu Ihren Zielen

Endlich, das Herzstück von Scrum! Sprints und das eingebaute Modell des Überprüfens und Anpassens sind integrale Scrum-Features. Es ist genau dieser Sprint-Prozess, durch den Sie die drei agilen Säulen der Verbesserung – Transparenz, Überprüfung und Anpassung – erreichen, die wir in [Kapitel 1](#) diskutieren.

Indem Sie Ihr Projekt in handliche Teile zerlegen und dann Ihren Fortschritt mit dem empirischen Scrum-Modell bewerten, können Sie sich ständig vorwärts bewegen. Dies ermöglicht Ihnen die

Wendigkeit und Leichtigkeit der Anpassung, die im Wasserfallmodell so schmerzlich fehlt.

Jedes Scrum-Teammitglied hat in den Sprints die gleichen Aufgabe: die Maximierung der Effektivität bei der Lieferung potenziell auslieferfähiger Produkte.

Sprints definieren

Sprints sind die Essenz von Scrum, wie bereits in [Kapitel 1](#) erläutert. Sie bilden ein konsistentes Zeitfenster für die Produktentwicklung durch das Entwicklungsteam. Jeder Sprint enthält die folgenden Elemente:

- ✓ Sprint-Planung inklusive Zielsetzung
- ✓ Daily Scrums
- ✓ Entwicklungszeit einschließlich regelmäßiger Überprüfung durch den Product Owner
- ✓ Sprint-Review
- ✓ Sprint-Retrospektive

Die gleichbleibende Dauer der Sprints ermöglicht es dem Entwicklungsteam, einen Entwicklungsrhythmus festzulegen. Außerdem können Scrum-Teams auf der Grundlage empirischer Daten wie der Velocity für die Zukunft planen. Sobald ein Sprint beendet ist, beginnt der nächste. Ein Fluss von konsistenten iterativen Feedbackschleifen wird geschaffen und erzeugt so ein ideales Umfeld für die Produktion und kontinuierliche Verbesserung.



Stellen Sie sich vor, Sie sind ein Läufer. Sie trainieren ständig für den 100-Meter-Lauf und sind unglaublich gut darin, aber plötzlich fordert Sie Ihr Trainer auf, einen Marathon zu laufen. Wenn Sie versuchen den Marathon in Ihrem 100-Meter-Lauf-Tempo zu schaffen, werden Sie den Marathon nicht beenden. Sie müssen Ihr Training modifizieren, um sich auf das Marathontempo einzustellen. Das erfordert ein angepasstes Coaching, einen Zeitplan und eine Ernährungsumstellung. Das gesamte Muskelgedächtnis und die Art der Ausdauer, die Ihr Körper entwickelt hat, müssen neu erlernt werden, damit Sie eine Strecke anderer Länge laufen können.

Sprintlänge planen

Da sich die Sprintziele während eines Sprints nicht ändern, hängt die Antwort auf die Frage »Wie lange sollte ein Sprint dauern?« von Ihrem Projekt ab und davon, wie lange Ihre Organisation ohne Änderungen auskommen kann. Dies sind die äußeren Rahmenbedingungen. Sie haben keinen Grund, über die so vorgegebene Dauer hinauszugehen. Wenn es beispielsweise für Ihre Organisation schwierig ist, eine Woche ohne Änderungen weiterzumachen, sollten Sie die Idee eines zweiwöchigen Sprints noch nicht einmal diskutieren. Sie werden nicht in der Lage sein, die Integrität der Stabilität des Sprints aufrechtzuerhalten, und Stabilität ist eine bedeutende Antriebsfeder für Performance in Scrum. Besprechen Sie stattdessen, wie viel kürzer Sie den Sprint machen können.

Außerdem ändern sich die Sprintlängen nach dem Start nicht und im Idealfall auch nicht während eines Projekts, es sei denn, sie werden verkürzt. Wenn ein Scrum-Team während eines Projekts die Sprintdauer ändert, entstehen erhebliche Kosten: Ihre frühere Velocity ist nicht mehr relevant. Leistung ist keine gerade mathematische Linie, die geschnitten, gewürfelt und wieder zusammengesetzt werden kann. Nur weil ein Scrum-Team die Sprintdauer von einer auf zwei Wochen verdoppelt, bedeutet das

nicht, dass das Team automatisch das Doppelte seiner bisherigen Velocity aus einwöchigen Sprints, erreichen kann.

Kürzere Sprints verkürzen die Zeit zwischen den Rückmeldungen der Stakeholder und ermöglichen es den Scrum-Teams, früher und häufiger zu überprüfen und anzupassen. Längere Sprints haben eine abnehmende Rendite, da die Dringlichkeit aufgrund der vielen Tage, die dem Team noch zur Verfügung stehen, geringer ist. Auch Wochenenden und längere Sprintmeetings können sich negativ auf die Effizienz auswirken.

Die Kapazität eines Entwicklungsteams während eines einwöchigen Sprints kann höher oder niedriger als die Hälfte der bisherigen zweiwöchigen Geschwindigkeit sein. Sie wissen es erst, nachdem Sie ein paar Sprints durchlaufen haben und können sich über die Velocity erst nach einer großen Anzahl abgeschlossener Sprints sicher sein.

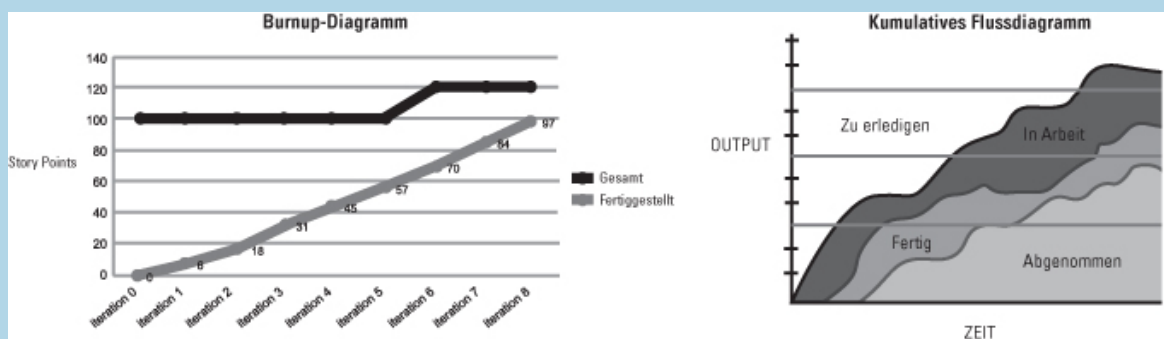


Zwar sind die Kosten für die Änderung der Sprintlänge während eines Projekts erheblich, doch die Kosten für die Änderung eines Sprintziels während eines Sprints liegen wahrscheinlich noch höher. Wenn ein Sprintziel vor dem Ende eines Sprints irrelevant wird (beispielsweise aufgrund von Richtungsänderungen des Unternehmens oder Änderungen im Markt), kann ein Product Owner entscheiden, den Sprint abubrechen. Aber seien Sie sich bewusst, dass der Abbruch wertvolle Entwicklungsressourcen verschwendet und für das Scrum-Team und die Organisation sehr traumatisch ist. Je kürzer die Feedbackschleife (das heißt die Länge des Sprints), desto unwahrscheinlicher ist es, dass ein Product Owner einen Sprint abbricht.

Kumulative Flussdiagramme

Kumulative Flussdiagramme (Cumulative Flow Diagrams, CFDs) geben Scrum-Teams Einblick in die Muster ihres Outputs und helfen, mögliche Engpässe in ihren Prozessen zu identifizieren.

Ein Burnup-Diagramm, wie es auf der linken Seite der Abbildung gezeigt wird, ist ein Typ eines kumulativen Flussdiagramms, das die Anzahl der Anforderungen zeigt, die das Team innerhalb einer bestimmten Zeitspanne erfüllen konnte. Kumulative Flussdiagramme, die auf der rechten Seite der Abbildung gezeigt werden, sind eine Momentaufnahme des Arbeitsaufwandes in einem bestimmten Stadium des Prozesses, sodass Indikatoren wie Lead, Zyklus, Arbeit und Wartezeiten aufgedeckt werden können, um Ineffizienzen zu identifizieren.



Die Änderung der Sprintlängen während eines Projekts führt im Laufe der Zeit zu inkonsistenten Leistungsindikatoren. Neben dem Verlust der Relevanz der bisherigen Velocity bei wechselnden Sprintlängen werden auch CFD-Indikatoren wie Lead-, Zyklus-, Arbeits- und Wartezeiten irrelevant.



Aus der Wissenschaft wissen wir, dass man seinen Verstand nicht abschalten kann. Er funktioniert immer. Wenn Sie Ihrem Entwicklungsteam eine kleine Anzahl von Problemen zu lösen geben und ihm sagen können, dass es morgen mit diesen Problemen konfrontiert wird, wird es bei der Arbeit bewusst darüber nachdenken. Ob es will oder nicht, es wird auch dann unbewusst an sie denken, wenn es nicht an der Arbeit ist. Aus diesem Grund ist die Stabilität von Sprints so wichtig. Nach dem Start eines Sprints müssen die Entwickler darauf vertrauen können, dass der Umfang stabil bleibt, sodass sie sich voll und ganz darauf konzentrieren können, was für diesen Sprint getan werden muss, egal, ob sie bei der Arbeit sind oder nicht. Hatten Sie schon einmal eine Erleuchtung beim Zähneputzen? Das ist die Dynamik, über die wir hier reden. Aber Sie brauchen zwei Elemente: eine begrenzte Anzahl von Problemen und das Vertrauen, dass Sie sich diesen Problemen morgen stellen werden. Wenn ein Entwickler an einem Tag an Projekt A, Projekt C oder Projekt wer weiß was arbeitet, wird das nicht passieren. Ein Entwickler wird sich erst dann geistig engagieren, wenn er ins Büro kommt und entdeckt, was vor ihm liegt. Ein Grund, warum agile Projekte so innovativ sind, liegt darin, dass sie diese Stabilität aufweisen und damit einen größeren Anteil am Denken der Entwickler haben.



Bei unseren Kunden haben wir festgestellt, dass die einwöchige Sprintdauer ein schöner Rhythmus ist. Dieser gibt dem Entwicklungsteam freie Zeit, vermeidet »Wochenendbetrügereien«, um mehr Arbeit zu erledigen, als das Team leisten kann, und ist dennoch lang genug, um jede Woche echte Fortschritte zu erzielen. Dieser kürzere Feedback-Zyklus ermöglicht es den Scrum-Teams auch, häufiger zu überprüfen und anzupassen. Aus diesen Gründen sollten Scrum-Teams immer nach Möglichkeiten suchen, ihre Sprintlänge auf verantwortbare Weise zu verkürzen.

Entscheidend ist es, Sprints durchzuführen, die es Ihrem Entwicklungsteam ermöglichen, bei jedem einzelnen Sprint ein konkretes, getestetes und abgenommenes Produkt zu erstellen. Nach jedem Sprint haben Sie etwas Reales, das Sie den Stakeholdern zeigen können.

Eine Warteschlange aus der Mathematik verwenden

Die Warteschlangentheorie ist das mathematische Studium des Wartens in Warteschlangen. Studien haben ergeben, dass Kunden, die in einem Restaurant auf einen Tisch warten, doppelt so lange warten, wenn sie am Anfang etwas Befriedigendes bekommen. Wenn sie beispielsweise gebeten werden, an der Bar zu sitzen, und die Möglichkeit haben, ein Getränk zu bestellen oder sogar etwas probieren können, bleiben sie länger und warten, bis der noch besetzte Tisch geräumt ist.

Das gleiche Prinzip gilt für Sie bei Scrum. Am Ende eines jeden Sprints haben Sie etwas Funktionierendes und Greifbares, das Sie den Stakeholdern überreichen können. Dieses Konzept des »frühzeitig und häufig« ist eine bekannte Technik zur Steigerung der Kundenzufriedenheit.

Dem Sprint-Lebenszyklus folgen

Jeder Sprint hat den gleichen Ablauf: Sprint-Planung, Daily Scrums, ein Sprint-Review und eine Sprint-Retrospektive. Sprints sind Entwicklungszyklen, die sich wiederholen, bis Ihr Projekt abgeschlossen ist. Anforderungen (oft in Form von User Stories) werden in jedem Sprint entwickelt, getestet, integriert und abgenommen. Der Prozess wird Sprint für Sprint wiederholt. [Abbildung 5.5](#) zeigt einen einwöchigen Sprintlebenszyklus.

Phase 4: Sprint-Planung

Beschreibung: Festlegen der iterationsspezifischen Ziele und Tasks

Besitzer: Product Owner und Entwicklungsteam

Häufigkeit: Zu Beginn jedes Sprints

Abbildung 5.5: Der einwöchige Sprintlebenszyklus



Wenn Scrum-Teams offshore mit Teammitgliedern in unterschiedlichen Zeitzonen (beispielsweise USA und Indien) arbeiten, müssen Vorkehrungen getroffen werden, damit alle Teammitglieder an jedem der Sprinttreffen teilnehmen können. Um die Zeitzonendifferenzen zu berücksichtigen, können die inländischen Teammitglieder am Sonntagabend an der Sitzung teilnehmen, während es für die Offshore-Teammitglieder Montagmorgen ist. Am Ende des Sprints beenden die einheimischen Teammitglieder den Sprint am Freitagmorgen und nehmen den Rest des Tages frei, während sich das Offshore-Team dem Sprint-Review und der Retrospektive am Freitagabend anschließt. Diese Zeiteinteilung kann bei jedem Sprint vertauscht werden, damit die Teammitglieder nicht immer an Sonntag- oder Freitagabenden arbeiten müssen.

Das Entscheidende ist, dass das Scrum-Team nach jedem Sprint neue Dinge lernt. Änderungen ergeben sich; sie sind unvermeidlich. Darauf zu reagieren und sich darauf einzustellen, sollte als Fortschritt und nicht als Misserfolg betrachtet werden.

In Scrum sind Änderungen einfach, da am Ende eines jeden Zyklus das Geschaffene komplett fertiggestellt wurde. Wenn Sie in Ihren nächsten Sprint gehen und Einträge aus dem Produkt-Backlog bearbeiten, spielt es keine Rolle, ob dieser Eintrag seit vier Monaten, vier Wochen oder vier Minuten im Produkt-Backlog vorhanden ist. Alt oder neu, jeder Eintrag wird nicht nach der

Reihenfolge, in der er eingegangen ist, sondern nach der Reihenfolge, in der er dem Kunden den höchsten Wert liefert, priorisiert.

Laufen Sie nicht den Wasserfällen hinterher

Scrum unterscheidet sich von der traditionellen Wasserfallmethode, bei der sich alle Elemente in einem Paket befinden und bei dem das Paket als Ganzes heranreift. Wenn also in der Mitte des Projekts Änderungen vorgenommen werden, sind die Auswirkungen dieser Änderungen nebulös und über mehrere Anforderungen hinweg verwoben. Diese unbekannten Auswirkungen führten dazu, dass »Veränderung« im Wasserfallansatz zu einem anstößigen Wort wurde.

Manche Scrum-Neulinge, die am Wasserfallverfahren festhalten, betrachten einen Sprint fälschlicherweise einfach nur als kürzeres Zeitfenster als zuvor, in dem Aufgaben erledigt werden, die Teil eines allgemeinen, vorher aufgestellten Gesamtplans sind:

- ✓ Sie legen zu Beginn des Projekts mehrere Sprints fest und identifizieren die genauen Anforderungen und Aufgaben, die in den einzelnen Sprints erledigt werden müssen.
- ✓ Sie halten sehr kurze Sprint-Reviews als Statusberichte über den Produkt-Backlog ab und verwenden diese nicht dafür, ein fertiges Produktinkrement zu demonstrieren.
- ✓ Sie halten kurze Sprint-Planungs-Meetings ab, bei denen es nur darum geht, den ursprünglichen Plan zu bestätigen.

Die erhöhte Häufigkeit der Überprüfung des ursprünglichen Plans erscheint einem Team, das neu im agilen Denken ist, angenehm. Dies geht am entscheidenden Punkt völlig vorbei. Das ist immer noch Wasserfall.

Der Sprintlebenszyklus ermöglicht die einfache Einbindung und Anpassung von Veränderungen, die auf der Realität

basieren. Das ist Empirie.



Unserer Erfahrung nach ist die Arbeitswoche von Montag bis Freitag ein natürliches, biorhythmisches Zeitfenster. Teams brauchen die Pause am Wochenende, die natürlich zu den Lebensgewohnheiten passt. Vermeiden Sie daher unkonventionelle Sprintmuster von Mittwoch bis Dienstag oder dergleichen.

Planung Ihrer Sprints: Phase 4

Die Planung von Sprints ist Phase 4 in der Roadmap to Value. Alle Arbeiten, die während dieses speziellen Sprints durchgeführt werden sollen, werden hier geplant. Jede Sprint-Planungssitzung ist auf maximal zwei Stunden pro Woche des Sprints begrenzt. Wenn Sie also beispielweise Sprints von einwöchiger Dauer verwenden, haben Sie maximal zwei Stunden Zeit, um Ihren Sprint zu planen.

Sprintziele

Für jeden Sprint wird ein Ziel erstellt. Der Product Owner initiiert die Zieldiskussion und identifiziert das zu erreichende Geschäftswertziel. Nachdem sich das Team über das Ziel im Klaren ist, wählt der Product Owner die Anforderungen aus dem Produkt-Backlog aus, die das Ziel am besten unterstützen. Wie beim Release-Ziel steuert das Sprintziel die zu entwickelnden Anforderungen, nicht umgekehrt.

Das Sprintziel selbst muss das Release-Ziel unterstützen, das seinerseits die Produktvision unterstützt. Diese Zielunterteilung und gemeinsame Ausrichtung sind unerlässlich, um sicherzustellen, dass Sie zielgerichtete Entwicklung betreiben.

Das Entwicklungsteam ist entscheidend bei der Erstellung des Sprintziels. Weil das Entwicklungsteam die eigentliche Arbeit

erledigt, legt es das Wie und das Wie viel fest, während der Product Owner die Richtung oder das Was festlegt.

Wenn Ihr Entwicklungsteam seine ermittelte und stabile Velocity kennt, kann diese als Input für die Bestimmung des Arbeitsaufwandes verwendet werden, die für den Sprint übernommen werden kann.

Das Entwicklungsteam kann die Velocity auch zum Ausdehnen, zum Testen seiner Grenzen oder zum Zurückziehen verwenden, wenn es Schwierigkeiten hat, die gesteckten Ziele zu erreichen. Wenn das Team bequem 34 Story Points erreicht hat, kann es die Latte auf 38 oder 40 Points legen. Wenn es darum kämpft, 25 Points zu erreichen, kann es die Velocity auf 23 senken, während es versucht herauszufinden, welche organisatorischen Hindernisse beseitigt werden müssen.

Beide Phasen der Sprint-Planung finden in der einzelnen Sprint-Planungsbesprechung zu Beginn des Sprints statt. Phase I ist die Phase, in der der Product Owner mit Hilfe des Entwicklerteams und unter Anleitung des Scrum Masters bestimmt, was erreicht werden muss (das Ziel). In Phase II legt das Entwicklungsteam fest, wie das Sprintziel zu erreichen ist und entwickelt den eigentlichen Sprint-Backlog mit den unterstützenden Aufgaben.



Es kann sein, dass die Sprint-Planungsbesprechung nicht immer reibungslos verläuft, vor allem am Anfang. Sie können wie Alice im Wunderland in einen Kaninchenbau fallen, Tangenten entdecken und verschiedene Einschätzungen der Möglichkeiten ausgraben. Hier ist ein starker und geschickter Moderator in Form des Scrum Masters gefragt. Es ist seine Aufgabe, dafür zu sorgen, dass die Besprechung auf Kurs bleibt und die Diskussion nicht zu hitzig wird.

Phase I

Zu Beginn der Phase I wird das Sprintziel bestimmt. Das Entwicklungsteam muss dieses vollständig verstehen, da es die Grenzen und die Richtung für die Arbeit vorgibt, die während des

gesamten Sprints zu leisten sind. Der Product Owner wählt dann einen Teil des Produkt-Backlogs aus, der das Ziel unterstützt. Dies ist nicht unbedingt der endgültige Sprint-Backlog, sondern die prognostizierte Funktionalität, die, bei Fertigstellung, das Sprintziel erfüllen würde. Also das, woran das Entwicklungsteam arbeiten wird, um das Sprintziel zu erreichen und das tatsächlichen Sprint-Backlog zu ermitteln.

Phase I gibt dem Entwicklungsteam und dem Product Owner eine weitere Möglichkeit, bestehende Anforderungen zu klären oder neue zu identifizieren, die zur Erreichung des Sprintziels notwendig sind. Dies wäre auch der Zeitpunkt, um die endgültige Aufwandsschätzung der bereits geklärten und der eventuell neuen Anforderungen vorzunehmen. Denken Sie an diese Richtschnur: Wenn die Anforderungen größer als 8, dann sind sie zu groß für einen Sprint (siehe [Kapitel 4](#)).



Wir übernehmen gerne sechs bis zehn Einträge aus dem Produkt-Backlog in einen einzelnen Sprint. Dies ist in der Regel die richtige Balance, um ein Produktinkrement mit hoher Funktionalität zu liefern, und dies dient als Lackmus-Test, um zu prüfen, dass jeder einzelne Eintrag ausreichend aufgeschlüsselt ist.



Die einzigen Anforderungen, die in der Sprint-Planung diskutiert werden sollten, sind die, die mit einem Wert zwischen 1 und 8 der Fibonacci-Folge geschätzt wurden. Dies ist keine Scrum-Regel, aber sie passt zu unserem Affinitätsschätzmodell (siehe [Kapitel 4](#)) und war für viele Teams eine effektive Möglichkeit, sich auf einwandfrei untergliederte Anforderungen zu konzentrieren, die innerhalb eines Sprints fertiggestellt werden können. Weitere Informationen zu Fibonacci-Zahlen finden Sie in [Kapitel 4](#).

Phase II

Wenn das Ziel und die unterstützenden Anforderungen durch das Scrum-Team festgelegt wurden, teilt das Entwicklungsteam diese

Anforderungen in einzelne Aufgaben auf. Damit beschreibt es, wie es aus den Einträgen im Produkt-Backlog das Produktinkrement erstellt. Die Aufgaben für jede Anforderung sollten explizit der Definition of Done des Teams entsprechen. Wenn die Definition of Done beispielsweise die Integration mit System A beinhaltet, sollte mindestens eine Aufgabe für die Anforderung »Integrationstest mit System A« sein.



Im Idealfall sollte jede Aufgabe an einem Tag erledigt werden können. Dies gibt dem Entwicklungsteam ein greifbares, realistisches Zeitziel, wenn es die Anforderungen in Aufgaben zerlegt. Außerdem wird so ein Maßstab festgelegt, durch den das Entwicklungsteam auf mögliche Probleme bei der Entwicklung aufmerksam gemacht wird. Wenn die Entwicklung einer Aufgabe mehrere Tage dauert, muss möglicherweise ein Problem mit der Aufgabe oder dem Entwickler gelöst werden.

Entwicklungsteams können die Aufgaben für jede Anforderung in Stunden abschätzen, wenn die Organisation diese Transparenz wünscht. Aber viele Teams verwenden einfach die Velocity und versehen die Produkt-Backlog-Einträge mit »fertiggestellt/nicht fertiggestellt«, um das Sprintrisiko adäquat zu visualisieren.

Das Entwicklungsteam ist gezwungen, hier am detailliertesten zu sein, was dazu beiträgt, den Geist zu schärfen. Sie können wirklich tief nach unten graben und schauen, was getan werden muss.

Am Ende der Sprintplanung sollte sich das Entwicklungsteam darüber im Klaren sein, wie die ausgewählten Sprint-Backlog-Einträge das Sprintziel unterstützen und welche Arbeiten zur Fertigstellung dieser Backlog-Einträge durchgeführt werden müssen. Dies bedeutet nicht, dass alle (oder irgendeine) der Aufgaben im Sprint-Backlog zu diesem Zeitpunkt zugewiesen werden. Vielmehr organisiert sich das Entwicklungsteam jeden Tag selbst, indem jeder Entwickler eine Aufgabe auswählt (Pull-Modell) und daran arbeitet. (Mehr zu Pull versus Push finden Sie

im Abschnitt »Arbeiten mit dem Sprint-Backlog« weiter hinten in diesem Kapitel.)

Ihr Sprint-Backlog

Das Sprint-Backlog wird in der Sprintplanung erstellt und ist die sortierte Liste der Anforderungen und Aufgaben, die zur Erreichung des Sprintziels notwendig sind.

Ein Sprint-Backlog kann folgende Informationen enthalten:

- ✓ Das Sprintziel und die Termine
- ✓ Eine priorisierte Liste der Anforderungen (zum Beispiel User Stories), die im Sprint entwickelt werden sollen
- ✓ Der geschätzte Aufwand (das heißt die Story Points), der erforderlich ist, um jede einzelne Anforderung zu entwickeln
- ✓ Die Aufgaben, die erforderlich sind, um die einzelnen Anforderungen zu entwickeln
- ✓ Die geschätzten Stunden für jede Aufgabe (falls erforderlich)
- ✓ Ein Burndown-Diagramm, das den Status der im Sprint entwickelten Arbeit zeigt

Der Vorteil des Burndown-Diagramms

Burndown-Diagramme ermöglichen die visuelle Darstellung des Fortschritts eines Sprints. Sie zeigen den Umfang der geleisteten Arbeit im Vergleich zu dem, was noch zu tun bleibt.

[Abbildung 5.6](#) zeigt ein Beispiel:

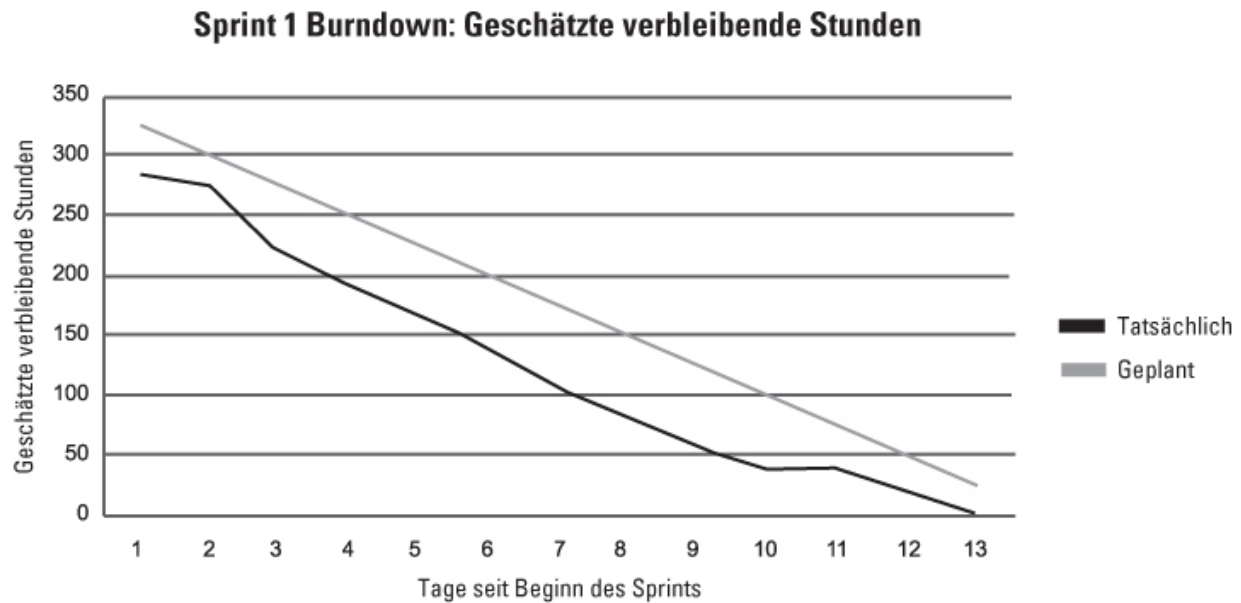


Abbildung 5.6: Sprint Burndown-Diagramm

- ✓ Die vertikale Achse stellt die noch zu erledigende Arbeit dar.
- ✓ Die horizontale Achse stellt die noch verfügbare Zeit im Sprint dar.

Ihre Sprints zeigen eine diagonale Linie von der linken oberen Ecke zur rechten unteren Ecke, die zeigt, wie ein gleichmäßiger konsistenter Burndown aussehen würde. In den meisten Fällen werden Sie in der Praxis jedoch kein perfekt gleichmäßiges Burndown vorfinden.

Einige Burndown-Diagramme verwenden eine weitere Linie, die die noch ausstehenden Story Points abbildet. So können Sie schnell und einfach den Status Ihres Sprints sowohl aus zeitlicher als auch aus relativer Sicht betrachten.

Sie können Ihr eigenes Burndown-Diagramm mit Microsoft Excel erstellen oder das aus der Sprint-Backlog-Vorlage verwenden, die Sie von dieser Seite herunterladen können:

<https://platinumedge.com/blog/anatomy-sprint-backlog>.

Das Burndown-Diagramm wird aus dem Sprint-Backlog generiert. Das Sprint-Backlog sollte täglich aktualisiert werden und nur das Entwicklungsteam kann dies tun. Am Ende eines jeden Tages

aktualisiert jeder Entwickler seine Aufgabe (gleichgültig ob auf einer Karteikarte, in einer Tabellenkalkulation oder in einem elektronischen Tool), indem er die Anzahl der verbleibenden Stunden (nicht die Anzahl der abgeschlossenen Stunden) eingibt, die zur Erfüllung der Aufgabe verbleiben. Das war's schon. Eine Zahl. Es dauert Sekunden und die Ergebnisse sind von unschätzbarem Wert. In [Abbildung 5.7](#) finden Sie ein Beispiel für ein Sprint-Backlog.

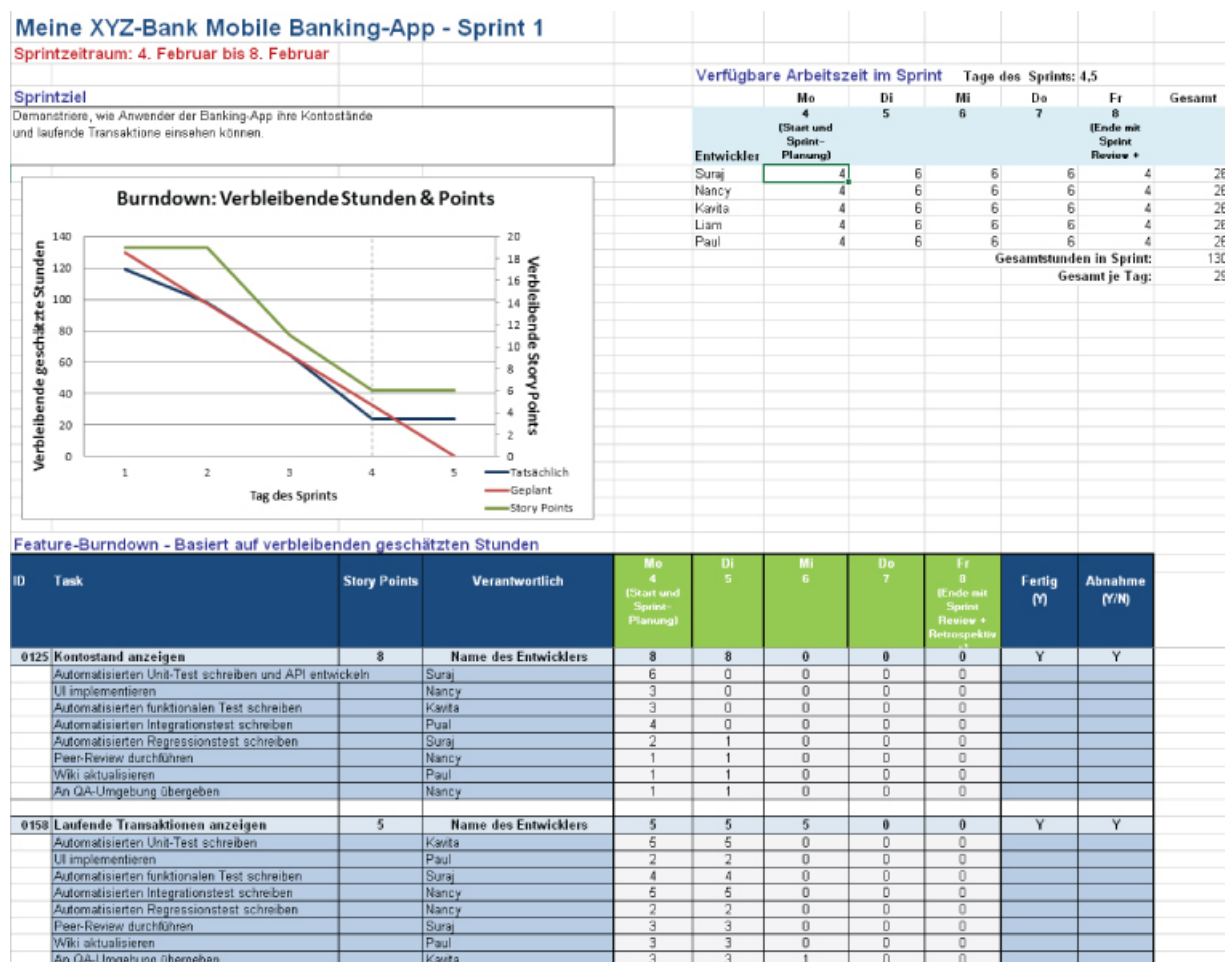


Abbildung 5.7: Das Sprint Backlog ist ein wichtiges Scrum-Artefakt.

Das Sprint-Burndown-Diagramm ist ein sogenannter Informationsverbreiter (Information Radiator), der jeden, der es wissen will, über den Status des Sprints informiert. Burndown-Diagramme werden automatisch generiert, wenn die Mitglieder

des Entwicklungsteams am Ende eines jeden Tages die verbleibende Zeit für ihre eine aktive Aufgabe aktualisieren.

Das Burndown-Diagramm zeigt die Summe der verbleibenden Zeit aller noch offenen Anforderungen im Sprint-Backlog. Verglichen mit der Trendlinie bietet es einem Scrum-Team ein tägliches Level an Statusdetails, das man mit herkömmlichen Projektmanagement-Techniken nicht erreichen kann.

Die Backlog-Kapazität festlegen

Wie viel Kapazität hat ein Tag? Wenn Sie sich die Anzahl der Stunden pro Tag ansehen, die ein Mitglied des Entwicklungsteams seiner Hauptaufgabe widmen kann – der Entwicklung! – weniger als acht. Jede Organisation hat einen gewissen Overhead. Wir stellen fest, dass in den meisten Unternehmen zwischen fünf und sieben Stunden am Tag effektiv gearbeitet werden kann.



Durchschnittlich werden 16 Stunden pro Woche für unklare Zielsetzungen, schlechte Teamkommunikation und ineffektive Meetings verschwendet.

Wie viel Kapazität hat ein Sprint wirklich? In einem einwöchigen Sprint verbringen Scrum-Teams bis zu zwei Stunden mit der Sprint-Planung, bis zu einer Stunde im Sprint-Review und bis zu 45 Minuten in der Sprint-Retrospektive. Das sind etwa vier Stunden, die in Sprint-Meetings verbracht werden. (Müssen Sie alle vier Stunden nutzen? Nein. Können Sie das Limit für ein bestimmtes Meeting überschreiten? Nein.)

Das macht vier der fünf Scrum-Ereignisse aus (gehen Sie davon aus, dass ein maximal 15 Minuten dauerndes Daily Scrum die Entwicklungszeit nicht beeinflusst), aber vergessen Sie nicht die Ausarbeitung des Produkt-Backlogs. Unserer Erfahrung nach wenden Entwicklungsteams durchschnittlich 10 Prozent ihrer Zeit pro Sprint für die Verfeinerung des Produkt-Backlogs auf. Dies entspricht etwa drei bis vier Stunden in einem einwöchigen Sprint.

So verbringt jeder Entwickler bei einem einwöchigen Sprint zwischen sieben und acht Stunden in Sprint-Events, was einen ganzen Arbeitstag für eine effiziente Organisation und etwa eineinhalb Tage für eine weniger effiziente Organisation bedeutet.



Gibt es in Scrum irgendwelche Puffer? Sicher.

Angenommen, einem Entwicklungsteam stehen in einem Sprint 165 Stunden zur Verfügung. Sie sollten nicht damit rechnen, dass 164 Stunden davon genau nach Plan laufen. Der Puffer ist von Team zu Team unterschiedlich, aber Sie sollten ihn transparent machen.

Die Kapazität für einen Entwickler für einen einwöchigen Sprint beträgt also zwischen 18 und 27 Stunden, abhängig davon, wie viele Stunden der effektive Arbeitstag der Organisation ausmacht. Berücksichtigen Sie dies bei der Ermittlung der Kapazität eines Entwicklungsteams während der Sprintplanung. Dies setzt voraus, dass keine bezahlten Feiertage, Ferien oder andere Auszeiten geplant sind, die die Entwickler an der Entwicklung hindern.

Wer sagt, dass Scrum führungslos ist? Disziplinierter kann man nicht sein.

Ein engagierter und effektiver Scrum Master hat eine unglaubliche Wirkung auf die Kapazität eines Entwicklungsteams. Durch die Beseitigung der organisatorischen Hindernisse, kann die effektiven Arbeitszeit pro Tag von fünf auf sieben Stunden erhöht werden. So summiert sich die zusätzliche effektive Arbeitszeit in einem einwöchigen Sprint auf weitere neun Arbeitsstunden pro Entwickler. Bei einem siebenköpfiges Entwicklungsteam ist das eine potenzielle Effizienzsteigerung von 63 Stunden. Scrum Master schaffen Mehrwert.

Was passiert, wenn das Entwicklungsteam am Ende der Sprint-Planung feststellt, dass die Anzahl der geschätzten Stunden für ihre Aufgaben aus dem Sprint-Backlog größer ist als die Kapazität des Teams? Versteckt es sich und macht Überstunden? Nein, der Product Owner hat eine Entscheidung zu treffen: Welche Sprint-

Backlog-Einträge werden zurück in das Produkt-Backlog verschoben, damit die Anzahl der benötigten Gesamtstunden wieder kleiner ist als die Kapazität des Entwicklungsteams? Anhaltende Überstunden führen zu schlechter Teammoral, schlechterer Qualität und langfristigen Produktivitätsverlusten.



Der Wert des iterativen Planungsprozesses ist innerhalb der Sprint-Planung gut sichtbar. Bis die zu erledigende Arbeit skizziert und auf die Aufgabenebene heruntergebrochen ist, haben Sie dies auf eine Weise getan, die Zeitverschwendung minimiert und den Unternehmenswert sowie den ROI maximiert. Denn die Roadmap, die von der Produktvision bis zum Sprint reicht, hat eine kontinuierliche Priorisierung und progressive Ausarbeitung der jeweils wichtigsten Einträge des Produkt-Backlog ermöglicht.

Arbeiten mit dem Sprint-Backlog

Oft stellen wir fest, dass Entwicklungsteams abgelenkt werden und aufgrund häufig anzutreffender Fehler das Ziel verfehlen. Befolgen Sie diese Praktiken, um diesen Fehlern bei der Arbeit mit dem Sprint-Backlog entgegenzuwirken:

- ✓ Stellen Sie sicher, dass die Anforderungen in Aufgaben aufgeteilt sind, die genau und vollständig Ihrer Definition of Done entsprechen (siehe [Kapitel 4](#)).
- ✓ Der Product Owner sollte eine Anforderung erst dann akzeptieren, wenn sie die Sprint-Definition-of-Done vollständig erfüllt.
- ✓ Das gesamte Entwicklungsteam arbeitet im Idealfall immer nur an einer Anforderung und stellt diese fertig, bevor es mit einer weiteren beginnt. Diese Technik wird Swarming genannt. Das Swarming kann durch folgende Aktivitäten erreicht werden:
 - Jedes Teammitglied arbeitet an unterschiedlichen Aufgaben, die sich alle auf die gleiche Anforderung

beziehen.

- Zwei Personen arbeiten als Paar an einer Aufgabe, um die Qualität zu sichern.
- Teammitglieder nutzen das Shadowing, um die Cross-Funktionalität zu erhöhen.

Da die Entwicklungsteams jeweils um eine Anforderung schwärmen, ist die Cross-Funktionalität gewährleistet, und jeder Sprint hat am Ende etwas Greifbares zu bieten.

- ✓ Jede Anforderung muss vollständig entwickelt, getestet, integriert und vom Product Owner abgenommen werden, bevor das Team mit der nächsten Anforderung weitermacht.
- ✓ Weisen Sie einzelnen Mitgliedern des Entwicklungsteams nicht mehrere Aufgaben zu.

Push versus Pull

Traditionelles Projektmanagement folgt dem Push-Modell der Zuweisung von Aufgaben an Einzelpersonen, wenn diese identifiziert werden. Jeder Einzelne verwaltet und konzentriert sich ausschließlich auf die Aufgaben in seiner persönlichen Warteschlange. Diese Warteschlange baut sich im Laufe der Zeit auf und es wird versucht, die Aufgabenlast auf die Teammitglieder zu verteilen, um eine Über- oder Unterbelastung zu vermeiden. Die Schwierigkeit bei Push-Systemen ist, den Status der Dinge zu kennen, außer es wurde noch nicht begonnen oder schon alles fertiggestellt. Dies trägt tendenziell auch dazu bei, dass die Teammitglieder in Silos arbeiten und nicht als funktionsübergreifende Teams.

Pull-Modelle wie Scrum verwalten alle Arbeiten in einer gemeinsamen Warteschlange und die Elemente werden von den einzelnen Entwicklern von oben nach unten aus der Warteschlange entnommen (was eine korrekte Reihenfolge und Priorisierung durch den Produktbesitzer voraussetzt). Sie benötigen keine Umverteilung der Aufgabenlast und bieten den Entwicklern eine natürliche Pause, um zu evaluieren, wo

sie anderen bei offenen Aufgaben helfen können, bevor sie eine neue »ziehen«. Außerdem arbeiten Entwicklungsteams bei einem Pull-Modell mit größerer Wahrscheinlichkeit viel näher, wenn nicht sogar genauer, an der ursprünglichen Reihenfolge als bei einem Push-System. Engpässe sind seltener und der gesamte Workflow ist besser vorhersehbar.

Push-Systeme fördern individuell zentrierte (Silo-)Ziele. Pull-Systeme unterstützen die Zusammenarbeit zur Erreichung produktzentrierter Ziele.

Zum Beispiel können Sie im Lebensmittelgeschäft zwischen einer Push- und einer Pull-Option wählen, wenn Sie zur Kasse gehen:

- ✓ Die Push-Option wird gewählt, wenn man sich an eine Kassenschlange mit einer für diese zuständigen Schlange Kassiererin anstellt. Die Schlange wird Ihnen »zugewiesen«, weil Sie sie gewählt haben und dort anstehen. Sie stecken dort fest, es sei denn, eine andere Warteschlange leert sich und Sie können sich dort anstellen, bevor es jemand anderes tut.
- ✓ Sie können auch das Pull-System wählen und so Ihre Chancen erhöhen, so schnell wie möglich das Geschäft verlassen zu können. Wählen Sie dazu die Self-Checkout-Option, bei der vier bis sechs einzelne Selbstbedienungskassen zur Verfügung stehen. Es gibt nur eine Warteschlange. Wenn eine Kasse frei wird, geht der erste in der Warteschlange zu dieser Kasse. Statt sich auf eine bestimmte Kassenschlange festzulegen, können Sie die nächste freie Selbstbedienungskasse nutzen, sobald sie frei ist.

Das Entwicklungsteam koordiniert täglich die Prioritäten und entscheidet, wer was macht. Ein Entwickler sollte immer nur an einer Aufgabe arbeiten, bis diese vollständig erledigt ist. Dies wird als *Pull-Mechanismus* bezeichnet. Greifen Sie nicht auf die

traditionelle Methode eines Managers zurück, der Teammitgliedern Aufgaben zuweist.



Das Ausschwärmen auf die Anforderungen entstammt dem Lean-Konzept der Work in Progress (WIP)-Limits. Wenn ein Entwicklungsteam gleichzeitig viele unterschiedliche Arbeiten zu erledigen hat, verzögert es das Ergreifen der notwendigen Maßnahmen, um diese Arbeit abzuschließen. Ihr WIP-Limit sollte idealerweise nur eine Anforderung zur gleichen Zeit für das Entwicklungsteam und nur eine Aufgabe zur gleichen Zeit pro Entwickler sein. Das Entwicklungsteam stellt in der Regel fest, dass so ihre Aufgaben schneller erledigt werden, als wenn sie alle gleichzeitig begonnen hätten. Nur eine Anforderung gleichzeitig offen zu haben, ist auch ein effektiver Weg, Prozessengpässe aufzudecken, die dann adressiert und behoben werden können, um einen schnelleren Durchsatz zu erhalten.

Sprints priorisieren

Jeder Sprint hat seinen eigenen Lebenszyklus, wie in [Abbildung 5.5](#) weiter vorne in diesem Kapitel gezeigt. Innerhalb jedes Sprints hat auch jede Anforderung ihre eigene Priorisierung und ihren eigenen Lebenszyklus. Jede Anforderung und Aufgabe wird entwickelt, getestet, integriert und genehmigt, bevor das Team zum Element mit der nächsthöheren Priorität übergeht. Dies ist in [Abbildung 5.8](#) dargestellt.

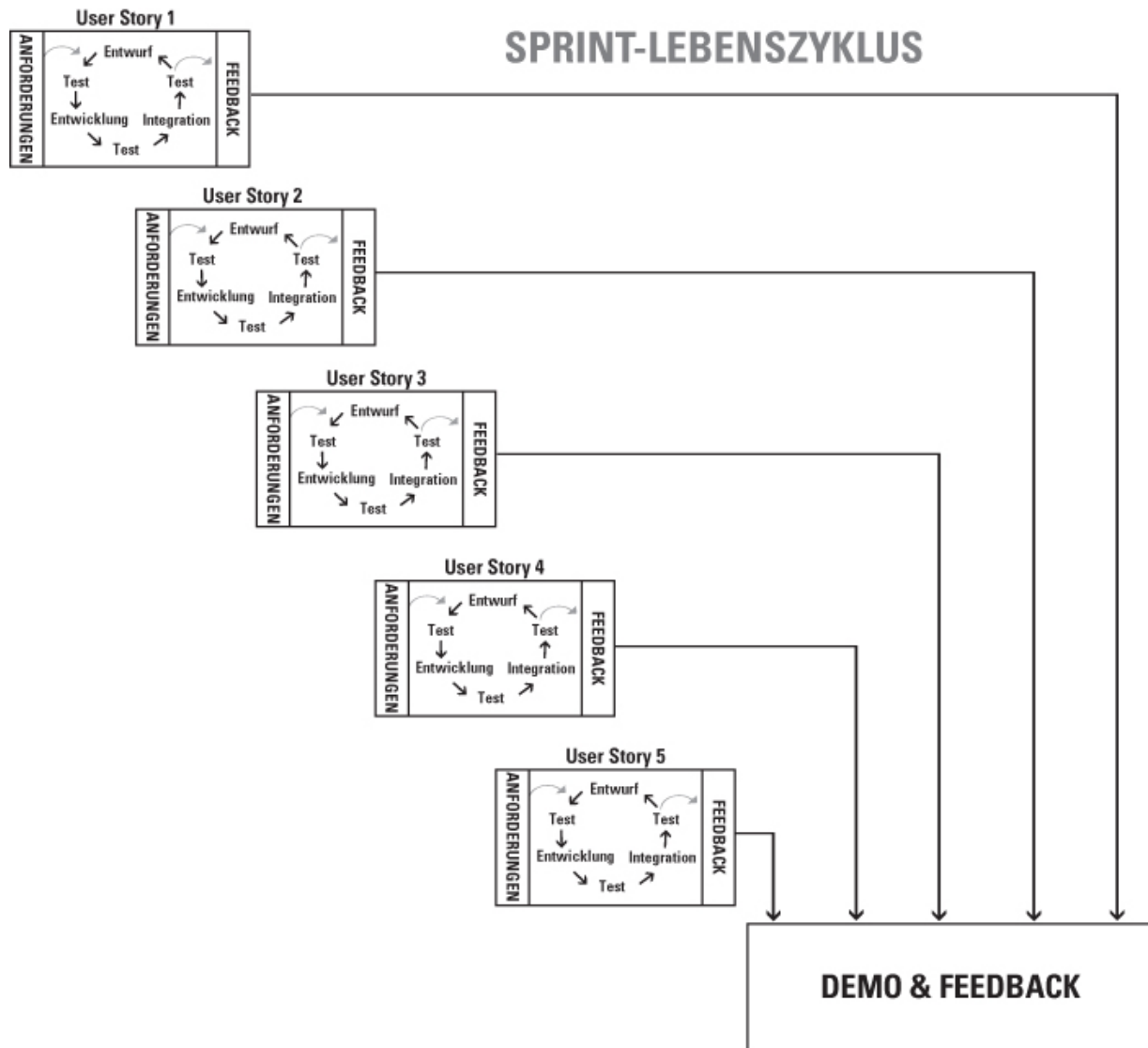


Abbildung 5.8: Priorisierung innerhalb eines Sprints

Die Einträge im Sprint-Backlog werden von oben nach unten priorisiert und in dieser Reihenfolge entwickelt. Das Entwicklungsteam arbeitet jeweils nur an einer Anforderung. Wenn diese Anforderung erfüllt ist, macht das Team mit dem Eintrag mit der nächstniedrigen Priorität weiter, anstatt einen beliebigen anderen aus der Liste auszuwählen, der einfacher oder interessanter sein könnte.

Kapitel 6

Das Beste aus Sprints herausholen

IN DIESEM KAPITEL

- Nutzen aus dem Daily Scrum ziehen
- Fertig werden durch Swarming
- Effektive Sprint-Reviews durchführen
- Kernelemente der Sprint-Retrospektive entdecken

Sprints sind die Essenz von Scrum, daher lohnt es sich, ihnen ein ganzes Kapitel zu widmen. Das Wesentliche wissen Sie bereits: Sprints sind Zeitfenster mit fester Dauer, die so gestaltet sind, dass Ihr Entwicklungsteam einen Entwicklungsrhythmus entwickeln kann. Außerdem hegen und pflegen Sprints die Prämisse der Überprüfung und Anpassung.

Aber das ist nicht alles, was Sprints können. In diesem Kapitel stellen wir Ihnen das Daily Scrum vor – unschätzbare 15 Minuten täglich, in denen Sie Ihre kurzfristigen Ziele wie nie zuvor fokussieren und organisieren. Moderiert vom Scrum Master hält das Daily Scrum Ihr Projekt auf Kurs, da das Scrum-Team mit Hindernissen umgeht und die Prioritäten des Tages koordiniert.

Dieses Kapitel stellt Ihnen außerdem das Sprint-Review und die Sprint-Retrospektive vor. Diese beiden Meetings heben das Konzept von Überprüfung und Anpassung auf eine neue Ebene. Stakeholder überprüfen das entwickelte Produkt und geben dem Product Owner sofortiges Feedback; das Scrum-Team beurteilt selbst, wie der Sprint verlief, und nimmt Verbesserungen am Prozess vor.

Daily Scrum: Phase 5

Das Daily Scrum ist eines der fünf Scrum-Ereignisse und Phase 5 der Roadmap to Value (siehe [Abbildung 6.1](#)).

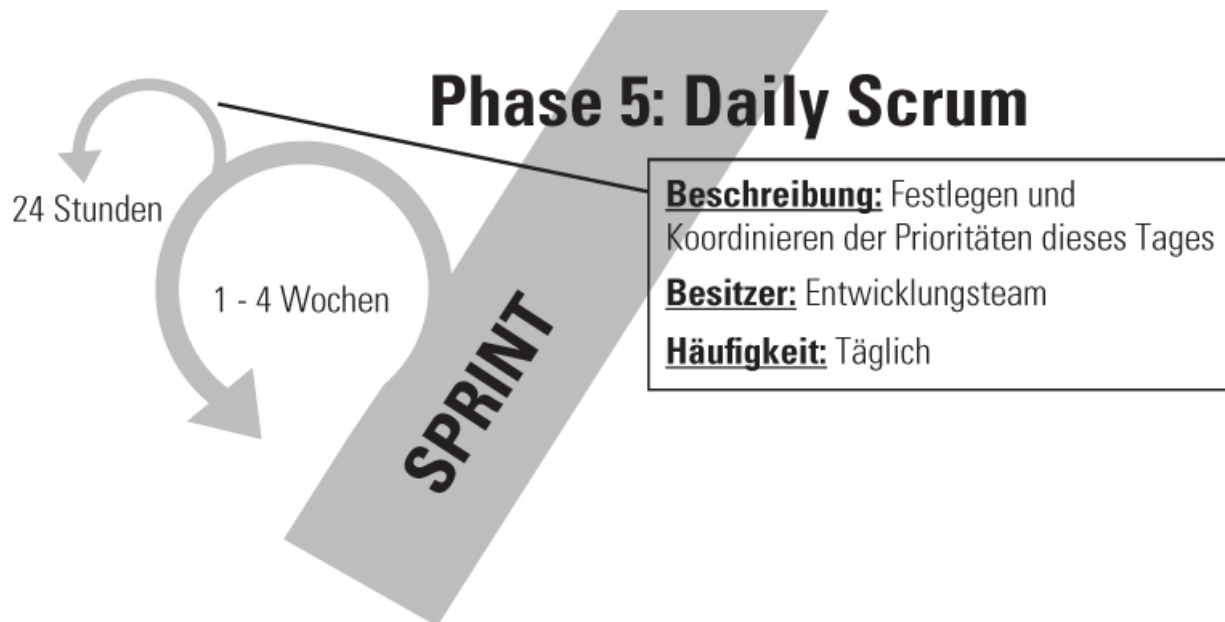


Abbildung 6.1: Das Daily Scrum ist ein integraler Bestandteil des Sprints und Phase 5 der Roadmap to Value.



Bei Scrum ist Planung extrem wichtig. Man setzt sich kein Ziel, vergisst es und kommt dann erst sechs Monate später wieder zusammen, um zu sehen, was passiert ist. Bei Scrum nehmen Sie tägliche Überprüfungen und Anpassungen vor, und dies während des gesamten Tages.

Da ein Entwicklungsteam jeden Tag »ausschwärmt«, um eine einzelne Anforderung in Angriff zu nehmen, ist Koordination entscheidend. Die Beseitigung von Hindernissen ist auch für Entwickler von entscheidender Bedeutung, die eng zusammenarbeiten, um innerhalb kurzer Zeiträume potenziell auslieferfähige Produktinkremente abzuliefern. Das Daily Scrum ist das Ereignis, bei dem sich das Entwicklungsteam koordiniert.



Wenn Sie Ihre Projekte wochenweise verwalten (das heißt, mit wöchentlichen Statusberichten, die der Projekt-Manager erstellt und die möglicherweise nicht sofort überprüft werden), dann wird sich Ihr Projekt auch wochenweise verzögern. Auch bei Scrum können sich immer noch Verzögerungen ergeben, aber nur um einen Tag, da Scrum-Teams den Arbeitstag mithilfe des Sprint-Backlogs, des Daily Scrum und durch die tägliche und direkte Interaktion managen. Wir schauen uns diese Unterschiede in [Kapitel 7](#) genauer an.

Definition des Daily Scrum

Wie der Name schon sagt, findet bei jedem Sprint ein Daily Scrum statt. Die hierfür zur Verfügung stehende Zeit beträgt maximal 15 Minuten, unabhängig von der Gesamtlänge des Sprints. In Scrum sollen Meetings, genau wie Artefakte, immer gerade ausreichend sein. Alles, was länger als 15 Minuten dauert, würde wertvolle Entwicklungszeit auffressen. Außerdem sind 15 Minuten ausreichend Zeit, um alles Notwendige zu erledigen.

Der Zweck des Daily Scrum ist es, die Sprintaktivitäten des Tages zu koordinieren und Blockaden zu identifizieren, die das Entwicklungsteam daran hindern, sein Sprintziel zu erreichen. Jedes Mitglied des Entwicklungsteams nimmt daran teil, sodass jeder darüber Bescheid weiß, woran das gesamte Team arbeitet. Wenn während des Daily Scrum ein Hindernis zutage tritt, wird es nach dem Daily Scrum behandelt. Bei diesem Ereignis handelt es sich um ein Koordinationstreffen, es ist kein Problemlösungs- oder Beschwerde-Meeting.



Der Scrum Master beseitigt Hindernisse oder unterstützt deren Beseitigung. Einige Hindernisse müssen vom Product Owner beseitigt werden oder erfordern eine Diskussion zwischen einem Teammitglied und jemandem außerhalb des Teams.

Teilnehmer am Daily Scrum sind die Mitglieder des Entwicklungsteams und der Scrum Master. Der Scrum Master

sorgt dafür, dass das Meeting stattfindet, und das Entwicklungsteam steuert es. Der Product Owner sollte teilnehmen (und muss teilnehmen, wenn er vom Entwicklungsteam ausdrücklich dazu aufgefordert wird). Der Product Owner kann bei Bedarf Erläuterungen zur Priorisierung geben und jeder andere Beteiligte und Interessierte kann zuhören, darf aber nichts sagen. Auf diese Weise können Dritte Nutzen aus der täglichen Transparenz ziehen und in den täglichen Prozess eingebunden werden, aber sie können ihn nicht behindern oder entgleisen lassen.



Das Daily Scrum ist das Verfahren, mit dem sich das Entwicklungsteam selbst organisiert und verwaltet. Jeden Tag entscheidet das Team, wer was tut und wer wem hilft. Die Arbeit wird nicht von einem Projektmanager oder einem anderen Nicht-Entwickler diktiert.

Wenn sich das Daily Scrum wie ein Statusbericht-Meeting anfühlt oder die Mitglieder des Entwicklungsteams beginnen, sich an eine Person zu wenden (beispielsweise einen informellen Teamleiter oder den Scrum Master), hat das Meeting seinen Zweck verfehlt. Das Daily Scrum ist ein Peer-to-Peer-Meeting.

Der Schlüssel zum Erfolg ist die Klarheit darüber, was es bedeutet, an diesem Tag erfolgreich zu sein. Anschließend schwärmt das Team, falls nötig, aus, um die Arbeit mit der höchsten Priorität zu erledigen. Das Team findet in dieser Zeremonie nur dann einen Wert, wenn die Teammitglieder relevante Klarheit und Zielstrebigkeit über die Ziele des Tages erreichen. Daily Scrums bieten die Möglichkeit, im Hier und Jetzt Überprüfungen durchzuführen und Anpassungen vorzunehmen.



Ein häufiges Missverständnis über das Daily Scrum besteht darin, dass dies der Zeitpunkt ist, zu dem das Entwicklungsteam dem Product Owner Bericht erstattet oder zu dem der Product Owner neue Anforderungen einführt oder ein Sprint-Ziel aktualisiert. Lassen Sie das Daily Scrum nicht zu einem Business-Statusbericht-Meeting werden; es ist ein Koordinationstreffen, um eine hohe Performance zu ermöglichen.

Daily Scrum planen

Da das Daily Scrum nur 15 Minuten dauert, müssen alle pünktlich sein und sofort loslegen können. Es gibt einen direkten Zusammenhang zwischen dem späten Beginn eines Meetings und dem fehlenden Fokus, nachdem das Meeting angefangen hat. Sie haben verschiedene Möglichkeiten, die Pünktlichkeit zu fördern, zu denen unter anderem die folgenden gehören:

- ✓ **Starten Sie Ihr Daily Scrum eine halbe Stunde nach Beginn des normalen Arbeitstages.** Dieser Zeitplan gibt Ihrem Entwicklungsteam Zeit, sich Kaffee zu holen, E-Mails zu beantworten, die Erlebnisse des Vorabends zu besprechen und alle anderen Morgenrituale zu durchlaufen.
- ✓ **Ahnden Sie Verspätungen, aber auf eine freundliche und muntere Art und Weise.** Lassen Sie Teammitglieder für jede Minute, die sie zu spät kommen, einen bestimmten Betrag in einen Feierfonds einzahlen oder lassen Sie sie ein Lied aus den Kindertagen singen. Werden Sie kreativ und machen Sie die Strafe unangenehm genug, dass die Verspätungen aufhören.



Ein Scrum-Team in Portland hat die »20-Dollar-oder-20-Liegestütze-mit-Multiplikator«-Methode genutzt. Wenn ein Teammitglied das erste Mal zu spät kam, zahlte es 20 Dollar oder machte 20 Liegestütze. Das nächste Mal, wenn es zu spät kam, zahlte es 40 Dollar oder machte 40 Liegestütze und so weiter. Diese »Strafe« wurde von den Teammitgliedern selbst entwickelt und nicht von außen auferlegt. Dieser Ansatz hat für das Team gut funktioniert und erfolgreich dafür gesorgt, dass keiner mehr zu spät kam.

Ein wichtiger Störfaktor des Daily Scrum ist die Ablenkung. Wir haben mit einigen Teams zusammengearbeitet, deren regelmäßige Gewohnheit es ist, dass sich jedes Mitglied auf seinen eigenen Laptop-Bildschirm konzentriert. Einige Mitglieder nehmen sogar telefonisch teil, während sie zum Büro fahren. Achten Sie auf Indikatoren wie »Können Sie das noch einmal sagen? Ich habe es nicht mitbekommen.« Beim Daily Scrum sollte »kein Einsatz von Elektronik« die Regel sein.

Durchführung des Daily Scrum

Stellen Sie sich ein Scrum-Team vor, das sich zu Beginn des Tages um seinen Sprint-Backlog oder sein Taskboard versammelt. Jede Person sieht auf einen Blick den Fortschritt vom Vortag; dann wählt jede Person proaktiv eine neue Aufgabe für den aktuellen Tag aus. Die Teammitglieder koordinieren, wo Hilfe benötigt wird, um eine Aufgabe vor Ende des Tages zu erledigen; dann gehen sie direkt an die Arbeit.

Jedes Teammitglied sollte drei Aussagen darüber machen, wie es dem Team hilft, das Sprintziel zu erreichen:

- ✓ Gestern habe ich es geschafft, ...
- ✓ Heute konzentriere ich mich auf ...
- ✓ Die Dinge, die mich behindern, sind ...

Wir empfehlen, dass Scrum Master über die Moderation hinaus am Daily Scrum teilnehmen, indem sie sich mit den identifizierten

und/oder in Bearbeitung befindlichen Hindernissen auseinandersetzen. Der Scrum Master könnte, nachdem die Teammitglieder gesprochen haben, folgende Aussagen treffen:

- ✓ Gestern habe ich dieses Hindernis beseitigt.
- ✓ Heute kann ich dieses Hindernis beseitigen.
- ✓ Die Hindernisse, die ich nicht beseitigen kann, sind ..., und ich werde sehen, ob mir dies oder das dabei helfen kann, sie zu eliminieren.

Wie wir in [Kapitel 5](#) erörtern, sollten die Aufgaben so aufgeteilt werden, dass sie an einem Tag oder in einem kürzeren Zeitraum erledigt werden können. Selbst dann, wenn Entwickler tagelang sich selbst überlassen bleiben, ohne sich als Team zu koordinieren und auszuschwärmen, können sie sich in unnötigen Details oder Problemen verstricken, die mit Hilfe anderer leicht zu lösen wären.

Daily Scrums synchronisieren ein Team und jeder geht zur Arbeit, um sich gegenseitig zu helfen, die nötigen Aufgaben zu erledigen. Das Team ist gemeinsam für das Ergebnis verantwortlich. Wenn die Teammitglieder am nächsten Tag wiederkommen, freuen sie sich, über ihre Fortschritte zu berichten.

Daily Scrums effektiver machen

Die folgenden Strategien können dabei helfen, Ihre Daily-Scrum-Meetings schnell und effektiv abzuhalten:

- ✓ Starten Sie immer pünktlich. Im Abschnitt »Daily Scrum planen« weiter vorne in diesem Kapitel finden Sie einige Tipps zum Einhalten der Pünktlichkeit.
- ✓ Führen Sie das Meeting im Stehen durch. Studien haben gezeigt, dass Meetings im Stehen 34 Prozent kürzer sind als solche im Sitzen. Niemand hat die Chance, in einen Stuhl zu fallen und sich zu entspannen; es ist, als ob alle schon in Bewegung wären.

- ✓ Konzentrieren Sie sich auf die Koordination, nicht auf die Problemlösung. Hindernisse werden nach dem Daily Scrum beseitigt.



Wenn im Daily Scrum Hindernisse aufgedeckt werden, kann der Scrum Master mit ihnen umgehen, indem er direkt nach dem Daily Scrum eine After-Party veranstaltet. An dieser Veranstaltung sind nur diejenigen beteiligt, die einbezogen werden müssen, und sie dient dazu, alle Probleme anzugehen, die während des Daily Scrum thematisiert wurden.

Es kann auch ein Backlog dieser »Team-Themen« mitgehalten werden, sodass die Themen vom Team während der After-Party oder während der Team-Retrospektive behandelt werden können. Nicht alle Themen, die während des Daily Standup-Meetings angesprochen werden, müssen auch an diesem Tag behandelt werden.

- ✓ Der Scrum Master ist der Moderator des Meetings und sorgt bei Bedarf dafür, dass das Meeting zeitlich und inhaltlich auf Kurs bleibt. Er gewährleistet, dass nur die Mitglieder des Entwicklungsteams teilnehmen. Die Moderation des Scrum Masters sollte so unauffällig wie möglich sein.
- ✓ Behandeln Sie dringende Probleme und Prioritäten, die sich auf den aktuellen Tag beziehen und die das Erreichen des Sprintziels unterstützen.
- ✓ Versammeln Sie sich um das Taskboard, um Kontext und Fokus zu gewährleisten.
- ✓ Weisen Sie keine feste Sprechreihenfolge zu, denn wenn die Leute die Reihenfolge kennen, neigen sie dazu abzuschalten, bis sie an der Reihe sind. In manchen Fällen tauchen sie erst dann auf, wenn sie an der Reihe sind.



Wir werfen einem beliebigen Mitglied des Entwicklungsteams ein quietschendes Hundespielzeug zu, wenn diese Person sprechen soll. Spricht jemand zu lange, verwenden wir einen Wurfwecker mit Alarm. Eine Alternative wäre, ein Paket DIN A4- Papier (das über 2 Kilo wiegt) zu werfen und die Person solange sprechen zu lassen, wie sie das Paket mit ausgestreckte Armen halten kann. Diese Taktiken sorgen dafür, dass das Daily Scrum schnell und vorwärtsgerichtet bleibt; außerdem machen sie Spaß.

Lassen Sie keine vagen Aussagen zu und verlassen Sie sich nicht auf die Erinnerungen der Teammitglieder daran, was sich im Sprint-Backlog befindet (siehe nächster Abschnitt, »Team-Taskboard«).

Team-Taskboard

Ein Taskboard ist eine Möglichkeit, das Sprint-Backlog darzustellen. Obwohl es für Scrum-Teams üblich ist, ihren Sprint-Backlog digital zu verwalten, brauchen Sie nur etwas Platz an der Wand oder am Whiteboard, Karteikarten, Haftnotizen und Klebeband. [Abbildung 6.2](#) zeigt ein Beispiel für ein Taskboard.

RELEASEZIEL:

SPRINTZIEL:

RELEASEDATUM:

SPRINT-REVIEW:

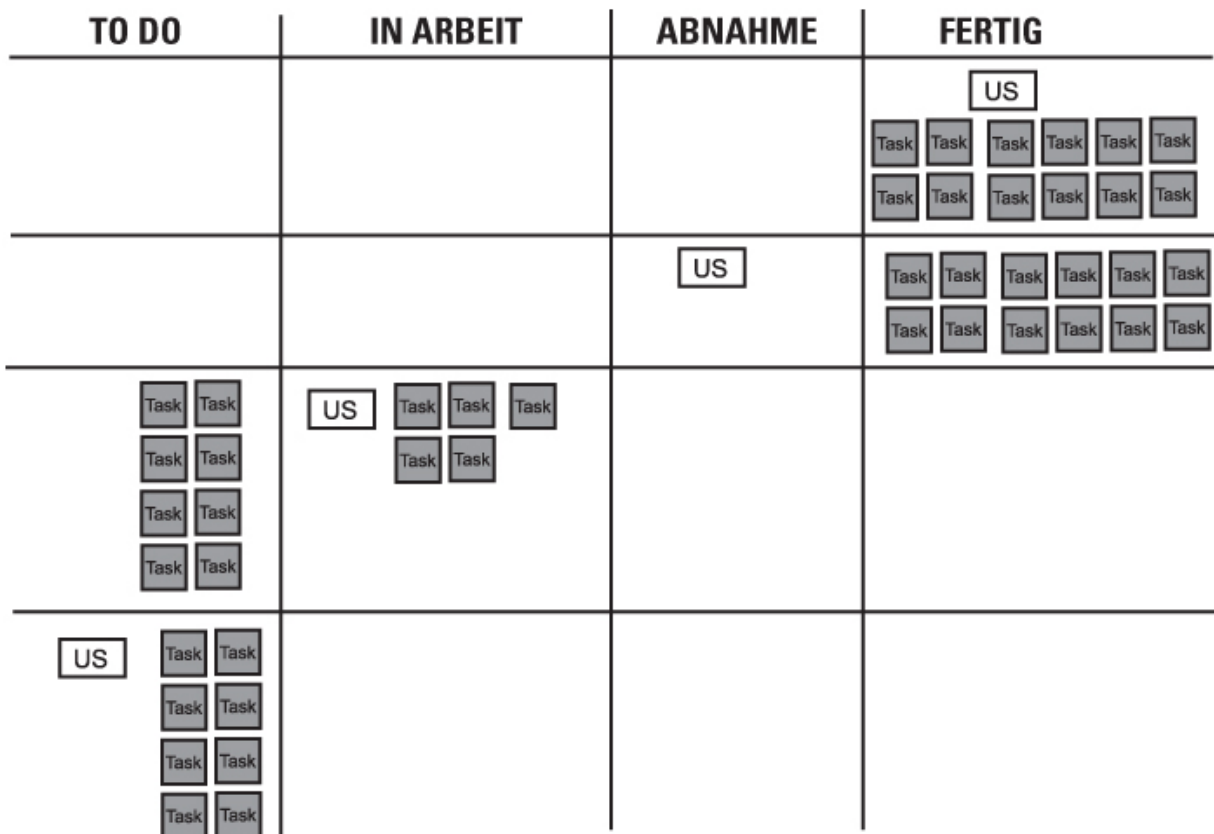
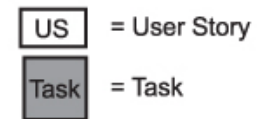


Abbildung 6.2: Ein Team-Taskboard

Das Taskboard erhöht genau wie die Produkt-Roadmap das Engagement und die Flexibilität, weil es greifbar ist.

Ein physisches Taskboard ist hervorragend, weil es eine schnelle und effektive Möglichkeit bietet, den Status eines gesamten Sprints anzuzeigen. Das Taskboard im Blickfeld des Entwicklerteams und des Product Owners zu behalten, stellt sicher, dass jeder sofort weiß, was getan wird, was nicht getan wird und alles dazwischen.

Verwenden Sie diese Grundelemente:

✓ Oben

- Das spezifische Sprintziel
- Das Release-Gesamtziel

Das Releasedatum und Sprinttermine können ebenfalls aufgenommen werden.

✓ Spalten (von links nach rechts)

- *To do*: Anforderungen und Aufgaben, die im Sprint noch entwickelt werden müssen.

Die Entwickler nehmen einen Eintrag vom Anfang der Liste, um mit einer neuen Aufgabe zu beginnen. Wenn zwei Entwickler dieselbe Aufgabe übernehmen wollen, können sie diese als Paar entwickeln; alternativ können sie auch das Shadowing nutzen oder entscheiden, wer diese Aufgabe am besten lösen kann.

- *In Arbeit*: Die Produkt-Backlog-Einträge und Aufgaben, an denen das Entwicklungsteam arbeitet.

Jede Aufgabe kann mit verschiedenfarbigen Punkte oder Aufklebern markiert werden, um den Eigentümer anzugeben oder um Aufgaben zu identifizieren, die durch ein Hindernis blockiert sind. In dieser Spalte sollten außerdem die WIP-Limits angezeigt werden, sofern sie verwendet werden. Nachdem die Entwickler eine Aufgabe erledigt haben, schauen sie hier nach, wem sie helfen können. Andernfalls nehmen sie die nächste Aufgabe aus der Spalte *To do* und überprüfen mit dem Team, ob es die richtige Aufgabe ist.

- *Abnahme*: Anforderungen, die auf die Annahme durch den Product Owner warten.

Wenn die Anforderung abgelehnt wird und genügend Zeit im Sprint verbleibt, geht sie zurück in die Spalte *In Arbeit*. Andernfalls wird die Anforderung wieder in das Produkt-Backlog verschoben und wird bei einem späteren Sprint berücksichtigt (siehe den Abschnitt »Umgang mit nicht fertiggestellten Anforderungen« weiter hinten in diesem Kapitel).

- *Fertig*: Die Anforderungen, die der Product Owner als fertiggestellt akzeptiert hat.

Nur die Mitglieder des Entwicklungsteams können die Anforderungen von *To do* zu *In Arbeit* zu *Abnahme* verschieben und nur der Product Owner kann sie aus der Spalte *Abnahme* in *Fertig* verschieben. Nachdem eine Anforderung vom Product Owner akzeptiert und in die Spalte *Fertig* verschoben wurde, verschiebt das Entwicklungsteam gleichzeitig die betreffenden Tasks. Wenn eine Anforderung abgelehnt wird, verschiebt das Entwicklungsteam die Tasks zurück zu *In Arbeit*, um sie zu überarbeiten, oder es legt neue Aufgaben an und reagiert so auf die Begründung für die Ablehnung der Anforderung.



Anforderungen in der Spalte *Abnahme* dürfen sich nicht anhäufen. Im Idealfall sollte eine Karte, nachdem sie in *Abnahme* platziert wird, noch am selben Tag in *Fertig* platziert oder abgelehnt werden, damit sie weiterentwickelt werden kann. Wenn es zu einer Verzögerung kommt, muss der Product Owner dahingehend gecoacht werden, dass sich die Stories nicht ansammeln, während sie darauf warten, akzeptiert zu werden. Es gibt keinen Grund für die Verzögerung, wenn der Product Owner ein engagiertes Scrum-Teammitglied ist, das jederzeit für eine Klärung zur Verfügung steht, die Anforderungen zu einer einzelnen Aktion oder Integration ausgearbeitet wurden und die Anforderungen die Definition of Done erfüllt haben. Für die Mitglieder des Entwicklungsteams ist es wichtig zu wissen, wann ihre Arbeit getan ist, damit sie mit der nächsten Anforderung weitermachen können.

Das Kanban-Board

Das Wort *Kanban* kommt aus dem Japanischen und bedeutet Schild, Plakat, visuelles Signal oder Karte. In den 1950er Jahren formalisierte Toyota dieses Konzept, um den Fluss der Lagerbestandteile in seinen Produktionslinien zu

standardisieren. Im Wesentlichen enthalten Kanban-Boards Karten, die einzelne Werke darstellen. Jede Karte dient als Statusmeldung und zeigt an, wann neue Arbeiten angenommen werden können.

Diese Praxis hat dazu beigetragen, schlanke Denk- und Fertigungspraktiken zu inspirieren und viele agile Teams verwenden heute so etwas wie ein Kanban-Board-System. Taskboards funktionieren ähnlich wie Kanban-Boards, indem sie einen visuellen Status (Signal) liefern, wo genau sich jede Aufgabe (Arbeit) im Gesamtprozess befindet. Diese Transparenz erleichtert es den Entwicklungsteams, Disziplin zu wahren und die Anforderungen zu erfüllen.

Swarming

In [Kapitel 5](#) haben wir das Swarming-Konzept im Zusammenhang mit dem Sprint-Backlog vorgestellt. Beim Swarming arbeiten alle Mitglieder des Entwicklungsteams während des Sprints jeweils nur an einer Anforderung. Obwohl dieses Prinzip nicht spezifisch für Scrum ist, ist es ein so effektiver Weg für Teams, ihren Sprint-Backlog abzuarbeiten, dass es hier erörtert werden muss.

Einer der Hauptvorteile von Scrum besteht darin, dass die Entwicklungsteams die Anforderungen so erstellen, dass sie ihre eigene Definition of Done erfüllen, um so innerhalb einer relativ kurzen Zeitspanne ein potenziell auslieferfähiges Produktinkrement zu produzieren. Das Team überprüft den Prozess auf der Grundlage der gewonnenen Erkenntnisse und wiederholt diesen Zyklus immer wieder. Ziel ist es, so viele Anforderungen wie möglich fertigzustellen, und nicht, mit der Bearbeitung von so vielen Anforderungen wie möglich zu starten, ohne sie jedoch fertigzustellen.

Durch Swarming können die Teams die folgenden Vorteile nutzen:

- ✓ Maximierung der Erfolgchancen, weil die Fähigkeiten und Fertigkeiten des gesamten Teams auf eine einzige Anforderung ausgerichtet sind.

- ✓ Fertigstellung des Zyklus von Planung, Entwurf, Entwicklung und Testen bis zur Fertigstellung für jede Anforderung.
- ✓ Probleme und Hindernisse heute lösen.
- ✓ Dramatische Reduzierung der Einführung von Fehlern in ein Produkt durch Pairing und Single-Tasking (versus Multitasking).
- ✓ Eliminierung von Single Points of Failure in den Bereichen Wissen, Prozesse und Kompetenzen.
- ✓ Fertigstellung der wichtigsten Anforderungen komplett und zuerst.

Wenn Teammitglieder sehen, dass alle Mitentwickler an einer Aufgabe arbeiten und es keine weiteren Aufgaben für die gleiche Anforderung (die User Story) mehr gibt, ist es ganz natürlich, dass sie es für produktiver halten, mit der Entwicklung einer neuen Anforderung zu beginnen, als anderen Entwicklern bei der laufenden Anforderung zu helfen. Diese Tendenz kann jedoch aus dem Ruder laufen und dazu führen, dass Teams mehrere Anforderungen begonnen haben, aber keine einzige fertiggestellt ist. Durch Shadowing, Paarprogrammierung, Recherchieren oder Unterstützung in welcher Form auch immer vermeiden Entwicklungsteams dieses Risiko.

Dieser Prozess stellt sicher, dass bei jedem Sprint *etwas* komplett entwickelt wird und den Stakeholdern zur Verfügung steht. Jeder Sprint bringt auslieferfähige Ergebnisse. Die Anstrengungen des Entwicklungsteams sind fokussiert, die Teamarbeit wird verstärkt und der iterative Prozess von Scrum wird mit Leben gefüllt.



Bleiben Sie fokussiert. Hören Sie auf, nur zu beginnen, sondern fangen Sie an, die Dinge zu beenden.

Umgang mit nicht abgenommenen Anforderungen

Wenn eine Anforderung in der Spalte *Abnahme* vom Product Owner abgelehnt wird, haben die Entwickler zwei Möglichkeiten:

- ✓ **Die aktuellen Aufgaben beenden und dann auf die abgelehnte Anforderung ausschwärmen:** Diese Option könnte dann besser sein, wenn im Sprint genügend Zeit bleibt, um sowohl die aktuellen Aufgaben als auch die abgelehnte Anforderung fertigzustellen.
- ✓ **Die Arbeit an den aktuellen Aufgaben beenden, um auf die abgelehnte Anforderung auszuschwärmen:** Diese Option könnte dann besser sein, wenn im Sprint nicht genügend Zeit bleibt, um sowohl die aktuellen Aufgaben als auch die abgelehnte Anforderung zu beenden.

Der Product Owner entscheidet über die Priorität, wenn er mit dieser Entscheidung konfrontiert wird. Neben der im Sprint verbleibenden Zeit können weitere Faktoren die Entscheidung des Product Owners beeinflussen. Weil das Team sein Lernen überprüft und sich während eines Sprints anpasst, kann es sein, dass die abgelehnte User Story für das Erreichen des Sprintziels weniger wertvoll ist als die Anforderung, die gerade bearbeitet wird. Hierdurch kann das Risiko, dass die aktuelle Anforderung nicht fertiggestellt wird, höher sein kann als das Risiko, die abgelehnte Anforderung nicht fertigzustellen.

In jedem Fall bleibt das gesamte Scrum-Team (einschließlich des Product Owners) während des gesamten Sprints durch die enge tägliche Abstimmung mit dem Product Owner fokussiert und auf Kurs.



Scrum-Teams sollten sich immer selbst pushen. Wenn Entwicklungsteams jedes Mal 100 Prozent ihres Sprint-Backlogs bewältigen, gehen sie vielleicht nicht an ihre Grenzen. Ein hoher Prozentsatz fertiggestellter Sprint-Backlog-Einträge sollte das Ziel sein, aber man sollte nicht erwarten, dass Scrum-Teams jedes Mal 100 Prozent erreichen. Scrum-Master helfen dem Scrum-Team ähnlich wie Luftfahrtingenieure, Wege zu finden, den Luftwiderstand zu reduzieren, um in jedem Sprint effektiver zu werden und mehr zu erreichen. Solange die Teams fertigstellen, was sie mit jedem Sprint beginnen, und die Velocity erhöhen, erkennen sie den Vorteil der kontinuierlichen Verbesserung von Scrum.

Multitasking und Thrashing

Vor kurzem hat Microsoft eine Studie über die Auswirkungen von Multitasking durchgeführt, die zeigt, dass Multitasking nicht funktioniert. Im Durchschnitt dauert es 15 Minuten, Ihr Gehirn wieder auf das Niveau zu bringen, auf dem es war, bevor Sie den Anruf oder die E-Mail beantwortet haben. Studien haben ebenfalls gezeigt, dass eine Unterbrechung von nur 4,4 Sekunden die Anzahl der Fehler verdreifacht, die bei nachfolgenden Aufgaben gemacht wurden, die eine Sequenzierung erfordern. Die Reduzierung von Multitasking in Ihrem Entwicklungsteam verschafft Ihnen einen soliden Vorsprung bei der Erreichung der 30 bis 40 Prozent schnelleren Markteinführung eines Produkts, die wir so oft erlebt haben.

Thrashing tritt auf, wenn Entwickler zwischen Projekten, Anforderungen und Aufgaben hin und her springen und den Kontext dabei wechseln. Thrashing erhöht die Zeit, die benötigt wird, um Aufgaben zu erledigen, um 30 Prozent. Wenn Sie nicht genug Leute haben, die die Arbeit als engagierte Entwickler im Schwarm übernehmen können, dann

haben Sie auch nicht genügend Zeit, sie auf mehrere Tasks zu verteilen.

Umgang mit nicht fertiggestellten Anforderungen

Selbst hochfunktionale Entwicklungsteams, die gut schätzen, schwärmen und sich während jedes Sprints an ein Work-in-Progress(WIP)-Limit von 1 halten, können am Ende eines Sprints mit nicht fertiggestellten oder nicht begonnenen Anforderungen im Sprint-Backlog enden. Dieses Ergebnis kann in Ordnung sein, wenn die Teammitglieder die Anforderungen mit höchsten Prioritäten bis zur Fertigstellung erfüllt haben und über funktionierende Produktinkremente verfügen, die ausgeliefert werden können.

Aber was macht das Team mit den restlichen Anforderungen?

Wenn eine Anforderung nicht begonnen oder begonnen und nicht fertiggestellt wurde, legt der Product Owner sie vollständig in das Produkt-Backlog zurück (natürlich unter Beibehaltung aller Notizen, Aufgaben und Dokumentationen) und priorisiert sie dann gegenüber dem Rest des Produkt-Backlogs. Der Product Owner kann die Anforderung entsprechend ihrer neuen Priorität einem zukünftigen Sprint zuordnen.

Basierend auf den während des Sprints abgeschlossenen Aufgaben kann es sein, dass die nicht begonnene oder nicht fertiggestellte Anforderung nicht mehr notwendig ist oder dass sie einen nicht mehr so hohen Wert hat wie zuvor. Was erledigt wurde, könnte ausreichen; es kann dann an der Zeit sein, zu einer anderen Funktion überzugehen.

Gleichgültig welcher Aufwand auch immer für die Anforderung erbracht wurde, sie wird beim Ermitteln der Velocity des Teams für diesen Sprint berücksichtigt. Wenn es die Anforderung in einen zukünftigen Sprint schafft, muss sie ausgearbeitet, geklärt und auf der Grundlage der verbleibenden Arbeit neu geschätzt werden. Sie können keine Story Points sammeln oder zwischenspeichern.

Eine Ausnahme kann auftreten, wenn Sie nach der Bearbeitung einer unvollendeten Anforderung feststellen, dass Sie diese aufteilen können. Sie beenden einen Teil der Anforderung während des Sprints, der andere Teil geht zurück in das Produkt-Backlog, um neu bewertet und priorisiert zu werden.

Die Lektion besteht darin, Swarming zu nutzen, um während des Sprints zu komplett fertig zu werden.

Sprint-Review: Phase 6

Die nächste Station unserer Roadmap ist Phase 6, das Sprint-Review (siehe [Abbildung 6.3](#)). Dieses Scrum-Ereignis ist integraler Bestandteil des Überprüfungs- und Anpassungsprozesses von Scrum und findet am Ende eines jeden Sprints statt.

Phase 6: Sprint-Review

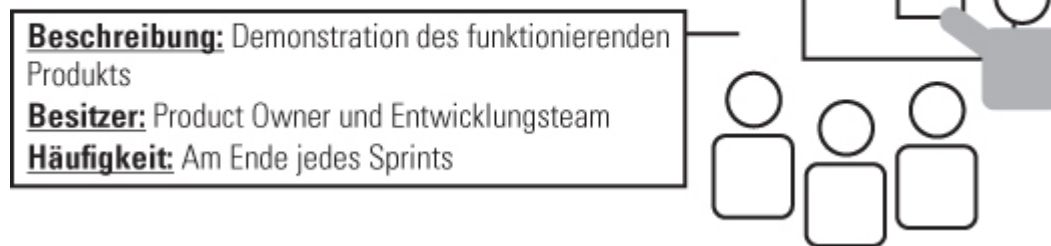


Abbildung 6.3: Der Sprint-Review ist ein Scrum-Event und Phase 6 der Roadmap to Value.

Der Zweck des Sprint-Reviews besteht darin, dass der Product Owner ein organisatorisches Feedback darüber erhält, ob sich die Entwicklung des Produkts in die richtige Richtung bewegt. Diese Überprüfung ist auch eine großartige Gelegenheit für das Entwicklungsteam, sich zu erheben und zu zeigen, was es erreicht hat. Teammitglieder erhalten volle Anerkennung für das, was sie erreicht haben und was nicht.

Dieses Treffen am Ende eines jeden Sprints stellt sicher, dass die Stakeholder über die Ergebnisse des Sprints auf dem Laufenden sind und ein Forum haben, in dem sie ihr Feedback im Beisein des Entwicklungsteams direkt an den Product Owner weitergeben

können. Auch die Stakeholder haben ein funktionierendes, nutzbares Produkt in der Hand.

Der Sprint-Review-Prozess

Das Sprint-Review, dessen Dauer auf eine Stunde je Sprintwoche begrenzt ist, findet am Ende des letzten Tages des Sprints statt. Berücksichtigen Sie diesen Zeitaufwand während der Sprint-Planung.

Die Teilnehmer am Sprint-Review sind das gesamte Scrum-Team und die Stakeholder; sie nehmen diese Rollen ein:

- ✓ **Scrum Master:** Unterstützt das Meeting und stellt sicher, dass es fokussiert bleibt und die Zeit eingehalten wird.
- ✓ **Product Owner:** Stellt kurz das Sprint-Ziel vor und wie gut das Scrum-Team das Ziel erreicht hat, informiert die Stakeholder darüber, welche Elemente aus dem Backlog fertiggestellt wurden und fasst zusammen, was für das anstehende Release noch zu tun ist.



Das Sprint-Review ist nicht der Ort, an dem der Product Owner Feedback über die fertiggestellte Funktionalität gibt. Bei diesem Ereignis erhält der Product Owner Feedback von den Stakeholdern über die Richtung, die das Produkt nimmt. Der Product Owner akzeptiert oder lehnt die Anforderungen ab, sobald sie abgeschlossen ist, und nicht erst am Ende des Sprints; er genehmigt die Anforderungen, bevor sie den Stakeholdern vorgeführt werden.

- ✓ **Mitglieder des Entwicklungsteams:** Zeigen und erläutern die fertiggestellten Anforderungen.
- ✓ **Stakeholder:** Stellen Fragen und geben Feedback.

Der Prozess beginnt damit, dass sich das Entwicklungsteam auf das Review vorbereitet. Berücksichtigen Sie bei der Vorbereitung auf das Sprint-Review die folgenden Richtlinien:

- ✓ Das Entwicklungsteam bereitet mit minimal möglichem Zeitaufwand (nicht mehr als 20 Minuten) das Sprint-Review vor, und zwar mit dem Ziel, fertiggestellte Anforderungen zu präsentieren.
- ✓ In einem Sprint-Review sollten keine formalen Folien verwendet werden. Vielmehr sollte das Entwicklungsteam seine Zeit damit verbringen, das Produkt zu entwickeln, statt mit der Vorbereitung einer Präsentation.
- ✓ Es werden nur die Anforderungen demonstriert, die gemäß der Definition of Done als fertig gelten und vom Product Owner genehmigt wurden.
- ✓ Das Entwicklungsteam zeigt die nutzbare Funktionalität der Anforderung – also wie sie in der realen Welt funktioniert.



Wenn Sie Ihre Zeit damit verbringen, Stakeholdern zu zeigen, was hätte getan werden können oder sollen, dann ist dies eine manipulierte Demonstration, mit der Sie niemandem einen Gefallen tun. Stakeholder erwarten nie weniger, sie erwarten immer mehr. Wenn Sie es so aussehen lassen, als ob Ihr Produktinkrement funktioniert, obwohl es nicht wirklich funktioniert, erhöhen Sie Ihre Arbeitsbelastung für den kommenden Sprint. Sie müssen dann im kommenden Sprint nicht nur die ganzen neuen und geplanten Aufgaben erledigen, sondern auch etwas funktionsfähig machen, das Sie bereits als funktionierend demonstriert haben. Präsentieren und demonstrieren Sie daher nur Produktinkremente, die wirklich funktionieren.

Stakeholder-Feedback

Entscheidend für den Erfolg des Sprints ist das Feedback der Stakeholder. Ein ständiger Kommunikationskreislauf hält das Projekt auf Kurs und produziert, was die Stakeholder wollen. Zwar können die Stakeholder den Mitgliedern des Entwicklungsteams nicht sagen, wie sie Anforderungen entwickeln sollen, doch sie können dem Product Owner Feedback über die Anforderungen

und Funktionen geben, die sie entwickeln wollen, und darüber, wie gut die Implementierungen den Bedürfnissen der Kunden entsprechen.

Diese Rückkopplungsschleife dient auch einem anderen Zweck: Sie sorgt dafür, dass das Entwicklungsteam am Projekt involviert und damit emotional beteiligt bleibt.

Feedback spielt in Scrum an unterschiedlichen Stellen eine wichtige Rolle. [Abbildung 6.4](#) zeigt, wie viele Feedback-Ebenen im Scrum-Framework enthalten sind. Jedes Mal, wenn Feedback eintrifft, wird es in das Produkt-Backlog eingespeist und in den Sprint-Planungssitzungen berücksichtigt – echte Überprüfung und Anpassung.

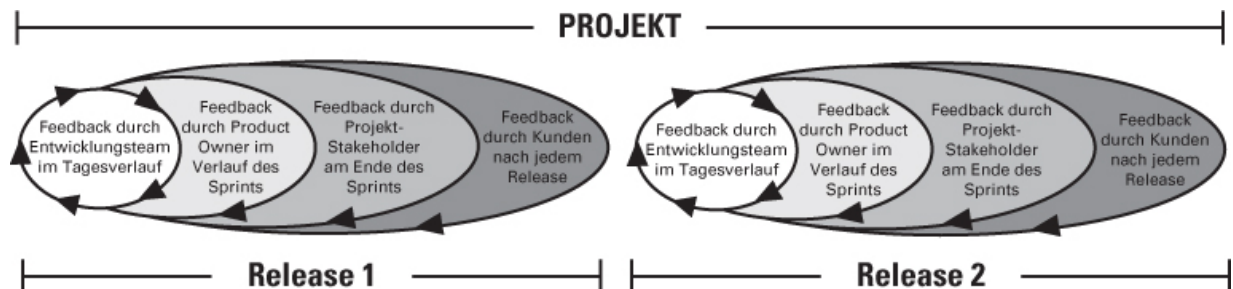


Abbildung 6.4: In einem typischen Scrum-Projekt gibt es mehrere Feedback-Ebenen.

Produktinkremente

Das Produktinkrement ist das letzte der drei Scrum-Artefakte. (Wir besprechen das Produkt-Backlog in [Kapitel 3](#) und das Sprint-Backlog in [Kapitel 5](#).)

Innerhalb eines einzigen Sprints ist das Produktinkrement ein funktionierendes Produkt, das vom Product Owner als fertiggestellt bewertet wird und nun potenziell auslieferfähig ist. Es ist *potenziell* auslieferfähig, da der Product Owner möglicherweise erst später entscheidet, dass das Produkt wirklich ausgeliefert wird. Jedoch ist es auslieferbar, sobald der Product Owner es ausliefern will.

Ein Produktinkrement wurde

- ✓ entwickelt,
- ✓ getestet,
- ✓ integriert,
- ✓ dokumentiert.

Während des Sprint-Review-Meetings wird den Stakeholdern dieses Produktinkrement demonstriert. Der Sprint-Lebenszyklus aus Überprüfung und Anpassung wird fortgeführt, wenn Feedback entgegengenommen und in Anforderungen übersetzt wurde. Diese Anforderungen können dann bei der Verfeinerung des Produkt-Backlogs verbessert werden; sie können bei zukünftigen Sprints vorrangig berücksichtigt werden und schließlich zu neuen Produktinkrementen werden.

Beim Sprint Review geht es darum, das Produkt zu verbessern. Die Sprint-Retrospektive zeigt, wie Scrum-Teams diese kontinuierliche Verbesserung für ihr Team und ihren Prozess umsetzen können.

Sprint-Retrospektive: Phase 7

Die siebte und letzte Phase in der Roadmap to Value ist die Sprint-Retrospektive (siehe [Abbildung 6.5](#)). Dieses Scrum-Event findet nach jedem Sprint statt.

Phase 7: Sprint-Retrospektive

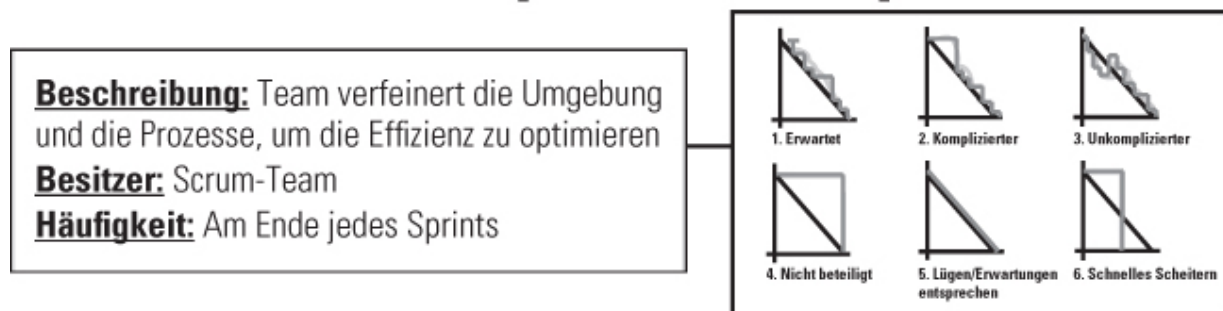


Abbildung 6.5: Die Sprint-Retrospektive, die siebte und letzte Phase der Roadmap to Value

Ziel der Sprint-Retrospektive ist es, dem Scrum-Team – Scrum Master, Product Owner und Entwicklungsteammitgliedern – die Möglichkeit zu geben zu beurteilen, was in dem gerade beendeten Sprint gut gelaufen ist und was verbessert werden kann. Bei diesem Prozess handelt es sich ein weiteres Mal um Überprüfung und Anpassung, jedoch dieses Mal mit Fokus auf den Menschen, Prozessen und Werkzeugen, die das Scrum-Team verwendet.

Das Ergebnis der Retrospektive sollten Aktionspläne zur kontinuierlichen Verbesserung von Scrum, Menschen, Prozessen und Werkzeugen in jedem Sprint sein. Obwohl das Scrum-Framework einfach ist – drei Rollen, drei Artefakte und fünf Ereignisse (siehe [Kapitel 1](#)) – und nicht optimiert werden muss, hat jedes Scrum-Team aufgrund seiner Produkt-, Organisations- und Entwicklungsmethoden Eigenheiten und Nuancen. Durch den Prozess der Überprüfung und Anpassung können Sie diese Individualitäten auf die Projektziele ausrichten.

Da die Sprint-Retrospektive um Input und Feedback von allen Scrum-Teammitgliedern bittet, erhöht sie die Eigenverantwortung durch Engagement und Zielstrebigkeit. Der Teamgeist wird gefördert, was wiederum zu einer Steigerung der Produktivität und Velocity führt, was Selbstmanagement bedeutet.



In Sprint-Retrospektiven ist es wichtig, ein vertrauensvolles Umfeld zu schaffen. Die Meinung jedes Einzelnen wird gehört und respektiert und nichts wird persönlich genommen. Vertrauen ist der Schlüssel, damit die Retrospektive nicht zu einem Labyrinth von Euphemismen oder Politik wird. Der Scrum Master spielt bei der Schaffung einer vertrauensvollen Umgebung eine zentrale Rolle.



Die Sprint-Retrospektive kann Probleme innerhalb des Teams aufdecken. Ein erfahrener Scrum Master kann das Meeting so unterstützen, dass diese Themen in einem angemessenen, emotional wenig aufgeladenen Umfeld behandelt werden. Bei der Sprint-Retrospektive geht es nicht ums Dampfablassen, sondern um die Erstellung umsetzbarer Verbesserungspläne. Halten Sie Ausschau nach passiv-aggressiver Ansprache und persönlichen Agenden.

Der Sprint-Retrospektive-Prozess

Die Sprint-Retrospektive findet am Ende jedes Sprints, nach dem Sprint-Review und vor der nächsten Sprintplanung statt. Für jede Woche, die der Sprint dauert, werden für dieses Ereignis 45 Minuten eingeplant. Bei einem zweiwöchigen Sprint dauert die Retrospektive also 90 Minuten.

Das gesamte Scrum-Team nimmt teil und nach dem Ermessen des Teams können weitere Personen (wie Kunden und Stakeholder) eingeladen werden, wenn das Team glaubt, dass diese Personen wertvolle Erkenntnisse über notwendige Verbesserungen einbringen können.

Als Vorbereitung auf die Retrospektive sollte jeder überlegen, wie der Sprint gelaufen ist und alle Ideen und Bedenken notieren. Verwenden Sie wie immer einfache Werkzeuge wie beispielsweise Haftnotizen; vermeiden Sie formale Präsentationen.

Obwohl die Rolle des Scrum Masters darin besteht, das Meeting zu fördern und zu unterstützen, nehmen alle auf einer Peer-to-Peer-Ebene teil. Der Zweck der Sprint-Retrospektive ist es, den Sprint zu überprüfen, der gerade zu Ende gegangen ist.

- ✓ Identifizieren Sie, was im Sprint mit den Prozessen, Tools und der Teamdynamik gut gelaufen ist.
- ✓ Diskutieren und entdecken Sie Verbesserungsmöglichkeiten.

- ✓ Definieren Sie einen Aktionsplan für die Umsetzung der Verbesserung(en).

Denken Sie während der Retrospektive auch daran, das zu betonen, was gut gelaufen ist, und räumen Sie diesen Themen die gleiche Zeit ein. Es ist wichtig, sich auf das Positive zu konzentrieren und zu erkennen, was gut funktioniert, damit Sie es weiterhin tun können. Freuen Sie sich als Team über Erfolge. Gerade bei der ersten Scrum-Implementierung ist es wichtig, die Erfolge wahrzunehmen, und zwar sowohl die großen als auch die kleinen.



Ein effektiver Weg, die Dinge positiv anzugehen und zu vermeiden, dass Menschen während einer Retrospektive isoliert werden, ist die Sandwich-Technik. Beginnen Sie mit den positiven Dingen, arbeiten Sie sich durch die negativen und schließen Sie wieder mit positiven.

Die Diskussion in der Retrospektive ist handlungsorientiert und konzentriert sich nicht auf Rechtfertigungen. Wenn man das Wort *weil* hört, ist das ein deutliches Signal dafür, dass sich die Diskussion in eine Rechtfertigung verwandelt hat, warum jemand etwas auf eine bestimmte Weise getan hat. Schauen Sie nach vorne, indem Sie sagen: »Das ist es, was ich erlebt habe, und das ist es, was in Zukunft besser funktionieren könnte.« Sagen Sie nicht: »Ich habe es so gemacht, weil... «

Der Derby-und-Larsen-Prozess

Esther Derby und Diana Larsen haben ein ausgezeichnetes Buch mit dem Titel *Agile Retrospektiven: Übungen und Praktiken, die die Motivation und Produktivität von Teams deutlich steigern* geschrieben (erschienen im Vahlen-Verlag). Dort finden Sie weitere Tipps und Techniken zu Sprint-Retrospektiven und anderen agilen Praktiken.

In ihrem Buch weisen Derby und Larsen darauf hin, dass es mehr darauf ankommt herauszufinden, was gut gelaufen ist und welche Verbesserungen vorgenommen werden müssen, als am Ende

eines jeden Sprints immer die gleichen drei Fragen zu stellen. Die Unterstützung der Retrospektive erfordert Vorbereitung und Strategie, um maximale Beteiligung, Offenheit und nützliche Daten von den Teammitgliedern zu erhalten. Das Derby-und-Larsen-Modell zur Strukturierung einer Retrospektive besteht darin, diese Fragen zu beantworten:

- ✓ Was, glauben Sie, ist gut gelaufen?
- ✓ Was möchten Sie ändern?
- ✓ Wie sollen wir diese Veränderung umsetzen?

Um die Effektivität der Retrospektive zu maximieren, empfehlen wir das Derby-und-Larsen-Verfahren:

1. **Ankommen**

Sie wollen von Anfang an Grundregeln für eine produktive Kommunikation aufstellen und Erwartungen und Ziele klären. Bereiten Sie das Team auf eine offene und ehrliche Diskussion vor.

2. **Daten sammeln**

Entscheidungen, die auf oberflächlichen, schlechten oder unvollständigen Daten basieren, können mehr schaden als nützen. Sie wollen wichtige Themen aufdecken, Erinnerungen wecken und Erfahrungen austauschen, die es zu besprechen gilt. Sie wollen nicht nur wissen, was die Leute denken, sondern auch, wie sie sich damit fühlen.

3. **Einsichten erzeugen**

Viele Teams sammeln Daten, tun aber nichts damit. So wie die besten Entwürfe von selbstorganisierenden Teams stammen, kommen die besten Einsichten von Teams, die sich Zeit nehmen, um zu erforschen, was die Daten bedeuten.

4. **Entscheiden, was zu tun ist**

Nur durch Handeln kann Veränderung und Anpassung stattfinden. Handeln erfordert einen Plan. Die Entscheidung,

was zu tun ist, verlagert den Fokus des Teams auf den nächsten Sprint.

5. Retrospektive abschließen

Der Abschluss bietet die Möglichkeit, die Retrospektive durch Aktivitäten zu vertiefen, die die Effektivität der Retrospektive bewerten und Wege zu ihrer Verbesserung aufzeigen. Sie ermutigt auch dazu, Wertschätzung auszudrücken.



Jeder Aspekt der Retrospektive kann durch eine Reihe von Aktivitäten erleichtert werden, die das individuelle Denken und die Gruppendiskussion anregen. Probieren Sie eine Internet-Suche nach *Triple Nickels*, *Five Whys*, *SMART Goals*, *Temperature Reading*, *Team Radar* und *Mad Sad Glad*. Dies sind alles gute Aktivitäten für eine Retrospektive.



Um die Diskussion in Retrospektiven anzuregen, organisieren Sie Aktivitäten rund um spezifische Fragen wie die folgenden:

- ✓ Was hält uns davon ab, unsere Velocity von 36 auf 38 zu erhöhen?
- ✓ Hat jeder das nötige Werkzeug, um den Job zu erledigen?
- ✓ Wiederholen sich irgendwelche Hindernisse?
- ✓ Ist unser Daily Scrum effektiv im Erkennen von Hindernissen und in der Koordination der täglichen Prioritäten?
- ✓ Fehlen unserem Team bestimmte Fähigkeiten, und wenn ja, wie können wir sie erlangen?

Manche Scrum-Teams müssen überredet und geschubst werden, um sich zu engagieren und zu beteiligen. Sie mögen zunächst zögern zu sagen, was sie wirklich fühlen. Andere wollen vielleicht sofort reden und sind voller Ideen und Anregungen. Ein

einfühlsamer, proaktiver Scrum Master passt sich an die Arbeit mit diesen Arten von Gruppen oder Gruppen, die irgendwo dazwischen liegen, an, um die besten Ergebnisse zu erzielen.



Setzen Sie für einen Sprint nur eine Aktion um. Am Anfang mag es verlockend sein, alle Themen umzusetzen, die das Team diskutiert. Finden Sie stattdessen eine Aktion, die sowohl wirkungsvoll als auch einfach zu implementieren ist. Ernten Sie zunächst die niedrig hängenden Früchte.



Die Ergebnisse aus der Retrospektive sollen als Verbesserungsposten in das Produkt-Backlog eingestellt werden. Das Scrum-Team sollte zustimmen, dass mindestens eine Verbesserungsaktion in jeden Sprint geht. Bringen Sie mindestens ein prioritäres Retrospektivobjekt in den nächsten Sprint, vielleicht aus der letzten Retrospektive. Warum also warten? Es geht doch um Überprüfung und Anpassung. Verzögern Sie also nicht den Anpassungsteil nicht!

Überprüfung und Anpassung

Bei Scrum geht es darum, die richtigen Dinge zur richtigen Zeit zu planen. Es geht darum, auf veränderte Märkte und Erfahrungen zu reagieren. Es geht darum, ständig zu lernen und zu bewerten, Risiken zu minimieren und den Wert bei jedem Schritt zu maximieren – und zwar an jedem Ort, an dem gearbeitet wird.

Die im offiziellen Scrum Guide (<http://scrumguides.org>) enthaltene Betrachtung zu Überprüfung und Anpassung ist eine gute Möglichkeit, dieses Kapitel abzuschließen. Wir haben die Kursivschrift zur Hervorhebung hinzugefügt:

Den größten Nutzen bringen Überprüfungen, wenn sie gewissenhaft durch kompetente Prüfer dort vorgenommen werden, *wo die Arbeit verrichtet wird.*

Der Scrum Guide sagt weiter, dass Anpassungen so schnell wie möglich vorgenommen werden. Anpassungen erfolgen, sobald ein Prüfer feststellt, dass die Arbeit die Grenzwerte überschritten hat und ein unzulässiges Produkt verursacht.

Kapitel 7

Überprüfen und anpassen: So korrigieren Sie Ihren Kurs

IN DIESEM KAPITEL

- Gewissheit benötigen
- Die Feedbackschleife verstehen
- Transparenz aufrechterhalten
- Mit Antimustern und äußeren Kräften umgehen
- Den Kurs korrigieren
- Tests in der Feedbackschleife nutzen
- Eine Innovationskultur aufbauen

Oftmals finden Sie Ihr Produkt während der Produktentwicklung an einem unerwarteten Ort wieder. In diesem Kapitel untersuchen wir, wie Scrum kontinuierliches Lernen und Verbesserung ermöglicht. Wir befassen uns auch damit, wie Sie trotz Unsicherheit arbeiten können, anstatt darauf zu hoffen, dass Sie diese wegplanen können. Denn agil zu sein, bedeutet, dass man lernen und sich anpassen kann – flexibel genug, um das beste Produkt mit den richtigen Eigenschaften und der besten Qualität zu bauen.

Bedürfnis nach Sicherheit

Das Bedürfnis des Managements nach Ergebnissicherheit kann die Abwärtsspirale oder den Verfall vieler Projekte und großer Ideen vorantreiben. Sie können sich weigern, die grundlegende Realität der Unsicherheit zu akzeptieren und sich mit der relativen

Sicherheit des Bekannten zu Lasten des Besseren zufriedengeben. Die Kraft des empirischen Ansatzes kann solange mit Unsicherheiten klarkommen, bis Sie die benötigten Informationen haben, um mehr Sicherheit über das Ergebnis Ihres Projekts oder Ziels zu erlangen. So kann es beispielsweise in sechs Monaten von heute an gerechnet schwierig sein, sicher zu wissen, ob eine benötigte Komponente, die von einem anderen Team gebaut wird, rechtzeitig fertig wird, wenn das andere Team nicht spätestens in drei Monaten mit der Entwicklung beginnt. Zwei Wochen vor der voraussichtlichen Lieferung haben Sie jedoch eine viel klarere Vorstellung über die Einsatzbereitschaft der Komponente.



Empirie bedeutet Lernen oder Verifizieren durch Beobachtung. Scrum ist also ein empirischer Ansatz, da Sie bei jedem Schritt Ihre Ergebnisse überprüfen und sich sofort anpassen, um bessere Resultate zu erzielen.

[Abbildung 7.1](#) zeigt ein Prognosetool, das unter dem Namen Cone of Uncertainty (auf Deutsch etwa Kegel der Unsicherheit) bekannt ist. Das Grundprinzip beruht darauf, dass Ergebnisse über einen längeren Zeitraum schwer vorherzusagen sind; je näher Sie jedoch Ihrem Ziel kommen, desto mehr nimmt die Sicherheit zu.

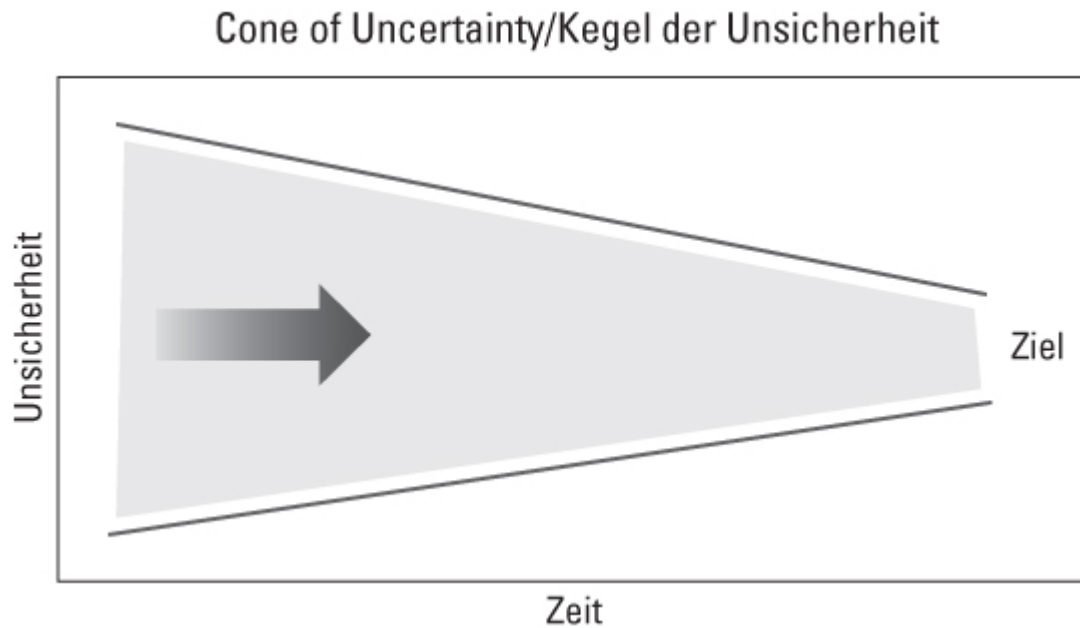


Abbildung 7.1: Kegel der Unsicherheit

Mit dem Cone of Uncertainty versuchen Meteorologen den Verlauf von Wirbelstürmen über mehrere Tage vorherzusagen. Der empirische Ansatz berücksichtigt die Wirklichkeit einer gewissen Unsicherheit und bietet einen Rahmen für das Management der damit verbundenen Risiken durch kontinuierliche Verbesserung und Anpassung.

Die Feedbackschleife

In der Volkswirtschaftslehre ist eine *Rückkopplungsschleife* als das Resultat oder die Ergebnisse eines Prozesses oder eines Zyklus zur Information des nächsten Prozesses oder Zyklus definiert. Diese Feedbackschleife speist die Daten in den empirischen Prüfprozess ein. [Abbildung 7.2](#) untersucht den Feedback-Prozess von Scrum-Teams.

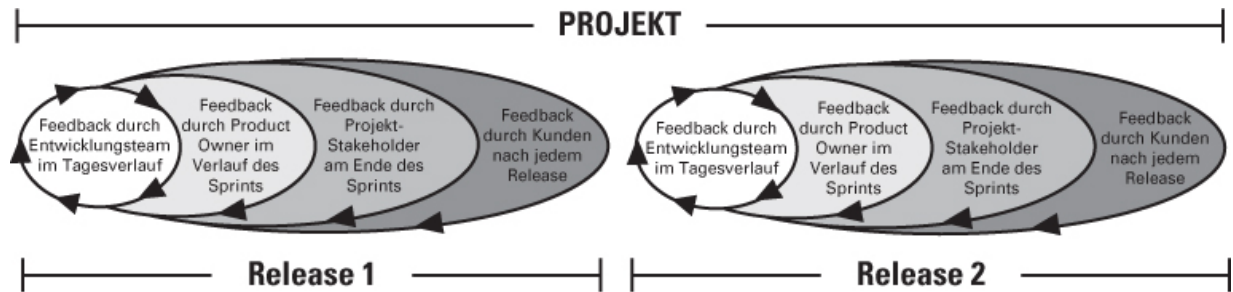


Abbildung 7.2: Feedback-Prozess

Die Idee hinter einer Feedbackschleife ist, dass Sie mithilfe der aus Erfahrung gewonnenen Informationen kontinuierlich Verbesserungen vornehmen können. Wenn Sie eine Möglichkeit haben, die gewonnenen Erkenntnisse auf zukünftige Situationen anzuwenden. Ein fester Bestandteil des Scrum-Prozesses ist zu erkennen, dass die bei jedem Sprint gewonnenen Kenntnisse und Erfahrungen in späteren Sprints wieder eingebracht werden müssen, damit sich der zu entwickelnde Plan anpassen kann. Zu Beginn eines Projektes werden Produkt-Roadmap und Sprint-Pläne immer auf Basis der besten zu diesem Zeitpunkt verfügbaren Informationen erstellt. Die Pläne werden vom gesamten Team akzeptiert und über die Stakeholder unter die Leute gebracht. Die strikte Einhaltung eines umfangreichen Anfangsplans kann die im Prozess gemachten Erfahrungen ignorieren und Innovationen im Team ersticken. Stattdessen nutzt Scrum Kunden- und Teamfeedback zur Verbesserung des Plans.

Sie fragen sich vielleicht, warum Sie überhaupt planen sollten, wenn Sie doch wissen, dass der ursprünglicher Plan wahrscheinlich lücken- oder fehlerhaft ist. Sie erstellen einen Plan als ersten Schritt in Ihrem Lernprozess. So wie eine Flugbesatzung einen Flugplan einreicht, der auf den erwarteten Bedingungen basiert, um die Besatzungsmitglieder und die Fluglotsen wissen zu lassen, was sie erwartet, legen die Produkt-Roadmaps die Preflight-Erwartungen des Scrum-Teams fest. Nach dem Start des Flugs bewerten die Mitglieder die tatsächlichen Bedingungen wie Wetter, Wind und Zeitplanung, um den Flugplan nach Bedarf anzupassen. Gleichmaßen geben Sie

in Scrum jeden Tag Feedback, und wenn Sie mehr Informationen haben, passen Sie Ihre Ideen und Pläne an.

Timing der Entscheidungen

Wir sind nicht unbedingt dafür, alle Entscheidungen erst zum letztmöglichen Zeitpunkt zu treffen. Ein Lean-Entwicklungskonzept besagt, dass Entwickler bis zum letztmöglichen Zeitpunkt mit Entscheidungen warten sollten, weil sie immer dazu lernen. Andere argumentieren, dass sich durch das Warten nur eine geringe Wertsteigerung ergibt. Ein einfaches heuristisches Modell ist, Entscheidungen in drei Töpfe zu packen:

- ✓ Dinge, die *bekannt* sind
- ✓ Dinge, die man *wissen könnte*
- ✓ Dinge, die man *nicht wissen kann*
- ✓ Entscheidungen über das, was bekannt ist, hinauszuzögern, bringt vermutlich keinen Mehrwert. Auch wenn eine Entscheidung nicht aus dem Topf »kann man nicht wissen« in »bekannt« verschoben werden kann, bringt das Verschieben der Entscheidung keinen Mehrwert. Verzögerung ist angemessen bei Dingen, die man wissen könnte, das heißt, die man in Erfahrung bringen kann. Das Warten auf mehr Informationen, führt wahrscheinlich zu einer besseren Entscheidung.



Feedbackschleifen können positiv (weil sie sich aus Erfolgen ergeben) oder negativ (aus Fehlern) sein. Erfolg kann Erfolg hervorbringen, aber auch Misserfolg kann für den Erfolg entscheidend sein, wenn man daraus lernt. Deshalb fragen wir, was gut funktioniert hat und was bei jedem Sprint verbessert werden kann. Es ist wichtig, weiterhin Scrum-Erfolge zu unterstützen und nach Veränderungen zu suchen, die sich positiv auswirken können.

Transparenz

Transparenz ist in der modernen Gesellschaft allgegenwärtig. Von Open-Source-Software bis hin zur offenen Zusammenarbeit bei Ideen wird Transparenz großzügig interpretiert. In [Kapitel 1](#) erläutern wir, warum Transparenz ein Grundpfeiler für Verbesserungen ist. Es ist wichtig, den hohen Wert von Transparenz für den Prozess des Überprüfens nicht zu übersehen. Ohne eine Kultur der Transparenz werden Entscheidungen mit ungenauen Daten getroffen. Viel zu oft beobachten wir, dass Organisationen gute Fake News mehr würdigen als die eigentliche Wahrheit und sie daher auf der Grundlage schlechter Informationen Entscheidungen treffen.

Nachfolgend einige Grundprinzipien, mit denen Sie in Ihren Projekten für Transparenz sorgen können:

- ✓ Nur die Fakten. Zeigen Sie insbesondere den Stakeholdern nur das, was auch stimmt. Zeigen Sie also nur fertiggestellte Arbeiten, die Ihre Definition of Done erfüllen.
- ✓ Seien Sie offen und klar über den Plan, das Design, den Prozess und den Fortschritt.
- ✓ Machen Sie alle Informationen für jedermann leicht zugänglich.
- ✓ Veröffentlichen Sie die Roadmap, den Release-Plan und die Definition von Done an einer Wand, wo alle Beteiligten sie sehen können.

- ✓ Fördern Sie ein Umfeld von faktischen Kennzahlen und Ergebnissen.
- ✓ Führen Sie so früh wie möglich eventuelle unangenehme Gespräche.



Es ist üblich, ehrliche und unangenehme Gespräche zu vermeiden, aber das Vermeiden unangenehmer Situationen kann zu Problemen führen. (Tatsächlich wurden ganze Bücher und Kurse geschrieben, um den Menschen beizubringen, wie man mit diesen Situationen umgeht.) Wir möchten, dass Sie verstehen, wie wichtig es ist, die zusätzlichen Kosten zu kennen, die mit der Vermeidung oder Verzögerung unangenehmer Momente verbunden sind. Die direktesten dieser Kosten ist Zeitverschwendung, weil man den falschen Weg einschlägt und hierdurch eine bessere Vorgehensweise verzögert. Scheuen Sie sich nicht, diese unbequemen Gespräche zu führen; führen Sie sie, sobald Sie merken, dass etwas angesprochen werden muss.

Sich vor Verschleierung hüten

Einige der Organisationen, die wir weitergebildet und gecoacht haben, hatten eine Arbeitskultur, die Teamtransparenz verhinderte. Achten Sie auf Antimuster, die Teams zwingen, die Realität als Mittel zum Selbstschutz zu verbergen. Achten Sie außerdem darauf, dass einige externe oder beauftragte Organisationen aus der Vermeidung sachlicher Berichterstattung eine Kunst gemacht haben. Als Führungskraft müssen Sie das spezifische Wissen Ihrer Scrum-Teams schätzen und ihm vertrauen. Ermöglichen Sie Teamerfolg, anstatt den Fortschritt zu hinterfragen.

Als Geschäftsführer einer Boutique-Softwareentwicklungsfirma musste David diese Lektion auf die harte Tour lernen. Während er an die Konzepte autonomer Teams glaubte, fiel es ihm schwer, dem Fortschritt seines Teams zu vertrauen, als die

Stakeholder nach Erfolgsgarantien fragten. Die einzige Antwort, die David von seinen Teams bezüglich der Produktqualität oder dem Projekt-Timing hören wollte, lautete: »Ja, wir werden pünktlich sein«. Aus Angst vor seinem Zorn konnte das Scrum-Team ihn nicht um Hilfe oder Klärung bitten. Die Antworten waren in Wahrheit über die Scrum-Artefakte verfügbar, aber David war zu beschäftigt, um nachzusehen. Nachdem er erkannt hatte, dass er kein Umfeld schuf, das den Erfolg des Teams ermöglichte, änderte er sich. Er lernte, den Scrum-Boards zu folgen, anstatt nach einem Statusbericht zu fragen. Dies ermöglichte es ihm, sich seinem Team anzuschließen und gemeinsam erfolgreich zu sein, anstatt sich aus Angst vor einem möglichen Misserfolg zurückzuziehen. Er konnte dann so früh wie möglich die unangenehmen Gespräche mit Stakeholdern führen. Seine Veränderung in Richtung mehr Ownership war ein großer Erfolg für seine Teams. Im Laufe der Zeit begannen die Teams, darauf zu vertrauen, dass er weiterhin ihren Erfolg unterstützte. Die Qualität und Pünktlichkeit des Teams machte damit einen großen Sprung nach vorne. David zeigte zunächst einige bekannte Antipatterns, die den Prozess des Überprüfens und Anpassens behinderten.

Antimuster

Der Begriff *Antimuster* (oder Antipattern) beschreibt eine gut gemeinte Lösung für ein Problem, die aber unbeabsichtigte negative Folgen hat. Es ist wichtig für eine Organisation, die Scrum einführt oder weiter optimiert, Scrum zu erlauben, Antipatterns als Teil des Prozesses von Überprüfung und Anpassung aufzudecken. Ebenfalls ist es wichtig, Scrum nicht an eine fehlerhafte Kultur oder Praxis anzupassen. Nehmen Sie stattdessen Änderungen in der Kultur vor, um den Erfolg Ihrer Scrum-Praxis zu erleichtern.

Möglicherweise kennen Sie einige der folgenden Beispiele für Antipattern, die in der Verwaltung und Softwareentwicklung vorkommen:

- ✓ **Paralyse durch Analyse:** Aufgrund ständiger Analyse und mangelnder Bereitschaft, Unsicherheiten zu akzeptieren, kommt man nicht voran.
- ✓ **Schall und Rauch:** Die Illusion von erreichten Zielen, wie zum Beispiel die Überbewertung von Qualität oder Vollständigkeit.
- ✓ **Möwen-Management:** Als Führungskraft hereinschneien, einen Haufen Lärm machen und dann wegfliegen, in der Hoffnung, alle mit einem Gefühl der Dringlichkeit motiviert zu haben, aber stattdessen wird Angst und Panik hervorgerufen.

Scrum-Teams und agile Führungskräfte müssen nach den Ursachen und kulturellen Antipatterns suchen, die die Feedbackschleife blockieren. Sobald die Antipatterns identifiziert sind, sollten sie als Hindernis des Scrum-Prozess an die Führungsebene eskaliert werden. Die Beseitigung dieser kulturellen Hindernisse muss eine Priorität sein, da sie mehrere Scrum-Teams innerhalb der Organisation behindern können.

Externe Kräfte

Ein Team kann von Dingen außerhalb des Teams beeinflusst werden, die dessen Fähigkeit das Produkt zu liefern, einschränken. Diese Kräfte wirken wie Gegen- oder Seitenwind, der ein Flugzeug vom Kurs abbringt oder zu einer Verspätung führt.

Für Scrum-Teams können diese Kräfte sich ändernde Vorschriften, sich entwickelnde Architekturen oder neu priorisierte Funktionen sein. Es mag einem Team zwar nicht möglich sein, diese Dinge zu ändern, aber verdeutlicht die Auswirkungen dieser Kräfte und zeigt oft alternative Handlungsmöglichkeiten auf.

Eine häufige Entschuldigung dafür, dass Scrum in einer bestimmten Umgebung nicht funktionieren kann, ist, dass externe Kräfte es unmöglich machen, dass Scrum erfolgreich sein kann. Aber Scrum kann überall erfolgreich sein, solange es genügend Einsatz und Anstrengungen der Beteiligten gibt, um es zum Funktionieren zu bringen. In späteren Kapiteln geben wir Ihnen einige Beispiele für Organisationen, die Scrum erfolgreich einsetzen und die wahrscheinlich mehr Einschränkungen aufweisen als Ihre Organisation.

Kurskorrektur während des Fluges

In diesem Abschnitt kommen wir auf die Analogie der Flugbesatzung und des Flugplans zurück, um die Dynamik bei der Planung und Neuplanung anhand empirischer Daten zu beleuchten. Eine Flugbesatzung plant immer die Details einer bevorstehenden Mission auf der Grundlage der dieser Mission zugewiesenen Parameter. Die Crew schätzt den Treibstoffverbrauch, die Flugzeit und die Ausführungsdetails anhand aller bekannten und erwarteten Details. Am Tag der Mission tragen die Besatzungsmitglieder alle aktuellen Wetterberichte und Missionsänderungen zusammen.

Wenn das Flugzeug in der Luft ist, werten die Mitglieder der Flugbesatzung kontinuierlich Geschwindigkeiten und Kursangaben aus und vergleichen die aktuelle Position des Flugzeugs mit den Flugplänen. [Abbildung 7.3](#) zeigt eine geplante Route und eine während des Flugs gemessene Position.

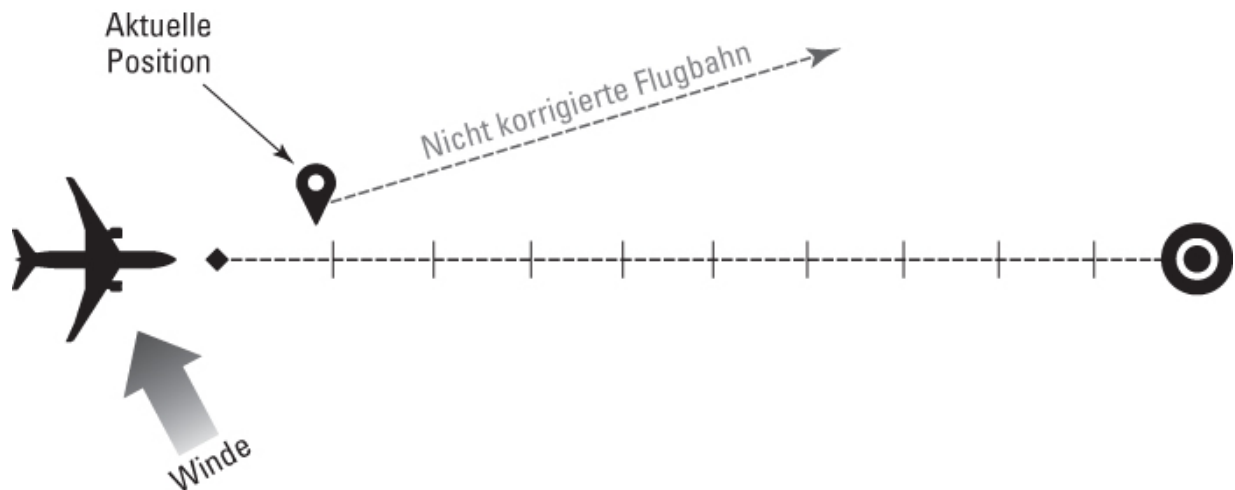


Abbildung 7.3: Flugbahn

Jedes Mal, wenn Besatzungsmitglieder die aktuelle Position des Flugzeugs ermitteln, untersuchen sie, warum sie sich vom Plan entfernen. Sie betrachten die Daten, um mögliche Ursachen für die Kursanpassung zu ermitteln, wie zum Beispiel unerwartete Windrichtungen oder -geschwindigkeiten oder eine Differenz zwischen dem geplanten und dem tatsächlichen Kurs und der Geschwindigkeit. Als Nächstes nutzen sie die neuen Informationen, um den Plan anzupassen und das Ziel der Mission zu erreichen – und das alles bei einer Geschwindigkeit von mehr als 800 km/h. Die Flugbesatzung wiederholt diesen Überprüfungs- und Anpassungsprozess immer wieder, um den Erfolg seiner Mission sicherzustellen.

Bei Scrum dienen Sprints als perfekte Zeiten, um Pläne neu zu bewerten und zu verbessern. In einer transparenten Organisation werden neue Informationen begrüßt und für den Produkterfolg genutzt.



Es ist wichtig, den kostspieligen Overhead von redundanten oder bürokratischen Parallelprozessen zu vermeiden. Wir haben mit wohlmeinenden Organisationen zusammengearbeitet, die mehr Aufwand für die Ermittlung des Status und die Erstellung von Präsentationsfolien aufwenden als für das Erstellen des Produkts; alles in der Hoffnung, die Kontrolle zu behalten oder Unsicherheiten zu vermeiden. Das Erstellen von Statusberichten ist eine unnötige parallele Aktivität. Eine Organisation, die agil geworden ist und Scrum vollständig implementiert hat, lässt diese Artefakte los und geht den neuen Weg.

Testen in der Feedbackschleife

In Scrum testen Sie bei jedem Sprint, indem Sie Code-Tests, Funktionstests und andere Tests durchführen, um Benutzer-Feedback zu erhalten. Vergleichen Sie diese Art von Projekt mit einem typischen Wasserfallprojekt, bei dem das Testen erst am Ende des Projekts stattfindet. Fehler in letzter Minute zu finden, verursacht entweder heldenhafte und teure Fehlerbehebungen oder Verspätungen im Zeitplan. Wie ein Flugzeug, dessen Weg unkorrigiert bleibt, endet ein Produkt, das nicht validiert ist, weit ab vom Kurs.

Das Feedback, das Sie während des Sprints durch Tests und die Kundenvalidierung erhalten werden, gibt den notwendigen Aufschluss über die tägliche Priorität. Korrekturen am Produkt werden in der Regel noch am selben Tag vorgenommen, während sie am einfachsten und kostengünstigsten zu beheben sind und den weiteren Fortschritt nicht behindern.



In einem Scrum-Team ist jeder für alles verantwortlich. Es liegt nicht an einem spezialisierten Tester, alles für alle anderen zu testen. Das gesamte Team besitzt das Ergebnis, und alle Mitglieder legen Wert auf Qualität.

Innovationskultur

Viele Unternehmen, egal ob neu oder alt, sagen, dass sie innovativ sein wollen. Sie sehen wahrscheinlich die Marktvorteile kreativer Ansätze. Doch die Befehls- und Kontrollstrukturen, mit denen sie Produkte und Prozesse verwalten, behindern die Innovationen, nach denen sie suchen. Die grundlegenden Überzeugungen von Organisationen über den Zweck bestimmen, ob ihre Teams in einer Innovationskultur arbeiten.

Das Scrum-Framework funktioniert am besten in einer Kultur der Selbstorganisation, der Bestimmung und der Innovation. Produkt- und Prozessinnovationen werden durch das Modell der Überprüfung und Anpassung gefördert, und der Innovationsgedanke steht im Mittelpunkt der Teamarbeit. Der Einsatz der Feedbackschleife ist ein guter Weg, zur Förderung von Innovationen.

Im Folgenden finden Sie einige Möglichkeiten, um eine innovative Kultur zu erzeugen:

- ✓ Versetzen Sie Teams in die Lage, Konventionen oder Einschränkungen in Frage zu stellen.
- ✓ Beseitigen Sie organisatorische Barrieren für Kreativität.
- ✓ Überdenken Sie, wie Sie motivieren.
- ✓ Suchen Sie kreative Querdenker.



Dem Psychologen und Philosophen Edward de Bono wird die Prägung des Begriffs *Querdenken* zugeschrieben, das er als eine Mentalität definierte, die Konventionen und Zwänge in Frage stellt. Ein Querdenker kann seine bisherigen Überzeugungen über Beschränkungen und Konventionen überwinden. Wer sich mit Querdenken beschäftigt, ignoriert nicht die Existenz von Zwängen, sondern sucht nach Ideen und Lösungen, die nicht sofort ersichtlich sind.

Anscheinend ist eine populäre moderne Sichtweise, dass Ingenieure und Entwickler roboterartige, logikbasierte oder lineare Denker seien. Nach unserer Erfahrung trifft dies auf die effektivsten Wissensarbeiter nicht zu. Sie sind Schöpfer mit ausgefeilten kognitiven Fähigkeiten, die nicht durch die komplexen Algorithmen ersetzt werden können, die sie entwickeln. Und sie gedeihen in einer Kultur der Innovation. Da wir dazu tendieren, Geschäftsfunktionen durch Automatisierung zu ersetzen, müssen wir Raum für Querdenker schaffen, die über wahrgenommene Grenzen hinausgehen können. Ohne Kreativität kommen wir nicht weiter und machen die gleichen Fehler.

Zu Kreativität motivieren

In seinem TED-Vortrag

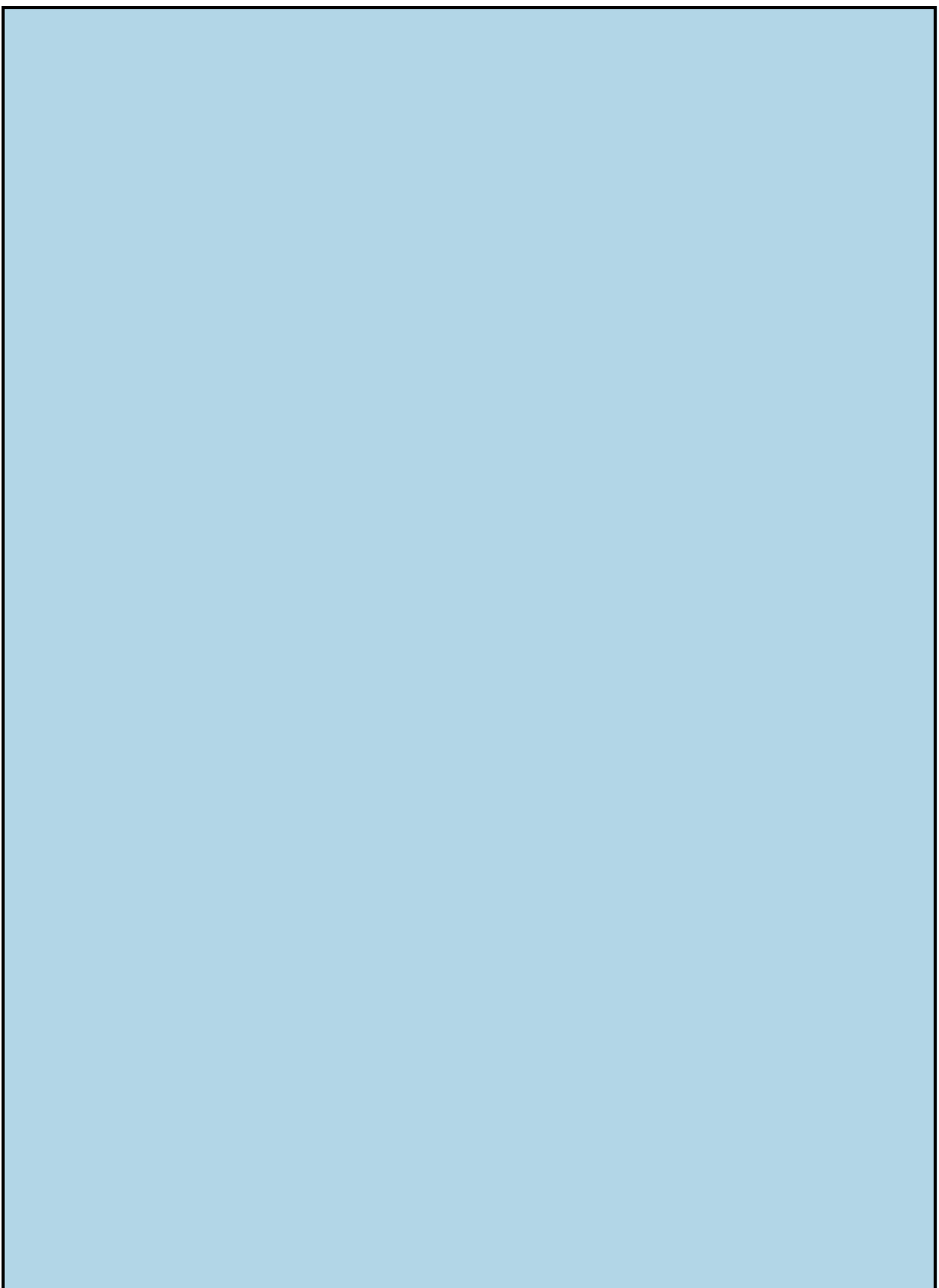
(www.ted.com/talks/dan_pink_on_motivation#t-1094472) stellte der Wirtschaftsautor Dan Pink bisherige Ideen zur Motivation der Kreativität in Frage. Anhand neuerer sozialwissenschaftlicher Forschungen stellte er einen Fall vor, der die Wirksamkeit von Zuckerbrot und Peitsche oder extrinsischen Belohnungsmotivatoren in Frage stellt.

Intrinsische Motivatoren wie Autonomie, Einflussnahme und Ziel sind effektiver darin, Teammitglieder zu ermutigen, sich als Querdenker zu engagieren.

Teil III

Scrum in der Industrie





IN DIESEM TEIL ...

Wenden Sie Scrum außerhalb der Softwareentwicklung an.

Vermeiden Sie übermäßiges Upfront-Design bei der Produktion von Hardware und anderen materiellen Produkten.

Verwenden Sie Scrum, um den Bildungssektor zu verbessern und um Leben zu retten und zu schützen.

Passen Sie sich an die Disruption im Verlagswesen und den digitalen Medien an.

Kapitel 8

Software-Entwicklung

IN DIESEM KAPITEL

Software und Scrum kombinieren
Flexibilitätsvorteile der Softwareentwicklung nutzen
Refactoring zur Optimierung und Vereinfachung
Den Wandel in der Softwarebranche nutzen

Sie machen kein Projekt, sondern nur Aktionsschritte, die damit zusammenhängen. Wenn genügend richtige Schritte unternommen wurden, ist eine Situation entstanden, die Ihrem ursprünglichen Bild des Ergebnisses so nahekommt, dass Sie das Projekt als »erledigt« bezeichnen können.

David Allen

Softwareentwicklung ist der Kontext, in dem die Erfinder von Scrum agile Werte und Prinzipien formalisiert haben und in dem das Scrum-Framework geboren wurde. Daher ist es nicht verwunderlich, dass Scrum leicht auf diese Branche übertragen werden kann.

Mit dem Fortschritt der Technologie wächst auch ihre Komplexität. Es entstehen neue Herausforderungen. Projektmanagement-Frameworks, die Bestand haben, sind diejenigen, die Veränderungen ermöglichen und fördern.

In diesem Kapitel zeigen wir Ihnen einige der Herausforderungen, vor denen die Softwareindustrie steht, und beschreiben, wie Scrum dabei hilft, diese zu bewältigen. Sie sehen die Anwendung von Scrum in aktuellen Beispielen und verstehen, wie das

Framework es Ihnen ermöglicht, ebenso flink und schnell zu sein wie diese exponentiell wachsende Welt der Technologie.

Scrum und Softwareentwicklung: Für einander geschaffen

Softwareentwicklung ist von Natur aus ein kreativer Prozess. Alles ist möglich, denn Ideen, Konzepte und Realität kennen keine Grenzen. Notwendigerweise sind die eingesetzten Designlösungen oft so kreativ wie die Produkte, die sie bedienen.

Aufgrund seiner empirischen Natur passt Scrum perfekt in dieses Umfeld. Scrum sagt Ihnen nicht, wie Sie etwas tun sollen. Es lässt Sie einfach klarsehen und macht deutlich, was Sie tun, und dies können Sie dann beurteilen.

Es gibt eine Vielzahl von Programmiersprachen, Tools, Methoden und Plattformen, um diese komplexen Probleme zu lösen. Scrum sagt Ihnen nicht, welche Sie verwenden sollen. Vielmehr lässt dieses Framework die Teams selbst entscheiden, welche Wege für die eigene Situation am besten geeignet sind. Scrum tut dies durch uneingeschränkte Transparenz, häufige Überprüfung und sofortige Anpassung.

Die Natur von Scrum erlaubt es Ihnen, jeden Tag, in jedem Sprint und bei jedem Release Lösungen zu finden. Sowohl das Produkt als auch der Prozess werden im Hinblick auf Kreativität und Spitzenleistung gefördert.

Traditionelle Projektmanagementmethoden und -Frameworks basieren darauf, die Zukunft genau vorhersagen zu können. Sie sind linear und sequenziell. Sie wirken wie ein Wasserfall. (Lesen Sie mehr über Wasserfallansätze in *Agiles Projektmanagement für Dummies* von Mark C. Layton und Steven J. Ostermiller, ebenfalls bei Wiley-VCH erschienen.) Es ist wichtig zu verstehen, dass die Anforderungen an Technologie und Design weit über

diesen veränderungsfeindlichen Wasserfallansatz hinausgewachsen sind.

Das Wasserfall-Projektmanagement beinhaltet die schrittweise Reifung eines festgelegten Anforderungspakets in verschiedenen Phasen, wobei eine Phase vor der nächsten abgeschlossen wird. So werden zum Beispiel alle Anforderungen an das gesamte Projekt festgehalten, bevor jegliche Entwicklung beginnt. Außerdem werden alle Entwicklungsarbeiten beendet, bevor erst am Ende des Projekts umfassende Tests durchgeführt werden. Bei Scrum werden diese Phasen während eines Projekts wiederholt durchlaufen, sodass ein kontinuierlicher Zyklus von Design, Entwicklung, Test und Integration entsteht und man jeden Tag, in jedem Sprint und bei jedem Release Feedback erhält.

Mit zunehmender Komplexität der Projekte wurden die natürlichen Grenzen des Wasserfallmodells deutlich. Es gab mehrere Phasen und lange Verzögerungen, bevor überhaupt Code geschrieben wurde. Die frühe Leistung war kein Indikator für die spätere Leistung, da die Aufgaben jeder Phase grundlegend unterschiedlich waren; die Code-Tests wurden an das Ende des Projekts verschoben, wenn das Team am wenigsten Zeit oder Geld hatte. Kunden interagierten erst mit dem Produkt, wenn es zu spät war, ihr Feedback zu integrieren. Projekte wurden verzögert oder nie abgeschlossen, und oft war das Ergebnis mangelhaft.

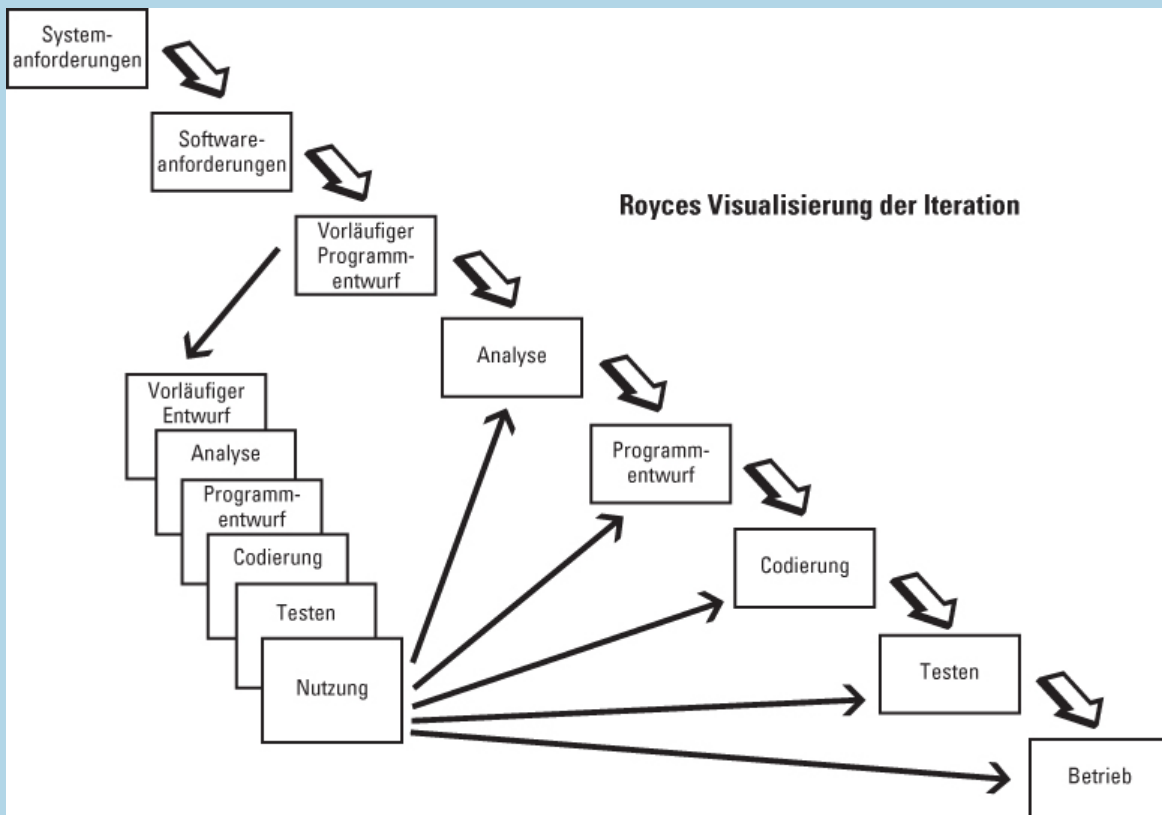
Scrum entstand aus der Notwendigkeit, bessere Wege zu finden, um Software zu entwickeln. Beachten Sie jedoch, dass das Scrum-Framework auf jedes Projekt angewendet werden kann, in dem Sie die Arbeit einkapseln und die Priorität der Elemente festlegen können.

Winston Royce hatte Recht

Interessanterweise betonte der Mann, der fälschlicherweise mit der Einführung des Wasserfall-Frameworks bei großen Softwareprojekten in Verbindung gebracht wurde, die

Grundprinzipien von Agilität und Scrum: Überprüfen und Anpassen.

Winston Royce war Informatiker und Direktor des Lockheed Software Technology Center. Er veröffentlichte ein Papier mit dem Titel »Managing the Development of Large Software Systems«, in dem er verschiedene Methoden beschrieb, darunter Wasserfall und eine frühe Iteration von Agilität, wie in dieser Abbildung gezeigt.



Royce warnte deutlich vor den Fallstricken, die später mit einem Wasserfall überwunden werden sollten. Es ist merkwürdig, dass genau das System, das er fälschlicherweise als Erfindung bezeichnet hat, das gleiche ist, vor dem er gewarnt hat. Er schrieb:

»Die Testphase, die am Ende des Entwicklungszyklus stattfindet, ist das erste Ereignis, bei dem unter anderem Timing, Speicherung, Input/Output-Transfers im Gegensatz zur

Analyse erlebt werden und erfahrbar sind ... Doch wenn diese Phänomene den verschiedenen äußeren Zwängen nicht gerecht werden, ist immer eine umfassende Neugestaltung erforderlich. Ein einfacher oktaler Patch oder die Überarbeitung eines isolierten Codes wird diese Art von Schwierigkeiten nicht beheben. Die erforderlichen Designänderungen sind wahrscheinlich so störend, dass die Softwareanforderungen, auf denen das Design basiert und die die Begründung für alles liefern, verletzt werden.«

Royce sagte, dass es riskant sei und Fehlern Tür und Tor öffne, die Tests als separate Phase am Ende durchzuführen. Er schlug einen iterativen Ansatz vor, der häufig getestet und auf der Grundlage der Ergebnisse angepasst wurde. Jeder Schritt ist dabei mit dem vorherigen verknüpft.

Software-Flexibilität und Refactoring

Scrum funktioniert aus verschiedenen Gründen in der Softwareentwicklung gut. In diesem Abschnitt sehen wir uns einige der Gründe an.

Wie wir in [Kapitel 4](#) erörtern, steigen die Kosten für die Fehlerbehebung exponentiell an, je später die Fehler gefunden werden. Fehler frühzeitig zu erkennen und zu beheben, ist zwar schwierig, aber machbar. Wenn Sie Defekte erst spät entdecken, können die angehäuften technischen Schulden kostspielig werden.

Auf jedem Schritt des Weges Qualität einzubauen, um Fehler zu beseitigen und zu reduzieren, ist ein Vorteil von Scrum. In den Abschnitten über Sprints in [Kapitel 5](#) stellen wir fest, dass das Testen jeder Anforderung und deren Integration mit anderen Anforderungen integraler Bestandteil des Prozesses ist. Trotzdem passieren Fehler.

Software kann sehr flexibel sein. Obwohl es nicht kostenlos ist, einen Entwickler Code-Refactoring durchführen zu lassen, ist der materielle Effekt virtuell, sodass die Arbeit in vielen Fällen relativ leicht erledigt werden kann.



Code-Refactoring ist der Prozess der Änderung von bestehendem Code, ohne sein beobachtbares Verhalten zu ändern. Ziel ist die Optimierung nicht-funktionaler Aspekte des Systems.

Refactoring ist in Scrum (auf positive Weise) unausweichlich, und Sie werden häufiger Refactoring vornehmen als je zuvor, weil Sie auf Veränderungen reagieren und diese willkommen heißen. Glücklicherweise diszipliniert der empirische Ansatz von Scrum Teams, kontinuierlich zu überprüfen und anzupassen, um den Zustand und die Qualität ihrer Produkte zu verbessern. Das Erkennen des Bedarfs an technischen Verbesserungen und deren Priorisierung gegenüber neuen Entwicklungen sind Teil dieser laufenden Wartung.

Code-Refactoring ist wie die Wartung Ihres Autos. Man hört das Rasseln, sieht das kleine rote Licht oder irgendetwas riecht verschmort. Sie wissen, dass diese Probleme größer werden, wenn Sie nichts tun. Sie können zwar so tun, als ob Sie eine Wahl hätten, diese haben Sie aber nicht. Es muss etwas geschehen. Wie beim Code-Refactoring unternehmen Sie die notwendigen kleinen Schritte, um sicherzustellen, dass Ihr Fahrzeug reibungslos läuft. Vielleicht braucht das Auto einfach mehr Öl oder eine Schraube muss angezogen werden. Aber wenn Sie warten, könnte Ihr Motor oder ein anderes Bauteil kaputt gehen, was zu viel höheren Kosten führen und einen größeren Zeitverlust mit sich bringen würde.

Refactoring ermöglicht es Ihnen, ein funktionierendes Modul zu erstellen, das skalierbar, erweiterbar und stabil ist. In der heutigen Welt, in der der Erste gewinnt, ist diese Art von Effizienz in der Qualitätskontrolle unbezahlbar. Führen Sie früh und oft Code-Refactoring durch.

Häufige und bedarfsgerechte Releases

Die Markteinführung neuer Produkte erfordert heutzutage keine Verpackung und keinen Versand mehr. Dank Internet, Cloud, automatisiertem Testen und kontinuierlicher Integration haben Sie die Flexibilität, Ihre Releases on demand und so oft wie nötig zu planen, um anderen Faktoren wie Marktreife, Kundenerwartungen und Marketingkampagnen gerecht zu werden.

Sie haben jetzt ein viel höheres Maß an Flexibilität in Bezug auf das von Ihnen erstellte Produkt und den Zeitpunkt, zu dem Sie es an Ihre Kunden releasen. Das Geld, das bisher für Papierprodukte ausgegeben wurde, wird nun für dringendere und gewinnbringendere Zwecke verwendet. Die Schaltfläche *Hier herunterladen* macht das Leben in vielerlei Hinsicht einfacher. In vielen Fällen, wie beispielsweise bei Software as a Service (SaaS, später in diesem Kapitel behandelt), werden Updates automatisch bereitgestellt, wodurch der Download-Schritt entfällt.

Da Scrum den Schwerpunkt auf die Produktion von auslieferbaren Produkten nach jedem Sprint oder sogar innerhalb von Sprints legt, haben Sie noch mehr Flexibilität bei der Planung von Produktreleases. Sie müssen nicht bis zum Ende eines Projekts warten, um es zu veröffentlichen. Sie können phasenweise und inkrementell releasen, und zwar geprüfte, integrierte und abgenommene Funktionalität freigeben.



Technologieriesen wie Facebook, Amazon und Google stellen mehrmals täglich Code in der Produktionsumgebung bereit – im Falle von Amazon 3.000 Mal täglich. Das ist schnell!

Anpassen der Release-Größe

Egal wie oft Sie releasen, kleiner ist in der Regel besser. Genau wie die Flexibilität durch häufige und bedarfsgerechte Releases ist auch die Größe des Produkts, das Sie freigeben, bei Scrum

unglaublich flexibel und basiert auf Funktionalität, Zielterminen und Marktbedingungen. Scrum funktioniert in allen Release-Größen, angefangen bei einer winzigen Anforderung, die eine größere, zuvor veröffentlichte Anforderung abrundet, bis hin zu einer großen neuen Produktgeneration, die das Licht der Welt erblickt.

Unabhängig davon, ob Sie eine einzelne Codezeile oder ein ganzes Bündel an Funktionen freigeben, das Produkt hat einen gründlichen Prozess durchlaufen. Qualität ist die Essenz der Flexibilität – und die Essenz von Scrum. Wenn die Kosten für Veränderungen und Verbesserungen zu groß werden, hat dies einen negativen Einfluss auf die Qualität.

Überprüfen und Anpassen beim Release

Durch die Möglichkeit, sowohl den Zeitpunkt als auch die Größe Ihrer Releases anzupassen, erhalten Sie sofortiges Kundenfeedback. Jedes Mal, wenn Sie ein Produkt freigeben, können Sie die Ergebnisse beurteilen und notwendige Änderungen vornehmen. Sowohl die Abteilung Analytics als auch der Kundendienst können Daten schnell erfassen und aus diesen Eingaben können Sie die zukünftige Entwicklung und Erweiterungen feinabstimmen.

Mit Scrum überprüfen und passen Sie täglich an, nach jedem Sprint und nach jedem Release. Diese Ereignisse können so oft stattfinden, wie Sie es möchten. In der Tat, je öfter Sie freigeben, desto öfter können Sie prüfen und anpassen.

Sie können sogar die Häufigkeit und den Fokus des Feedbacks anpassen. Kleinere Releases haben möglicherweise ein gezielteres Feedback. Größere Versionen können einer umfassenderen Auswertung unterzogen werden und aus dieser Auswertung können Sie kleinere, zielgerichtete zukünftige Versionen erstellen.



Die Sprint-Retrospektive (siehe [Kapitel 6](#) und [7](#)) zur Überprüfung und Anpassung Ihres Prozesses ist auch in Releases wichtig. Sie passen Ihren Prozess bei jeder Iteration an und nicht nur einmal in einem Meeting am Ende des Projekts, wenn alle erleichtert sind, dass es vorbei ist, und bereit sind, mit neuen Projekten zu beginnen.

Die Kunden haben eine große Auswahl, denn täglich kommen neue Produkte auf den Markt. Um zu überleben und um Konkurrenten zu übertreffen, muss man mindestens so flink sein wie diese. Scrum stellt genau hierfür den Rahmen zur Verfügung.

Veränderung positiv sehen

Obwohl Scrum aufgrund seiner Effektivität und Schnelligkeit in vielen Situationen eine Selbstverständlichkeit ist, ist dies bei hierarchisch strukturierte Organisationen anders. Schon diese Tatsache allein kann zu Problemen führen. Neue Ideen einzubringen und sich von alten Gewohnheiten zu lösen, kann eine Herausforderung darstellen. Einige Unternehmenskulturen sind offen für Veränderungen, andere haben damit größere Schwierigkeiten. Es liegt in der Natur des Menschen, Trost in seinen Gewohnheiten zu finden, und es erfordert Anstrengung und Energie, diese Gewohnheiten abzulegen.

Um die Menschen in der Organisation dazu zu bringen, die Veränderung anzunehmen, müssen Sie ihnen verdeutlichen, wie wertvoll Veränderung ist. Wenn die Beteiligten verstehen, was für sie selbst drin ist, akzeptieren sie die Änderungen leichter. Wenn die Beteiligten die Vorteile von frühzeitigen Releases und Anpassung erkennen, wird die Umstellung schneller vonstattengehen.

Um Ihnen den Übergang Ihres Unternehmens zum Scrum-Modell zu erleichtern, gehen wir in den folgenden Abschnitten auf einige spezifische Fragen ein.

Herausforderungen des Entwicklungsteams

In traditionellen Top-Down-Managementkulturen funktioniert die Implementierung von Scrum unter Beibehaltung des Status quo nicht. Das Entwicklungsteam kann durch Doppelarbeit frustriert sein, wenn es nicht nur Scrum implementiert. Manager können bei Projekten, die nicht im Voraus geplant sind und die nicht die gewohnte Vielzahl von Berichten und Analysen bieten, aus dem Ruder laufen.

Mit Scrum können Unternehmen die Überprüfung und Anpassung von einem Sprint zum anderen einfach gestalten. Es kann schwierig sein, Manager dazu zu bringen, sich angemessen (als Stakeholder) an den ersten Sprint-Reviews zu beteiligen. Ebenso herausfordernd ist es, die Menschen dazu zu bringen, funktionierende Software als neuen Maßstab des Erfolgs zu sehen (siehe den Kasten »Spotify versus Healthcare.gov« weiter hinten in diesem Kapitel).

Scrum ist eine neue Arbeitsweise für das Team sowie für alle Stakeholder und Einflussnehmer. Wenn die Organisation ausharrt und sich Zeit lässt, damit sich die Vorteile von Scrum manifestieren, sieht sie jedoch sofortige, praktische und qualitativ hochwertige Ergebnisse. Die Organisation entdeckt, dass sie nach jedem Sprint etwas Reales hat, mit dem gearbeitet werden kann, und dass sich der Fortschritt in einem logischen und schnellen Tempo ansammelt. Der Zeitaufwand für die Erstellung von Berichten und die Verwaltung des Status wird durch die Erstellung echter, brauchbarer Produkte ersetzt. Der Status ist klar, weil er echt ist.

Businessaspekte und Technik aufeinander ausrichten

Traditionell sind die geschäftliche Seite einer Organisation und das Software-Engineering getrennt, und oft sind sie uneins. Seit

Jahrzehnten arbeiten diese Gruppen als Silos innerhalb derselben Organisation:

- ✓ Das Unternehmen dient dem Markt und den Aktionären und steht unter extremem Druck, schnell Werte zu liefern und sich an einen sich ständig verändernden Markt anzupassen.
- ✓ Die Technologen wollen nachhaltige Architektur und Grundlagen mit geringstmöglichen technischen Schulden.

Scrum bringt diese beiden Seiten zusammen. Ein Vertreter des Unternehmens (Product Owner), ein vollwertiges Mitglied eines Scrum-Teams (auf der gleichen Ebene wie die anderen Teammitglieder) und Stakeholder sind während des gesamten Sprints involviert und geben direkt am Ende jeder Iteration Feedback (Sprint-Review). Sie werden nie eine bessere gemeinsame Ausrichtung, die Erleichterung der Zusammenarbeit und gemeinsame Ownership finden. Dank Scrum treffen die richtigen Leute die richtigen Entscheidungen, und zwar über die richtigen Dinge zur richtigen Zeit. Mit dem Finger auf andere zeigen ist nicht mehr relevant. Der Fortschritt ist für alle Beteiligten jederzeit sichtbar. Auf dem Weg dorthin gibt es Überprüfungs- und Anpassungspunkte. Selbstorganisation und Selbstmanagement bedeuten, dass alle daran beteiligt sind.

Reale Veränderung

Vor nicht allzu langer Zeit haben wir für einen unserer Software-Kunden ein Scrum-Training durchgeführt. Der Kunde arbeitete an einem Forschungs- und Entwicklungsprojekt mit neuen Technologien und erhielt von der Geschäftsleitung eine Frist zur Fertigstellung eines umfassenden Leistungsumfangs.

Das Team erarbeitete zunächst recht schnell einen Wirksamkeitsnachweis für Features und Anforderungen, mit dem das Management für die Durchführung des Gesamtprojektes gewonnen werden konnte. Das Produkt-Backlog war jedoch vage, und das Team verbrachte Monate

damit, eine Demoversion für das Management zu entwickeln. Hierfür war kein Budget eingeplant.

Der Entwicklungsprozess war langsam und voller technischer Überraschungen. Das Team verpasste nicht nur den ersten Releasetermin, sondern auch die nächsten. An dieser Stelle haben wir das Team geschult und das Scrum-Framework implementiert.

Zuerst hielt das gesamte Projektteam an den traditionellen Praktiken fest (vorheriges Design, festgelegter Umfang und Zeitplan, Umbenennung von Wasserfallartefakten und Ereignissen in Scrum-Terminologien). Dennoch trieb das Team das Projekt voran. Es wurden konsistent Sprint-Reviews mit Stakeholdern abgehalten und eine Reihe von harten, unbequemen Gesprächen über verpasste Termine geführt und darüber, wie die Aspekte von Scrum helfen könnten, diese Herausforderungen zu lösen.

Nach einigen Sprint-Reviews, der Klärung der Rollen (vor allem, dass man einen engagierten und gut ausgebildeten Product Owner bekommt) und der Beseitigung einiger Hindernisse begann die Transparenz, die Scrum innewohnt, sichtbar zu werden. Die Mauern zwischen dem Scrum-Team, den Stakeholdern und den Kunden (die tatsächlichen Nutzer waren an jedem Sprint beteiligt) wurden abgerissen; die Kommunikation wurde verbessert und das Projekt wurde mit einer angemessenen Vision, einer Roadmap und einem Produkt-Backlog neu geplant. Auf Basis der neuen Velocity des Teams und neuer Schätzungen des Produkt-Backlogs wurde ein neuer Releaseplan erstellt. Lange Rede kurzer Sinn, von diesem Zeitpunkt an dauerte es Monate und nicht Jahre, bis das minimal funktionsfähige Produkt (MVP, Minimal Viable Product) auf den Markt kam. Der neue Projektplan basierte auf Priorität, Kundenfeedback und empirischen Daten, wie beispielsweise der Velocity.

Während der ersten Sprint-Reviews identifizierte das Team einen wichtigen Stakeholder (einen Kunden, der das Produkt

verwenden würde), der nicht an den Sprint-Reviews teilgenommen hatte. Er wurde eingeladen, und sein Engagement lieferte entscheidendes Feedback zum Produkt. Als die anderen Stakeholder die Bereitschaft und Schnelligkeit des Scrum-Teams bei der Überprüfung und Anpassung beobachteten, stieg das Vertrauen sprunghaft an, die Eigenverantwortung und die Zusammenarbeit (einschließlich der Stakeholder) verbesserten sich erstaunlich.

Am Ende spiegelte das Produkt genauer wider, was der Kunde wünschte; zusätzliche Funktionen wurden priorisiert und weggelassen (das hatten ein Stakeholder und der Product Owner entschieden, nicht das Entwicklungsteam); ein MVP wurde in vollem Funktionsumfang freigegeben und die Stakeholder waren voll in die Schulung der Benutzer der Software involviert. Tatsächlich arbeiteten die Stakeholder mit dem Product Owner bei jedem Aspekt des Rollouts zusammen.

Upfront-Engineering

Die Neigung von Entwicklungsleitern, Projektleitern und Entwicklern, alles vorher, also upfront, zu planen, hat tiefe Wurzeln:

- ✓ Die meisten Hochschulen unterrichten immer noch das Upfront-Engineering (siehe »Der Marshmallow-Test« in [Kapitel 1](#)). Direkt nach dem Studium planen Ingenieure zuerst und entwickeln erst danach.
- ✓ Der Wasserfall erzwingt die Denkweise »alles planen und dann alles entwickeln«. Dieser Prozess basiert darauf, zuerst eine Phase, etwa den Entwurf, abzuschließen, bevor mit der nächsten Phase, zum Beispiel der Entwicklung, weitergemacht wird. Beim Wasserfall ist das Upfront-Engineering Teil des Fundaments, auf dem alles andere ruht.

Vor diesem Hintergrund werden Sie feststellen, dass die Vorstellung, etwas zu ändern, nachdem ein Projekt begonnen hat, als negativ betrachtet wird. Aber in unserer Denkweise und im Rahmen von Scrum ist Veränderung ideal. Alle Veränderungen sind Lernprozesse. Änderungen werden jeden Tag, nach jedem Sprint und in jedem Release integriert. Sie werden merken, wie Sie nach Veränderung suchen. In Scrum bedeutet Veränderung Fortschritt.

Auch wenn Management und Entwickler Zeit und Energie in die Planung investiert haben, ist es schwer, diese Pläne fallen zu lassen und möglicherweise versunkene Kosten zu erleiden.



Irreversible Kosten (auch *Sunk Costs* oder *versunkene, verlorene Kosten* genannt) sind Finanzmittel, die für ein Projekt ausgegeben wurden und nicht rückgängig gemacht werden können. Im Wasserfall waren versunkene Kosten häufiger anzutreffen, da Projekte im Vorfeld geplant und finanziert wurden. Wenn sie im Verlauf des Projekts scheiterten, hatten die Unternehmen nichts vorzuweisen. Mit Scrum haben Sie die Möglichkeit, schnell zu scheitern oder inkrementell die gesamte Funktionalität zu erstellen. Anstatt im Vorhinein viel Geld in Projekte zu stecken, können Sie auf dem Weg dorthin Mittel bereitstellen. Wenn das Projekt früher endet als geplant, verlieren Sie weniger Geld in Form von versunkenen Kosten oder treffen eine großartige Geschäftsentscheidung, die die Rendite Ihrer begrenzten Investition maximiert. Mit Scrum sind die Tage des katastrophalen Scheiterns vorbei.

Ungefähr 64 Prozent der Software-Features werden nie oder nur selten genutzt. Denken Sie an die Verschwendung von Zeit, Energie und Ressourcen, die mit der Planung, dem Design, der Entwicklung, dem Testen und der Dokumentation dieser Features verbunden ist. Mit Scrum wird das vorherrschende Paradigma, alles im Voraus zu planen, aufgehoben.

Wir haben gesehen, wie Entwicklungsteams, die das Wasserfallmodell verwenden, Wochen und Monate mit der Planung verbracht haben. Wir haben gesehen, wie Scrum-Teams Features und Systeme der gleichen Größenordnung in Stunden geplant haben.

Weil Scrum-Teams verstehen, dass sie nicht alles im Voraus wissen, planen sie nicht, was sie nicht wissen. Sie bewerten und passen sich an, wenn neue Informationen gesammelt und Prioritäten neu bewertet werden. Da die Transparenz dauerhaft ist, verringert sich die Wahrscheinlichkeit einer Überreaktion beim Management.

Sich entwickelnde Architektur

Emergent Architecture, was sich am besten mit sich entwickelnde Architektur übersetzen lässt, ist die schrittweise Ausarbeitung der Architektur Ihrer Anwendung oder Ihres Unternehmens, basierend auf den Anforderungen, die sich derzeit in der Entwicklung befinden, um Stabilität und Skalierbarkeit zu erreichen und zu erhalten. Das Ziel ist, die Architektur auf der Grundlage der Realität einer funktionierenden Software zu überarbeiten, anstatt das gesamte System auf der Grundlage von Vermutungen zu entwickeln.

Viele Programmierer betrachten das Konzept des sich entwickelnden Designs als Zeichen des Scheiterns. Sie wollen, dass das gesamte Produkt im Voraus geplant wird und kein Code (oder Stein) umgedreht wird. Jedoch ist dieser Ansatz weder realistisch noch praktisch.

Sie beginnen mit Architekturentscheidungen auf Makroebene, basierend auf der ganzheitlichen Betrachtung der Anforderungen, die Sie aus der Produkt-Roadmap haben. Dann, in Ihrem ersten Sprint, fragen Sie, was die Anforderungen und die Architektur für diesen Sprint sind. Lassen Sie das Sprint-Ziel den Sprint steuern, und wenn dieses Ziel ein Code-Refactoring erfordert, warten Sie nicht. Ihr Produkt sollte in jedem Sprint funktionieren. Mit Scrum führen Sie häufiger Code-Refactoring durch, und genau das

wollen Sie erreichen. Anforderungsänderungen sind nicht Ihr Feind, ebenso wenig wie Code-Refactoring.

Der Anfang des Projekts ist der Zeitpunkt, zu dem Sie am wenigsten darüber wissen. Zu diesem Zeitpunkt sollten Sie keine detaillierten Planungen vornehmen oder Teile des Systems entwickeln, die nicht verwendet werden. Sie können ein Lean-Prinzip auf dieses Szenario anwenden: Entscheiden Sie im letzten verantwortbaren Moment. Je früher im Projekt Sie eine Entscheidung treffen, desto größer ist das Risiko, dass die Entscheidung falsch ist und desto größer ist die Wahrscheinlichkeit, dass Sie unnütze Arbeit machen. Je später Sie sich entscheiden, desto mehr Informationen stehen Ihnen für Ihre Entscheidung zur Verfügung und desto geringer ist Ihr Fehlerrisiko.



Refactoring sollte in jede Aufwandsschätzung der Anforderungen integriert werden. Wenn neue Anforderungen ganz oben auf der Liste stehen und verfeinert werden, entsteht in den Köpfen des Entwicklungsteams das Design. Als Teammitglieder schätzen und planen sie das Refactoring.



Wenn Ihre Sprints eine Woche dauern, werden Sie kleine Chargen der Architektur überarbeiten. Wenn Sie bis zum Ende des Projekts warten, wie in traditionellen Frameworks, refaktorisieren Sie monatelang oder jahrelang die Architektur. Refaktorisieren Sie oft. Refaktorisieren Sie früh.

Dank Scrum gewonnene Erkenntnisse

7Shifts ist ein Unternehmen, das eine Software zum Erstellen von Dienstplänen für Restaurants entwickelt hat und das von Jordan Boesch und seiner Frau Andrée Carpentier gegründet wurde. Als sie ihr Geschäft aufbauten, trafen sie auf einen Kunden, der ihnen unbeabsichtigt und ohne einen Cent für ihre Produkte auszugeben, half, einen Wasserfall in eine Scrum-Lösung umzuwandeln.

Dieser potenzielle Kunde, ein Restaurantbesitzer, bat um einige Features in seiner App, von denen eines ein Budget-Tool war. Jordan entwarf alle Funktionen bis zur Fertigstellung, einschließlich der Budget-App, und präsentierte sie dem Kunden. Aus irgendeinem Grund unbeeindruckt, sagte der Kunde »Nein danke« und sie hörten nie wieder etwas von ihm. Es hatte also eine Menge vorheriger Planung und Entwicklung ohne Rückmeldung eines tatsächlichen Anwenders gegeben.

Etwas verwirrt machte Jordan mit seiner Anwendung weiter und band auch die Budget-Tool-App ein, ohne sie weiter zu inspizieren. Eines Tages führte er einige Analysen des Tools durch, um zu sehen, wie häufig seine Kunden es benutzten. Die Daten zeigten, dass nur 2 Prozent der Kunden es verwendeten.

Jordan machte sich daran, die App zu verbessern, indem er zunächst die bestehenden Kunden fragte, warum sie das Budget-Tool nicht benutzen. Sie erklärten, dass sein Programm von ihnen verlangte, einen geschätzten Dollarbetrag manuell einzugeben, obwohl er mit einigen Optimierungen leicht automatisch hätte berechnet werden können.

Jordan veröffentlichte ein Update der App und hatte damit erstaunlichen Erfolg. Er hatte gelernt, nur mit dem zu planen, was ihm bereits bekannt war, frühzeitig mit dem Kunden zu sprechen, zu überprüfen und anzupassen, bis die Kunden zufrieden waren.

Scrum-Anwendungen in der Software

Viele Technologieunternehmen setzen Scrum bereits erfolgreich ein. Sie können damit schneller und flexibler sein, um wettbewerbsfähig zu bleiben.

Es ist eine Sache, die Theorie des Scrum-Erfolges zu verstehen; es ist eine ganz andere, Scrum in Aktion zu sehen. Obwohl wir Sie nicht mit in den Praxisalltag nehmen können, können wir in den folgenden Abschnitten den Praxisalltag zu Ihnen bringen. In diesem Abschnitt zeigen wir Ihnen einige Herausforderungen, vor denen Unternehmen stehen und beschreiben, wie die einzigartige Anpassungsfähigkeit von Scrum zum Tragen kommt.

Entwicklung von Videospielen

Dieser genauso beliebte wie komplexe Bereich der Softwareentwicklung kann aus vier Gründen Scrum mit großem Erfolg nutzen:

- ✓ **Flexibilität:** Scrum beginnt mit grundlegenden Funktionen und erweitert das Spiel um mehr Komplexität. Diese Entwicklung geschieht organisch, deshalb entwickeln Sie immer das Wichtigste.
- ✓ **Spaßfaktoren finden:** Entwickler können unterhaltsame, lustige und Spaß bringende Elemente in kleinen, iterativen Dosen hinzufügen. Die Features, die den größten Fun-Faktor (ein entscheidendes Element eines jeden Spiels) bringen, werden zuerst hinzugefügt.
- ✓ **Kosteneinsparungen:** Die meisten Spiele scheitern und erreichen nicht den Break-even-Punkt. Die Kostenersparnis von Scrum kann Unternehmen retten. Selbst wenn die Finanzierung mitten im Projekt gekürzt wird (vorausgesetzt, dass die minimale Spielzeit eingehalten wird), sind die wichtigsten 50 Prozent der Features voll funktionsfähig und können, auch wenn sie noch nicht marktfähig sind, zumindest wiederverwendet werden.
- ✓ **Regelmäßiges Feedback:** Dieses Feedback wird täglich durch den Product Owner und bei jedem Sprint-Review von den Stakeholdern (zum Beispiel Herausgeber/Verlag, Marketing-Mitarbeiter) gegeben.

Beim Spielen geht es darum, dass die Kunden Spaß haben und am Ball bleiben. Feedback ist innerhalb von Scrum schnell integriert, sodass das Ergebnis genau den Wünschen der Benutzer entspricht.

Entwicklungsphasen

Bei Spielen folgt die technische Entwicklung in der Regel den drei Phasen, die in [Abbildung 8.1](#) dargestellt sind.

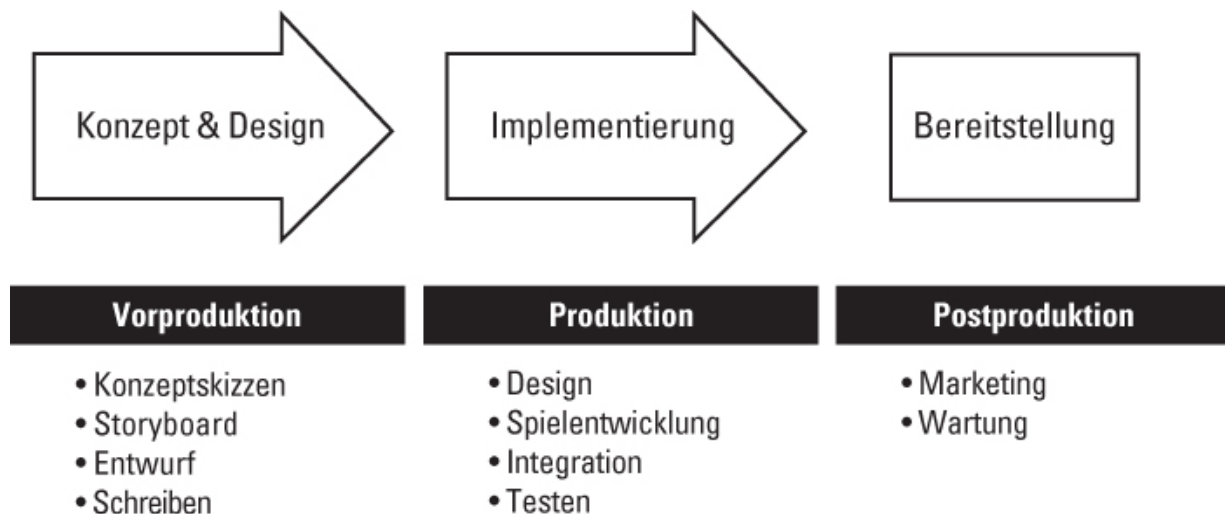


Abbildung 8.1: Der klassische Ablauf der Entwicklung von Videospielen

- ✓ **Präproduktion:** Hier kommen Künstler, Regisseure und Ingenieure zusammen, um die Idee zu entwickeln. Der Sinn dieser Prototyping- und Proof-of-Concept-Phase ist es, festzustellen, ob das Spiel eine gute, lustige Idee ist, bevor das Unternehmen die Produktionsphase startet und mehr Geld ausgibt.

Speziell für Handyspiele können Entwickler schnell an Prototypen arbeiten, ohne dabei auf alle künstlerischen Details einzugehen. Für alle Spiele in dieser Phase können Teams die Idee des Spiels validieren, sie definieren, grafische Konzepte entwickeln, die Finanzierung sicherstellen und ein Entwicklungsteam zusammenstellen.

- ✓ **Produktion:** Der Proof of Concept wird von Regisseuren, Künstlern und Ingenieuren entwickelt und umgesetzt.

- ✓ **Postproduktion:** Professionelle Tester testen das fertige Produkt, es folgt der Betatest und es beginnen Vertriebs-, Marketing- und Supportaktivitäten.

Wie dieser Prozess impliziert, arbeiten die Spieleanbieter in der Regel mit Studios unter traditionellen Wasserfallverträgen und -strukturen zusammen und bezahlen das Studio nur bei bestimmten langfristigen Meilensteinen. Diese Regelung impliziert oft, dass der Spieleanbieter über lange Zeiträume keine Fortschritte sieht.

Bei Verwendung des Scrum-Modell sieht die Spieleentwicklung eher wie in [Abbildung 8.2](#) aus, die das aus Gründen der Qualitätssicherung erforderliche kontinuierliche Testen darstellt. Nach jeder Aufgabe – wie Konzeptgrafiken, Storyboarding, Schreiben oder Prototyping – wird die Aufgabe getestet und weiterentwickelt.

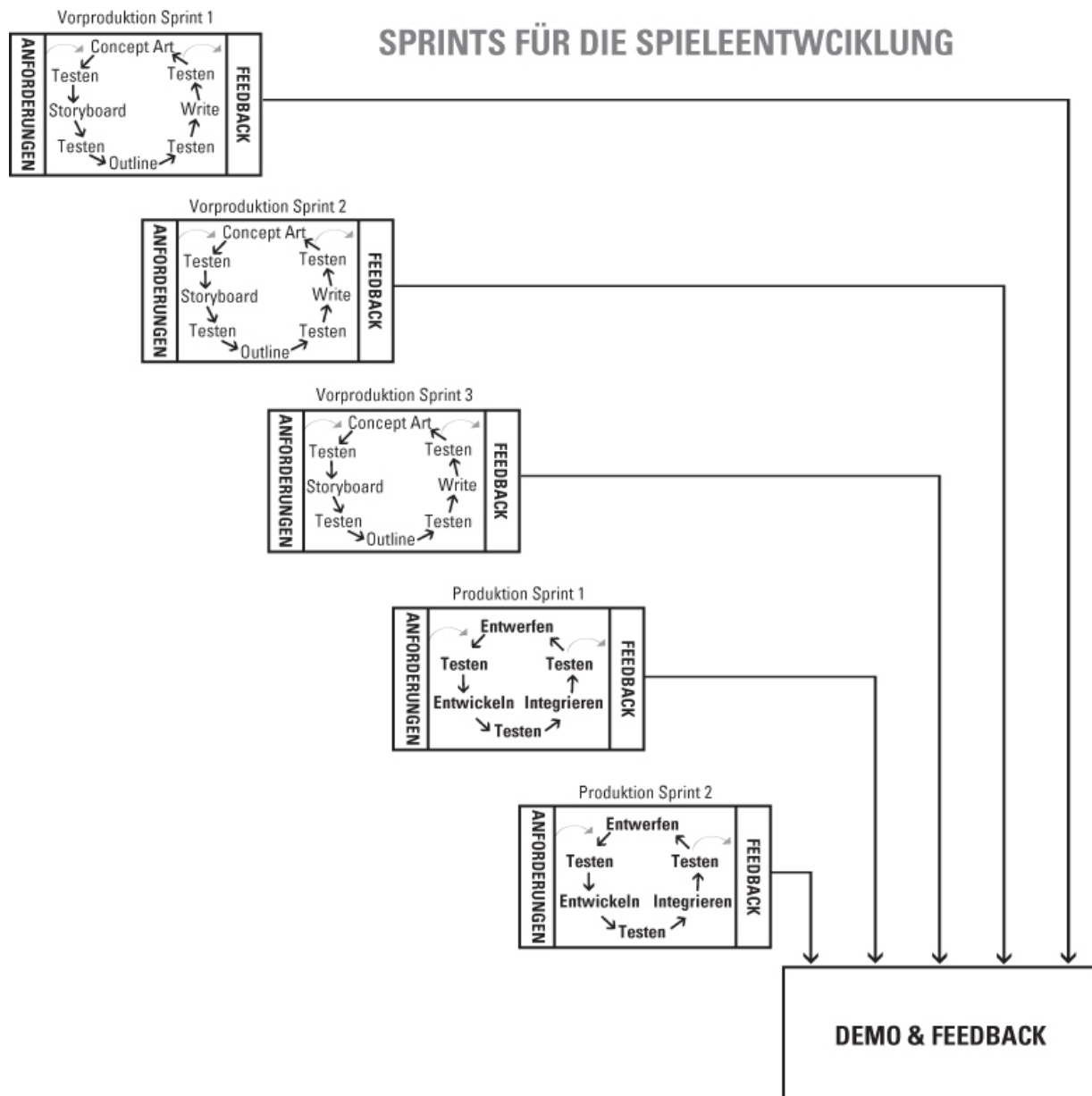


Abbildung 8.2: Der Spielentwicklungsprozess mit Scrum

Mit Scrum können die Spieleanbieter die Spielelemente, die keinen Spaßfaktor bringen, schneller erkennen und entfernen. Außerdem können sie die Qualität besser kontrollieren, da sie regelmäßig an Sprint-Reviews teilnehmen und die Studios inkrementell bezahlen, entsprechend der Lieferung von funktionierenden Spielinkrementen.

Kurze Sprints sorgen für einen permanenten Fluss ausgelieferter Inhalte, der die Konsumenten anzieht und bindet. Scrum erlaubt

es Stakeholdern wie Spieleanbietern und Fokusgruppen, schneller spielbare Spiele verfügbar zu machen. Dieser Prozess ermöglicht es Entwicklern, den Spaßfaktor hervorzuheben und die nicht unterhaltsamen Spielelemente schnell zu eliminieren (was auch zur Vermeidung unnützer Kosten für deren künstlerische Umsetzung führt).



Die Definition of Done der Vorproduktion und der Produktion können sich unterscheiden. Achten Sie darauf, beiden Phasen im Vorfeld zu klären. Die Definition of Done der Präproduktion-Sprints kann andere Arten und Ebenen von Dokumentation und grafischen Standards beinhalten als die Definition of Done der Produktionsphase.



Idealerweise sollten Spiele nicht mit weniger als zehn Stunden Spielzeit für ein Einzelspieler-Spiel ausgeliefert werden. Obwohl diese Spielzeit das Release-Ziel sein kann, können die Publisher nach jedem Sprint die funktionierende Software überprüfen und wertvolles Feedback geben.

Marketing

Spiele werden traditionell als Einzelhandelsprodukte verkauft, aber der Online- und digitale Vertrieb verlagert den Schwerpunkt vom Einzelhandel auf die digitale Auslieferung. Es besteht ein enormer Bedarf, Inhalte ständig und schnell zu aktualisieren, hinzuzufügen und zu verbreiten. Diese Fähigkeit macht den Unterschied zwischen überleben und nicht überleben aus. Schnelligkeit bei der Markteinführung ist entscheidend.

Die Spieleverkäufe sind überwiegend saisonabhängig, wobei mehr als 50 Prozent des Umsatzes der Branche in der Vorweihnachtszeit erzielt werden. Diese Tatsache ist ein wichtiger Faktor in der entscheidenden Phase, mit der viele Teams konfrontiert sind, da die Weihnachtssaison eine harte Deadline auferlegt.

Das prioritätsgesteuerte Produkt-Backlog von Scrum stellt sicher, dass die Spielelemente mit der höchsten Priorität und dem größten Spaßfaktor in kürzester Zeit geliefert wird, was nicht nur das Management glücklich macht, sondern auch den Burnout der Mitarbeiter reduziert.

Grafiken

Grafiken sind ein wesentlicher Teil der Entwicklungsarbeit in Videospielen. Konsolen- und Computerspiele erfordern viele Grafik- und Animationselemente, diese machen bis zu 40 Prozent oder mehr der Kosten eines Spieltitels aus. Technische und spielerische Probleme in der späten Produktion sind kostspielig.

Da die künstlerische Gestaltung das teuerste Element ist, ist es verlockend, sie ins Ausland auszulagern. Probleme mit Zeitzonen und Kommunikation führen jedoch fast immer zu Verzögerungen, zu Nacharbeit und zur Unzufriedenheit der Kunden mit der Qualität des fertigen Produkts.

Scrum ermöglicht es Künstlern und Entwicklern, im selben Team und nicht in Silos zu arbeiten, mit den gleichen Zielen und in Echtzeit darüber zu kommunizieren, was funktioniert und was nicht. Um die Zusammenarbeit zu optimieren, sollten die Teams im gleichen Land und am gleichen Standort angesiedelt sein. Onshore-Teams und Kollokation führen zu geringen Verzögerungen und Nacharbeitskosten.

Durch die zeitgesteuerte Lieferung (Sprints) der grafischen Bausteine bleiben alle Beteiligten im Zeitplan, wobei die Fristen als Team festgelegt werden und ein stetiger Fluss von Ergebnissen auf den Markt kommt, wodurch die Verschwendung auf ein Minimum reduziert wird.

Viele große Spiele erfordern große Entwicklungsteams mit 15 oder mehr Entwicklern, 10 oder mehr Grafikern und 30 oder mehr Mitgliedern der Qualitätssicherung. Eine ausführliche Erörterung darüber, wie Scrum über größere Projekte hinweg skaliert wird, finden Sie in [Kapitel 13](#).

Dienstleistungen

Entwicklungsfirmen, die ihre Produkte als Dienstleistung, »as-a-Service«, vermarkten, verwenden Scrum, um ihre Produkte schnell und effektiv freigeben zu können, indem sie die Möglichkeiten des Internets nutzen.

SaaS

Software as a Service (SaaS) ist ein Modell zur Verteilung von Softwareanwendungen, die von einem Service Provider gehostet und den Benutzern über das Internet zur Verfügung gestellt werden. Es ist keine Installation auf PCs oder Geräten erforderlich. Kunden zahlen in der Regel eine monatliche Gebühr oder haben ein Abonnement abgeschlossen.

Sogar mobile Anwendungen, die einen Download erfordern, bieten schnelle Downloads, die unterwegs, über Wi-Fi oder über ein mobiles Netzwerk während des Roaming durchgeführt werden können. Installations-CDs, manuell eingegebene Lizenzschlüssel und stundenlange Installationsschritte bei eingestecktem Gerät gehören der Vergangenheit an. Die Cloud ermöglicht es den Scrum-Teams, auf der Basis von Echtzeitdaten Überprüfungen und Anpassungen vorzunehmen und Änderungen schnell bereitzustellen, um so auf Feedback zu reagieren.

IaaS

Infrastructure as a Service (IaaS) ist ein Cloud-Computing-Modell, das Unternehmen die zur Unterstützung des IT-Betriebs erforderliche Ausrüstung zur Verfügung stellt. Dazu gehören unter anderem Hardware, Server, Speicher und andere Netzwerkkomponenten. IaaS-Anbieter besitzen die Geräte und sind für deren Beschaffung, Betrieb und Wartung verantwortlich. Kunden zahlen in der Regel pro Nutzung, basierend auf dem Datenvolumen oder der Größe der Geräte. Dieselben Produkte ermöglichen es anderen Unternehmen, ihre IT-Abteilungen zu verwalten und ihre Infrastruktur auf eine Scrum-ähnliche Art und Weise zu betreiben.

So dauerte beispielsweise die Installation eines neuen Datenbankservers früher Wochen oder Monate. Der hauseigene Techniker musste die Geräte beschaffen und dann bauen, konfigurieren und installieren, bevor die Datenbankingenieure mit der Installation der Datenbank und der Datenmigration beginnen konnten. IaaS-Lösungen ermöglichen es nun, diesen Prozess virtuell und in Minutenschnelle durchzuführen.

PaaS

Platform as a Service (PaaS) ist ein Cloud-Computing-Modell, das eine Computing-Plattform und Stack-Lösungen (Satz von Software-Subsystemen) bietet. Kunden zahlen in der Regel Lizenzgebühren und/oder Gebühren pro Nutzung.

Mit IaaS- und PaaS-Diensten wie Amazon können IT-Organisationen innerhalb weniger Minuten einen neuen Server in Betrieb nehmen, der dann umgehend für die Daten und Installation bereitsteht.

Kundenspezifische Projekte

Manche Software kommt von der Stange, manche wird individuell auf die Bedürfnisse des Kunden zugeschnitten. Wie geht Scrum mit kundenspezifischen Anpassungen um? Benötigen Sie individuelle Teams für jeden Kunden?

Wir haben mit einem Kunden zusammengearbeitet, der eine Logistiksoftware anbietet, die umfangreiche Anpassungsmöglichkeiten vorsieht, damit sie mit besonderen Ausrüstungskonfigurationen funktioniert und Software von Drittanbietern und Legacy-Komponenten integrieren kann. Vor dem Einsatz von Scrum lagen die Projektteams, die den Kunden die Anpassungsdienste zur Verfügung stellten, so weit zurück, dass durch die lange Vorlaufzeit für die Inbetriebnahme durch einen neuen Kunden manche Aufträge nicht abgeschlossen werden konnten.

Die Lösung bestand darin, die Projektteams, die die Anpassung durchführten (und die noch kein Scrum einsetzten) als Stakeholder zu den Sprint-Reviews des Scrum-Teams für das

Kernprodukt einzuladen. Diese Änderung ermöglichte es diesen Stakeholdern, das Produkt frühzeitig zu antizipieren. Die Stakeholder konnten Feedback über die Nutzung der Software durch die wichtigsten Kunden geben. Dies erlaubte es dem Scrum-Team, das Kernprodukt so weiterzuentwickeln, dass für die einzelnen Kunden weniger Anpassungen erforderlich waren. Das Team sah, wie bestimmte Features eingesetzt wurden und wie sie dieses Wissen in Gesprächen mit anderen Kundenvertretern nutzen konnten.

Spotify versus Healthcare.Gov

Um die Effekte der Nutzung von agilen und auf Scrum basierenden Frameworks innerhalb von Services Computing und SaaS zu verstehen, vergleichen Sie zwei Beispiele: Spotify und Healthcare.gov.

Was auch immer Ihre politischen Ansichten über den Affordable Care Act sind (Bundesgesetz der USA, das unter anderem den Zugang zur Krankenversicherung regelt, auch als Obamacare bekannt) –die Website Healthcare.gov ist eine sehr große Bereitstellung einer Cloud-Computing-Lösung mit unzähligen Problemen, die eine umfassende Lösung erforderten. Jeff Sutherland, einer der Erfinder von Scrum, nannte diesen Rollout »missglückter Wasserfall«.

Healthcare.gov war ein agiles Projekt am Frontend, aber die Entwickler haben das zweite Agile Prinzip außer Acht gelassen: funktionierende Software. Die Leute am Frontend trugen ihren Teil dazu bei, aber das Fehlen von Last- und Leistungstests auf der eigentlichen Hardware machte die Software unbrauchbar.

Obwohl Dutzende von renommierten Beratungsfirmen an dem Projekt arbeiteten, fand zwischen ihnen nur eine minimale Koordination statt. Die Anwendung des Programms wurde landesweit eingeführt, aber wenn es Bundesstaat für Bundesstaat getestet worden wäre (getreu und agil), hätten viele der Probleme früh und schnell gelöst werden können.

Spotify, der Online-Musikdienst, machte es besser und erzielte beachtliche Erfolge. Spotify hat Scrum und Agilität so weit implementiert, dass alle seine Scrum Master agile Trainer sein müssen.

Spotifys Konkurrenten sind Google, Amazon und Apple. Daher musste Spotify flinker, schneller und billiger sein, um überhaupt daran denken zu können, den Konkurrenzstreit mit diesen Mitwettbewerbern anzugehen. Das Unternehmen verfügt über Standorte und Scrum-Teams auf der ganzen Welt und jedes Team ist für ein einzelnes Produkt verantwortlich. Aber auch diese Struktur war nicht schnell genug, sodass das Unternehmen die kontinuierliche Bereitstellung entwickelte und das Produkt mehrmals während eines Sprints in die Produktionsumgebung überführt.

Wenn Sie diese beiden riesigen Cloud-Computing-Dienste vergleichen, von denen der eine das Wasserfallmodell verwendet und der andere Scrum einsetzt, dann erkennen Sie die Notwendigkeit, funktionierende Software häufig zu liefern, zu überprüfen und anzupassen.

Kapitel 9

Produktion materieller Güter

IN DIESEM KAPITEL

Sich vom Wasserfall wegentwickeln
Verbesserung von Bauprojekten
Einsatz von Scrum-Lösungen in der Fertigung
Entwicklung von Hardware mit Scrum

Sogar ein Fehler kann sich als das Einzige erweisen, was für einen beachtlichen Erfolg notwendig ist.

Henry Ford

Angeichts der Ursprünge von Agilität und Scrum liegt die Vermutung nahe, dass beides nur für die Entwicklung von Software verwendet werden sollten. Tatsache ist, dass nichts weiter von der Wahrheit entfernt sein könnte. Der Artikel »New New Product Development Game«, der im Jahre 1986 im Harvard Business Review erschien und in diesem Buch in den [Kapiteln 1](#) und [8](#) erwähnt wird, wurde als Antwort auf die Erfahrungen in der Fertigungsindustrie geschrieben. Der Artikel verwendet als Fallbeispiele Hardware, Elektronik und Automobilprodukte, die von Unternehmen wie Fuji-Xerox im Jahr 1978, Canon im Jahr 1982 und Honda im Jahr 1981 entwickelt und gebaut wurden.

Die Entwicklung und Produktion von Sachgütern funktioniert wunderbar mit Scrum, und wir zeigen Ihnen in diesem Kapitel, wie es geht. Situationen und Produkte können sehr unterschiedlich sein, aber die Werkzeuge und Techniken bleiben die gleichen. Indem Sie die von uns beschriebenen Verfahren des Frameworks befolgen, werden Sie leicht erkennen, dass Scrum für die meisten

Projekte geeignet ist. Alles, was Sie brauchen, ist eine Liste von Anforderungen, die priorisiert werden können.

Der Sturz des Wasserfalls

Wie wir in [Kapitel 8](#) darlegen, findet die Produktentwicklung mit zunehmender Geschwindigkeit statt, und nicht nur die Softwarebranche sieht grundlegende Veränderungen darin, was entwickelt wird und wie lange die Entwicklung dauert. Auch die langfristige manuelle Fertigung verändert sich. Die Entwicklungszyklen sind in den meisten Branchen aufgrund der Zeit- und Kostenersparnis durch den Einsatz von Technologie kürzer und schneller. Unternehmen wie Apple, deren Produkte sicher materiell sind, bringen mehrmals im Jahr innovative Produkte der nächsten Generation auf den Markt.

Einige argumentieren, dass das Scrum-Framework nicht auf materielle Produkte angewendet werden kann und sagen: »Wir können nicht den zehnten Stock bauen, wenn wir erkennen, dass wir im fünften Stock etwas ändern müssen!« Manchmal haben sie recht.

Solche Einwände macht man auch in der Softwareentwicklung immer wieder. Eine Designänderung kann so kostspielig sein, dass sie nicht realistisch umgesetzt werden kann, und es müssen andere Optionen in Betracht gezogen werden, einschließlich der Option, ohne Änderungen weiterzumachen. Vielleicht wäre eine andere Datenbank vorzuziehen, aber sie ist zu kostspielig, also beschließt der Product Owner, die Idee zu ignorieren und so weiterzumachen wie bisher.

Bei Scrum geht es darum, die Realität aufzudecken und gute Entscheidungen auf der Grundlage empirischer Erkenntnisse zu treffen, nicht darum, die Anforderungen nach Lust und Laune des Kunden auszutauschen. Ob es sich um die Entwicklung von Software für die Produktion von Flugzeugen, Zügen oder Automobilen handelt, es gelten immer die Grundlagen von Scrum: die Organisation eines Scrum-Teams mit entsprechenden Rollen,

häufige Überprüfungs- und Anpassungspunkte und zu erzwingen, dass Entscheidungen im letzten verantwortbaren Moment getroffen werden (das heißt, auf verschiedenen Detailebenen die richtigen Dinge zur richtigen Zeit zu planen).



Aufgrund der Feedbackschleife hat der Kunde über den Product Owner Einfluss auf einige Geschäftsentscheidungen. Auf diese Weise muss das Entwicklungsteam nicht alle Markt- und Kundennuancen verstehen, die für die Geschäftsentscheidungen notwendig sind. Der Product Owner filtert das Rauschen aller Beteiligten heraus, sodass die Entwickler ihre Zeit nur für die Anforderungen mit der höchsten Priorität verwenden. Product Owner und Entwickler arbeiten im Team, um die Problemlösungen, die der Product Owner präsentiert, umzusetzen. Wenn alle auf die gleichen Ziele hinarbeiten, wird man weniger mit dem Finger auf jemanden zeigen und stattdessen die Verantwortlichkeit gemeinsam getragen.

Engagierte Kunden sind zufriedener Kunden, die am Ende bekommen, was sie wollen. Mit Scrum finden Feedbackschleifen regelmäßig, konsistent und so häufig wie möglich statt, sodass Kunden das bekommen, was sie wollen, und zwar zu den Kosten, die sie dafür zu zahlen bereit sind; hierdurch bleiben Kunden und Entwicklungsteams auf Kurs. Hindernisse können frühzeitig erkannt und behoben werden und die Verantwortung wird von denjenigen übernommen, die die Arbeit tun *und* von denen, die Entscheidungen treffen, wozu auch die Kunden gehören.



Obwohl die folgenden Abschnitte dieses Kapitels als Leitfaden für den Einsatz von Scrum in bestimmten Branchen dienen, lassen sich die Scrum-Rollen, die dieses Framework so effektiv machen, leicht auf neue Projekttypen anpassen.

Baugewerbe

Der Einsatz von Scrum in der weiten Welt der materiellen Güter ist enorm. Die im Bereich der Software erzielbare Geschwindigkeitssteigerung und Kosteneinsparung ist auch in anderen Bereichen möglich. Scrum im Baugewerbe ist ein hervorragendes Beispiel.

Scrum funktioniert mit jedem Projekt, vorausgesetzt, Sie können die zu erledigende Arbeit auflisten und priorisieren. Natürlich haben Bauprojekte diese Eigenschaften. Sie bieten auch branchenspezifische Herausforderungen. Jedes Segment einer Branche tut das. Anstatt zu sagen: »Oh, Scrum wird wegen X und Y und Z nicht funktionieren«, sagen wir: »Das sind die Herausforderungen, vor denen wir stehen, also schauen wir uns an, wie Scrum sie lösen und uns schneller und billiger auf den Markt bringen wird«.

In den folgenden Abschnitten skizzieren wir spezifische Probleme in der Baubranche und beschreiben, wie Scrum dabei hilft, diese zu lösen. Sie sehen, wie Sie mit den traditionellen Problemen, mit denen Sie während Ihrer gesamten beruflichen Laufbahn in der Bauwirtschaft zu tun hatten, umgehen können, indem Sie Scrum verwenden.

Die besten Angebote

Der Wettbewerb im Ausschreibungsverfahren für Bauprojekte ist hart und es gibt keine Anzeichen dafür, dass sich dies ändert und einfacher wird. Kombinieren Sie diese Tatsache mit einem stetigen Anstieg der Baukosten, und Sie haben eine Formel, um Dinge abzukürzen, und für die Möglichkeit, wesentliche Schritte zu überspringen.

Nichts davon ist neu in dieser Branche. Ihr Angebot kann tadellos sein und alles bieten, was der Kunde zu einem vernünftigen Preis benötigt, aber Sie wissen, dass Ihre Mitbewerber Sie unterbieten können, indem sie Dinge auslassen, von denen Sie wissen, dass Ihr Kunde sie benötigt. Scrum kann sowohl Ihnen als auch Ihrem Kunden helfen. Denn je besser der Kunde bedient wird, desto besser ist das Ergebnis, desto besser ist es für Ihr Unternehmen und desto größer ist die Chance, weiterempfohlen zu werden.

Scrum bietet die folgenden Verbesserungen:

- ✓ **Transparenz:** Lassen Sie die Kunden entscheiden, wofür sie ihr Geld ausgeben wollen. Sie können Änderungen vornehmen, überprüfen und im Laufe des Projekts Anpassungen durchführen. Aber sie werden über jeden Schritt des Weges umfassend informiert, da sie an den regelmäßigen Sprint-Review-Sitzungen, in denen der Fortschritt kontrolliert wird, teilnehmen, Feedback geben und Fragen stellen.
- ✓ **Anpassungsfähigkeit:** Scrum ermöglicht es dem Team, Probleme, die eine Kurskorrektur erforderlich machen, frühzeitig zu erkennen. Selbst wenn Sie mehrere Häuser komplett bauen, ist es besser, Probleme zu lösen, die bei der Fertigstellung eines einzelnen Hauses identifiziert wurden, als diese erst dann zu erkennen, wenn zehn Häuser gleichzeitig denselben Fertigstellungspunkt erreichen. Wenn man dieses Beispiel weiterdenkt, ist die Anpassung an die Rückmeldung des Kunden über eine nicht tragende Wand, die an der falschen Stelle steht, viel einfacher zu beheben, als wenn bereits die Elektroinstallation und die Klempnerarbeiten durchgeführt und die Gipskartonplatten verputzt sind.
- ✓ **Tägliche Koordination des Scrum-Teams:** Das Vorhandensein eines festen Forums, in dem Probleme erörtert werden können, begrenzt Engpässe und verhindert, dass aus kleinen Themen große Probleme werden. Wenn Acme Pipe Co. es versäumt hat, die benötigten Materialien zu liefern, kümmert sich der Scrum Master sofort um eine Lösung und darum, dass sich das Problem nicht wiederholt – sofort bedeutet sofort, nicht erst am Ende der Woche, wenn das Problem im wöchentlichen Statusreporting-Meeting auftaucht.

In [Kapitel 1](#) erörtern wir drei Säulen der Verbesserung durch Agilität: Transparenz, Überprüfung und Anpassung. Obwohl alle drei Säulen überall in Scrum gute Dienste leisten, ist das Thema Transparenz ganz klar (Wortspiel beabsichtigt) im Bieter- und Bauprozess von zentraler Bedeutung.

Scrum-Rollen im Baugewerbe

Im traditionellen Projektmanagement trägt die Rolle des Projektmanagers eine große Verantwortung, die breit gefächert, allumfassend und oft zu viel ist, als dass eine Person sie allein tragen könnte. Der Projektleiter trägt so viele Hüte – koordiniert unter allen Beteiligten –, dass Effizienz und Qualität unter den Folgen leiden können.

Scrum stellt verschiedene, klar definierte Rollen zur Verfügung, die dem Projektmanager helfen, die Last zu tragen. Das Gleichgewicht wird wiederhergestellt, wenn die Mitglieder des Scrum-Teams neue Aufgaben übernehmen:

- ✓ Die Rolle des Projektmanagers verschwindet.
- ✓ Der Product Owner repräsentiert den finanzierenden Kunden und die Stakeholder. Der Product Owner kann die Person sein, die traditionell der Projektleiter war, oder der Projektarchitekt oder Ingenieur, je nachdem, wie diese Rollen in der jeweiligen Organisation implementiert sind. Unabhängig davon, welche Rolle der Product Owner zuvor gespielt hat, ist seine Rolle im Scrum die gleiche: das »Was« und das »Wann« des zu erbauenden Produkts.
- ✓ Die Rolle des Scrum Masters erleichtert die Kommunikation und Interaktion, beseitigt Hindernisse für das Entwicklungsteam, stellt sicher, dass jeder Beteiligte den Prozess versteht und dass das Team so eingerichtet ist, dass es erfolgreich sein kann. Diese Rolle klingt sehr nach Vorarbeiter oder Oberaufsicht, nicht wahr?
- ✓ Das Entwicklungsteam besteht aus Subunternehmern, Ingenieuren, Architekten und einzelnen Handwerkern.

Kundenbeteiligung

Die traditionelle Ansicht lautet, dass Veränderungen innerhalb eines Projekts schlecht sind. Bei Scrum hingegen werden Veränderungen als Zeichen des Fortschritts angesehen. Sie

suchen nach Veränderung als eindeutigen Beweis dafür, dass Sie ein besseres Produkt entwickeln, das den Wünschen Ihres Kunden besser entspricht.

Bei traditionellen Bauprojekten (wie in so vielen anderen Bereichen) ist die Tendenz, die Kunden so wenig wie möglich einzubeziehen, um Änderungen auf ein Minimum zu beschränken. Die Bauunternehmen sind immer noch im Umbruch. Je früher und häufiger der Kunde involviert ist, desto günstiger sind jedoch die Änderungen und desto wahrscheinlicher ist es, dass Sie etwas bauen, was der Kunde liebt.

Das andere Extrem ist, dass Sie keine schleiche Zunahme des Projektumfangs, auch *Scope Creep* genannt, wollen. Wenn die Kunden sehen, dass die Dinge Gestalt annehmen, wollen sie mehr Funktionen hinzufügen, was dem Projekt zusätzlichen Spielraum gibt. Dieser Umfang erhöht die Kosten, die möglicherweise nicht budgetiert wurden. Bei Scrum wollen Sie Kosten senken, nicht erhöhen.

Scrum verhindert diese Situation, indem es den Umfang auf Basis der Kosten – und damit der zur Verfügung stehenden Zeit – begrenzt und nicht nach dem, was eine schriftliche Anforderung bedeutet. Der Kunde kann jede neue Anforderung erhalten, die mit der Vision übereinstimmt, aber jede neue Anforderung wird durch den Verlust einer Anforderung mit niedrigerer Priorität ausgeglichen, es sei denn, diese Anforderung mit niedrigerer Priorität wird durch zusätzliches Budget finanziert.

Scrum erleichtert den Prozess der Kundenbeteiligung durch:

- ✓ Strukturierung des Kundenfeedbacks innerhalb von Sprint-Zyklen bei natürlichen Pausen und Meilensteinen im Projekt, wie zum Beispiel Inspektion von Elektro- oder Sanitärinstallationen, bevor die Trockenbauwände montiert werden, damit die Armaturen sicher dort installiert werden, wo der Kunde es wünscht. Diese Praxis erlaubt es Ihnen, alle notwendigen Anpassungen vorzunehmen, bevor es zu spät

ist. Der Kunde überprüft am Ende jedes Sprints die ausgeführten Arbeiten und gibt dann wertvolles Feedback.

- ✓ Zunahme der Eigenverantwortung für das Projekt durch Entscheidungen, die sich an den Kosten und dem Return on Investment innerhalb dieser Feedbackschleifen orientieren.
- ✓ Sichtbarmachung von Kundenänderungsanfragen während des gesamten Projekts, indem gezeigt wird, wie sich diese Anfragen auf den verbleibenden Zeitplan und die Kosten auswirken.

Letztendlich ermöglicht der Transparenz-, Überprüfungs- und Anpassungsprozess eine bessere Kommunikation. Wenn alle Beteiligten die Informationen haben, die sie brauchen, wird ein Umfeld für exzellente Entscheidungen geschaffen.

Das Subunternehmer-Dilemma

Seit dem Bau der Pyramiden in Ägypten kämpfen Projektmanager und Subunternehmer darum, koordiniert und auf Kurs zu bleiben. Aufgrund der Struktur, in der diese wertvollen Parteien operieren, treten immer noch viel zu häufig Probleme auf.

Die Herausforderungen sind vielfältig. Traditionell war der Projektablauf schwierig zu koordinieren, da es zahlreiche Subunternehmer gibt und nur wenige Auftragnehmer. Durch mangelnde Koordination, Kommunikation und damit Eigenverantwortung für das Ergebnis sowie die menschliche Natur konzentrieren sich die beteiligten Parteien auf ihre eigenen Vorteile und Ergebnisse und nicht auf das gesamte Projekt.

Jeder Subunternehmer ist sehr spezialisiert, daher ist ein hohes Maß an Koordination zwischen den Subunternehmern und den Verantwortlichen, Stakeholdern und potenziellen Nutzern erforderlich. Die Wasserfallmethode bietet diese Umgebung einfach nicht. Scrum glücklicherweise schon.

Hier sind einige Möglichkeiten, wie Scrum die Beziehung zwischen Bauunternehmer und Subunternehmer erleichtert:

- ✓ Durch regelmäßige, strukturierte Scrum-Events (Daily Scrum, Sprint-Reviews und Sprint-Retrospektiven) wird die Kommunikation verbessert und das Feedback aller Beteiligten gefördert.
- ✓ Sprints und Releases sind zeitgesteuert und finden häufig statt; jede Partei weiß genau, was bei jedem Schritt erwartet wird.
- ✓ Subunternehmer liefert Input dazu, wie viel Arbeit sie in einem Sprint vollenden können (basierend auf den empirische Daten darüber, was sie in den vorhergehenden Sprints fester Länge fertigstellten). Die Bauunternehmer können ihren Beitrag zu den Prioritäten leisten, aber die endgültige Entscheidung liegt bei den Personen, die die Arbeit ausführen.
- ✓ Subunternehmer werden dazu ermutigt, ihr Wissen und ihre Erfahrungen auszutauschen, um den Prozess effizienter zu gestalten (das heißt, selbstorganisierte Teams bringen ihr spezifisches Wissen ein, um die besten Entwürfe zu entwickeln).
- ✓ Bauunternehmer erhalten Wissen und Erfahrung von den Personen, die die Arbeit ausführen: den Subunternehmern.
- ✓ Subunternehmer können Ineffizienzen und Probleme frühzeitig (durch Sprint-Retrospektiven) ansprechen. Dieses Feedback ist für die Beseitigung von Hindernissen entscheidend.
- ✓ Liefer- und Beschaffungsfragen lassen sich leichter koordinieren, da alle Beteiligten durch aktive, persönliche Gespräche (Verfeinerung des Produkt-Backlogs, Daily Scrum und tägliches Feedback des Product Owners) gemeinsam auf den Weg kommen.
- ✓ Die Qualität wird durch verstärkte Koordination verbessert.
- ✓ Materialfluss und -bereitstellung lassen sich über die Verfeinerung des Produkt-Backlogs und die Sprint-Planung, einschließlich der Aufgabenplanung der Anforderungen, besser koordinieren.

Der Schlüssel ist, dass unterschiedliche Rollen und Jobs zu einem kommunizierenden, koordinierten Scrum-Team zusammengefasst werden. Mit Scrum haben Sie einen lebenden Organismus. Alle Beteiligten verlassen sich auf den jeweils anderen und verbessern sich gegenseitig.



Einzelne Subunternehmer-Teams nutzen Scrum, um sich und ihre Arbeit zu organisieren. Das gesamte Projektteam besteht jedoch aus mehreren Subunternehmern. Wie bringen Sie all diese Teams dazu, als ein Team zu arbeiten? Nun, es ist vielleicht nicht realistisch, alle Teams jeden Tag in den gleichen Raum zu bekommen, aber es wäre möglich, alle Vorarbeiter jeden Tag in den gleichen Raum zu bekommen, um die Prioritäten des Tages zu koordinieren, Hindernisse zu identifizieren und sie zu beseitigen, um sicherzustellen, dass die Arbeiten ungehindert voranschreiten.

Arbeitssicherheit

Bedenken hinsichtlich der Sicherheit der Arbeitnehmer sind immer wichtig, besonders jetzt, da die Zahl der Projekte in Gebieten mit hohem Verkehrsaufkommen und hoher Bevölkerungsdichte zunimmt. Eine klare Kommunikation von Risiken und Lösungen ist daher wichtiger denn je. Leben hängen davon ab.

Scrum erhöht die Sicherheit der Mitarbeiter durch folgende Maßnahmen:

- ✓ Die Sicherheitsbestimmungen sind relevant und gründlich. Die Aufnahme dieser Vorschriften in die Definition of Done bedeutet, dass dokumentierte, validierte Sicherheitsmaßnahmen vorhanden und bei Kontrollen einsehbar sind.
- ✓ Häufige Sprint-Retrospektiven ermöglichen es den Arbeitnehmern, Sicherheitsbedenken und -lösungen vorzubringen. Gute Scrum Master können diesen Prozess erleichtern, um sicherzustellen, dass die entsprechenden Sicherheitsthemen auftauchen.

- ✓ Erfahrene Mitarbeiter können ihr Know-how bei der Bereitstellung von Lösungen einbringen.
- ✓ Die Daily Scrums bieten ein Forum für dringende Fragen.

Es ist wahrscheinlich, dass diese Dinge bereits auf verschiedene Weise erledigt werden, nur nicht mit Scrum-Begriffen. Die Transparenz der Sicherheitsaspekte ist notwendig und bietet einen Mechanismus zur Überprüfung und Umsetzung von Anpassungsmaßnahmen.

Fly-over in Bangalore

Ein Artikel der Scrum Alliance mit dem Titel »A Real-Life Example of Agile, Incremental Delivery of an Infrastructure Project in Bangalore, India« (<https://stage-kentico.scrumalliance.org/community/articles/2014/july/real-life-example-of-agile-in-infrastructure-project>) liefert ein Beispiel für den Scrum-Prozess im Baugewerbe. Das Projekt war der Bau einer Überführungsstraße an einer stark befahrenen Kreuzung in Bangalore, Indien. Viele High-Tech-Unternehmen hatten an dieser Kreuzung große Niederlassungen.

Normalerweise würde ein Fly-over-Projekt dieser Größenordnung 18 Monate dauern. Dabei werden auf beiden Seiten der Hauptverkehrsstraße provisorische Straßen angelegt, während beide Überführungen gleichzeitig gebaut werden. Natürlich staut sich der Verkehr während des Baus, da temporäre Straßen benutzt werden und keine Überführungen zur Verfügung stehen, bis beide am Ende des Projekts geöffnet werden.

Dieses Projekt wurde jedoch mit Scrum in mehreren Schritten durchgeführt:

- ✓ Die Anforderung mit der höchsten Priorität wurde als eine Seite der Überführung für die erste Freigabe identifiziert. Deshalb wurde auf einer Seite der Hauptstraße eine

provisorische Straße gebaut und genutzt. Gleichzeitig begann der Bau der Überführung auf der ersten Straßenseite.

- ✓ Nach der Fertigstellung wurde die erste Überführung für den Verkehr in beide Richtungen freigegeben und mit dem Bau der gegenüberliegenden Überführung begonnen. Obwohl der Verkehr immer noch behindert war, wurden Staus reduziert, da mindestens eine Überführung funktionsfähig war. Mit der Fertigstellung der ersten Überführung wurde auch ein auslieferbares Produktinkrement geschaffen. Das Swarming fand nur auf dieser Seite der Überführung statt.
- ✓ Die gleiche provisorische Straße wurde beim Bau der zweiten Überführung benutzt. Zeit und Geld wurden gespart, da keine neue Straße benötigt wurde. Vielleicht hätte eine zweite provisorische Straße im Vorfeld geplant werden können, aber die erste provisorische Straße hat sich in der zweiten Phase als nützlich erwiesen. Überprüfung und Anpassung vermeiden Verschwendung.
- ✓ Als die zweite Überführung fertiggestellt war, waren beide Überführungen offen; der Verkehr wurde zur Hauptverkehrsstraße zurückgeführt und die provisorische Straße gesperrt.

Der Prozess scheint einfach zu sein, nicht wahr? Hier sind einige der Ergebnisse:

- ✓ Durch die schrittweise Ausführung von jeweils nur einer Überführung konnte die Gesamtbauzeit von 18 Monaten auf 9 Monate reduziert werden (eine Verkürzung der Markteinführungszeit um 50 Prozent).
- ✓ Die Verkehrsbehinderungen wurden reduziert, da zuerst die eine und dann die andere Überführung geöffnet wurde.
- ✓ Es musste nur eine provisorische Straße gebaut werden.

- ✓ Nachdem eine Überführung in Betrieb war, wurde das Ausfallrisiko reduziert. Wäre die Finanzierung zur Mitte des Projekts gekürzt worden, da eine Seite voll funktionsfähig war (das heißt, es wurden ein auslieferfähiges Inkrement erstellt), wäre diese Kürzung keine Katastrophe gewesen. Die risikoreichsten Themen wurden im Vorfeld behandelt.
- ✓ Der Gesamtwirkungsgrad wurde verbessert, da zu jedem Zeitpunkt weniger bewegliche Teile vorhanden waren.

Scrum im Wohnungsbau

Scrum im Hausbau ist real, es wird üblich und seine Vorteile sind genauso stark wie in anderen Branchen. Ein Haus kann mit dem Scrum-Framework definiert, geplant, geschätzt und erbaut werden.

In der ersten Sitzung mit dem Bauunternehmer geht es darum, die Art des Hauses zu beschreiben, das Sie wünschen – Eigenschaften wie die Anzahl Schlafzimmer, der Typ des Hauses und die Anzahl der Garagen, die Qualität der Ausstattung und des Bodenbelags, die Landschaftsgestaltung, den ungefähren Betrag, den Sie aufwenden möchten und Ihren Zeitrahmen. Dieses Treffen, das etwa eine Stunde dauert, vermittelt dem Bauunternehmer ein umfassendes Verständnis dessen, was Sie wünschen (die Vision und die Roadmap).

Mehrere Wochen nach dem ersten Planungsgespräch treffen sich die Käufer mit einem Architekten, um genauere Angaben zu den Plänen zu erhalten, wie zum Beispiel gewünschte Raumaufteilungen, Fußböden, Keller, Grundstücke, Bodenbeläge und Schränke. Gestaltungselemente wie Farben, Geräte und Armaturen sind noch nicht ausgewählt. Es wird nur das entschieden, was nötig ist, um die Kosten des Hauses abzuschätzen und einen Plan und eine Kostenabschätzung zu erstellen. Es handelt sich also um risikoreichere Positionen, deren

spätere Änderung schwierig und teuer wird (das Produkt-Backlog).

Von diesem Zeitpunkt an wird erwartet, dass sich die weniger riskanten Dinge später ändern werden, sodass diese Entscheidungen bis zum letzten verantwortbaren Moment aufgeschoben werden. Es werden aber auch mögliche Risiken diskutiert, wie beispielsweise Überraschungen, die sich beim ersten Spatenstich und dem Erstellen des Fundaments ergeben könnten.

Mit dem Bauplan kommt ein Budget, das die Details so weit untergliedert, dass der Bauunternehmer den Gewinn berücksichtigt und der Käufer jedes Merkmal des Hauses sieht (User Stories).

Das Budget für jedes Merkmal, wie beispielsweise Schränke, basiert auf der Beschreibung, die der Käufer dem Bauunternehmer zu Beginn des Projekts gegeben hat. Innerhalb dieses Budgets kann der Käufer genau das auswählen, was er will. An dieser Stelle können Kompromisse gemacht werden oder die Farbabstimmung kann stattfinden, wenn der Käufer das eigentliche Haus betrachtet und nicht die Bilder in einer Broschüre. Obgleich sich eine bestimmte Holzqualität während der Planungsphase gut angehört hat, kann der Kunde seine Meinung ändern und eine weniger teure Option wählen und die Einsparungen verwenden, um noch etwas hinzuzufügen oder aufgrund der Einsparungen weniger finanzieren.

Fertigung

Interessanterweise verschwimmen heutzutage die Grenzen zwischen Software und Hardware. Materielle Produkte enthalten heutzutage eine überraschend große Menge an Software. Das Auto, das Sie fahren, der Kühlschrank, in dem Sie Ihren Eiskaffee kühlen, und der E-Reader, auf dem Sie dieses Buch lesen, sind voller Code.



Automobile sind heute mit der Internet of Things (IoT)-Technologie verbunden und werden zunehmend softwaregesteuert. Ein Tesla-Fahrzeug verbindet sich mit dem WLAN Ihres Hauses und lädt regelmäßig tolle neue Funktionen herunter. Nachdem ein Fehler in seinem Ladegerät gefunden wurde, hat Tesla einen Fix für seine 29.222 Fahrzeughalter per Software-Update durchgeführt. Wir stehen jetzt an der Schwelle zu selbstfahrenden Autos.

An Fertigungsprozesse werden neue Anforderungen gestellt, wobei es nicht nur darum geht, ein Widget mit einem anderen zu verbinden. Komplexe Systeme, von denen viele softwarebasiert sind, vernetzen und integrieren sich. Tatsächlich ist Software entscheidend für den Erfolg vieler hergestellter Produkte.

Diese Trends führen zu größeren, komplexeren Entwicklungsprojekten und Teams mit unterschiedlichen Talenten. Sie führen auch zu einer erhöhten Komplexität bei der Einbindung von Produktfamilien (nicht nur einzelner Produkte), zu einem erhöhten Fehlerrisiko bei steigender Komplexität und zu einer neuen Reihe von Compliance- und geforderten Standards.

Lean und Mean mit Toyota

In den 1940er Jahren entwickelte eine relativ unbekannte Autofirma namens Toyota einen Plan zur Herstellung von Autos bei gleichzeitiger Kostenkontrolle. Statt des üblichen Fließbandverfahrens, das hohe Gemeinkosten verursachte, wurde ein Just-in-Time-Prozess geschaffen.

Statt alles auf einmal zu produzieren, baute das Unternehmen nur das, was es im Moment für ein Projekt benötigte. So wie ein Supermarkt nur ausverkaufte Artikel ersetzt, baute oder beschaffte Toyota nur die Teile, die es brauchte. Da es keinen großen Lagerbestand gab, konnten die Kosten gesenkt werden.

Daher liegt eine der Wurzeln von Scrum im Bereich der Herstellung.

Überleben dank schneller Markteinführung

Ein Schlüssel, um den Wettlauf um Marktanteile zu gewinnen, ist die schnellste Markteinführung, was kein neues Konzept ist. Ein ebenso wichtiges Konzept ist es, mit der Innovation Schritt zu halten und sogar führend zu sein – konkret: Innovation, die sich an den Bedürfnissen und Wünschen der Kunden orientiert.

Sie können ein Produkt nur dann am schnellsten auf dem Markt einführen, wenn Sie im Vorfeld Kundenfeedback erhalten und gleichzeitig die Funktionen mit der höchsten Priorität und den höchsten Risiken angehen. Höhere Qualität wird durch frühzeitige Tests erreicht. Scrum ermöglicht eine frühe, qualitativ hochwertige Markteinführung. Klingt nach einem guten Platz für Scrum. Aus diesem Grund setzen Tesla und andere Hersteller jetzt Scrum Master und agile Trainer ein.

Neue Technologien sind verblüffend: Robotik, künstliche Intelligenz, 3D-Druck und Nanotechnologie, um nur einige zu nennen. Jede dieser Technologien führt zu neuen Komplexitäten in der Produktion. Scrum- und auf agilen Konzepten basierende Frameworks sind ideal für die Lösung komplexer Probleme.

Shareholder-Value

Die Bürokratien, die traditionelle Elemente der Fertigungsindustrie sind, legen Wert auf Effizienz, Kostensenkung und maximalen Shareholder-Value im Gegensatz zum Mehrwert für die Kunden. Diese Betonung ist ihre Achillesferse. Unternehmen, die sich in der Zukunft hervorheben werden, legen Wert auf den Kunden.

Scrum und der organische Feedbackzyklus legen Wert auf regelmäßiges Feedback über den Nutzen und die Vorteile, die der Kunde hat. Nach jedem Sprint haben Sie ein funktionierendes, auslieferfähiges Produkt, das Sie den Stakeholdern und Kunden

zeigen können. Auch wenn Sie während des Sprints kein Feedback erhalten, können Sie sofort Anpassungen vornehmen, sobald Sie das Feedback bekommen.

Wie im Agilen Manifest festgehalten, ist der primäre Maßstab für den Erfolg ein funktionierendes Produkt in den Händen eines Kunden – je früher, desto besser, und je häufiger, desto besser. Mit Scrum erreichen Sie beides.

Strategisches Kapazitätsmanagement

Bei der Produktion materieller Güter kann es sein, dass Sie am Ende eines einwöchigen Sprints kein endgültiges Produkt haben, dass Sie jemand in die Hand geben können. Das ist in Ordnung. Denken Sie nur daran, dass der Fortschritt vorzeigbar ist und dass der Feedbackzyklus so kurz wie möglich sein muss.

Die Idee ist, regelmäßiges und häufiges Feedback von den Nutzern zu erhalten. Die Besonderheiten sind von Produkt zu Produkt unterschiedlich. Arbeiten Sie daran, den Feedbackzyklus so kurz wie möglich zu halten und ihn, falls möglich, zu verkürzen.

Intels Übergang zu Scrum

Intel weist eine lange Geschichte beim Einsatz des Wasserfall-Projektmanagements auf, was nicht verwunderlich ist, da Intel ein Fertigungsunternehmen ist. Das Unternehmen entschied sich, Scrum für die Entwicklung der Pre-Silizium-Infrastruktur und -Readiness zu testen. Die Idee war, dass Scrum auch in anderen Fertigungsprozessen implementiert werden kann, wenn es für dieses Projekt funktioniert.

Intel stellte einen Scrum-Coach ein, der helfen sollte, alte Gewohnheiten zu erkennen und zu überwinden, neue zu integrieren und das Scrum-Framework im Allgemeinen richtig zu verstehen. Trotz des Coachings traten jedoch Übergangsprobleme auf. Nicht jeder Senior Manager nahm an den ersten Scrum-Meetings teil, die Akzeptanz durch wichtige

Leute war gering und es gab Beispiele aus der Praxis, warum Scrum-Arbeiten nicht identifiziert wurden.

Schließlich fand Intel heraus, dass der beste Weg, seine Teams zur Implementierung von Scrum zu bringen und davon zu profitieren, in der Befolgung der Scrum-Richtlinien bestand und nicht in der Anpassung von Scrum an den eigenen Prozess.

Pat Elwer verfasste eine Fallstudie mit dem Titel »Agile Projektentwicklung bei Intel: Eine Scrum-Odyssee«

(www.collab.net/sites/all/themes/collabnet/_media/pdf/cs/CollabNet_casestudy_Intel.pdf).

Zusammenfassend stellte die Fallstudie fest, dass die Teams herausfanden, dass Scrum dem Projekt auf vier verschiedene Arten helfen konnte:

- ✓ Die Zykluszeit reduzierte sich um 66 Prozent.
- ✓ Die Leistungen blieben im Zeitplan, ohne dass das Team seine Verpflichtungen versäumte.
- ✓ Verbesserte Mitarbeitermoral. Ironischerweise wurde das Team mit der niedrigsten Moral zum leistungsstärksten Team.
- ✓ Erhöhte Transparenz, die dazu führte, Hindernisse und unproduktive Gewohnheiten zu erkennen.

Nachdem die Teams und das Management begannen, an den Prozess zu glauben, zeigte sich schnell, wie mächtig Scrum war.

Hardware-Entwicklung

Es gibt viele Gemeinsamkeiten zwischen der Implementierung von Scrum in Software- und in Hardwareprojekten. Sie hören vielleicht, dass die Unterschiede zwischen den beiden Produktarten so groß sind, dass Scrum nicht mit beiden

funktionieren kann. Das stimmt nicht. Wann immer Sie die zu erledigende Arbeit kapseln und gegenüber anderen Arbeiten priorisieren können, können Sie Scrum zu Ihrem immensen Nutzen einsetzen.

Ein Schlüsselement von Hardware-Projekten, die mit Scrum durchgeführt werden, ist es, sich frühzeitig und oft auf Feedback zu konzentrieren. Vielleicht haben Sie am Anfang nicht viel, mit dem Sie arbeiten können, dennoch sollten Sie auch hier während jedes Sprints einsatzfähige Inkremente erstellen. So erreichen Sie mehr Ergebnisse, die Sie den Kunden und Stakeholdern zeigen können – wobei Sie deren Feedback berücksichtigen können.

Früherkennung risikoreicher Anforderungen

Wenn Sie Fehler und Probleme frühzeitig erkennen, sparen Sie Zeit, Geld und Ärger. Im Wasserfall wird erst am Schluss getestet. Aber warum sollte man nur am Ende testen, wenn man permanent testen und Fehler frühzeitig finden (und beheben) kann?

Scrum erzwingt die rechtzeitige Integration von Firmware und Software. Scrum bricht funktionale Silos auf und bringt Ingenieure dazu, gemeinsam Probleme zu lösen, anstatt später zusammenzukommen, um Probleme zu beheben.

Im Daily Scrum (das nicht länger als 15 Minuten dauert) können Sie Ihre Arbeitsergebnisse mit den Anforderungen koordinieren und gleichzeitig besprechen, welche Hindernisse noch im Weg stehen.



Open-Source-Hardware ist ein Geschenk an Scrum und an die Hardwareentwicklung als Ganzes. Da immer mehr Hardware-Projekte als Open-Source veröffentlicht werden, können Unternehmen schnell und kreativ bestehende Designs, Frameworks und Architekturen implementieren, mit denen sie ihre Produkte schneller auf den Markt bringen können.

Hardware-Entwicklung an Beispielen

In den folgenden Abschnitten stellen wir vier Beispiele für den erfolgreichen Einsatz des Scrum-Frameworks bei der Hardware-Entwicklung vor. Jedes Beispiel ist einzigartig, aber die Scrum-Prinzipien und Praktiken bleiben gleich. In den meisten Fällen können Sie Scrum verwenden, um schneller und mit höherer Qualität und Einfachheit zu entwickeln.

Johns Hopkins CubeSat

Das Johns Hopkins University Applied Physics Lab verwendete Scrum in seinem Projekt *Multi-Mission Bus Demonstrator (MMBD)* für den Bau von zwei CubeSat-Satelliten. Das Programm selbst wurde von der NASA gesponsert.

Drei Schlüsselemente wurden als entscheidend für den Erfolg des Projekts identifiziert:

- ✓ **Scrum-Teams:** Diese Teams bestanden aus der Leitungsebene der Subsysteme. Alle Teammitglieder arbeiteten am gleichen Ort und erhielten direkten Zugriff auf den NASA-Vertreter (den Product Owner, der vor Ort war), um so eine schnelle Entscheidungsfindung zu gewährleisten.
- ✓ **Betonung eines funktionierenden Systems:** Nur eine Entwurfsprüfung wurde gefördert und es wurden CAD-Modelle (Computer Aided Design) zur Simulation und Manipulation von Entwürfen verwendet. Während des gesamten Projekts wurden informelle Peer-Reviews durchgeführt und die Dokumentation war minimal.
- ✓ **Kultur der Innovation:** Veränderungen wurden begrüßt und aufgenommen. Es wurden Daily Scrums durchgeführt, um die besten Wege zu finden, den höchsten Anforderungen gerecht zu werden. Auf eine langfristige Planung wurde verzichtet, um besser auf die Realität zu reagieren.



Wie in der Fertigung sind auch bei Hardwareprojekten Veränderungen unumgänglich. Akzeptieren Sie diese Tatsache und machen Sie daraus einen Vorteil. Verleugnung führt zu nichts. Glücklicherweise ist eine Änderung der Hardware-Projektierung und -Entwicklung praktikabel und praktisch.

Wikispeed, das modulare Auto

Wikispeed ist ein von Ehrenamtlichen getragenes Autobau-Projekt, das Prototypen umweltschonender Autos entwirft. Durch die Bildung selbstorganisierender Teams und die Verwendung des Scrum-Frameworks baute das Unternehmen einen funktionsfähigen Prototyp mit einem Verbrauch von 1,35 Litern pro 100 Kilometer. Die Teams gewinnen Rennen, weil ihr Auto komplett modular aufgebaut ist. Sie können Komponenten austauschen, je nachdem, auf welcher Strecke und bei welchen Wetter das Auto eingesetzt wird. Die Fahrzeuge ihrer Konkurrenz weisen diese Fähigkeiten nicht auf.

Wikispeed hat aus seiner Scrum-Erfahrung vier wichtige Erkenntnisse gewonnen:

- ✓ **Kunden-Feedback:** Die Teams gingen zu den Nutzern, um regelmäßiges Feedback zu erhalten und testeten jede Woche, was sie konnten.
- ✓ **Sprints und Reviews:** Regelmäßige Sprintzyklen wurden durchgeführt, wobei am Ende jedes Zyklus angegeben wurde, was richtig und was falsch gelaufen war.
- ✓ **Transparenz:** Alle Teammitglieder kannten die Ziele und jeder konnte einen Vorschlag machen.
- ✓ **Peer-to-Peer-Kommunikation:** Niemand war der Chef, alle waren gleichberechtigt und trugen jeden Tag dazu bei, das Sprintziel zu erreichen.

Telefonica Digital

Telefonica Digital begann als Hardwarehersteller, der Wasserfall-Projektmanagement einsetzte. Als sich die Technologie änderte, begann das Unternehmen, mehr Software zu entwickeln. Als sich der Fokus von der Hardware auf die Software verlagerte, passte sich das Unternehmen vollständig an Agilität und Scrum an.

Als sich die Technologie und die Ausrichtung des Unternehmens wieder änderte, kehrte die Firma zurück zur Produktion von Hardware. Aufgrund der jahrelangen Erfahrung mit Scrum wollte das Unternehmen nicht auf das traditionelle Projektmanagement zurückgreifen und nutzte das Scrum-Framework für seine Hardwareprojekte.

Die erste Aufgabe war die Verlagerung des Standortes, damit die Mitarbeiter an einem Ort arbeiten konnten. Das Unternehmen wusste, wie wichtig es ist, dass das Team in räumlicher Nähe arbeitet. Telefonica entwickelte außerdem ein System, mit dem man die Geräte und Materialien einfach an dem Tag beschaffen konnte, an dem sie gebraucht wurden.

Das Unternehmen setzte Scrum-Teams ein und förderte die Cross-Funktionalität. Es wurde früh und oft getestet, sogar mit 3-D-Druck- und Prototyping-Technologien, um das Testen zu beschleunigen. Es wurde viel Open-Source-Hardware eingesetzt, die man zum Testen verwenden konnte. Manchmal beschleunigte dieser Prozess den Fortschritt des Hardware-Teams so, dass es auf die Software-Teams warten musste.

Saab-Mehrzweckkampfflugzeug

Saab Defense verwendet Scrum zur Entwicklung des Mehrzweckkampfflugzeug JAS 39E Saab Gripen. Der Bau dieses hochkomplexen Jets erfordert mehr als 1.000 Ingenieure und mehr als 100 Teams. Scrum bietet Prioritäten auf Teamebene und die nötige Transparenz, um regelmäßig und schnell sich ändernde Variablen aufzudecken, zu überprüfen und anzupassen. Saab skalierte Scrum über mehrere Teams und Abteilungen hinweg. In [Kapitel 13](#) finden Sie weitere Informationen zur Skalierung von Scrum.

Warum der Bau eines komplexen Kampfflugzeugs funktioniert hat:

- ✓ **Größere Autonomie:** Teams erhalten Freiheitsgrade innerhalb eines klar definierten Rahmens. Die Konzentration auf autonome Teams fördert die Entscheidungsfindung auf der untersten Ebene und reduziert die Bürokratie (siehe [Kapitel 13](#)).
- ✓ **Modulare Architektur:** Saab nutzte die modulare Architektur, um monolithische Abhängigkeiten zu reduzieren und sich auf Flexibilität für zukünftige Änderungen zu konzentrieren.
- ✓ **Kontinuierliche Verbesserung:** Retrospektiven nach jedem Sprint und kurze Feedbackschleifen von außen ermöglichen Produkt- und Prozessverbesserungen.

Kapitel 10

Dienstleistungssektor

IN DIESEM KAPITEL

Scrum im Gesundheitswesen nutzen
Schulen stärken
Sicherheit im In- und Ausland

Es gibt mehr als 9.000 AbrechnungsCodes für die verschiedenen Behandlungen und Pflegeeinheiten. Aber es gibt keinen einzigen AbrechnungsCode für die Einhaltung oder Verbesserung der Therapietreue oder dafür, die Patienten zu unterstützen, gesund zu bleiben.

Clayton M. Christensen: The Innovator's Prescription. A Disruptive Solution for Health Care

Scrum bietet in der Dienstleistungsbranche bietet ein enormes Potenzial. Wir sind auf Gesundheitsversorgung, Bildung und öffentliche Einrichtungen angewiesen, um unsere Zivilgesellschaft zu erhalten und zu verbessern. Dennoch gibt es Raum für schlanke Systeme zur Kosteneinsparung und Qualitätsverbesserung. Und in vielen Fällen wird bereits Scrum eingesetzt.

Jede Branche hat ihre eigenen Herausforderungen. Es gibt einzigartige Sachverhalte, die individuell behandelt werden müssen. Viele der veralteten Systeme für Entwicklung und Wartung wurden in einfacheren Zeiten entwickelt. Doch mit der zunehmenden Komplexität der Welt wächst auch der Bedarf an innovativen und flexiblen Projekt-Frameworks. Scrum ist ideal für den Umgang mit neuen Anforderungen. Wir zeigen Ihnen, wie

und mit welchen Ergebnissen. Auch wenn sich ein Teil der Beispiele auf Gegebenheiten in den USA beziehen lassen sich die Grundprinzipien von Scrum, die dort deutlich werden, auf andere Länder übertragen.

Gesundheitswesen und Scrum

In den letzten Jahren stand das Gesundheitswesen in den USA im Mittelpunkt des öffentlichen Interesses und der Nachrichten. Erschwingliche und zugängliche Gesundheitsversorgung wird oft als Grundprinzip einer zivilisierten Gesellschaft angesehen. Doch steigende Kosten, der Druck, die Entwicklungszeit ohne Qualitätseinbußen zu verkürzen, verschwendete Ausgaben und vermeidbare Todesfälle haben zu massiven Veränderungen in der Art und Weise geführt, wie die Amerikaner für medizinische Hilfe bezahlen und wie sie diese erhalten.

Für das Jahr 1970 wurden die Gesundheitsausgaben in den Vereinigten Staaten auf 75 Milliarden US-Dollar geschätzt. Im Jahr 2016 betrugen die Ausgaben 3,3 Billionen US-Dollar oder 10.348 US-Dollar pro Person und machten 17,9% des Bruttoinlandsprodukts aus.

Hinzu kommt eine Gesundheitskultur, in der die Kostenerstattung zunehmend an die Kundenzufriedenheit gekoppelt ist. Die Gesundheitstechnologie hat einen erweiterten und wichtigen Einfluss auf die klinischen Ergebnisse und die Patientenzufriedenheit. Es gibt neue Paradigmen und es bedarf neuer Methoden, um ihre Bedürfnisse zu befriedigen. Es ist nicht verwunderlich, dass Scrum häufiger als je zuvor zur Behandlung von Problemen im Gesundheitswesen eingesetzt wird.

Einige der höchstprioritären Herausforderungen für das Gesundheitswesen in den USA und vermutlich auch für viele andere Länder sind:

- ✓ Verfahrensfehler bei der Gesundheitsversorgung sind jetzt die drittgrößte Todesursache in Amerika, direkt hinter Herzkrankheiten und Krebs. Ein Forbes-Artikel aus dem Jahr

2016 besagt, dass täglich mehr als 500 Menschen an »Fehlern, Unfällen und Infektionen« in Krankenhäusern sterben.

- ✓ Geld wird für unnötige medizinische Tests und Behandlungen verschwendet. Das *Institute of Medicine Health* behauptet, dass ein sattes Drittel der Milliarden, die jedes Jahr für die Gesundheitsversorgung ausgegeben werden, verschwendet wird. *The American Journal of Obstetrics and Gynecology* stellt fest, dass allein für geplante Kaiserschnitte jährlich 1 Milliarde Dollar ausgegeben werden.
- ✓ Neue und umfangreichere Vorschriften erfordern eine schnelle und gründliche Anpassung.
- ✓ Die Forderung, auf elektronische Gesundheitsakten umzustellen, ist riesig, doch es fehlt an Verfahren, um diese Umstellung zu erreichen.
- ✓ Forschung und Entwicklung im Bereich neuer Behandlungsmethoden und Medikamente müssen weiterhin finanziert und innovativ sein.
- ✓ Es gibt Bedenken, dass neue medizinische Geräte und Systeme nicht gründlich auf Sicherheitsrisiken überprüft wurden. Viele von ihnen sind mit traditionellen Methoden entwickelt worden, bei denen das Design an erster Stelle steht und die Prüfung erst ganz zum Schluss stattfindet.

In den folgenden Abschnitten zeigen wir Ihnen, wie Scrum bei den oben beschriebenen Problemen helfen kann. Scrum wurde mit großem Erfolg im Gesundheitswesen eingesetzt. Es beginnt mit dem administrativen Buy-In, gefolgt von der Implementierung von Scrum-Teams und deren Rollen. Dann beginnt der Prozess, der wie jedes andere Scrum-Projekt der Roadmap folgt. Folgen Sie den Schritten, die wir in diesem Buch skizzieren, und beobachten Sie die positiven Veränderungen.

Im Allgemeinen bringt Scrum die folgenden Vorteile mit sich:

- ✓ **Schnelles und regelmäßiges Feedback:** Feedback ist entscheidend, um festzustellen, was akzeptabel ist und was nicht. Noch wichtiger ist das Feedback, wenn der Entwicklungszyklus die Einhaltung regulatorischer Rahmenbedingungen beinhaltet. Scrum beschleunigt den Zeitpunkt, an dem die interne Revision eine bessere Einhaltung der Vorschriften gewährleisten kann.
- ✓ **Schnellere Implementierung:** Im Gesundheitswesen geht es darum, Leben zu retten, aber Wettbewerbsvorteile können ein zusätzlicher Vorteil sein.
- ✓ **Schnellere Monetarisierung:** Eine qualitativ hochwertige Gesundheitsfürsorge profitabel zu machen, ermöglicht schnellere Veränderungen, die mit größerer Wahrscheinlichkeit anhalten.
- ✓ **Bessere Mitarbeiterbindung:** Fokussierte Mitarbeiter weisen nachweisbar eine höhere Arbeitszufriedenheit auf und bleiben länger beim Unternehmen.
- ✓ **Weniger Produktfehler:** Höhere Sichtbarkeit führt zu höherer Qualität und höherer Kundenzufriedenheit.

Scrum wird im Gesundheitswesen benötigt, um Veränderungen zu fördern, die die klinische Entscheidungsfindung in der hochwirksamen Patientenversorgung und Betriebswirtschaft unterstützen. Das Gesundheitswesen kämpft darum, sich rasch zu entwickeln und gleichzeitig die sich ständig ändernden Anforderungen der Regulierungsbehörden zu erfüllen.

Scrum hat dazu beigetragen, die Entwicklungszeit für neue administrative und klinische Systeme zu verkürzen und die Qualität und Effizienz zu steigern. Das gegenwärtige Umfeld im Gesundheitswesen befindet sich im Spannungsfeld zwischen privater und öffentlich finanzierter Gesundheitsversorgung. Vorschriften und Gesetze werden geändert und verfeinert. Daher ist ein hohes Maß an Flexibilität und Transparenz notwendig, um in diesen turbulenten Gewässern zu überleben.

Schnelle Markteinführung

Viele der Krankheiten, unter denen unsere Eltern und Großeltern als Kinder litten, sind dank der Fortschritte bei der Hygiene und den Behandlungen nicht mehr alltäglich, aber neue Krankheiten werden entdeckt und erforscht. Intensive Forschung und Fortschritt sorgen weiterhin für Heilung und verhindern chronische und unheilbare Krankheiten. Aber Gesundheitsexperten wollen mehr und schneller Leben retten.

Die Erwartungen der Kunden an das Gesundheitswesen und die Wunder der Wissenschaft steigen. Die Kostenträger im Gesundheitswesen recherchieren gründlich, um auch die besten Medikamente und Behandlungen zu finden und sie zu kaufen.

Pharmaunternehmen stehen unter ständigem Druck, neue Medikamente zu entwickeln, um im harten Wettbewerb mithalten zu können. Doch sie müssen dies in einem kostensparenden und ökonomischen Umfeld tun. Um wettbewerbsfähig und erfolgreich zu sein, müssen Unternehmen in dieser Branche (wie in jeder anderen in diesem Buch behandelten Branche) die schnellsten, wendigsten und kostengünstigsten sein.

Dennoch hat der Output der Pharmaindustrie insgesamt stagniert. Die meisten Pharmaunternehmen reagieren auf neue Situationen und technologische Fortschritte mit den gleichen Projektmanagement-Frameworks, die sie bereits in den 1940er Jahren verwendeten. Sie sind von neuen Geschäftsmethoden umgeben und gleichzeitig, häufig in traditionellen Managementmentalitäten stecken geblieben.

Scrum kann auf folgende Weise zu positive Veränderungen führen:

- ✓ Jedes Experiment beginnt mit einer Hypothese. Im Scrum ist diese Hypothese das Release-Ziel.
- ✓ Beim Test der Hypothese wird untersucht, ob die Prämissen wahr oder falsch sind. Jede Prämisse ist ein Sprintziel. Das Sprint-Backlog wird in Aufgaben und Experimente aufgeteilt,

die notwendig sind, um jede Prämisse vollständig zu erforschen. Die Definition of Done für den Sprint beschreibt die Art der Aufgaben und Aktivitäten, die durchgeführt und abgenommen werden müssen, damit eine Prämisse als vollständig getestet und erforscht betrachtet werden kann.

- ✓ Am Ende eines jeden Sprints präsentiert das Team seine Ergebnisse den Stakeholdern, die Fragen stellen und die Sprintschlussfolgerungen des Teams bestätigen. Zu den Interessengruppen sollten auch Vertreter der Regulierungsteams gehören. Wenn etwas übersehen wurde, weiß das Scrum-Team frühzeitig Bescheid und kann beim nächsten Sprint Folgeaktivitäten durchführen, um das Feedback, falls es eine hohe Priorität hat, zu berücksichtigen.
- ✓ Nachdem alle Prämissen validiert wurden, bleibt die Hypothese bestehen oder ist ungültig. So oder so kommt die Hypothese zu einer freizugebenden Schlussfolgerung.
- ✓ Alle Maßnahmen, die zur Überprüfung der Hypothese erforderlich sind, sind die Anforderungen im Produkt-Backlog und werden entsprechend priorisiert.
- ✓ Für neue Produkte und Verbesserungen an alten Produkten ermöglicht Scrum eine schnelle Iteration. Das Mantra »Scheitere schnell und du scheitert preiswert« (fail fast, fail cheap) bedeutet, dass Sie beim Ausprobieren von neuen Ideen und Prozessen so flink sein können, wie Sie wollen.
- ✓ Selbstorganisierende Scrum-Teams ermöglichen es Entwicklern, die besten Forschungsergebnisse und Lösungen zu finden.
- ✓ Die Einbeziehung von Kunden, Unternehmen und Behörden während des gesamten Prozesses bedeutet kontinuierliches Feedback und Orientierung. Aus diesem Grund wird die Verantwortung geteilt und das Engagement steigt.

Weniger Fehler, höhere Qualität

Obwohl die heutigen Generationen über die beste Medizintechnik und Behandlungen verfügen, die es je gab, ist immer noch Raum für Verbesserungen. Unglaublicherweise sind Fehler bei der Gesundheitsversorgung jetzt in Amerika die dritthäufigste Todesursache.

Die Fehler im Gesundheitswesen, die Schaden verursachen, reichen von komplizierten chirurgischen Komplikationen, die unerkannt bleiben, bis zu kleineren Fehlern in Bezug die Dosis oder die Art der Medikation.

Niemand kennt die genauen Kosten von Fehlern im Gesundheitswesen, weil das Kodierungssystem, das zur Aufzeichnung von Sterbedaten verwendet wird, nicht imstande ist, Kommunikationsausfälle, Diagnosefehler und schlechtes Urteilsvermögen zu erfassen – all das kann Leben kosten. Aber wir wissen, dass die Fehler im Gesundheitswesen sowohl menschliche als auch wirtschaftliche Opfer fordern. Die Zahl der Todesfälle in den USA wird auf mehr als 300.000 pro Jahr geschätzt; die Anzahl der Personen mit ernsthaften Schäden, deren Ursache medizinische Fehler sind, ist zehnmal so hoch. Die wirtschaftlichen Kosten medizinischer Fehler können auf 20,8 Milliarden Dollar geschätzt werden.

Ein Bericht des in den USA erscheinenden *Journal of Health Care Finance* schätzt, dass die Kosten dieser medizinischen Fehler, wenn man die verlorene Produktivität am Arbeitsplatz mit einbezieht, jährlich bis zu 1 Billion Dollar betragen können. In vielen Fällen können diese Fehler dazu führen, dass ein Gerät oder Medikament sehr kurzfristig für immer verboten wird, was zu einem Totalverlust für ein Unternehmen führt, das während des Entwicklungszyklus Milliarden von Dollar investiert hat.

Die Ursachen für diese Fehler können unterschiedlich sein, aber die im Scrum-Framework eingebaute Qualitätssicherung kann Probleme und deren Lösungen frühzeitig aufspüren. Scrum macht es einfach, Systemineffizienzen, Engpässe im Arbeitsablauf, Kommunikationsausfälle und mangelndes zeitnahes Feedback zu

erkennen und zu beheben. Im Folgenden finden Sie einige Möglichkeiten, wie Scrum helfen kann:

- ✓ Mit zunehmender Kommunikation werden Ineffizienzen sichtbar. Scrum deckt diese Ineffizienzen durch Daily Scrums und Sprint-Reviews auf; echte Lösungen können ausprobiert, getestet und angepasst werden.
- ✓ Obwohl approbierte Ärzte die einzigen Personen sind, die Diagnosen stellen dürfen, unterstützen selbstorganisierte Teams (bestehend aus dem gesamten Personal) die notwendige Behandlung.
- ✓ Das Feedback ist schnell und Anpassungen können schnell implementiert und erneut getestet werden.
- ✓ Mit steigender Qualität sinken unnötige Risiken und verschwendete Kosten.
- ✓ Da der Input aller Mitarbeiter und kerngruppen gesucht und genutzt wird, erhöhen sich das Engagement und die Moral. Buy-in oder Eigentum ist im Gesundheitswesen, in dem viele Menschen überlastet und von Burnout betroffen sind, enorm wertvoll.

Kostensenkung

Dieser Begriff allein reicht aus, um das Interesse jedes Mitarbeiters in der Verwaltung des Gesundheitswesens zu wecken. In gewisser Weise hilft jedes Problem und jede Lösung, die wir in diesem Kapitel erörtern, Kosten zu sparen. Vermeidbare Krankheiten und Todesfälle, erhöhte Effizienz in der Versorgung und im administrativen Ablauf, Erleichterung bei der Einhaltung von Vorschriften und so ziemlich alles andere, was Sie nennen können, kann verbessert werden, um Kosten zu sparen.



Kosteneinsparungen sind ein Markenzeichen von Scrum. Scrum hilft Ihnen nicht nur bei der Beschleunigung der Markteinführung, die einen großen Teil der Einsparungen ausmacht, sondern rationalisiert auch Prozesse, beseitigt

Ineffizienzen und ermöglicht Ihnen, bessere Geschäftsentscheidungen zu treffen. Das *Center for American Progress* konstatierte im Jahre 2012: »Die Verwaltungskosten im US-Gesundheitssystem belaufen sich auf schätzungsweise 361 Milliarden Dollar jährlich – 14 Prozent aller Gesundheitsausgaben in unserem Land. Mindestens die Hälfte dieser Ausgaben wird als Vergeudung eingeschätzt. «

Beachten Sie Einsparungen im Zusammenhang mit unnötigen Gesundheitskosten. Eine Studie des *Institute of Medicine* aus dem Jahr 2012 schätzt, dass in den Vereinigten Staaten jährlich 750 Milliarden Dollar für unnötige Gesundheitsversorgung verschwendet werden. Ursachen sind unter anderem ineffiziente und unnötige Dienstleistungen, überhöhte Preise, überhöhte Verwaltungskosten und Betrug.

Einige der Möglichkeiten, wie Scrum helfen kann, Kosten zu senken, sind:

✓ **Sprint-Retrospektiven:** Die Sprint-Retrospektiven im Scrum stellen routinemäßig folgende Fragen:

- Was ist in unserem Reklamationsprozess überflüssig?
- Wie können wir wiederholte Patientenbesuche minimieren?
- Welche zusätzlichen Schritte können wir unternehmen, um das Patientenrisiko noch weiter zu reduzieren?
- Wo sind die Engpässe?
- Was wäre, wenn das Formular XYZ elektronisch wäre?
- Was hindert uns daran, unsere tägliche Patientenbehandlungsrate um 20 oder 50 Prozent zu erhöhen?

✓ **Tägliche Koordination der Prioritäten:** Bei Daily Scrums können beispielsweise die folgenden Fragen gestellt werden:

- Ist das für diesen Patienten heute notwendig?

- Warum kann dieser Patient nicht heute gesund und geheilt nach Hause gehen?
 - Wie können wir dieses Hindernis beseitigen?
- ✓ **Transparenz und Einfachheit:** Mit Scrum möchten Sie die Menge der nicht erledigten Arbeit maximieren, besonders wenn sie schädlich ist. Eine Verringerung der auf Wunsch der Mutter eingeleiteten Frühgeburten (in der 37. bis 39. Woche statt der 40.) würde beispielsweise bis zu 500.000 Tage auf der Intensivstation für Neugeborene einsparen.

Einhaltung der Vorschriften

Mit den Änderungen des Affordable Care Act im Jahr 2010 kam es in den USA zu einer Vielzahl neuer Regelungen. Diese Vorschriften dienen dem Schutz von Patienten und Leistungserbringern und jede Organisation im Gesundheitswesen muss Wege finden, diese einzuhalten. Die Implementierung von Lösungen zur Einhaltung neuer regulatorischer Vorschriften kann eine Herausforderung darstellen und sogar mühsam sein, und die Tatsache, dass weniger Ressourcen zur Verfügung stehen, macht den Prozess noch schwieriger.

Hier sind einige der Möglichkeiten, wie Scrum Anbietern von Gesundheitsdienstleitungen hilft, neue Vorschriften zu verstehen und einzuhalten:

- ✓ Ein Produkt-Backlog kann nach Geschäftswert und Risikominimierung erstellt und priorisiert werden. Dann haben die Scrum-Teams und die Organisation als Ganzes die Basis, auf der sie in komplexen Zeiten von Regulierungsänderungen Geschäfts- und Ressourcenentscheidungen treffen können.
- ✓ Früher waren die technischen und betriebswirtschaftlichen Aspekte von Lösungen im Gesundheitswesen vielleicht nicht immer aufeinander abgestimmt; schließlich waren deren Zielvorgaben oft unterschiedlich. Da bei Scrum die Businessaspekte von einem Product Owner abgedeckt wird und Entwicklungsteams gebildet werden, um Lösungen zu

entwickeln, sind die Ziele aufeinander abgestimmt und stimmen plötzlich überein. Scrum-Teamrollen sind Peer-Rollen. Die Teammitglieder arbeiten zusammen, um das richtige Umfeld für die Abstimmung von Zielen und Vorhaben zu schaffen, was ideal für die Umsetzung von Änderungen im Umgang mit Vorschriften ist.

- ✓ Selbstorganisierte Teams definieren die besten Lösungen. Wenn neue Vorschriften eingegrenzt und priorisiert werden, ist es einfacher, Lösungen zu finden und schnell umzusetzen.
- ✓ Die sich entwickelnden regulatorischen Anforderungen können in die Definition of Done des Scrum-Teams integriert werden. Auf diese Weise können Product Owner und Stakeholder sicher sein, dass sie im Falle von Kontrollen und Inspektionen eingehalten wurden.

Scrum ist kein Notausstieg für Vorschriften. Denn die Vorschriften haben das Wohl des Patienten im Blick. Scrum ist ein Instrument, um auf Veränderungen mit taktischer Flexibilität und strategischer Stabilität zu reagieren und so das Ziel – eine bessere Gesundheitsversorgung – zu erreichen.

Risiko- und Problemmanagement

Die Dokumentation ist für das Risiko- und Problemmanagement im Gesundheitswesen von entscheidender Bedeutung, insbesondere während der Durchführung von behördlichen Kontrollen oder bei gerichtlichen Schritten.

Im traditionellen Projektmanagement wird die Dokumentation von Risiken und die Verfolgung von Problemen in der Regel durch die Erstellung und Pflege eines Risiko- und Problemprotokolls erreicht, das auf den beim Projektstart festgelegten Anfangsanforderungen basiert.

Da sich die Anforderungen im Laufe der Implementierung ändern, sind sie schwer zu managen und bewirken oft schwerfällige Prozesse im Änderungsmanagement.

Bei Scrum können Sie das Risikomanagement in Ihren Prozess integrieren, indem Sie es zu einem Teil Ihrer Definition of Done machen. Die Anforderung ist erst dann fertiggestellt, wenn spezifisch genannte Risiken identifiziert, gemildert oder akzeptiert und dokumentiert sind.

Der Fokus von Scrum auf kontinuierliche Überprüfung liefert bessere Ergebnisse als herkömmliche Projektmanagementmethoden.

Herstellung und Sicherheit medizinischer Geräte

Wie die pharmazeutische Industrie agiert auch die medizintechnische Industrie in einem wettbewerbsintensiven und kostenbewussten Umfeld. Viele sind von traditionellen Projektmanagement-Mentalitäten geplagt. Die Vorplanung in Kombination mit späten Backend-Tests bedeutet, dass Fehler spät erkannt werden und die Kosten für deren Behebung steigen – vorausgesetzt, die Fehler werden erkannt, bevor das Produkt auf den Markt kommt (siehe [Kapitel 4](#)).

Mit Scrum können Sie:

- ✓ Produkttests in jede Iteration einbauen, Fehler frühzeitig erkennen, ausmerzen und bessere Designs umsetzen.
- ✓ Entwickler und Ingenieure an einem Ort arbeiten lassen (Kollokation), damit sie sofort erreichbar sind und alltägliches Feedback in Echtzeit erhalten.
- ✓ Anforderungen im Projektverlauf hinzufügen. Oft wissen die Kunden am Anfang nicht genau, was sie wollen. Nachdem die ersten Anforderungen mit den obersten Prioritäten entwickelt wurden, erhalten die Kunden dank Scrum ein umfassenderes Bild.
- ✓ Ständig überprüfen und anpassen. Erstellen Sie nur das Wichtigste. Testen Sie regelmäßig und häufig. Sammeln Sie

regelmäßig und häufig Feedback.

- ✓ Sich täglich mit dem Product Owner über die geleistete Arbeit abstimmen.
- ✓ Bei den regelmäßigen Sprint-Reviews Feedback von den Stakeholdern erhalten. Durch dieses Feedback sind die Stakeholder während des gesamten Entwicklungsprozesses mit einbezogen.
- ✓ Sich bei der Synchronisation von Software- und Hardwareentwicklung frühzeitig und oft synchronisieren.
- ✓ Teilen Sie die Definition of Done für beide Teams und schließen Sie die Integration und die Tests während jedes Sprints ein, anstatt am Ende.

GE IMAGING

Der Unternehmensbereich Bildgebung von General Electric stellte in seinem traditionellen Ansatz zur Entwicklung medizinischer Geräte vor der Implementierung von Scrum die folgenden Probleme fest:

- ✓ **Vorhersagbarkeit:** Das Unternehmen führte 12- bis 24-monatige Projekte durch, meist mit erheblichen Verzögerungen.
- ✓ **Schleichende Zunahme des Funktionsumfangs (Scope Creep):** Business-Vertreter drängten in der Regel darauf, Funktionen hinzuzufügen, die über den ursprünglichen Rahmen hinausgingen.
- ✓ **Phasenweiser Ansatz:** Minimale Kundeninteraktion bis zur ersten Testrunde trug wesentlich zu Verzögerungen und steigenden Projektkosten bei.
- ✓ **Silo-Teams:** Zwischen funktionalen, technischen und geschäftlichen Einheiten der Organisation bestanden künstliche Barrieren.

Scrum löste diese Probleme auf folgende Weise:

- ✓ Die Product Owner waren integraler Bestandteil der Scrum-Teams, wodurch sich die Silos erheblich verringerten und die Prioritäten im gesamten Unternehmen aufeinander abgestimmt wurden. Das ganze Unternehmen begann, gemeinsam daran zu arbeiten, das richtige Produkt pünktlich an die Kunden zu bringen.
- ✓ Es wurden kurze Sprints durchgeführt, in denen die tatsächlichen Produktinkremente vervollständigt und demonstriert wurden. Die Rückmeldungen brachten das Projekt früher als üblich zum Abschluss.
- ✓ Die Führungsspitze war begeistert. Durch die Beobachtung von Erfolgen in einem Joint Venture, das Scrum einsetzte, sahen die Führungskräfte, was mit Scrum möglich war, und ihre Unterstützung beschleunigte die Akzeptanz.
- ✓ Aufgrund der hohen Regulierung der Branche baute GE regulatorische Schritte in die Definition of Done und in die Abnahmekriterien ein. Die Entwicklungsteams stellten sich bei der Schätzung und Ausführung auf diesen Aufwand ein.

Durch die Integration von Scrum in den Herstellungsprozess von Medizintechnik konnte GE die Kosten senken und die Markteinführungszeit verkürzen. Am wichtigsten war, dass das Unternehmen Fehler reduzierte, wodurch die Risiken in puncto Reputation und Regulierung verringert wurden. Die Chance, dass ein Produkt jemandem schaden würde, sank, da das Produkt so oft getestet und integriert worden war.

Bildung und Scrum

In diesem Buch weisen wir wiederholt darauf hin, dass Scrum in einer Vielzahl von Branchen eingesetzt werden kann – in der Tat in allen Branchen. Scrum ist ein Framework zur Entwicklung von Produkten und kein softwarespezifisches Werkzeug.

Bildung ist eine solche Branche. Kinder sind die Zukunft, denn auch sie werden wieder Kinder bekommen. Die Ausbildung junger Menschen und die Förderung ihrer Fähigkeit, selbst kreative, innovative Lösungen und Entscheidungen zu treffen, sollte die oberste Priorität jeder Gesellschaft sein.

Das staatliche Bildungswesen entstand in einem anderen historischen Kontext: Es ging darum, für die Grundbildung der Arbeiter zu sorgen. Die Anforderungen an die Ausbildung waren einfach, weil die Arbeitswelt einfach war. Heute jedoch ist die Komplexität der Arbeitswelt exponentiell gestiegen; dementsprechend können die heutigen Schülern und Studenten zwischen einer Vielzahl von Berufen und Rollen wählen.

Aufgrund dieses Wandels in der Arbeitswelt muss Bildung die Kinder auf einen Einstieg und eine effektive Teilnahme in der Arbeitswelt vorbereiten. Viele Lehrer sind immer noch darin geschult, Materialien auf die alte Art und Weise vorzubereiten und zu vermitteln, aber es gibt einen besseren Weg.

Scrum im Unterricht hilft Kindern, sich kreativ und flexibel an Veränderungen anzupassen, Projekte und Ideen zu priorisieren und neue Lösungen zu finden. Technik und Medien verändern sich so schnell, dass die Informationen, die Kinder erhalten, immer im Fluss sind. Mit Scrum lernen sie, wie man aus Veränderungen Vorteil zieht.

Herausforderungen in der Bildung

Das Bildungswesen steht heute vor anderen Herausforderungen als jemals zuvor. Der Lehrplan erweitert sich, leistungsschwache Schüler benötigen mehr Aufmerksamkeit und die Klassenzimmer werden größer. Trotz dieses Chaos kann man mit Scrum Fortschritte erzielen.

In vielerlei Hinsicht verwenden die Lehrer von Natur aus Scrum, aber sie verwenden Begriffe, die außerhalb ihres Berufs nicht sonderlich bekannt sind, wie Ziele, Scaffolding, Minilektionen, Lernmodalitäten und Reflexion.

Die Lehrer legen bereits für jede Lektion Ziele fest. Sie verwenden die Standards als Ziel und setzen Ziele für die Lektion, da manche Standards mehrere Schritte erfordern. Diese Schritte werden als Scaffolding (Unterstützung der Lernenden auf ihrem Weg der Aneignung einer nächsten Kenntnis oder Fähigkeit) bezeichnet. Die Lehrer unterteilen die Lektion in Iterationen, sodass die Schüler erfolgreich sein und jeden notwendigen Aspekt erfüllen können. Währenddessen nutzen sie formative und summative Beurteilungen, um das Lernen zu messen und zu modifizieren. Schüler und Lehrer reflektieren am Ende jeder Lektion wie in einer Retrospektive.

Als Nächstes werden die Lehrer ausgebildet, um kollaborative Lernmodelle zu nutzen und Projekte durchzuführen, die die Schüler motivieren und ihr Engagement aufrechterhalten.

Erweiterung des Curriculums

Die Anforderungen an den Lehrplan für Lehrer steigen seit Jahrzehnten. Mehr Informationen müssen abgedeckt werden und die Lehrer können bei dem Versuch Schritt zu halten, überfordert sein. Dieser Druck auf den Lehrplan kommt aus vier Quellen: Schulsystem, staatliche Behörden, Eltern und Schüler.

Viele Themen und Erfahrungen wurden früher von den Eltern zu Hause vermittelt. Die Schule war für die Grundlagen da. Heute decken die Schulsysteme Themen ab, von denen man sich früher nicht vorstellen konnte, dass sie im Unterricht behandelt werden. Einige wenige, die im letzten Jahrzehnt eingeführt wurden sind:

- ✓ Mobbingprävention
- ✓ Umgang mit sexueller Belästigung
- ✓ Body-Mass-Index-Bewertung (Monitoring im Hinblick auf Übergewicht)
- ✓ Finanzielle Allgemeinbildung
- ✓ Unternehmerische Kompetenzentwicklung
- ✓ Gesundheits- und Wohlfühlprogramme

An diesen zusätzlichen Inhalten ist nichts auszusetzen. In der Tat, dienen sie einem wichtigen Zweck. Aber die Lehrer sollen diese Themen und viele weitere neue zusätzlich zu den bisherigen Grundlagen des Lesens, Schreibens und Rechnens in den Lehrplan aufnehmen. Jedoch stehen hierfür keine zusätzlichen Unterrichtsstunden zur Verfügung und die Lehrer arbeiten oft mit begrenzten Ressourcen und Mitteln.

Die Lernenden selbst haben mit einer Fülle von Erfahrungen und Kenntnissen, die vor einigen Jahrzehnten noch Science-Fiction gewesen wären. Sie besitzen oft schon in jungen Jahren Computerkompetenzen und wurden durch Fernsehen, Filme und Musik mit »Erwachsenen«-Themen und -Botschaften konfrontiert. Obwohl die zuvor erwähnten Lehrmethoden von Natur aus Scrum sind, ist das Spektrum der gelehrteten Inhalte gewachsen. Das übergeordnete Ziel der Bildung bleibt jedoch dasselbe: gut ausgebildete Schüler.

Effektive Priorisierung sowie Überprüfungs- und Anpassungsmethoden sind unerlässlich, um diesen wachsenden Lehrplan sinnvoll zu verwalten. Die Lehrer setzen neue Prioritäten, da sie jeden Tag sorgfältig planen und die Schüler in die Lage versetzen, sich selbst zu organisieren und zu verwalten. Sie folgen nicht mehr einem starren Plan, der am Anfang des Schuljahres feststeht, sondern führen regelmäßig Beurteilungen durch und passen ihre Zeitpläne an. Dabei verwenden sie manchmal mehr Zeit als geplant für ein Thema, um einer Klasse zu helfen, wenn sie mit den Inhalten Schwierigkeiten hat. Manchmal benötigen sie weniger Zeit als geplant, wenn eine Klasse ein Konzept schneller als erwartet versteht.

Auch die Lehrer erkennen den Wert des Refactoring. Früher schrieb man eine Klassenarbeit, bekam eine Note dafür, und das war's. Weil die Pädagogen aus Erfahrung gelernt haben, wurden Anpassungen vorgenommen, damit die Schüler aus ihren Fehlern lernen und für ihr Lernen belohnt werden. Kontinuierliche Feedbackschleifen ermöglichen das Lernen, gefolgt von einer Bewertung (Vortest oder Quiz), mit Feedback und weiteren Lerneinheiten, gefolgt von der abschließenden Bewertung. Die

Note ist nicht das einzige Ziel. Die Schüler beginnen zu verstehen und zu schätzen, was sie lernen, und sie nutzen dieselben Ereignisse, die auch in Scrum verwendet werden.

Leistungsschwache Schüler fördern

Lehrer sind sehr hilfreich, wenn es darum geht, die Schüler durch Lerneinheiten zu bringen. Manchmal arbeiten die Lehrer jedoch nach einem strukturierten Lehrplan, der von ihnen verlangt, dass sie sich mit neuen Themen und Fächern beschäftigen, auch wenn noch nicht all ihre Schülerinnen und Schüler das Vorherige richtig verstanden haben. Obwohl sie vielleicht mehr wertvolle Zeit mit jedem Schüler verbringen möchten, haben sie dafür keine Zeit, da sie den gesamten Lehrplan abarbeiten müssen.

Daher ist die benötigte individuelle Aufmerksamkeit, um Schülern mit schlechten oder mittelmäßigen Noten zu helfen, manchmal aus Zeitmangel nicht verfügbar. Die Lehrer müssen in ihrem Lehrplan immer weiter vorankommen. Große Klassen können dieses schwierige Problem verstärken.

Die Folge davon ist, dass leistungsschwache Schüler im nächsten Schuljahr in die Klasse kommen, ohne das Material des Vorjahres vollständig verstanden zu haben, oder sie bleiben sitzen und das gleiche Material wird ohne neue Lerntechniken erneut unterrichtet. Es werden zwar Evaluierungen durchgeführt, aber der Unterricht kehrt nicht notwendigerweise zu den Inhalten zurück, die nicht gut verstanden wurden.

Der Erfolg bei der Förderung einzelner Schüler entsteht durch die Zusammenarbeit zwischen Schule, Lehrern, Eltern und Schülern. Wenn das Scrum-Framework für jeden Schüler mit Teammitgliedern wie dem Lehrer, dem Schüler, dem Betreuer, einem Schulverwalter und einem Tutor angewendet würde, könnte dieses Team iterative Ziele setzen, die zu einem Endziel führen, mit regelmäßigen Feedbackschleifen, während es den Fortschritt und den Prozess überprüft und an die Bedürfnisse des Schülers anpasst.

Mehr Schüler pro Lehrer

Die Klassen werden größer und die Aufmerksamkeit der Lehrer für den einzelnen Schüler nimmt ab. Es ist eine große Aufgabe, eine ganze Klasse und den wachsenden Lehrplan zu verwalten und gleichzeitig Unterricht anzubieten, der dem jeweiligen Fach gerecht wird und den einzelnen Schüler berücksichtigt.



Lehrer verbringen Zeit damit, Beziehungen aufzubauen, eine Lektion zu modifizieren, Lektionen durchzuführen und den Schülern persönlich zu helfen. Zeit zu finden, ist die wichtigste Herausforderung im Bildungsbereich. Da die Haushaltsmittel weniger werden, aber die Lehrmethoden und -systeme gleich bleiben, ist es unvermeidlich, dass die Bildung im Klassenzimmer darunter leidet. Scrum kann helfen, dass Lehrer ihre Zeit besser nutzen und ihre Anpassungsfähigkeit erhöhen.

Scrum im Klassenzimmer

Scrum funktioniert im Bildungsbereich wie überall sonst auch. Da Scrum auf Zielen und Visionen basiert, ermöglichen fokussierte Lernziele den Schülern, ihren Lernstil an ihre Lerngeschwindigkeit anzupassen. Durch die Fokussierung auf jeweils ein Lernziel können die Schüler jedes Fach und jedes Niveau meistern, bevor sie weitermachen. Mit Scrum können Lehrerinnen und Lehrer Lerninhalte mit hoher Priorität hervorheben, auf die sich die Schülerinnen und Schüler konzentrieren.

Scrum bleibt in großen Klassen wirksam, da für Projekte kleinere Scrum-Teams gebildet werden können. Unter der Berücksichtigung von Alter und der Natur der einzelnen Schülergruppen basieren, können diese Teams die Bildungsziele selbstorganisierend und selbstgesteuert erreichen.

Überprüfung und Anpassung funktionieren bemerkenswert gut in Lernumgebungen. Studien haben gezeigt, dass Schüler mit Unterrichtsplänen, auf die sie sich einstellen können und die es erlauben, an bestimmten Inhalten zu arbeiten, mit denen sie sich

schwertun, ein besseres Verständnis erreichen können. Iterative Unterrichtsmodelle wurden mit bemerkenswerten Ergebnissen in den Klassenzimmern eingeführt:

- ✓ Der Lehrer umreißt die Einheit oder das Projekt für alle Schüler und es werden Gruppen gebildet (Sprint-Planung).
- ✓ Der Lehrer führt den täglichen Unterricht durch und leitet Lernaufgaben in Gruppenarbeit, wobei er auch ein Zeitlimit setzt (Sprint).
- ✓ Der Schüler und der Lehrer bewerten die Leistung des Schülers am Ende des Zeitfensters (Sprint-Review).
- ✓ Wenn die Schüler ein vorher festgelegtes Niveau erreichen, wechseln sie zu neuem Material. Wenn sie unter diesem Niveau liegen, wird eine Sprint-Retrospektive durchgeführt.
- ✓ Anhand des Feedbacks aus dem Sprint-Review werden Problemthemen für jeden Schüler identifiziert.
- ✓ Neue Teams werden gebildet, um die Fähigkeit des Lehrers zu maximieren, den Schülern bei den individuellen Herausforderungen zu helfen. Diese Teams arbeiten zusammen, um ihr Verständnis für das Thema zu verbessern.
- ✓ Das Material für diese neuen Gruppen wird überarbeitet (Produkt-Backlog-Verfeinerung). Die Schülerinnen und Schüler entwickeln Ideen, wie sie das Thema besser erlernen können.
- ✓ Ein weiterer Sprint wird durchgeführt.
- ✓ Die Leistungen der Schüler werden bewertet.

Dieser Prozess wiederholt sich, wodurch die Leistung der Schüler schrittweise verbessert wird.

Studien haben gezeigt, dass sich die Leistungen der Schüler deutlich verbessern. Auch machten die Schüler weniger Fehler, hatten mehr Selbstvertrauen und verbrachten weniger Zeit mit der Suche nach Problemlösungen.

Scrum-Teams aus Schülern müssen sich nicht verändern, sie können stabil bleiben. Auf diese Weise können

funktionsübergreifende Teams gebildet werden, die sich aus Schülern zusammensetzen, die in einigen Bereichen über höhere Fähigkeiten verfügen als andere. Nachfolgende Sprints verwenden die gleichen Techniken wie bei der Paarprogrammierung und dem Shadowing. Durch das Lernen von anderen Schülern erwirbt jeder Schüler neue Fähigkeiten und fortgeschrittene Schüler erweitern ihre Fähigkeiten.

Standards auf verschiedenen Ebenen schreiben bestimmte Anforderungen dafür vor, wie der Lernerfolg in bestimmten Fächern nachgewiesen wird (das heißt die Definition of Done), aber jede Schule oder jeder Lehrer kann diese vorgegebenen Richtlinien der Definition of Done anpassen, um bestimmten Umständen gerecht zu werden. Was die Schüler lernen und wie sie ihren Lernerfolg präsentieren, sind wichtige Elemente für den Gesamterfolg des Sprints. Der Erfolg wird je nach Situation variieren, aber die Definition of Done sollte jeder kennen, insbesondere die Schüler.

Ein Beispiellansatz mit Scrum

Wie funktioniert das Scrum-im-Klassenzimmer-Konzept?

Die Kinder lieben es. Sie werden zu aktiven Teilnehmern mit Whiteboards und Haftnotizen und der Verantwortung, dafür zu sorgen, dass alles an seinem Platz ist.

Nachfolgend sehen Sie, wie ein Lehrer Scrum im Klassenzimmer verwendet:

1. Richten Sie ein Projekt mit einer Dauer von ein bis fünf Tagen ein.

Dieses Projekt ist eigentlich ein Sprint. Der Lehrer verwendet den ursprünglichen Lehrplan, sichert sich jedoch das Engagement der Schüler, indem er die Anforderungen aufführt. Wenn es im Projekt beispielsweise darum geht, das Periodensystem der Elemente zu verstehen, können die einzelnen Anforderungen die Abschnitte der Tabelle sein. Aufgaben können die einzelnen Elemente selbst sein.

2. Die Schüler bilden Scrum-Teams und die Teammitglieder entscheiden, wer die Scrum-Rollen einnimmt.
Diese Teams haben ein hohes Maß an Autonomie.
3. Ein Projekt-Backlog wird erstellt, oft mit Hilfe eines Taskboards.
Die Kinder schreiben die Anforderungen auf. Der Lehrer dient als Scrum Master und hält alle auf Trab.
4. Es werden Haftnotizen auf einem Whiteboard, einer Wand oder einer Tafel verwendet, um Kinder in die Sortierung und Priorisierung der Anforderungen einzubeziehen.
Kinder lieben das Gefühl, das etwas vollendet wird und sie können deutlich sehen, wie die Haftnotizen mit den Aufgaben aus dem Bereich *Zu erledigen* über den Bereich *In Arbeit* in den Abschnitt *Fertig* wandern.
5. Jede Projektsitzung beginnt mit einem Daily Scrum.
Die Schüler verwenden die Scrum-Terminologie, wie zum Beispiel Daily Scrum, Scrum Master und Sprint-Reviews.
6. Am Ende eines jeden Sprints werden sowohl das Review als auch die Retrospektive durchgeführt.
Das Engagement verbessert sich und die Schüler können bei Bedarf Ziele für neue Projekte einbringen.

Kinder erfreuen sich an diesem praktischen Prozess und genießen den Fortschritt. Sie sehen schnell Fortschritte und sind erstaunlich flexibel in der Anpassung. Kurz gesagt, Kinder denken von Natur aus im Sinne des iterativen Prozesses (siehe [Kapitel 1](#)). Scrum ist nur eine formalisierte Form des Umgangs mit der natürlichen Verkabelung des Gehirns.

eduScrum

In Alphen aan de Rijn in den Niederlanden wird Scrum in einem Gymnasium und einer Sekundarschule eingesetzt. Das Programm heißt eduScrum und die Pädagogen, die es verwenden, erleben erstaunliche Ergebnisse.

Lehrer und Schüler verwenden Scrum in allen Fächern und bilden Scrum-Teams, in denen alle Rollen vorkommen. Drei Lehrer führen als Scrum Master das Kommando, jeder von ihnen unterstützt den Prozess in seiner Klasse. Die Scrum-Master führen auch Daily Scrums zu den Projekten durch, ebenso wie Sprint-Reviews und Retrospektiven.

Die Schule sammelte Daten über die Ergebnisse dieses Prozesses mit 230 Schülern im Alter von 12 bis 17 Jahren. Die Ergebnisse waren beeindruckend. In den Niederlanden liegen die Testergebnisse zwischen 1 und 10. Eine Note von 5,5 ist gut, aber die Schüler streben nach 6,7 oder besser. Die Schüler, die an Scrum-Teams teilnahmen, übertrafen immer wieder diejenigen, die nicht an Scrum-Teams teilnahmen, und zwar um 0,8 bis 1,7 Punkte – ein deutlicher prozentualer Anstieg.

Die Hälfte der Schüler sagte, dass sie das Thema besser verstehen, mehr Spaß haben (was für das Lernen sehr wichtig ist), härter und schneller arbeiten können und das Gefühl haben, dass sie auf eine bessere Art und Weise lernen.

Die Lehrer bemerkten auch ein größeres Engagement ihrer Schüler und eine positivere Erfahrung. Interessanterweise berichten Unternehmen und Organisationen oft von einer verbesserten Arbeitsmoral, wenn sie Scrum einsetzen.



Ein interessanter Nebeneffekt von Scrum in den Schulen war, dass die Schüler einfach mehr Spaß am Unterricht und am Lernen hatten. Hierdurch gingen sie gerne zur Schule und wollten sich mehr beteiligen. Und diese schüchternen, ruhigen Typen? Sie blühten auf, als ihre Fähigkeiten und Beiträge in den Reviews und Retrospektiven wahrgenommen wurden.

Auch die Teamatmosphäre verbesserte sich. Weit über die Hälfte der Schüler lernte, besser zusammenzuarbeiten, und entwickelte Vertrauen in ihre Teammitglieder. Sie waren offen für das Geben und Empfangen von Feedback und die Lehrer bemerkten eine entspanntere Atmosphäre.

Blueprint

Blueprint High School in Chandler, Arizona, ist eine gemeinnützige Organisation, die Sonderpädagogik entwickelt und umsetzt. Die Schulleitung entschied, dass ein Diplom nicht ausreichen würde, um die Schüler auf das 21. Jahrhundert vorzubereiten. Die Schüler benötigen auch Fähigkeiten in den Bereichen Zusammenarbeit, Kreativität, Verantwortlichkeit, kritisches Denken und Teamarbeit. Interessanterweise fördert Scrum all diese Eigenschaften auf natürliche Weise.

Die Scrum-Rollen an der Blueprint High School wurden flexibel auf den individuellen Teamkontext abgestimmt. Manchmal war der Lehrer der Product Owner oder Scrum Master, manchmal übernahmen die Schüler diese Rollen. Mit zunehmender Reife und Erfahrung übernahmen die Teammitglieder automatisch die Rolle des Product Owners. Die Lehrer gaben einfach die Art der Projekte vor, die abgeschlossen werden sollten; dann übernahmen die Schüler und definierten ihre eigenen Ziele, Implementierungen und Reviews. Die Zusammenarbeit und die Teamfähigkeit nahmen zu.

Der Erfolg von Scrum in der High School führte dazu, dass Blueprint den Ansatz auch in der Grundschule einsetzte, was dort zu ähnlichen Ergebnissen führte. Obwohl Blueprint nicht die Art von quantitativer Studie durchgeführt hat, die eduScrum durchgeführt hat, gab es viele positive anekdotische Ergebnisse, darunter mehr Engagement der Schüler, mehr Spaß, größeres Einfühlungsvermögen, unabhängiges Denken und eine positivere Bildungserfahrung. Eine Lehrerin sagte, dass ihre Kinder sogar kommen würden, wenn sie krank wären, weil sie keinen einzigen Schultag verpassen wollten.

Militär und Strafverfolgung

Viele agile Experten und Scrum-Kenner schätzen die agilen Prinzipien und das Scrum-Framework aufgrund ihrer Erfahrung im Militär. Militärische Organisationen werden fälschlicherweise als

stark zentralisierte Entscheidungsträger wahrgenommen. Militärstrategen wissen jedoch seit langem, dass zentralisierte Entscheidungsfindung zu einer Niederlage auf dem Schlachtfeld führt. Ein Kommandant kann unmöglich alle Teile des Schlachtfeldes sehen oder schnell genug kommunizieren, um eine chaotische, sich schnell verändernde Situation zu verstehen oder sich neu ergebende Chancen auszunutzen. Kluge Befehlshaber wissen, dass sie auf allen Ebenen Leute befähigen müssen, rechtzeitig Entscheidungen zu treffen.

1871 beobachtete der deutsche Militärstrategie Helmuth von Moltke (der Ältere) weise: »Kein Schlachtplan überlebt die erste Feindberührung«. Heute ist sein Ansatz zur dezentralen Entscheidungsfindung beim Militär als Mission Command, Auftragstaktik, bekannt.

Durch die Doktrin der Auftragstaktik können Führungskräfte agile Entscheidungen treffen und sich unter unsicheren Bedingungen anpassen. Die Prinzipien werden von den meisten westlichen Militärorganisationen, auch dem U.S. Marine Corps und der U.S. Army, praktiziert. Die U.S. Army betreibt das Mission Command Center of Excellence, um Führungskräfte in den Methoden der dezentralen Entscheidungsfindung auszubilden.

Traditionelle militärische Organisationen haben riesige Befehls- und Kontrollstrukturen. Immer mehr militärische Führer erkennen, dass all die Führung und Kontrolle vielleicht doch nicht der ideale Ansatz ist.

Im Jahr 2007 ermöglichte General David Petraeus kleinen, funktionsübergreifenden Teams, agile Entscheidungen zu treffen, die die Aufstandsbekämpfung vor Ort in Bagdad veranschaulichten. »Sicherheit der Bevölkerung, besonders in Bagdad und in Partnerschaft mit den irakischen Sicherheitskräften, ist der Fokus der militärischen Bemühung,« sagte Petraeus.

Spezielle Einsatzkräfte, auch Special Forces genannt, sind speziell ausgebildet, um anpassungsfähig, selbstständig und in Unsicherheit operieren zu können. In den USA werden mit

Special Forces oft namhafte Spezialeinheiten wie die Green Berets und Navy Seals bezeichnet.

Agile Teams sind in vielerlei Hinsicht vergleichbar mit Spezialeinheiten. Beide Arten von Teams sind klein, gut ausgebildet, hochprofessionell, zusammenhängend und funktionsübergreifend. Spezialeinheiten sind klein und funktionsübergreifend, sodass sie sich an unterschiedliche Situationen anpassen können. Crossfunktionalität bedeutet, dass jedes Mitglied des Teams mehr als eine Sache tun kann; im Idealfall besitzt jeder im Team alle notwendigen Fähigkeiten.

Wie agile Teams sind auch Spezialeinheiten stabil und langlebig. Durch ihre hart erarbeitete Erfahrung wissen ihre Mitglieder, wie man zusammenarbeitet und einander vertraut, und sie tun alles, was nötig ist, um die Mission zu erfüllen. Diese Mission kann die Beherrschung einer Fremdsprache, den Aufbau von Beziehungen zu lokalen Dorfbewohnern oder sogar den geschickten Einsatz von tödlicher Gewalt erfordern. Ihr Auftrag kann eine beliebige Kombination dieser Dinge erfordern, aber die Durchführung der Mission kann etwas völlig anderes beinhalten. Spezialeinheiten wissen, wie man lernt und sich anpasst.

Eine weitere Ähnlichkeit besteht darin, dass sowohl agile als auch Spezialeinheiten ein Gefühl der gegenseitigen Rechenschaftspflicht schaffen. In Situationen mit hohen Einsätzen, ob im Geschäft oder an einem Kriegsschauplatz, ist der Wunsch, ein Teammitglied nicht im Stich zu lassen, wichtiger als jeder Termin.

Wie Scrum-Sprints werden Missionen für militärische Teams in der Regel so untergliedert, dass sie schnell, längstens innerhalb von Wochen, durchgeführt werden können. Längere Missionen verschleißten die Soldaten, verbrauchen die Vorräte und erfordern eine erhebliche kontinuierliche Unterstützung. Sie finden einen exponentiellen Zusammenhang zwischen der Dauer einer Mission und den Kosten und der Ausfallrate. Kurze Missionen sorgen für mehr Konzentration, Teamgeist und Erfolg.

Eines von vielen Beispielen, bei denen es einem Militärkommandanten gelang, seinen Befehls- und Kontrollansatz zu ändern, ist Admiral Nelson in Trafalgar. Anstatt die strikte Einhaltung der auf seinem Flaggschiff aufgezogenen SignalfLAGgen zu fordern, übertrug Nelson den Schiffskapitänen wesentliche Befugnisse und sagte: »Falls Signale weder gesehen noch perfekt verstanden werden können, kann kein Kapitän etwas falsch machen, wenn er sein Schiff neben das des Feindes stellt.«

Scrum und das FBI

Cyberangriffe aller Art, von Kreditkarten-Hacking bis hin zu Regierungs- und militärischen Sicherheitsbedrohungen, nehmen weltweit zu. Militär- und Cyberkrieg-Experten sind sich einig, dass die Kontrolle dieser wachsenden Bedrohung ein großes Anliegen ist. Programme für die Bekämpfung von Cyberbedrohungen, die sich an die sich ständig ändernden Herausforderungen anpassen können, werden benötigt – und zwar schnell. In vielen Fällen ist es eine Frage der nationalen Sicherheit.

Das FBI hat zunehmend aktiv an Lösungen gearbeitet, um auf diese Cyberangriffe auf nationaler und krimineller Ebene reagieren zu können. Die Führung weiß, dass es nicht ausreicht, alte Techniken zum Umgang mit diesen übermodernen Bedrohungen einzusetzen. Neue Wege des Technologieeinsatzes, der Personalgestaltung und der Zusammenarbeit mit Partnern sind gefragt.

Kein Wunder, dass sie sich dieser schwierigen Aufgabe mit Scrum nähern.

Nach den Terroranschlägen vom 11. September 2001 begann das FBI damit, den Informationsfluss und die Koordination mit allen relevanten Stellen zu verbessern. Nach einigen Fehlstarts entwickelte man das Programm Sentinel: ein umfassendes Software Case Management System. Ziel ist es, ein bestehendes System aus einer Mischung aus digitalem

und papiergebundenem Informationsfluss durch ein rein digitales zu ersetzen.

Zehn Jahre lang wurde das Projekt mit den vom FBI vorgeschriebenen Wasserfallmethoden durchgeführt, einschließlich umfangreicher Vorentwürfe und festgelegter Anforderungen. Ein neuer CIO, Chad Fulgham, verlangte von den beauftragten Entwicklern eine inkrementelle Lieferung. Hier sind einige der Symptome des bis dahin verwendeten Wasserfallprozesses:

- ✓ Nur 4 von 18 Workflows waren live, aber mit Fehlern.
- ✓ Weniger als 1 Prozent der mehr als 14.000 Formulare wurden im neuen System erstellt.
- ✓ Die bisherigen Kosten beliefen sich auf 405 Millionen Dollar, mit geschätzten weiteren 350 Millionen für die folgenden sechs Jahre.
- ✓ Alle gelieferten Funktionen wurden innerhalb der Organisation als optional angesehen.

Scrum wurde drei Monate lang eingesetzt und man konnte erreichen, was mit dem Wasserfallansatz in 10 Jahren nicht möglich war: 5% des Budgets und 80% weniger Personal. Insgesamt wurde der Personalüberhang um mehr als 50 Prozent reduziert, es wurden User Stories erstellt und 21 Sprints von zweiwöchiger Dauer geplant – was zu einer Verkürzung des Projektzeitplans um 85 Prozent führte.

Kapitel 11

Medienlandschaft im Umbruch

IN DIESEM KAPITEL

Veränderungen im Verlagswesen durch Überprüfung und Anpassung bewältigen

Digitale Disruption in der Nachrichten- und der Medienwelt

Die Geschichte wird gut zu mir sein, denn ich beabsichtige, sie zu schreiben.

Winston Churchill

In jedem sich rasant ändernden Umfeld sind schnelle und flexible Innovationstaktiken gefragt: Das Verlagswesen und die Nachrichtenmedien durchlaufen vor unseren Augen massive Veränderungen, traditionelle Produkte und Leser verändern sich und niemand weiß, wo der Wandel enden wird.

Verlage und Nachrichtenagenturen müssen weiterhin aktuelle Produkte monetarisieren und neue Einnahmequellen erschließen: Das Ausprobieren einer neuen Form von Native Advertisement, bei der die Werbung zum Inhalt wird (wie beispielsweise Anzeigen, die in einen Feed von Nachrichtenartikeln eingestreut werden), erfordert eine schnelle Bearbeitungszeit und eine schnelle Reaktion auf Kundenfeedback. Die Förderung eines neuen Buches über die bestehenden Social-Media-Kanäle eines Autors oder die Entwicklung neuer umsatzfördernder Ideen für Kundenfeedback erfordern einen disziplinierten Feedback-Zyklus, gezielte Entwicklung und enge Interaktion mit Kunden.

Wenn das Tor in Bewegung bleibt, macht es keinen Sinn, auf die Stelle zu zielen, an der es sich früher befand, denn wir wissen

nicht, wie sich das Verlagswesen und die Nachrichten weiterentwickeln werden. Daher müssen wir überprüfen und anpassen, während wir unterwegs sind. Die erfolgreichsten Player in der heutigen Medienlandschaft machen genau das.

Eine sich verändernde Landschaft im Verlagswesen

Enorme Veränderungen sind zu verzeichnen. Klassische Buchhandlungen leiden, die Leser haben andere Wege gefunden, um nach Büchern und Autoren zu suchen. Selbst Bibliotheken erhalten weniger öffentliche Mittel und ein großer Teil ihrer Regalfläche wird auf Computer und andere Medien umgestellt.

Hinzu kommt, dass die Leser heutzutage wählen können, ob sie auf Desktops, mobilen Plattformen und, was immer seltener vorkommt, in traditionellen Hardcover- und Taschenbüchern lesen.

In diesem neuen Verlagsumfeld ändern sich die traditionellen Erlösmodelle für Werbung und Abonnements, sodass neue Modelle benötigt werden, die die Vorteile dieser neuen digitalen Welt nutzen können.

Die Musikindustrie erlebte einen ähnlichen umfassenden Wandel: Traditionelle Albumkäufe flogen aus dem Fenster, als iTunes einflog, kostenlose und oft illegale Musikdownloads riefen eine neue Debatte über das Urheberrecht hervor (und bereicherten dabei eine Vielzahl von Anwälten), und die Vertriebswege für Musik sind heute dramatisch anders als noch vor 10 oder 20 Jahren. Dennoch entstehen ständig neue Songs und Alben und das Musikleben geht weiter.

Das Gleiche gilt für das Verlagswesen: Für manche Unternehmen sind massive Veränderungen in der Branche erschreckend, andere wiederum erkennen neue Chancen, es wird nach Flexibilität gesucht, Überprüfung und Anpassung werden als vorrangig angesehen. Manche Verlage sagen, dass sie aufgrund

der Schnelligkeit des Wandels nicht sicher sind, welche Daten sie für ihre Entscheidungen verwenden sollen. Der Sand bewegt sich unter ihren Füßen.

Scrum kann der Verlagsbranche zum Erfolg verhelfen: Gerade die Umstände, die so viele Menschen in der Branche beunruhigen – rascher Wandel, sich ändernde Konsumentenbedürfnisse und -wünsche und unsichere Einnahmequellen – sind es, in denen Scrum sich auszeichnet.

Prüfen, Anpassen und Refactoring

Überprüfung, Anpassung und Refactoring sind der Kern von Scrum, und sie passen in die Welt des Verlagswesens: Wie jede Branche, die wir in diesem Buch behandeln, hat auch das Verlagswesen besondere und Scrum-Lösungen.

Leser ändern sich

Was die Leser von einem Buch, einem Zeitschriftenartikel oder einem Zeitungsfeed erwarten, ändert sich: Sofortige Information und sofortige Befriedigung sind die Regel. Nur wenn der Autor bereits Bestseller geschrieben hat und eine große Fangemeinde besitzt, werden sich die meisten Leser hinsetzen und sich einem 1000-Seiten-Wälzer widmen.

Das schnelle Feedback der Leser, kurze Artikel, Blogs und analytische Tracking-Tools für die Klickraten und Antworten der Leser ermöglichen den Zugriff auf eine Flut von Daten, sodass Nachrichtenmedien, Verlage und einzelne Autoren schnell erkennen können, worauf die Leser reagieren, und sich entsprechend anpassen können.

Dieser schnelle Feedbackzyklus führt nicht nur zu besseren Inhalten, sondern auch zu schnellerer Monetarisierung: Während Sie im Prozess Überprüfungen und Anpassungen vornehmen, können Sie die Wege verfolgen, die zu mehr Klicks und Hits

führen, und somit mehr Einnahmen durch Werbung und Verkauf erzielen.

Autoren ändern sich

Hugh Howey hat mit seinem New-York-Times-Bestseller *Wool* auf unterschiedliche Weise mit alten Traditionen gebrochen. Er veröffentlichte die Original-Kurzgeschichte für 99 Cent pro Exemplar auf Amazon und erhielt eine so überwältigende positive Resonanz, dass er weiterschrieb.

Howey schrieb und veröffentlichte nach und nach fünf Fortsetzungsgeschichten; nach jedem Teil erhielt er Feedback von den Lesern. Er kombinierte sie zu dem Buch *Wool Omnibus* (im Selbstverlag bei Broad Reach Publishing erschienen, die deutsche Ausgabe erschien unter dem Titel *Silo*, Band 1, im Piper Verlag), das auf der Bestsellerliste der *Times* landete und ihm einen traumhaften siebenstelligen Geldstrom einbrachte.

Fortsetzungsgeschichten werden in Sprints produziert. Das Feedback kommt von den Lesern und potenziellen Käufern des Buches. Einige Autoren nutzen die Veröffentlichung im Selbstverlag, bei der jedes Detail überprüft und angepasst werden kann. Die gesamte Erfolgsgeschichte war eine unbeabsichtigte Variation von Scrum.

Nach der Veröffentlichung im Selbstverlag und dem Erreichen der Bestsellerliste unterzeichnete Howey einen Vertrag mit einem traditionellen Verlag, Simon & Schuster, und verkaufte in einem weiteren Schritt nur die Druckrechte an den Verlag, wobei er alle digitalen Rechte und Einnahmen für sich behielt.

Produkte ändern sich

Was und wie die Menschen lesen, ist im Wandel. Bildromane, Manga und interaktive Serien verbinden sich mit Romanen, Artikeln und Blogs zu einem weiten Netz, in dem die Autoren ihre Arbeit platzieren können. Nehmen Sie dann noch E-Reader, Smartphone-Apps sowie die kontinuierliche Abnahme der

Verkäufe in Papierform hinzu, und Sie haben eine Welt, die sich ändert.

Kein Autor und kein Leser bleibt von dieser literarischen Revolution unberührt: T.S. Eliots klassisches Epos *The Waste Land* (Das öde Land) hat eine eigene iPad-App, bei der nicht nur einfach der Text kopiert und eingefügt wurden. Die App enthält außerdem diese Features:

- ✓ eine gefilmte Aufführung des Gedichtes von Fiona Shaw
- ✓ Hörproben von T.S. Eliot und Schauspielern wie Alec Guinness und Viggo Mortensen
- ✓ interaktive Notizen, die dem Leser mit kulturellen Bezügen und poetischen Nuancen helfen
- ✓ Videos von Literaturexperten, die einen Einblick in das Meisterwerk geben
- ✓ Original-Manuskriptseiten, die dem Leser zeigen, wie sich das Gedicht unter der Leitung von Ezra Pound entwickelt hat

Die Verleger (Faber und Touch Press) sehen nicht einfach nur zu, klagen und wünschen sich die guten alten Zeiten zurück, sondern sie nutzen die Möglichkeiten und produzieren auf neue Art und Weise große Kunst.

Sowohl für Verlage als auch für Autoren ermöglicht der Feedbackzyklus von Scrum schnellen Input und damit eine schnellere Produkteinführungszeit, wodurch Produkte entstehen, die von den Kunden gewünscht und bezahlt werden. Wer diesen Änderungen positiv gegenübersteht und agile Frameworks nutzt, die dies unterstützen, wie beispielsweise Scrum, wird in diesem neuen Umfeld erfolgreich sein.



»Überleben der Schnellsten« ist das Mantra. Sehen Sie Veränderung als Chance und Sie werden an die Spitze kommen.

Scrum anwenden

Wie in der Softwarebranche können Inhalte einfach und häufig überprüft, angepasst und überarbeitet werden, bis sie für die Veröffentlichung bereit sind. Bei kurzen Werken ist der Prozess einfacher, aber das Prinzip lässt sich auch auf Werke jeglichen Umfangs anpassen.

Beim Erstellen von Inhalten für YouTube ist Überprüfen und Anpassen eine Selbstverständlichkeit: Sie können Inhalte posten und mit Hilfe von Analysen prüfen, wie viele Leute sie gesehen haben, ob sie Ihren Kanal abonniert haben, wie lange sie geblieben sind und welche Links sie verfolgt haben.

Auch bei längeren Stücken funktioniert der Prozess. Hier als Beispiel Marks erstes Buch, *Agiles Projektmanagement für Dummies* (erschienen im Wiley-VCH Verlag):

1. Mark legte die Vision für das Buch fest (ein Praxis-Ratgeber, den Leute, die Scrum machen, als Referenz verwenden können).
2. Er stellte eine Roadmap auf: die Gliederung des Buches.
3. Er unterteilte die Gliederung in verschiedene Buchteile (Releases) und jedes Kapitel wurde zu einem eigenen Sprint.
4. Er begann mit [Kapitel 1](#) und schrieb es auf Basis der Autorenrichtlinien für die Dummies-Reihe.

Nachdem er [Kapitel 1](#) an die Wiley-Lektorin geschickt hatte, teilte sie ihm mit, dass sie es richtig schlecht fand und warum (Feedback, Feedback, Feedback).

5. Er implementierte dieses Feedback in [Kapitel 1](#), reflektierte, was er gelernt hatte und nutzte das neue Wissen beim Schreiben von [Kapitel 2](#).

Er schickte [Kapitel 2](#) an die Wiley-Lektorin und erfuhr, dass es ihr nicht gefiel und warum dies so war (Feedback, Feedback, Feedback).

6. Als er [Kapitel 3](#) an seine Lektorin schickte, sagte sie ihm: »Es ist okay, aber es wäre besser, wenn Sie ... « (Feedback, Feedback, Feedback).

Dieser Zyklus wurde fortgesetzt, bis Mark und seine Lektorin bei [Kapitel 5](#) miteinander synchron waren.

Hätte er alle 20 Kapitel geschrieben und sie alle auf einmal an Wiley geschickt und die Lektorin hätte ihm gesagt, dass sie das Buch richtig schlecht fände, wäre es bis heute nicht auf dem Markt.

Das Scrum in Scrum für Dummies

Getreu dem Scrum-Motto »Überprüfen und Anpassen« hat Mark die Lernerfahrungen und Erkenntnisse, die er beim Schreiben seines letzten Buches gewonnen hat, beim Schreiben von *Scrum für Dummies* umgesetzt.

Er richtete das Buch als virtuelles Taskboard ein, seine Haftnotizen waren virtuell, und die Spalten von links nach rechts waren

- ✓ ein Produkt-Backlog, in dem alle Kapitel, Vorder- und Rückseite und alle anderen relevanten Informationen aufgelistet waren,
- ✓ To-do,
- ✓ in Arbeit,
- ✓ erledigt.

Jedes Kapitel enthielt Aufgaben wie Recherche, Schreiben, Bearbeiten (Feedback!), Überarbeiten, Überprüfen und Genehmigen, in einigen Fällen wurden mehrere Aufgaben erstellt (beispielsweise für mehrere Bearbeitungen).

Anforderungen waren Kapitel, Aufgaben waren Kapitelaktivitäten, das Verlagsteam (Entwicklungsteam) »schwärmte« für jeden Teil und jedes Kapitel aus und

wechselte dann zum nächsten, es gab Daily Scrums, in denen jede Partei umriss, was sie tat und welche Hindernisse auftraten. Nach den wöchentlichen Sprints folgten das Review und eine Retrospektive, in der die Teammitglieder ausarbeiteten, was besser gemacht werden konnte.

Die Klarheit war erstaunlich: Alle Mitglieder des Autorentams wussten, was sie wann tun sollten.

Nachrichtenmedien und Scrum

Die Nachrichtenmedien haben einen seismischen Wandel erlebt: Print ist online gegangen, die Werbung verändert sich mit jedem neuen Medium (wie Print, Radio, TV, Online, Social Media und mobiler Nutzung) und die Erfahrungen der Leser haben sich gewandelt: Viele Menschen bekommen beispielsweise keine Tageszeitung mehr nach Hause geliefert.

Aber im Scrum ist Veränderung gut, zumindest hilft Scrum, Veränderungen für Verbesserungen nutzbar zu machen.

Clayton Christensen prägte den Begriff der »disruptiven Innovation«. Diese kann als das beschrieben werden, was passiert, wenn ein neues Produkt oder eine neue Dienstleistung in einen bestehenden Markt eintritt und unerbittlich Anteile gewinnt, bis es etablierte Konkurrenten entwurzelt.

Die größten Herausforderungen für traditionelle Printmedienunternehmen bestehen darin, Wege zu finden, ihr aktuelles Produktangebot zu monetarisieren und digital umzusetzen. Print und Digital unterscheiden sich völlig voneinander. Intelligente Medienunternehmen, die sowohl eine Print- als auch eine digitale Präsenz haben, haben die beiden Seiten ihres Unternehmens voneinander getrennt, damit sie das tun können, was sie am besten können – die digitale Seite dieser Unternehmen, wie in [Kapitel 7](#) dargelegt, implementiert Scrum.

Brady Mortensen, ein Newsroom-Veteran und Senior Director (Product Owner) bei Deseret Digital Media, sagte:

Tatsächlich haben Nachrichtenorganisationen wahrscheinlich viele Jahre lang viele Scrum-Techniken praktiziert, ohne es zu merken. Daily-Scrum-Meetings sind keine Seltenheit: Bei lokalen TV-Sendern oder Tageszeitungen ist die „Sprint“-Länge ein Tag, und das Endprodukt ist eine Sammlung von Nachrichtensendungen und eine Zeitung. Am Ende eines jeden Zyklus entsteht ein nutzbares Produkt. Newsrooms leben und sterben durch diese Praktiken. Was traditionellen Nachrichtenorganisationen helfen würde, ist zu erkennen, dass das, was sie bereits tun, scrumartig ist, und dass sie Techniken noch mehr annehmen.

Die Unternehmen, die in diesem neuen Umfeld erfolgreich sind, sind diejenigen, die sich als flink erwiesen haben – und vor allem diejenigen, die in regelmäßigen Abständen ihre risikoreichsten Bereiche identifiziert haben (Überprüfung und Anpassung).

Organisationen wie *The Washington Post*, *The Chicago Tribune* und *National Public Radio* nutzen Scrum in ihren Redaktionen. Einige spezifische Techniken sind die folgenden:

- ✓ **Chicago Tribune:** Teams beginnen mit der Frage, wer die Benutzer sind, was sie brauchen und welche Funktionen sie einbinden können, um diese Anforderungen zu erfüllen.

Das Entwicklungsteam besteht aus Journalisten, Designern, Fotografen, Redakteuren und anderen, die mit der Entwicklung von Inhalten zu tun haben.

Scrum reduziert die Anzahl der Meetings, die gerade bei digitalen Medien überwältigend sein können.

- ✓ **NPR (National Public Radio):** NPR verwendet einen zweiwöchigen Sprintzyklus mit einer zweistündigen Sprint-Planungsbesprechung am Start.

Daily Scrums dauern 15 Minuten, Teams koordinieren, wer an welchen Geschichten arbeitet, und Hindernisse werden identifiziert und beseitigt.

- ✓ **Washington Post:** Diese Zeitung verfügt über eine spezielle agile Technik zur Entwicklung von Inhalten für seine Live-Blogging-Plattform.

Das Team beginnt mit einer Vision: Welche Wirkung will es auf den Anwender haben?

In den Daily Scrum-Meetings entscheidet das Team, woran es an diesem Tag arbeitet: Journalisten arbeiten in Paaren, ein Prozess, den die *Post* schon lange vor der Scrum-Implementierung genutzt hat. Zwei Journalisten sitzen am selben Schreibtisch und beenden das Projekt, was die Arbeit intensiviert und Ablenkungen begrenzt.

Das Ziel ist es, so schnell wie möglich mit den Nachrichten live zu gehen und dann Feedback von den Benutzern und der Gruppe zu erhalten. Auf der Grundlage dieses Feedbacks nimmt das Team Anpassungen und Korrekturen für den nächsten Zyklus vor.

Definieren, wann Inhalte als »done« gelten

Die Verwendung von Scrum für die Entwicklung von Produkten und Dienstleistungen, die keine Software sind, wie beispielsweise Inhalte für die Veröffentlichung, ist vergleichbar mit der Verwendung von Scrum für Software.

Um zu unserer Roadmap to Value zurückzukommen, sollte ein Verlags-Scrum-Team eine Vision haben, die angibt, was die Bedürfnisse der Leser sind, wie die Publikation die Bedürfnisse dieser Leser erfüllt, worin sie sich im jeweiligen Markt und der Branche von Mitbewerben unterscheidet und wie die Vision mit der Unternehmensstrategie verknüpft ist (mehr zu den Visionen siehe [Kapitel 2](#)).

Die Roadmap spiegelt diese Vision wider, indem sie die redaktionellen Schwerpunkte der Publikation, einschließlich saisonaler Überlegungen, umreißt: Das Produkt-Backlog ist eine

priorisierte und geordnete Liste der vorgeschlagenen Features, Serien und Geschichten, die recherchiert, entwickelt, bearbeitet und veröffentlicht werden sollen.

Die Vision ist der Rahmen, um zu definieren, was es bedeutet, Inhalte fertiggestellt zu haben, und zwar jeweils für eine Geschichte oder einen Artikel. Die Definition of Done könnte ungefähr so aussehen:

- ✓ Mit jedem Artikel ist es uns gelungen, unsere Vision [Aussage dazu, wie der Bedarf gedeckt wird und wie sich unser Angebot unterscheidet] zu verwirklichen, nachdem wir das Wer, Was, Wann, Wo und Warum in der Einleitung dargelegt haben. Diese Elemente sind deutlich vorhanden.
- ✓ Für Ausgewogenheit sorgen, die sicherstellt, dass beide Seiten eines Themas vertreten sind.
- ✓ Sicherstellen, dass Suchmaschinenoptimierungsstandards und -anforderungen in Artikel-Elementen wie Headline, Tags und Body erfüllt werden.
- ✓ Für die unterschiedlichen Standpunkte wurde mindestens eine Quelle zitiert.
- ✓ Der Text wurde zweimal auf Genauigkeit geprüft, um Floskeln und Verzerrungen zu vermeiden.
- ✓ Begleitinhalte für Social-Media-Beiträge sind vorbereitet.
- ✓ Es wurde überprüft, ob die sprachliche und inhaltliche Bearbeitung abgeschlossen und der Text korrekturgelesen wurde.

Ein Newsroom, in dem diese Kriterien für Kuratoren, Redakteure und Produzenten stets im Mittelpunkt stehen, bietet Konsistenz und Klarheit darüber, was erwartet wird und wie der Erfolg aussieht.

Das Nachrichtenmedien-Scrum-Team

Ein Content-Team-Direktor für eine große regionale Nachrichten-Website nutzt die folgende Scrum-Implementierung. Die Aufgabe dieses Teams besteht darin, tägliche Inhalte im Hinblick auf die Relevanz für die Leser zu bewerten, sie zu bearbeiten und auf der Seite zu veröffentlichen. Obwohl das Team keinen Programmcode entwickelt, wird dennoch Scrum eingesetzt. In der Anpassung von Scrum hat das News-Team die folgenden Themen abgedeckt:

- ✓ Was ist das Produkt? (News-Content, den Kunden lesen wollen)
- ✓ Was ist das Produkt-Backlog? (Potenzielle Nachrichten und zugehörige Medienbeiträge, die sich ständig ändern)
- ✓ Was ist das Release? (Kontinuierliche Lieferung; der Inhalt wird geliefert, sobald er bearbeitet und freigegeben ist)
- ✓ Was ist die Sprintdauer? (Der 24-Stunden-Nachrichtenzyklus)
- ✓ Wann ist die Sprint-Planung? (Zu Beginn eines jeden Morgens; Backlog-Stories, die weiter recherchiert werden sollen, werden diskutiert)
- ✓ Wann ist das Sprint-Review? (Ende eines jeden Tages oder vor der Sprint-Planung jeden Morgen, die einen Überblick über die veröffentlichten Artikel und die daraus resultierenden Analysen zur Überprüfung und Anpassung an die Artikel des nächsten Sprints enthält)
- ✓ Wer ist im Entwicklungsteam? (Kuratoren, Reporter, Fotografen, Redakteure, Grafiker und Videografen)
- ✓ Wer ist der Product Owner? (Der leitende Redakteur)
- ✓ Wer ist der Scrum Master? (Ein Mitglied des Entwicklungsteams, das sich mit Scrum auskennt, Erfahrung damit hat und die organisatorische Durchsetzungsfähigkeit besitzt, Hindernisse für das Team zu beseitigen)

- ✓ Wer führt die Backlog-Verfeinerung durch und wann? (Das gesamte Entwicklungsteam und der Product Owner während des gesamten Tages kuratieren und evaluieren vorgeschlagene Geschichten, einschließlich aktueller Nachrichten)

Wenn diese grundlegenden Fragen gestellt und beantwortet werden, wird das Gesamtbild sichtbar: Jede Rolle, jedes Artefakt und jedes Ereignis kann identifiziert und zugeordnet werden.

Sprint-Flexibilität

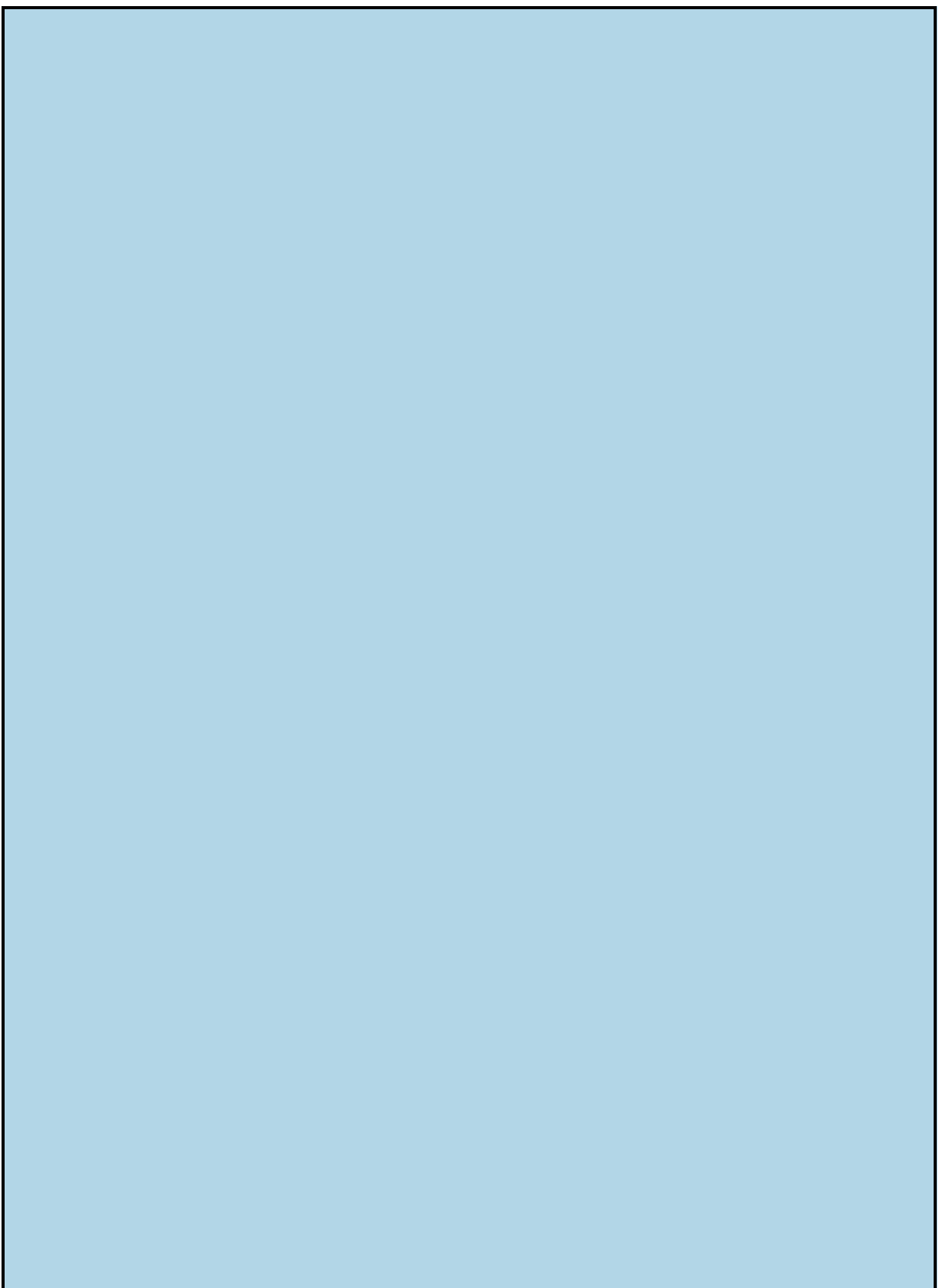
Tägliche Sprints in einer Nachrichtenorganisation bieten oft die nötige Flexibilität für tägliche Newsfeeds. Man kann Nachrichten nicht fünf Tage vorausplanen, aber man kann, zumindest in den meisten Fällen, einen Nachrichtentag planen. Während des Sprints können Eilmeldungen und unerwartete Nachrichten durch direkte Kommunikation zwischen den Teammitgliedern berücksichtigt werden.

Medien mit längeren Inhaltszyklen, wie beispielsweise Magazine (online und/oder in gedruckter Form), können längere Sprintzyklen für Inhalte verwenden, wobei jedes Feature – ein Abschnitt, ein Artikel, ein Kapitel oder ein anderes Segment – bei Bedarf in Anforderungen und Aufgaben unterteilt werden kann.

Teil IV

Scrum für betriebliche Funktionen





IN DIESEM TEIL ...

Optimieren Sie die Performance Ihres Projektportfolios.

Skalieren Sie Scrum auf mehr als 1.000 Projektteams.

Fördern Sie Projekte inkrementell und gewinnbringend.

Maximieren Sie Teams für Geschäftsfeldentwicklung und für den Kundendienst.

Steigern Sie den Wert und die Effizienz von IT und Wartung.

Kapitel 12

IT-Management und Operations

IN DIESEM KAPITEL

Die Migration großer Datenmengen erleichtern
IT-Tools aktivieren und sichern
Talente halten und fördern
Betriebswert verbessern
Innovative Wartung und Support

Jede hinreichend fortschrittliche Technologie ist von Magie nicht zu unterscheiden.

Sir Arthur C. Clarke

In [Kapitel 8](#) erörtern wir Scrum in der Softwareentwicklungsbranche und geben Beispiele für die Leistungsfähigkeit von Scrum in diesem Bereich. Einige Leute gehen davon aus, dass Informationstechnologie (IT) und Software ein und dasselbe sind, aber diese Annahme ist völlig falsch.

Softwareentwicklung hat mit der Erstellung von Softwareanwendungen zu tun, die auf Computersystemen und elektronischen Geräten laufen. In der Informationstechnologie geht es um Computer- und Telekommunikationssysteme zum Speichern, Senden und Abrufen von Informationen.

Nicht alle Organisationen haben eine Softwareentwicklungsabteilung. Es gibt aber kaum eine Organisation, die ohne einen Stab von IT-Fachleuten überleben kann, der sich um den reibungslosen Informationsfluss und um die Informationssicherheit kümmert.

Der IT-Betrieb kann von Unternehmen zu Unternehmen sehr unterschiedlich sein. Rollen und Verantwortlichkeiten können sich unterscheiden und selbst innerhalb einer einzelnen Gruppe kann die Art der Arbeit von Tag zu Tag und von Quartal zu Quartal variieren. Darüber hinaus erlebt die IT einen rasanten Wandel, da sich die Technologie weiterentwickelt und die Möglichkeiten immer größer werden.

In jedem wachsenden Bereich gibt es viele Herausforderungen. Wenn das System überlastet wird und die Dienste unterbrochen werden, kann es zu Verzögerungen oder sogar zum Abbruch der geschäftskritischen Arbeit kommen, was zu erhöhten Kosten, Ressourcenverschwendung, Umsatzeinbußen und sogar zum Verlust von Kunden führt. Der Arbeitsstau eines IT-Teams kann so schnell anwachsen, dass es nicht mehr mithalten kann.

In diesem Kapitel beschäftigen wir uns mit IT-Management und Operations und beschreiben, wie Scrum zeit- und kostensparende Lösungen für moderne Herausforderungen bietet.

Big Data und Migration im großen Maßstab

Der Menge der Daten ist erstaunlich, und selbst während wir dieses Kapitel schreiben, wird sie immer größer. Der Versuch, sich die Größe von Big Data vorzustellen, ist, als wollten Sie sich ein riesiges mathematisches Phänomen wie die Geschwindigkeit des expandierenden Universums vorstellen. Big Data ist so groß, dass es über das Vorstellungsvermögen der meisten Menschen hinausgeht.

Die folgenden Zahlen sind nur einige Beispiele dafür, wie groß Big Data sein kann:

- ✓ Walmart führt 1 Million Transaktionen pro Stunde durch und füttert Datenbanken mit mehr als 2,5 Petabyte (etwa 167-mal so viel wie die Daten aller Bücher in der Library of Congress).

- ✓ Facebook beherbergt mehr als 40 Milliarden Fotos.
- ✓ eBay verarbeitet täglich 50 Petabyte an Informationen.
- ✓ Die Entschlüsselung des menschlichen Genoms erfordert die Analyse von etwa 3,27 Milliarden Basenpaaren. Beim ersten Mal dauerte der Prozess zehn Jahre, jetzt dauert er eine Woche.
- ✓ Cisco schätzt, dass im Jahr 2016 die jährliche Datenmenge des globalen Internetverkehrs 1,2 Zettabyte (ZB) pro Jahr oder 96 Exabyte (EB) pro Monat betrug. Der jährliche globale Internetverkehr wird bis 2021 voraussichtlich 3,3 ZB pro Jahr oder 278 EB pro Monat erreichen.

Die Bedeutung von Big Data kann nicht hoch genug eingeschätzt werden. Viele dieser Daten sind sehr persönlich und sensibel und sie können sowohl das individuelle Leben als auch das Betriebsergebnis beeinflussen. Die Herausforderung besteht darin, Daten schnell, effektiv und korrekt zu erfassen, zu verwalten und zu interpretieren. Auch mögliche zukünftige Nutzungen dieser Daten müssen bei der Gestaltung der Speicher- und Abrufprozesse berücksichtigt werden. Diese Daten müssen zu nützlicher Intelligenz werden und dürfen nicht nur Daten bleiben.

Eine große Herausforderung besteht darin, dass 80 Prozent dieser Daten unstrukturiert sind (wie E-Mails, Blogs, Tabellenkalkulationen, Textdokumente, Bilder, Videos, Sprache und Suchprotokolle). Dieses unstrukturierte Segment wächst schneller als strukturierte Daten. Mit anderen Worten, die meisten Daten sind ein riesiges Durcheinander.

Datensicherheit und Datenschutz sind wichtiger denn je; gleichzeitig ist Sicherheit schwieriger denn je zu gewährleisten. Herkömmliche Datenmanagement-Frameworks und -Prozesse sind nicht in der Lage, diese Menge zu verarbeiten. Schnelligkeit, Flexibilität und sofortiges Feedback sind gefragt. Sechs Monate sind zu lang, um zu hoffen, dass ein neues, ungetestetes System funktioniert. Und die Chancen stehen gut, dass sich in sechs

Monaten die Anforderungen geändert haben oder eine neue Lücke entsteht.

Um mit diesem Daten-Tsunami fertigzuwerden, ziehen viele Firmen und Organisationen in die Cloud. Viele Unternehmen verfügen über eigene Cloudstrukturen oder virtualisierte Umgebungen. (In [Kapitel 8](#) finden Sie weitere Details zu Cloud-Computing.)

Was ist ein Zettabyte?

Ein Zettabyte ist eine Milliarde Gigabyte oder 10^{21} Bytes digitale Information. Hier ist die Skala der Datengrößen:

- ✓ 1 Byte entspricht einem einzelnen digitalen Zeichen eines Textes (zum Beispiel Buchstabe a).
- ✓ 1 KB sind ein Kilobyte oder etwa 1.000 Bytes digitale Informationen.
- ✓ 1 MB sind ein Megabyte oder 1 Million (1.000^2) Bytes digitale Informationen.
- ✓ 1 GB sind ein Gigabyte oder 1 Milliarde (1.000^3) Bytes digitale Informationen.
- ✓ 1 TB sind ein Terabyte oder 1 Billion (1.000^4) Bytes digitale Informationen.
- ✓ 1 PB sind ein Petabyte oder 1 Billiarde (1.000^5) Bytes digitale Informationen.
- ✓ 1 EB sind ein Exabyte oder 1 Trillion (1.000^6) Bytes digitale Informationen.
- ✓ 1 ZB sind ein Zettabyte, oder 1 Sextillion (1.000^7) Bytes digitale Informationen.

Data-Warehouse-Projektmanagement

Data-Warehouse-Projekte gelten traditionell als schwierig zu managen. Obwohl jedes Segment oder jede Phase des Projekts einen erkennbaren Anfang und ein erkennbares Ende hat, ist das Data Warehouse selbst nie fertig; es wächst und verändert sich ständig.



Ein Data Warehouse ist kein stacheldrahtumzäuntes Gebäude am Stadtrand. Vielmehr ist es ein Prozess oder Rahmen für den Umgang mit Daten innerhalb eines Unternehmens oder einer Organisation oder eine wissensbasierte Anwendungsarchitektur, die strategische und taktische Entscheidungen erleichtert.

Eine weitere Komplexität besteht darin, dass eine kontinuierliche Fusions- und Akquisitionstätigkeit unternehmensweite Datenintegrationsprobleme aufwirft. Neue Assets und Gruppen werden erworben oder ausgegliedert und entsprechende Daten und Prozesse müssen verwaltet werden. Die Pflege diverser Legacy-Anwendungen, die sich nicht gut integrieren lassen, kann kostspieliger sein als Konvertierungsprojekte.

Migration großer Datenmengen

Ein Beispiel für die erfolgreiche Umsetzung von Scrum ist ein Multimilliarden-Dollar-Energieunternehmen mit Aktivitäten in den USA und Brasilien. Häufige Fusionen und Übernahmen erforderten eine zeitnahe und genaue Integration der Daten. Die Geschäftsleitung benötigte Berichte über die neuen Einheiten und ihre neuen Produkte und Kunden. Konkret konzentrierte sich das Management auf Energieeffizienzprojekte, um dieses Kundensegment fest im Griff zu behalten.

Die Daten lagen in verschiedenen Formaten über mehrere Anwendungen hinweg vor, darunter Budget- und Finanzdaten in Tabellenkalkulationen, ein Customer Relationship Management (CRM)-Tool und Daten aus einer Vielzahl von Cloud-basierten Kundenbefragungen.

Der Scrum-Integrationsprozess ging in diese Richtung:

1. Rollen wurden eingerichtet.

Stakeholder wurden als Endnutzer identifiziert. Der Product Owner vertrat die Stakeholder und Sponsoren. Der Projektleiter wurde zum Scrum Master. Das Entwicklungsteam wurde durch den Datenarchitekten, den Systemarchitekten und einen ETL-Architekten (extract, transform, load) vertreten.

2. Die Stakeholder und der Product Owner legten den Ausgangspunkt mit der höchsten Priorität fest: die Budgetdaten.

3. ETL war der Umfang des ersten Projektblocks. Als die Daten verfügbar wurden, arbeitete der Product Owner mit den Stakeholdern zusammen, um Berichte zu identifizieren, die inkrementell implementiert werden konnten, und nahm sie in das Produkt-Backlog für die Priorisierung in kommenden Sprints.

4. Sprint 1 wurde ausgeführt. Budgetdaten wurden in das Data Warehouse geladen. Die Daten wurden mit Beispielberichten verifiziert.

5. Sprint 2 wurde ausgeführt. CRM-Daten wurden in das Data Warehouse geladen. Zur Validierung der Daten wurden Beispielberichte verwendet.

Nach dem zweiten Sprint identifizierten die Stakeholder neue Anforderungen in den Budgetdaten, nachdem sie diese mit den geladenen CRM-Daten verglichen hatten. Diese Anforderungen wurden in das Produkt-Backlog eingefügt und priorisiert.

6. Sprint 3 wurde ausgeführt. Die Umfragedaten wurden geladen und mit Beispielberichten validiert.
Beim dritten Sprint wurden neue Datenquellen entdeckt und geladen.
7. Sprint 4 wurde ausgeführt. Die Integration aller drei Datenquellen wurde mit Beispielberichten verifiziert.
8. In den übrigen Sprints wurden die Berichte nach Priorität erstellt.

Innerhalb weniger Monate, nicht Jahre, war die gesamte Datenbankmigration abgeschlossen. Das Data Warehouse war voll funktionsfähig und ein Prozess zur Verwaltung neuer Daten und Änderungen war vorhanden. Ein Grund für den Erfolg dieses Projekts war die Zusammenarbeit von Geschäfts- und Entwicklungsteams auf ein gemeinsames Ziel hin.

Der Prozess war schneller abgeschlossen, da er nicht das Wasserfallmodell verwendete, bei dem zuerst alle möglichen Daten aus allen möglichen Quellen geladen und dann das Gesamtprodukt getestet wird. Vielmehr wurden die Daten mit der höchsten Priorität zunächst phasenweise implementiert und dabei überprüft und angepasst. Neue Erkenntnisse flossen in den nächsten Sprint ein und die Daten mit der nächsthöheren Priorität wurden als Nächstes umgesetzt.

Unternehmensressourcenplanung

Enterprise Resource Planning (ERP) ist eine Suite von integrierten und dynamischen Softwareanwendungen, mit denen Organisationen und Unternehmen Geschäftsprozesse erfassen, verwalten, interpretieren und integrieren, einschließlich Planung,

Einkauf, Fertigung, Inventar, Marketing, Vertrieb, Finanzen und Personalwesen.

Die Implementierung eines ERP-Systems bedeutet in der Regel die gleichzeitige Entwicklung über verschiedene Funktionsbereiche (zum Beispiel Marketing, Vertrieb, Vorrat und Einkauf) hinweg, um einen bestimmten Geschäftsvorgang durchzuführen. Die Implementierung beinhaltet das Design und die Konfiguration vieler Module gleichzeitig. Diese Module werden zwar individuell entwickelt, müssen aber auch für den funktionsübergreifenden Einsatz konzipiert sein. So wird beispielsweise bei der Konzeption des Vertriebsmoduls sowohl auf vor- als auch auf nachgelagerte Prozesse geachtet.

Überlegen Sie, wie sich der Vertrieb in den gesamten End-to-End-Prozess einfügt. Sie beginnen mit dem Lagerbestand und müssen anschließend in der Lage sein, Ihre Aufträge zu fakturieren. Daher muss sich das Verkaufsmodul nahtlos in Ihr Bestandsmodul und Ihr Finanzmodul integrieren (und Ihr Bestandsmodul muss sich in Ihre Fertigungs- und Einkaufsmodule integrieren, die sich in Ihre Finanzmodule integrieren müssen).

Leider dauert der Entwurf und Bau dieser einzelnen Module traditionell Jahre, bevor die integrierte Testphase beginnt. Zu diesem Zeitpunkt wird für das Erkennen und Beheben von Lücken zwischen den Modulen noch mehr Zeit benötigt. Eine kleine Lücke zwischen Vertrieb und Finanzen kann zu monatelangem Mehraufwand führen. Dann kann es noch passieren, dass dieser Fix nicht perfekt in ein anderes Modul integriert werden kann. Wenn alle bis zur integrierten Testphase in Silos arbeiten, ist es schwierig, Lücken und nicht passende Komponenten frühzeitig zu erkennen.

Traditionell haben ERP-Anbieter diese Interdependenz durch das Einbinden einer bestimmten Entwicklungssequenz bewältigt. Tatsächlich mussten auch Parameter, die nicht verwendet werden sollten, in der vom ERP-Anbieter festgelegten Reihenfolge konfiguriert werden.

Mit skalierten Scrum-Teams (siehe [Kapitel 13](#)) können Sie diese Anpassung nun parallel durchführen, wobei sich jedes Scrum-Team auf einen bestimmten Funktionsbereich konzentriert und automatisierte Integrationstests verwendet, um sicherzustellen, dass der Geschäftsvorgang modulübergreifend funktioniert. Durch die Anwendung agiler Techniken können Integrationstests täglich (und zwar vom ersten Tag an) durchgeführt werden, und nicht erst Monate oder Jahre nach dem Projektbeginn.

Auch wenn diese modularen Abhängigkeiten als Verbindlichkeiten erscheinen mögen, machen sie es einfacher, die Arbeit in Blöcke zu unterteilen, die zu separaten Scrum-Teams passen, die wiederum synchronisierte Sprints durchführen. Die Priorisierung des Produkt-Backlogs wird auf Programmebene festgelegt und inkrementelle Anforderungsänderungen werden minimiert. Auch die Priorisierung des Sprint-Backlogs passt in diese Vorgehensweise. Sie können die Flexibilität von Scrum nutzen und die Umsetzung drastisch beschleunigen. (Wir befassen uns in [Kapitel 13](#) intensiver mit diesem Modell des vertikalen Schnitts.)

Die ERP-Systemarchitektur orientiert sich zunehmend an Software as a Service (SaaS; siehe [Kapitel 8](#)), was bedeutet, dass monolithische Komponenten modularer sind als bisher bei Client-Installationen.

Außerdem sind die Aufgaben, die zur Konfiguration von ERP-Systemen erforderlich sind, in der Regel repetitiv, sodass Kadenz und Schätzung frühzeitig ermittelt werden können und genaue Größen- und Zeitvorhersagen möglich sind.

In [Kapitel 13](#) erörtern wir die Skalierung von Scrum über mehrere Teams hinweg. Um mehr von einem ERP-Implementierungsprojekt auf einmal in Angriff zu nehmen und die Lieferung zu beschleunigen, können mehrere Teams gleichzeitig an jeder Business-Funktionskomponente arbeiten. Der effektive Einsatz von skaliertem Scrum ermöglicht es mehreren Scrum-Teams, ihre Definition of Done so zu strukturieren, dass Integration, Regression, Performance, Sicherheit und Regressionstests auf der Sprint-Ebene statt auf der Release-

Ebene stattfinden. Ein Abgleich der Definition of Done ist erforderlich, da ERP-Systeme schwer zu korrigieren sind, wenn Probleme und Konflikte erst während des produktiven Einsatzes erkannt werden. Teams lernen, ihre Definition of Done sehr genau zu befolgen.

Wir haben auch festgestellt, dass Scrum gut mit solchen Projekten funktioniert, wenn sie sich auf die Bereitstellung von Business Intelligence für das Unternehmen konzentrieren. Visuelle Berichte von Daten haben eine klare, geschäftsorientierte Anforderung an die Anwender. Die Arbeit der Datenaufbereitung (zum Beispiel Aggregation und Manipulation) umfasst die Aufgaben, die die Ablieferung eines Berichts an den angegebenen Benutzer (zum Beispiel Führungskraft oder Manager) unterstützen.

Mehrjährige ERP-Implementierungen waren früher üblich, aber Unternehmen können im heutigen schnelllebigen Markt nicht so lange warten. Unternehmen brauchen schnellere und kostengünstigere Lösungen. Die Kunden wollen so schnell wie möglich einen Return on Investment mit verbesserter Kundenzufriedenheit erreichen.

Iteration, Überprüfung und Anpassung durch Scrum ermöglichen es, die Implementierung zu verkürzen.

Konvertierung von Legacy-Systemen

Ein Kunde von uns, eine gemeinnützige Organisation, hat ein umfangreiches Projekt zur Umstellung mehrerer Altsysteme auf eine neue ERP-Lösung übernommen. Historisch gesehen wäre dieser Job ein Albtraum gewesen. Eine treffende Beschreibung wäre gewesen, einen quadratischen Stift in ein rundes Loch zu hämmern oder Katzen zu hüten. Aber auch diese Euphemismen stellen im Vergleich zur Kombination verschiedener Datenmanagementsysteme einfache Herausforderungen dar.

Die Roadmap, die der Kunde aufgestellt hatte, sah eine 18-monatige Projektlaufzeit vor. Die Vision war eine End-to-End-Lösung für das Management globaler Immobilienressourcen. Das Unternehmen unternahm die folgenden Schritte:

1. Ein Team sicherte sich einen freien Bereich, den es als Scrum-Teamraum einrichtet; eine Wand wurde als Taskboard genutzt. Ein anderes Team richtete sich nebenan in direkter Nähe ein, was die Zusammenarbeit vereinfachte.
2. Mit Hilfe der Stakeholder ermittelte der Product Owner das erste Minimum Viable Product (MVP) – eine Reihe der gängigsten Prozesse im Zusammenhang mit Mietaktivitäten. Die Anwender könnten sofort von diesen gemeinsamen Prozessen profitieren.
3. Das Release wurde geplant, einschließlich einiger Sprints, wobei das neue MVP als Ziel diente. Das Team ging an die Arbeit.
4. Das Team wurde um Entwickler erweitert, die sich um die Integrationen von Drittanbieterlösungen kümmerten.
5. Von Anfang an wurden Shadowing und Paarprogrammierung genutzt, da das System für alle neu war. Die Teammitglieder wollten so schnell wie möglich eine Cross-Funktionalität erreichen.
Das Team bat auch ein Mitglied des traditionellen Deployment-Teams, Vollzeit im Scrum-Team mitzuarbeiten. Er war Teil des Scrum-Teams und lernte die Konfigurationsschritte während der Sprints kennen. Bei der Bereitstellung des Systems in der Produktionsumgebung verlief die Implementierung dank des gemeinsamen Wissens und der Vorabtests reibungslos.
6. Das Team fuhr fort, neue Funktionen aus der Roadmap zu integrieren, während die Kunden von den Vorteilen der neu entwickelten Features für Mietaktivitäten profitierten.

Vieles, was aus diesem ersten Release gelernt wurde, wurde in zukünftigen Releases in andere Aspekte des neuen Systems wie Planung, Bau, Betrieb und Wartung, integriert. Durch den Scrum-Prozess konnte das Team den Kunden schnell etwas an die Hand geben und sich danach auf die Features mit den nächsthöheren Prioritäten konzentrieren.

Das Dilemma zwischen Dienstleistung und Kontrolle

Der Wert einer IT-Gruppe ist enorm. Diese Gruppe hält den Informationsfluss im gesamten Unternehmen aufrecht, während gleichzeitig die von ihr verwalteten Technologiesysteme immer anspruchsvoller und komplexer werden. Erschwerend kommt hinzu, dass andere Softwareanwendungen und Cloud-Ressourcen leicht verfügbar sind und die Mitarbeiter ihre eigenen Geräte zur Arbeit mitbringen (BYOD) oder externe Anwendungen einbringen (BYOA).



BYOD steht für *bring your own device* und bedeutet, dass eigene Geräte, insbesondere Smartphones, Laptops und Tablets, verwendet werden, die nicht von der IT unterstützt werden. *BYOA* steht für *bring your own application* und bezieht sich auf die Beschaffung und Nutzung von Softwaretools und Abonnements.

BYOD und BYOA sind weit verbreitet. Eine aktuelle Umfrage von Syntonic ergab, dass 60% der Unternehmen mit mehr als 100 Mitarbeitern eine BYOD-freundliche Politik haben. Wenn Mitarbeiter das Gefühl haben, dass sie nicht über die Tools verfügen, die sie von der IT benötigen, ist es für sie selbstverständlich, nach Lösungen anderswo zu suchen, insbesondere wenn diese Lösungen über SaaS und in Form von mobilen Apps so leicht verfügbar sind. Dieser Kontrollverlust wird für IT-Teams zu einem wichtigen Problem, wenn nicht unterstützte

Geräte und Anwendungen Sicherheitsrisiken verursachen oder in unternehmensweit zugelassene Lösungen eingreifen.

Tatsache ist, dass ein ausgewogenes Verhältnis zwischen der Bereitstellung der Werkzeuge, die die Mitarbeiter für ihre Arbeit benötigen, und der Aufrechterhaltung von Kontrolle und Sicherheit erreicht werden kann. Nachfolgend finden Sie einige Beispiele, wie Sie Scrum nutzen können, um häufige Probleme in der IT zu bewältigen:

- ✓ Die IT-Abteilung kann die persönlichen Geräte und Anwendungen, die Mitarbeiter verwenden, nicht kontrollieren, aber sie kann den Datenzugriff kontrollieren. Verwenden Sie Scrum, um Anwendungen zu entwickeln, die sicher auf die Daten zugreifen. Das Scrum-Framework schränkt die Features ein, sodass das MVP schnell identifiziert und implementiert werden kann.
- ✓ Lassen Sie IT-Mitglieder als Stakeholder an den Sprint-Reviews anderer Scrum-Teams teilnehmen, sodass die IT-Abteilung aus erster Hand weiß, woran die Entwicklungsteams arbeiten und welche Tools sie verwenden. Fragen können vor Ort gestellt werden und auf Bedürfnisse kann fundiert reagiert werden.
- ✓ Laden Sie Abteilungsvertreter zu IT-Sprint-Review-Meetings ein, um ihr Feedback zur IT-Roadmap zu erhalten. Sie können lernen, was für die Organisation bei der Entwicklung von Produkten wichtig ist. Auf diese Weise können die geschäftlichen und die IT-Interessen aufeinander abgestimmt werden.

Behandeln Sie das Problem, nicht das Symptom, indem Sie den Mitarbeitern die Werkzeuge an die Hand geben, die sie brauchen, wenn sie sie brauchen, so wie sie sie brauchen. Scrum hilft Ihnen dabei.

Sicherheitsherausforderungen

Sie haben vielleicht schon einige dieser Begriffe gehört: *Bots*, *Würmer*, *Malware*, *Phishing* und *Sicherheitslücken*. Für manche Menschen ist das Internet ein fruchtbarer Boden für böswillige Aktivitäten. Größere Probleme im Zusammenhang mit Sicherheitslücken bei Daten machen viel zu oft Schlagzeilen. Im Folgenden finden Sie nur einige Beispiele von Unternehmen und Organisationen, bei denen in letzter Zeit Kundendaten nicht gut geschützt waren:

- ✓ August 2017: Equifax (143 Millionen Menschen betroffen)
- ✓ Juni 2017: Republican National Committee (200 Millionen Menschen betroffen)
- ✓ Februar 2017: PayPal/TIO (1,6 Millionen Menschen betroffen)
- ✓ Oktober 2016: Uber (57 Millionen Menschen betroffen)
- ✓ Februar 2015: Anthem Healthcare (78,8 Millionen Menschen betroffen)

Zu oft wird Sicherheit erst nachträglich berücksichtigt. Manager sehen dies manchmal als schmückendes Beiwerk und Nice-to-have-Feature an und sagen: »Wir kümmern uns darum, wenn es ein Problem wird.« Die Sicherheitsrisiken nehmen zu und müssen daher früh statt später angegangen werden. Im Folgenden finden Sie einige gängige Möglichkeiten, wie Scrum dazu beitragen kann, sicherheitsrelevante Probleme aufzudecken und Verbesserungen im Umgang mit Sicherheitsproblemen zu ermöglichen:

- ✓ Verfeinern Sie Ihre Definition of Done. Sowohl auf der Release- als auch auf der Sprint-Ebene ist die Definition of Done entscheidend. Legen Sie fest, welche Sicherheitsanforderungen in jedem Zeitfenster erfüllt werden sollen. Durch das Ergänzen jeder Anforderung um eine sicherheitsrelevante Aufgabe, die Teil der Definition of Done ist, können die Kosten für die Verringerung des Sicherheitsrisikos über die gesamte Projektlaufzeit verteilt

werden. Stellen Sie die Sicherheit ab dem ersten geeigneten Moment in den Vordergrund.

- ✓ Unabhängig von Ihrer Rolle sollten Sie als Scrum-Teammitglied immer dann, wenn die Sicherheit nicht den von Ihnen gewünschten Stellenwert einnimmt, dies bei der nächsten Retrospektive ansprechen. Die Stakeholder des Unternehmens und/oder der Product Owner müssen möglicherweise in diesen Fragen geschult werden. Identifizieren Sie die Probleme und nehmen Sie sie in das Produkt-Backlog auf.
- ✓ Lassen Sie IT-Teammitglieder an den Sprint-Reviews des Projektteams teilnehmen und geben Sie Feedback zu allen Fragen rund um das Thema Sicherheit.
- ✓ Automatisieren Sie Sicherheitstests so weit wie möglich. Erwägen Sie die Automatisierung von Penetrationstests, Cross-Site-Scripting und Schwachstellen-Scans.
- ✓ Erwägen Sie die Gründung einer Sicherheitsgilde innerhalb der Organisation, die das Bewusstsein und die Entwicklung von Fähigkeiten in Bezug auf gemeinsame Sicherheitspraktiken und -technologien in der gesamten Organisation fördert.

Gilden

Gilden sind Gruppen von Menschen in einer Organisation, die Wissen, Werkzeuge und Praktiken austauschen. Eine Gilde ist wie eine Community of Practice (praxisbezogene Gemeinschaft), in der Menschen regelmäßig gemeinsam lernen und Wissen über gemeinsame Interessen oder Praktiken austauschen. Beispiele für Gilden sind Sicherheits-, Systemverwaltungs-, Datenbank-, Qualitäts- und Scrum-Teams.

Gilden sollten für jeden offen sein, der daran interessiert ist, andere zu betreuen und anzulernen oder der selbst mehr

lernen möchte, auch für diejenigen, die neu in der Fertigkeit oder Praxis sind.

Die Gilden sollten sich oft genug treffen, um Wissen aufzubauen und das Lern- und Qualifikationsniveau in einer Organisation zu erhöhen.

Stellen Sie sicher, dass mindestens eine Person in jedem Scrum-Team Mitglied der Sicherheitsgilde wird. Dies wird einen großen Beitrag dazu leisten, dass die Sicherheitspraktiken einheitlich sind und das Bewusstsein in allen Teams gestärkt wird.

Die Babyboomer-Ruhestands-Lücke

Die Nachwuchskräfte, die an die Stelle der Babyboomer treten, die in Rente gehen, haben eine andere Generationsmentalität als ihre Vorgänger. Persönliches Wachstum und Ehrgeiz besitzen oft einen größeren Stellenwert, als während der gesamten beruflichen Laufbahn bei der gleichen Firma zu bleiben. Das bedeutet, dass die langfristigen Mitarbeiter, auf die Sie sich immer verlassen können, immer weniger werden, was operative Herausforderungen mit sich bringen kann. Die folgenden agilen Praktiken können Ihnen helfen, neue Mitarbeiter zu binden und zu fördern:

- ✓ Erstellen Sie ein Mentoring-Programm für diejenigen, die in den Ruhestand gehen und deren jüngere Nachfolger. Dieses Programm steigert die Effizienz beim Wissenstransfer sowie die Akzeptanz und Loyalität.
- ✓ Geben Sie neuen IT-Mitarbeitern Projekte und Verantwortlichkeiten außerhalb ihrer Komfortzone. Lassen Sie sie an Fortbildungen teilnehmen und neue Dinge lernen, um ihr mentales und emotionales Engagement zu steigern. Gilden (siehe nebenstehende Kasten) erleichtern diesen Prozess

durch die Bereitstellung von Expertenpools in jedem Bereich. Senior- und Junior-Entwickler erweitern ihre Fähigkeiten durch die Zusammenarbeit.

- ✓ Nutzen Sie Techniken des crossfunktionalen Arbeitens, wie beispielsweise Paarbildung und Shadowing (siehe [Kapitel 4](#)).



Diese Aktivitäten können auch dazu beitragen, dass die Mitarbeiter längerfristig weiterwachsen. Menschen, die sich auf dem Weg befinden, die Organisation zu verlassen, widersetzen sich manchmal dem Wandel, da sie die letzten Jahre einfach nur hinter sich bringen wollen, bevor sie in Rente gehen. Gilden können eine Möglichkeit bieten, deren Fachwissen und ihren immensen Wert zu nutzen. Holen Sie diese Mitarbeiter aus ihren Komfortzonen heraus, indem Sie Workshops, Lunch-and-Learning-Sitzungen und andere Arten von Schulungen organisieren.

Ermutigen Sie neue Mitarbeiter zu wachsen und befähigen Sie sie dann mit den Mitteln, dies zu tun. Während Langeweile und Routine die Menschen motivieren können, sich dorthin zu begeben, wo das Gras noch grün ist, führen Herausforderungen, Wachstumschancen, Anerkennung und Sinnstiftung dazu, dass Mitarbeiter bleiben und sich engagieren.

Gewinn- und Verlustpotenzial

Der Geschäftswert der IT ist enorm. Ihr Wert muss dem Rest der Organisation regelmäßig mitgeteilt und demonstriert werden. Manchmal wird dieser Wert jedoch angesichts der Probleme übersehen, die die Technologie mit sich bringt. Eine große Chance liegt darin, den Wert der IT für alle Beteiligten klar zu kommunizieren – auch wenn diese Vorteile lediglich darin bestehen, Risiken abzumildern.



Die Schlüsselfrage lautet: »Verbessert diese Aufgabe oder Tätigkeit die Kernprioritäten unserer Organisation?« Wenn ja, behalten Sie sie bei und priorisieren Sie sie. Wenn nicht, finden Sie heraus, wie Sie diese Funktion beseitigen und sich auf unternehmenskritische Aufgaben konzentrieren können.

Die Erhöhung der Transparenz für das Unternehmen (durch Artefakte wie das priorisierten Produkt-Backlog der IT und Inkremente, die bei Sprint-Reviews demonstriert wurden) gibt der gesamten Organisation Aufschluss darüber, wie die IT arbeitet, um Engpässe im Informations- und Kommunikationsfluss zu beseitigen, und liefert die Werkzeuge zur Steigerung der Produktivität. Wenn die gesamte Organisation den Wert der IT versteht, werden die Interaktionen zwischen der IT und anderen Abteilungen kollaborativer und die Lösungen der IT werden zu eindeutigen Wegbegleitern der Unternehmensstrategie.

Genau wie ein Scrum-Team, das seine Velocity von 25 auf 27 erhöht, braucht es nur eine kleine prozentuale Einsparung an Betriebskosten, um im Laufe der Zeit eine große Ergebnisverbesserung zu erzielen.

Energieeffizienz ist heutzutage ein heißes Eisen und ein Beispiel dafür, wie viel Geld man mit der Zeit sparen kann. Eine breite Palette von Werkzeugen steht zur Verfügung, um Energiekosten zu sparen. Durch eine bessere Überwachung und Anwendung von energiesparenden Produkten können einige Prozentpunkte an Kosten eingespart werden; dies ermöglicht es der IT, mehr Gewinn zu erzielen.



IBM schätzt, dass IT- und Energiekosten zusammen bis zu 75 Prozent der Betriebskosten und bis zu 60 Prozent der Investitionen in einem Unternehmen ausmachen.

Es wird geschätzt, dass ein 2.300 Quadratmeter großes Rechenzentrum jedes Jahr mehr als 4 Millionen Dollar an Energie verbraucht. Zwar lassen sich diese Kosten nicht sofort eliminieren, aber durch inkrementelle Schritte der Überprüfung

und Anpassung können die Kosten schrittweise und angemessen gesenkt werden. Bei Kosten in Höhe von 4 Millionen Dollar macht auch eine kleine prozentuale Abnahme viel aus.

Innovation versus Stabilität

Unternehmen verlassen sich auf den IT-Betrieb, um Stabilität, Leistung und Verfügbarkeit zu gewährleisten, doch die IT muss immer innovativ sein, was bedeutet, dass sie sich schnell und häufig ändern muss. Stabilität und Veränderungen stehen miteinander im Konflikt. Dieser Konflikt wird durch eine engere Zusammenarbeit zwischen den Leuten, die sich um den IT-Betrieb kümmern (Operations) und den Entwicklungsteams gelöst. Anstatt neue Technologien zu entwickeln und sie über den Zaun zu werfen, damit die Abteilung Systemadministration sie unterstützen können, und umgekehrt, baut die operative Seite des Unternehmens Sandkästen oder Standards, innerhalb derer Entwicklungsteams bauen können.

Jedes Mal, wenn Entwicklungsteams im Innovationsprozess Codeänderungen vornehmen, können sie sich darauf verlassen, dass diese Codeänderungen innerhalb der zwischen Betrieb und Entwicklung vereinbarten Standards vorgenommen werden. Änderungen an Datenbankstrukturen oder -designs innerhalb von operativen Standards verhindern die Risiken, die mit den gleichen Änderungen in Entwicklungssilos verbunden sind, die mit wenig oder gar keiner Zusammenarbeit mit Operations vorgenommen wurden.

Diese Standards werden in die Definition of Done eines Teams aufgenommen. Die IT-Abteilung kann sich darauf verlassen, dass die Teams bei jedem Sprint und bei jeder Anforderung innerhalb dieses gesteckten Rahmens bleiben. Wenn Änderungen am »Sandkasten« vorgenommen werden müssen, beraten Entwicklung und Operations gemeinsam darüber und legen die neuen Grenzen fest.

Die Notwendigkeit einer besseren Koordination ist entscheidend, besonders in der Softwareentwicklung, wo DevOps ins Spiel kommt.

DevOps

Entwicklung und Betrieb (DevOps) ist eine zunehmend eingesetzte Lösung für eine Lücke in der Entwicklung von Softwareanwendungen. [Abbildung 12.1](#) zeigt, wie DevOps die Herausforderungen der Koordination angeht.

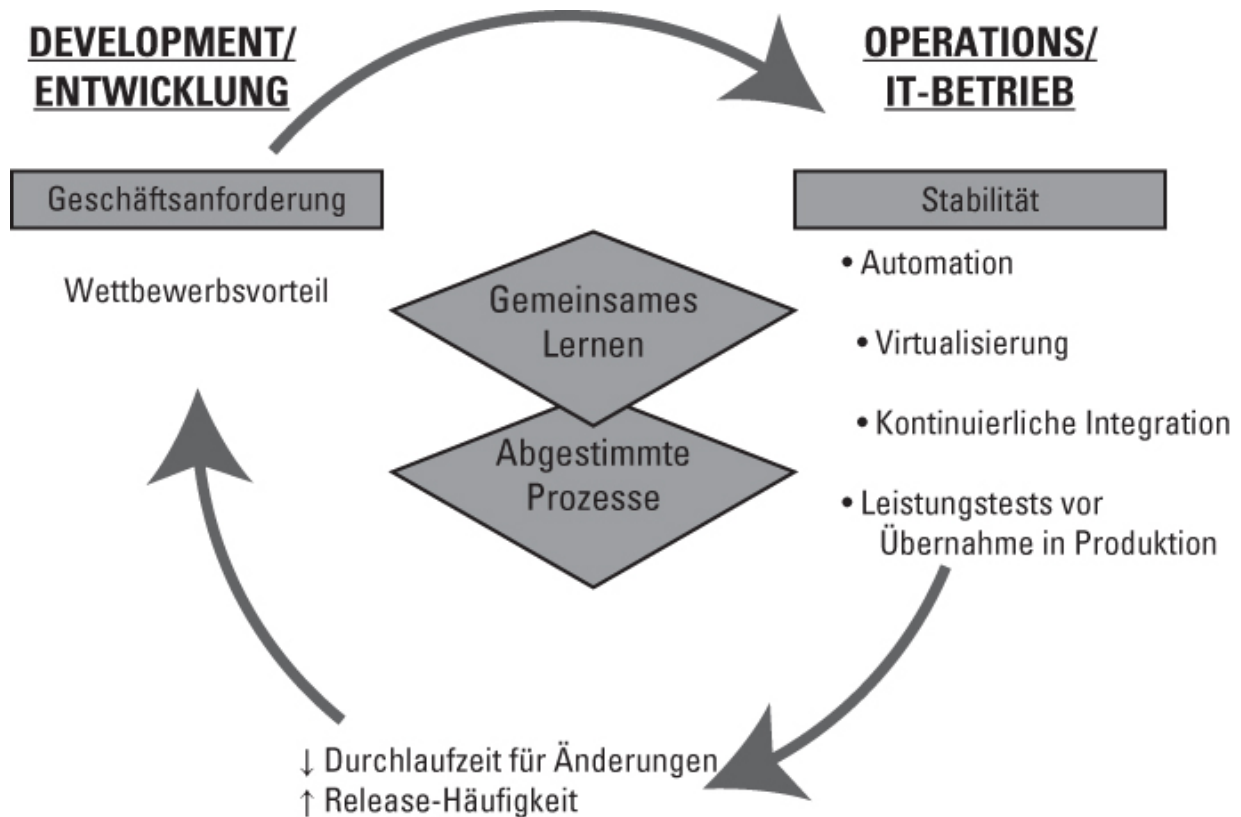


Abbildung 12.1: DevOps wägt eine Geschäftsanforderung gegen Stabilität ab.

Traditionelle IT-Aufgaben werden auf DevOps-Teammitglieder verlagert, wodurch nicht nur die IT von Aufgaben entlastet wird, sondern außerdem Entwicklungsteams mit Dev-Ops-Teammitgliedern in die Lage versetzt werden, ihre Produktentwicklung mit weniger IT-Abhängigkeiten produktiv einsetzbar zu machen. Virtualisierte Rechenzentren machen dies zunehmend möglich, da die IT die Plattform verwaltet, was die

Erstellung neuer virtualisierter Umgebungen durch jemanden mit DevOps-Fähigkeiten in einem Projekt-Scrum-Team ermöglicht.

IT-Abhängigkeiten können nun innerhalb des Scrum-Teams aufgelöst werden. Innerhalb der Sprints lassen sich höhere Qualität bei einer schnelleren Fertigstellung erreichen.

Wartung

Nach der Bereitstellung einer Anwendung leisten die Wartungsteams Support. Diese Unterstützung kann in Form der Beantwortung von Supportanfragen, der Behebung von Fehlern und der Implementierung kleinerer Verbesserungen zur Behebung von Funktionslücken in der Produktion erfolgen.

Wenn sich Scrum-Teams auf die Projektarbeit konzentrieren, egal ob Software oder IT, ist die Behebung von operativen Problemen störend. Projekt- und Wartungsarbeiten sind zwei verschiedene Paar Schuhe. Sie erfordern von den Entwicklern andere Formen von Arbeit und Aufwand, was auch eine andere Kadenz mit sich bringt. Wenn ein Projekt-Scrum-Team Betriebs- oder Wartungsaufgaben wahrnimmt, wird die Entwicklung neuer Funktionen häufig unterbrochen. Dies wirkt sich negativ auf die Release-Zeitpläne der Funktionen aus, die für den Kunden von geschäftlichem Wert sind. (Siehe [Kapitel 13](#) für weitere Informationen über die Kosten von Verzögerungen durch den Einsatz eines Entwicklungsteams für mehrere Projekte, das sogenannte Thrashing.)

Indem die Wartungs- und Neuentwicklungsarbeiten der Scrum-Teams getrennt werden, lässt sich die Wartung straffen und die Neuentwicklung kontinuierlich vorantreiben. Wir empfehlen eine Wartungs-Scrum-Teamstruktur, die diese beiden Funktionen trennt, sodass Unterbrechungen minimiert werden, ohne den Overhead zu erhöhen (siehe [Abbildung 12.2](#) und [Abbildung 12.3](#)).

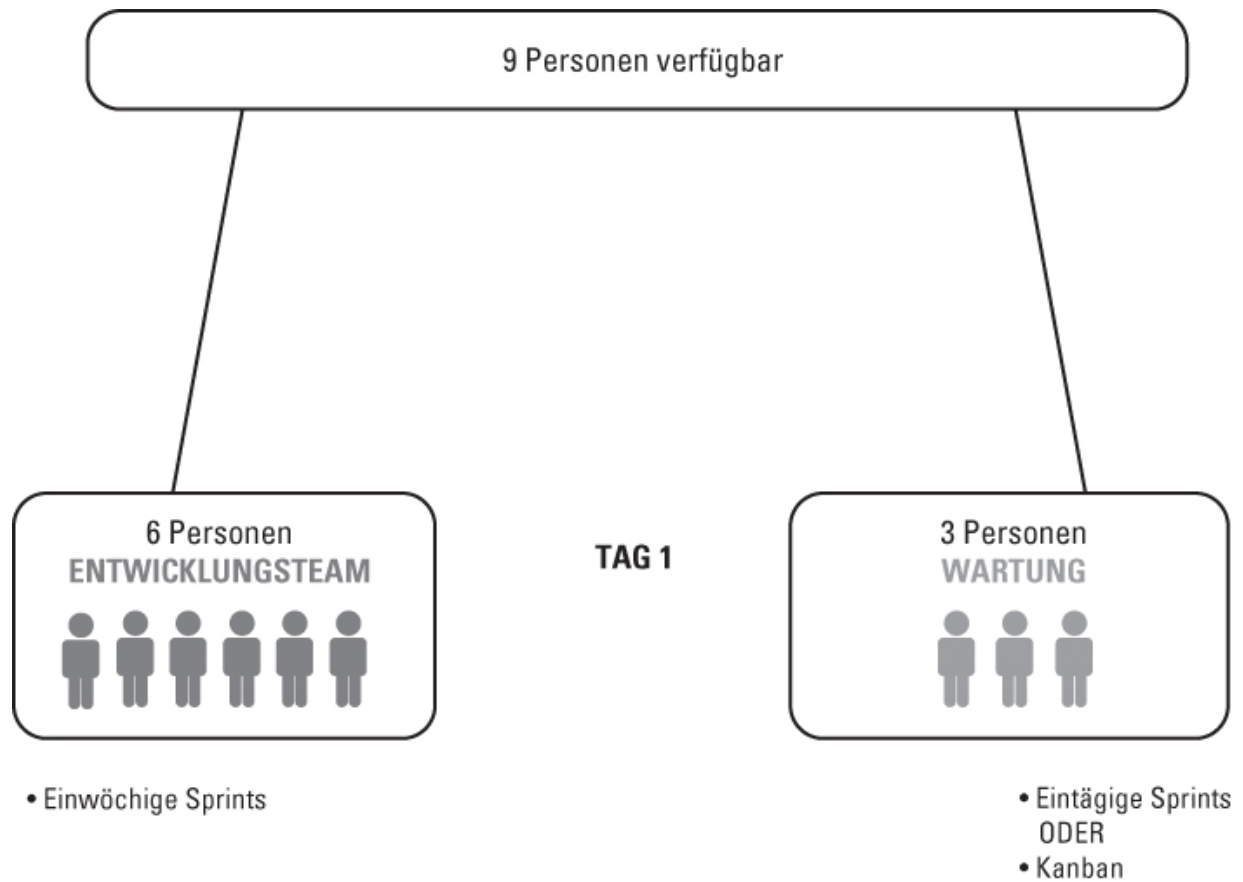


Abbildung 12.2: Eine gängige Praxis für die Trennung der Instandhaltungs- und Entwicklungsaufgaben in einem Scrum-Team

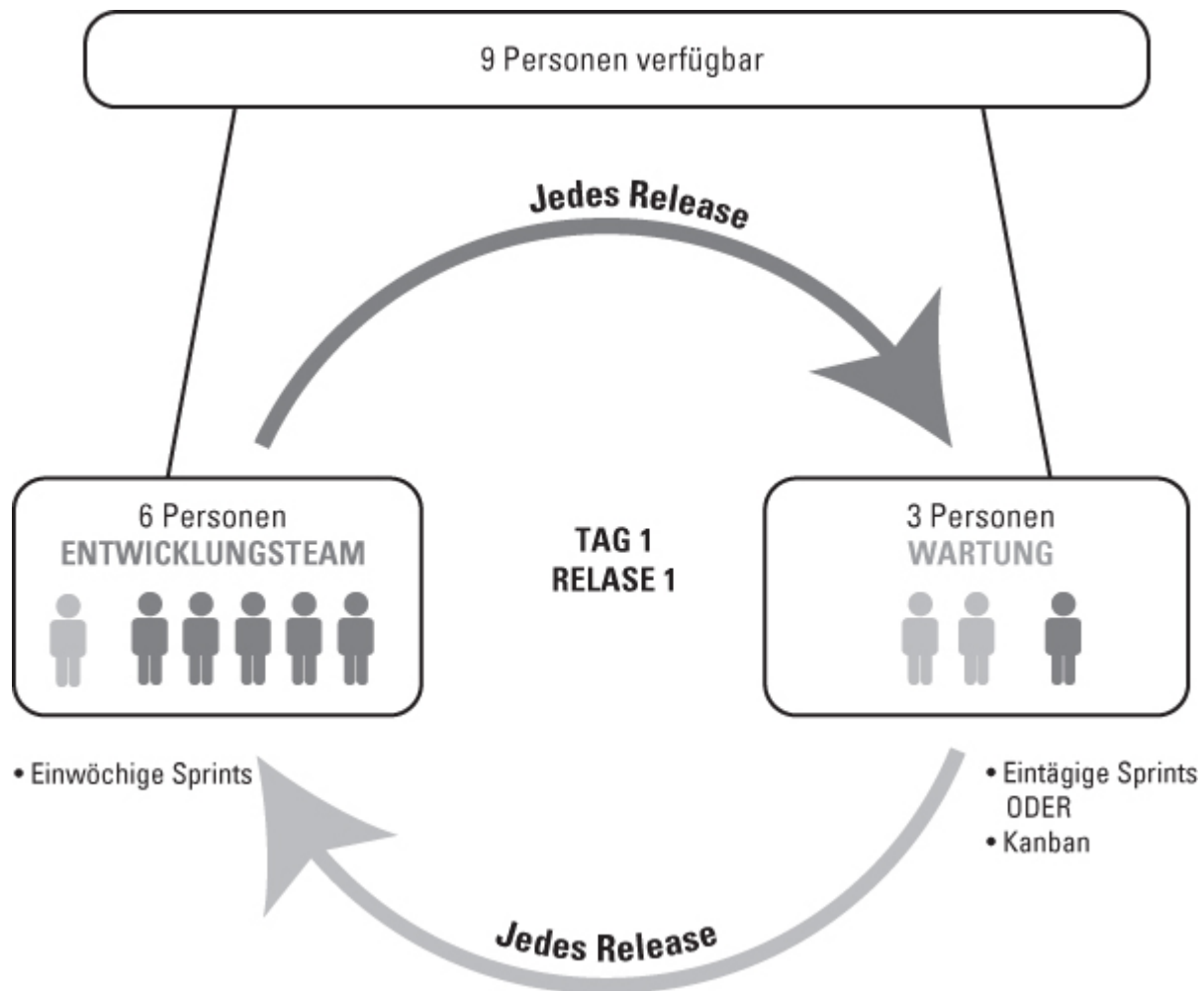


Abbildung 12.3: Die Teammitglieder werden in angemessenen Zeitabständen zwischen den Teams gewechselt, um die gegenseitige Befruchtung durch Wissen und Fähigkeiten zu gewährleisten.

Kanban innerhalb einer Scrum-Struktur

Das Kanban-Konzept erörtern wir in [Kapitel 6](#). Einige Scrum- und agile Teams nutzen ein Kanban-Board, um ihren Workflow zu visualisieren. So funktioniert der Prozess in [Abbildung 12.2](#) und [Abbildung 12.3](#):

- ✓ Zu Beginn eines neuen Release-Zyklus wird das Entwicklungsteam aufgeteilt. Eine kleine Untergruppe des

Teams, das genügend Erfahrung und Wissen aufweist, um das Produkt effektiv zu warten, wird zum Wartungsteam.

Wie in [Abbildung 12.2](#) gezeigt, können Sie, wenn Sie ein neunköpfiges Entwicklungsteam haben, drei Entwickler nehmen, die dann ein separates Wartungs-Scrum-Team bilden, sodass sechs Entwickler im bestehenden Scrum-Team neue Produkte entwickeln können. Ihre Teamgröße kann variieren. Auch können für das Wartungsteam weniger oder mehr als drei Mitglieder benötigt werden. (Obwohl das bestehende Scrum-Team jetzt kleiner ist, wird es nicht mehr zwischen der Entwicklung neuer Produkte und den Wartungsarbeiten hin und her wechseln. Weniger Leute bedeuten nicht weniger Leistung. Eigentlich sollte das Team durch den Wegfall des Thrashing in der Lage sein, seine Leistung deutlich zu steigern.)

- ✓ Das Wartungsteam führt eintägige Sprints oder Kanbans durch, um auf schnell wechselnde Anforderungen (oft täglich wechselnd) zu reagieren. Jeden Morgen trifft sich das Team und plant die Prioritäten für den Tag. Es führt die im Verlaufe des Tages die geplanten Arbeiten aus. Am Ende des Tages überprüft das Team, was es erreicht hat, und klärt, ob die Arbeit freigegeben werden kann. (Das Release kann jederzeit und muss nicht täglich erfolgen.)
- ✓ Wartungsteams können kleiner sein als neue Entwicklungsteams. Der Product Owner und der Scrum Master des Wartungsteams sind oft die gleichen wie für das Projektteam. Die Probleme des Hin- und Herwechsels sind minimal, falls sie überhaupt auftreten, da beide an einer einzigen Anwendung arbeiten.
- ✓ Bei jedem Release des Projektteams (oder wenn ein Team häufig und in angemessenen Abständen releast, wie beispielsweise alle 90 Tage), wechselt ein Mitglied des Entwicklungsteams in das Wartungsteam. Es wird dort vollzeitbeschäftigtes, engagiertes Mitglied, das den notwendigen Wissenstransfer zur Unterstützung des Release bereitstellt. Auch hier wird das Thrashing minimiert, da es auf

Release-Ebene durchgeführt wird und die Teammitglieder weiterhin am gleichen Produkt arbeiten. (In [Kapitel 13](#) finden Sie weitere Informationen über die Minimierung des Thrashing, um so die Stabilität und Rentabilität zu maximieren.)

- ✓ Bei jedem Release wechselt ein Mitglied des Wartungsteams beziehungsweise operativen Teams in das Entwicklungsteam und wird dort ein Vollzeitmitglied dieses Teams.
- ✓ Jeder führt letztendlich sowohl neue Entwicklungs- als auch operative Arbeiten durch.
- ✓ Die Produktkenntnisse werden zwischen den beiden Teams geteilt; dies ist ein entscheidender Vorteil.
- ✓ Die Crossfunktionalität nimmt durch das Rotationsprinzip ebenfalls zu.

Wartungsteams führen in der Regel eintägige Sprints oder Kanbans durch; sie dauern nicht wie bei Projekt-Scrum-Teams eine Woche oder länger. Wir empfehlen meistens den Einsatz von eintägigen Sprints anstelle von Kanban, da wir die erzwungenen Produkt- und Prozess-Berührungspunkte mögen, die sich durch Sprint-Review und Retrospektive von Scrum ergeben; Kanban kann ebenfalls effektiv sein, wenn es innerhalb einer Scrum-Struktur durchgeführt wird.

Die Kanban-Praxis der Visualisierung von Arbeitsabläufen passt genau zum Scrum. Außerdem hilft die Begrenzung der laufenden Arbeiten den Teams dabei, sich zu konzentrieren und ihre Arbeit zu erledigen.

Auch für Teams, die mit Kanban arbeiten, ist es sinnvoll, den Ablauf und die Schätzung auf Basis der Durchlaufzeit (verstrichene Zeit von der Anfrage bis zur Lieferung) und der Entwicklungsdauer (verstrichene Zeit vom Arbeitsbeginn bis zur Lieferung) zu steuern. Die Durchlaufzeit und die Entwicklungsdauer können für Scrum-Teams auch bei der Planung und der Kommunikation mit Stakeholdern hilfreich sein.



Die *Durchlaufzeit* (*lead time*) ist die Zeitspanne zwischen dem Eingang einer Anfrage und der Lieferung des fertigen Produkts. Die *Entwicklungsdauer* (*cycle time*) ist die Zeit zwischen dem Start und der Auslieferung des Produkts. Daher ist die Durchlaufzeit das, worum es den Stakeholdern oft geht.

Teams, die Kanban verwenden, kennen auch die Theorie der Einschränkungen, was bedeutet, dass das System des Teams durch mehrere Einschränkungen begrenzt ist. Bei Scrum ist die Velocity eine Einschränkung und der Scrum Master sucht ständig nach Möglichkeiten, organisatorische Hemmnisse zu entfernen, um die Velocity zu erhöhen und diese Einschränkung zu reduzieren.

Kanban fehlt es an einem erzwungenen Feedback-Mechanismus. Für Teams, die Kanban nutzen, ist es sehr einfach, diese Rückkopplungsschleife zu vermeiden und ständig untereinander und mit den Stakeholdern zu kommunizieren, indem sie sagen: »Ja, wir arbeiten gerade an Dingen und wir machen Fortschritte.« Das Kanban-Board ist sichtbar, aber möglicherweise sind die Anstrengungen für die Überprüfung und Anpassung von sowohl Produkt als auch Prozess nicht effektiv. Scrum stellt Kanban für diesen Prozess eine Struktur zur Verfügung.

Falls Sie sich bei eintägigen Sprints über den Overhead Sorgen machen, so ist dies nicht nötig. Ein eintägiger Sprint ist ein Fünftel eines einwöchigen Sprints. Das ist unglaublich effizient, der Zeitaufwand ist minimal. Hier ist die Aufteilung in Minuten:

- ✓ Tägliche Sprint-Planungsrunden dauern maximal 25 Minuten. Stellen Sie sicher, dass Sie die Stakeholder, die die Anforderung angefordert haben, mit einbeziehen, um zu klären, was das Entwicklungsteam über die Anforderungen wissen muss.
- ✓ Das tägliche Sprint-Review dauert maximal 15 Minuten.

- ✓ Die tägliche Sprint-Retrospektive dauert maximal 10 Minuten. Achten Sie darauf, jeden Tag zu überprüfen und anzupassen, auch dann, wenn es einen Notfall gibt und jeder einfach nur nach Hause will.
- ✓ Führen Sie außerdem Retrospektiven auf Makroebene durch, und zwar synchron zur Sprint-Retrospektive des Entwicklungsteams, da diese Praxis eine umfangreichere Überprüfung ermöglicht.
- ✓ Daily Scrums sind nicht notwendig, da Ihre Sprint-Planung am Morgen für die Koordination und Synchronisation der Prioritäten sorgt. Hindernisse müssen jedoch im Laufe des Tages angesprochen werden, sodass der Scrum Master wachsam sein und bekannte oder potenzielle Hindernisse im Laufe des Tages proaktiv verfolgen muss.

Das Entscheidende bei eintägigen Sprints ist die Aufteilung der Anfragen, sodass sie in einen Tag passen, was Übung erfordert. Am Anfang haben unsere Teams einwöchige Batch-Releases durchgeführt. Als sie zum ersten Mal zu ihrer neuen Ein-Tages-Kadenz übergingen, nahm die Anzahl der bis zum Ende der Woche abgeschlossenen Korrekturen ab. Die Kundenzufriedenheit stieg jedoch weiter an, da die Kunden nicht mehr eine Woche warten mussten, sondern täglich etwas erhielten. Warten und Zufriedenheit stehen zueinander im Widerspruch.

Kapitel 13

Portfoliomanagement

IN DIESEM KAPITEL

Portfolios mit Scrum priorisieren

Scrum auf unternehmerische Vorhaben anwenden

Scrum mit vertikalem Schnitt, SAFe und LeSS skalieren

Nichts ist so sinnlos, als etwas effizient zu tun, was man überhaupt nicht tun sollte.

Peter Drucker

Portfoliomanagement ist die gleichzeitige Koordination, Integration, Verwaltung, Priorisierung und Steuerung mehrerer Projekte über das gesamte Produkt oder die gesamte Organisation hinweg.

Die größte Problem beim Portfoliomanagement ist das Versagen der Führungsebene, Projekte richtig zu priorisieren und Talente angemessen zuzuordnen. Dieses Versagen der Führungskräfte wird dadurch überdeckt, dass Mitarbeiter gleichzeitig an mehreren Projekten arbeiten oder zwischen verschiedenen Projekten wechseln. Die Kommunikation scheitert, Prioritäten werden fallen gelassen und das Quietschende-Rad-Syndrom nimmt überhand (analog zur Redensart »Das Rad, das am lautesten quietscht, bekommt das meiste Fett« erhält der lauteste Stakeholder die meiste Aufmerksamkeit und Ressourcen).

Sie haben nur eine Möglichkeit, mit der Situation umzugehen, dass Sie mehr Projekte haben als Talente: Priorisieren Sie Effektivität vor Effizienz. Wenn eine Organisation hocheffizient ist, aber an den falschen Funktionen arbeitet, wie effektiv wird diese

Organisation sein? Es ist viel wichtiger, effektiv zu sein. Arbeiten Sie nur an den am Projekten mit den höchsten Prioritäten und in diesen Projekten ebenfalls nur an den Features mit den höchsten Prioritäten. Mit diesem Modell erzielen Sie Profitabilität.

In diesem Kapitel gehen wir auf die wichtigsten Herausforderungen im Portfoliomanagement ein und erörtern die verfügbaren Scrum-Lösungen. Wir schauen uns an, warum Lean Startup gut zu Scrum passt und wie man große Multi-Team-Projekte mit Scrum skalieren kann.

Herausforderungen im Portfoliomanagement

Die folgenden vier zentralen Herausforderungen verhindern ein gutes Portfoliomanagement, wenn sie nicht effektiv bewältigt werden:

- ✓ Mitarbeiterzuordnung und Priorisierung
- ✓ Abhängigkeiten und Fragmentierung
- ✓ Trennung zwischen Projekten und Geschäftszielen
- ✓ Verantwortlichkeit am falschen Platz

Mitarbeiterzuordnung und Priorisierung

Insbesondere der Mangel an Zeit, Geld und Leuten stellt die größte Einschränkung für das Portfoliomanagement dar. Der Mangel an Ressourcen zwingt Führungskräfte dazu, effektiv Prioritäten zu setzen, den organisatorischen Aufwand für Talente zu reduzieren oder die Belastung auf eine Ebene zu verlagern, die nicht in der Lage ist, sich zu wehren. Leider entscheiden sich viele Führungskräfte für Letzteres. Oft ist es so, dass die lauteste Stakeholder Stimme diejenige ist, die das Geschehen lenkt.

Unternehmensstrategie, nicht Launen oder die lauteste Stimme, sollte die Priorisierung vorantreiben.

Effektiv zu priorisieren bedeutet, nur die Projekte mit dem höchsten Wert und dem höchsten Risiko auszuwählen und jeweils ein Projekt an stabile Teams zu übergeben. Wenn Sie nur ein Scrum-Team haben, dann weisen Sie ihm ein Projekt nach dem anderen. Wenn Sie fünf Teams haben, dann weisen Sie jedem Team nur ein Projekt auf einmal zu. Wenn ein Projekt für ein Team zu umfangreich ist, lassen Sie so viele Teams wie nötig auf das Projekt mit der höchsten Priorität ausschwärmen, bis das Projekt fertiggestellt ist; anschließend übertragen Sie den Teams das nächste Projekt.



Rückwärts rennen ist ein Konzept, das wir in Schulungen verwenden. Es beschreibt, dass zu viel Zeit mit den falschen Dingen verbracht wird. Wenn Sie nicht effektiv sind, fahren Sie im Rückwärtsgang. Wenn Sie die Menschen dazu drängen, mehr zu arbeiten, Überstunden zu machen und sich über ihre Möglichkeiten hinaus zu engagieren, *erhöhen* Sie die Anzahl der Fehler, was die zukünftige Arbeit und die Kosten erhöht. Marks Webinar zu diesem Thema finden Sie unter <https://platinumedge.com/blog/video-mark-laytons-racing-reverse-presentation>.

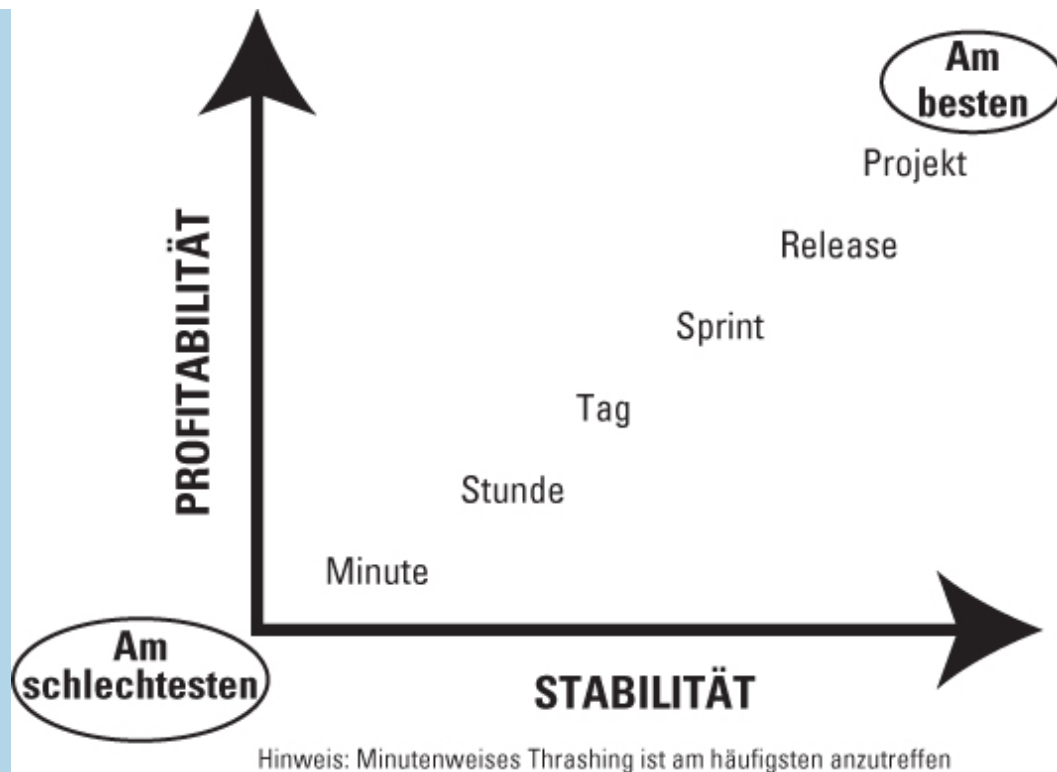
Das Rätsel der Projektpriorisierung kann mit einer Frage gelöst werden: Wenn die Priorität dieses Projekts nicht groß genug ist, um dafür engagierte Talente zur Verfügung zu haben, ist es dann gerade jetzt notwendig? Sollte die Antwort »nein« lauten, ist es am besten, zuerst die Projekte mit der höchsten Priorität abzuschließen und dann diese Frage erneut zu stellen.

Eine Hierarchie des Thrashing

Unabhängig davon, in welcher Form Thrashing in Ihrer Organisation auftritt, verwenden Sie Scrum, um die Auswirkungen zu minimieren. Denken Sie daran – je stabiler Ihr Team ist, desto höher ist Ihre Rentabilität. Im Folgenden

werden die verschiedenen Gesichter des Thrashing beschrieben:

- ✓ Das Team ist während der Projektlaufzeit stabil: Hier ist das Projekt stabil und ohne Thrashing, da das Team für das gesamte Projekt solange fokussiert ist, bis $TK + AK > W$ (siehe [Kapitel 5](#)). Dies ist die effizienteste Struktur, mit der schnell die bestmögliche Qualität produziert und die Rentabilität maximiert werden kann.
- ✓ Das Team ist für die Dauer der Releases stabil: Das Team konzentriert sich auf Projekt A, bis es für die Kunden etwas Freigabefähiges hat. Es wechselt dann zu Projekt B, bis es etwas für Kunden freigegeben haben, dann wechselt es zu Projekt C und so weiter. Dieser Ansatz ist schlechter als Stabilität für das gesamte Projekt.
- ✓ Das Team ist für die Dauer des Sprints stabil: Das Team konzentriert sich für den gesamten Sprint auf Projekt A. Dann wechselt es für einen Sprint gleicher Länge zu Projekt B, dann zu Projekt C und so weiter. Dieser Ansatz ist schlechter als die Stabilität für ein Release.
- ✓ Das Team ist für die Dauer des Tages stabil: Das Team arbeitet am Montag an Projekt A. Am Dienstag wechselt es dann zu Projekt B, am Mittwoch zu Projekt C und so weiter. Dieser Ansatz ist schlechter als die Stabilität für einen Sprint.
- ✓ Das Team ist nur für Stundenblöcke stabil: Das Team arbeitet jeden Morgen an Projekt A, am Nachmittag an Projekt B und so weiter. Dieser Ansatz ist schlechter als die Stabilität für den gesamten Tag.



- ✓ Thrashing des Teams Minute für Minute: Dies ist der gängigste Portfoliomanagement-Ansatz. Einzelne Teammitglieder beginnen mit der Arbeit an Projekt A; 45 Minuten später kommt es zu einem Notfall an Projekt B, der sie bis zum Mittagessen in Anspruch nimmt. Nach dem Mittagessen versuchen sie, Projekt A neu zu starten, aber es stellt sich etwas über Projekt C heraus, das den Nachmittag in Anspruch nimmt. Am nächsten Tag sind sie wegen der mentalen Remobilisierungszeit wieder am Anfang des Projekts A. Der Prozess geht weiter, bis ein Projekt im Einflussbereich einer einflussreichen Führungskraft so verspätet ist, dass das Team gezwungen ist, seine Abende und Wochenenden damit zu verbringen, genau an diesem Projekt zu arbeiten. Das ist der schlechteste Ansatz.

Abhängigkeiten und Fragmentierung

Bei gleichzeitig laufenden Projekten entstehen Engpässe durch Abhängigkeitsprobleme und Verzögerungen aufgrund von Schwierigkeiten bei der Integration. Im traditionellen Projektmanagement werden Menschen, Geld und Ausrüstung aufgeteilt. Es ist nicht ungewöhnlich, dass Manager Mitarbeiter auf mehrere Projekte verteilen (50 Prozent hier, 40 Prozent dort und die letzten 10 Prozent bei einem anderen Projekt), nur um das Gefühl zu haben, dass sie alle verfügbaren Ressourcen nutzen und die Organisation so effizient wie möglich wird.



Menschen verlieren mindestens 30 Prozent der Zeit bei der kognitiven Mobilisierung und Demobilisierung, die mit dem Hin- und Herschalten zwischen verschiedenen Aufgaben einhergeht. Wenn Mitarbeiter ihre Zeit auf mehrere Projekte verteilen, schadet das ihrer Produktivität.

In der Vergangenheit haben Projektmanager versucht, alles gleichzeitig zu tun, weil sie unter dem Druck der Geschäftsinhaber stehen, über mehrere Bereiche hinweg zu liefern. Dies führt dazu, dass sie gleichzeitig verschiedenen Zielen hinterherlaufen, ohne zu trennen und zu priorisieren, was zuerst getan werden muss. Abhängigkeitsprobleme führen zu Knoten in Projekten, da das Projekt mit den zentralen Abhängigkeiten nicht zuerst ausgeführt wurde.

Trennung zwischen Projekten und Geschäftszielen

Die meisten Projekte beginnen im Einklang mit den Geschäftszielen, aber im Laufe der Zeit entwickeln die Manager weiterhin Produktideen und sowie Hypothesen über die Richtung, die das Produkt einschlagen könnte. Ohne konsequente Feedbackschleifen mit Kunden und Stakeholdern und ohne tägliche Abstimmung mit dem Entwicklungsteam weitet sich der Projektumfang leider in alle möglichen Richtungen aus. Je mehr Zeit seit der ersten Planung vergangen ist, desto mehr bricht das

gemeinsame Verständnis der Ziele zusammen, bis niemand mehr die ursprünglichen Ziele kennt.

Diese schleichende Zunahme des Projektumfang spiegelt den Kern des Problems wider, und zwar die fehlende Prioritätensetzung. Mit Scrum erfolgen Priorisierung und Neupriorisierung kontinuierlich während des gesamten Projekts auf den Ebenen Roadmap, Release, Sprint und Daily Scrum. Im Produkt-Backlog befinden sich immer die Elemente mit der nächsthöheren Priorität; diese Elemente werden in der Reihenfolge ihrer Priorität als Nächstes fertiggestellt und releast.

Falsch zugeordnete Verantwortlichkeit

Der Überbringer schlechter Nachrichten für das Management zu sein, ist niemandes Lieblingsaufgabe. Selbst wenn ein Team weiß, dass sein Projekt schief läuft, können die Mitglieder zögern, dies dem Senior Management mitzuteilen, da es sich negativ auf ihre Karriere auswirkt. Je länger ein Team damit wartet, schlechte Nachrichten zu übermitteln, desto schlimmer werden die Probleme.

Obwohl die frühe und häufige Kommunikation von Hindernissen und Problemen der Stil von Scrum ist, unterstützt das traditionelle Projektmanagement die Mentalität »sag mir, wann es fertig ist«; Geschäftsinhaber wissen oft nicht, dass ihr Projekt vom richtigen Weg abgekommen ist, bis es zu spät ist, es zu korrigieren.

Verantwortlichkeit muss am richtigen Ort sein und nicht bei einem Projektmanager, der als Mittelsmann fungiert und der nicht an der Produkterstellung beteiligt ist. Die Transparenz von Scrum beseitigt dieses Problem. Der Product Owner ist Eigentümer der Unternehmensziele und -prioritäten. Das Entwicklungsteam entscheidet, wie diese Ziele umgesetzt werden können und in welchem Maße sie sich verpflichten können. Die Teammitglieder informieren den Product Owner im Verlauf des Sprints täglich über ihre Lösungen; die Stakeholder werden am Ende eines jeden Sprints auf den neuesten Stand gebracht.

Der Product Owner übernimmt keine Verantwortung für die Entwickler und die Entwickler übernehmen keine Verantwortung für den Product Owner. Jeder ist für sich selbst verantwortlich. Scrum bietet die Möglichkeit, Verantwortlichkeit, Transparenz und Eigenverantwortung angemessen zu verorten.

Scrum-Lösungen

Scrum schont im Sinne der Unternehmensinhaber die Ressourcen, indem es Elemente mit hohem Risiko und mit hoher Priorität an die erste Stelle setzt. Der Ansatz lautet: »Wenn Sie scheitern, dann scheitern Sie bitte früh und schnell.« Wenn Unternehmer nicht glauben, dass ein Projekt so realisiert wird, wie sie es sich wünschen, haben sie die Möglichkeit, frühzeitig den Stecker zu ziehen und wertvolle Ressourcen und Zeit zu sparen.

Wenn die tatsächlichen Kosten (TK) und die Alternativkosten (AK) den Wert des Projekts (W) übersteigen, ist es an der Zeit, mit etwas anderem weiterzumachen. Effektive Portfoliomanager verwenden die in [Kapitel 5](#) vorgestellte Formel $TK + AK > W$ als Trigger, um ein Projekt zu beenden.



Mit dem Scrum-Framework kann ein Portfoliomanager innerhalb weniger Sprints feststellen, ob ein Projekt realisierbar ist. Das Projekt schreitet zügig und effizient voran oder es wird aus dem Backlog entfernt und die Talente werden für Projekte mit höherer Priorität freigesetzt. Das Scheitern eines Projekts wird weniger zu einem Stigma, weil die Kosten des Scheiterns drastisch gesenkt wurden.



In [Kapitel 5](#) werden die Work-in-Progress-Grenzwerte für Entwicklungsteams diskutiert. Diese Limits gelten auch für das Portfoliomanagement. Sie können so viele Projekte offen haben, wie Sie Scrum-Teams haben. Ein stabiles Team kann sich auf eine Anforderung, ein Release und ein Projekt konzentrieren, bis es fertig ist, und dann das Projekt mit der nächsthöheren Priorität übernehmen. Wenn Sie mehrere Teams haben – großartig. Bringen Sie die Fähigkeiten der

Mitarbeiter mit den Projekten in Einklang. Weil Teams engagiert und stabil sind, werden Projekte schneller durchgeführt, als wenn Talente zwischen den Projekten wechseln. *Alle* Projekte werden im Rahmen eines dedizierten Teammodells abgeschlossen, bevor ein beliebiges Projekt im Rahmen eines Modells fertig wird, bei dem die Mitarbeiter zwischen Projekten wechseln.

[Abbildung 13.1](#) zeigt den Unterschied zwischen laufenden, nacheinander bearbeiteten Projekten (dedizierte Teams) und parallelen Projekten (Teams, die Thrashing nutzen).

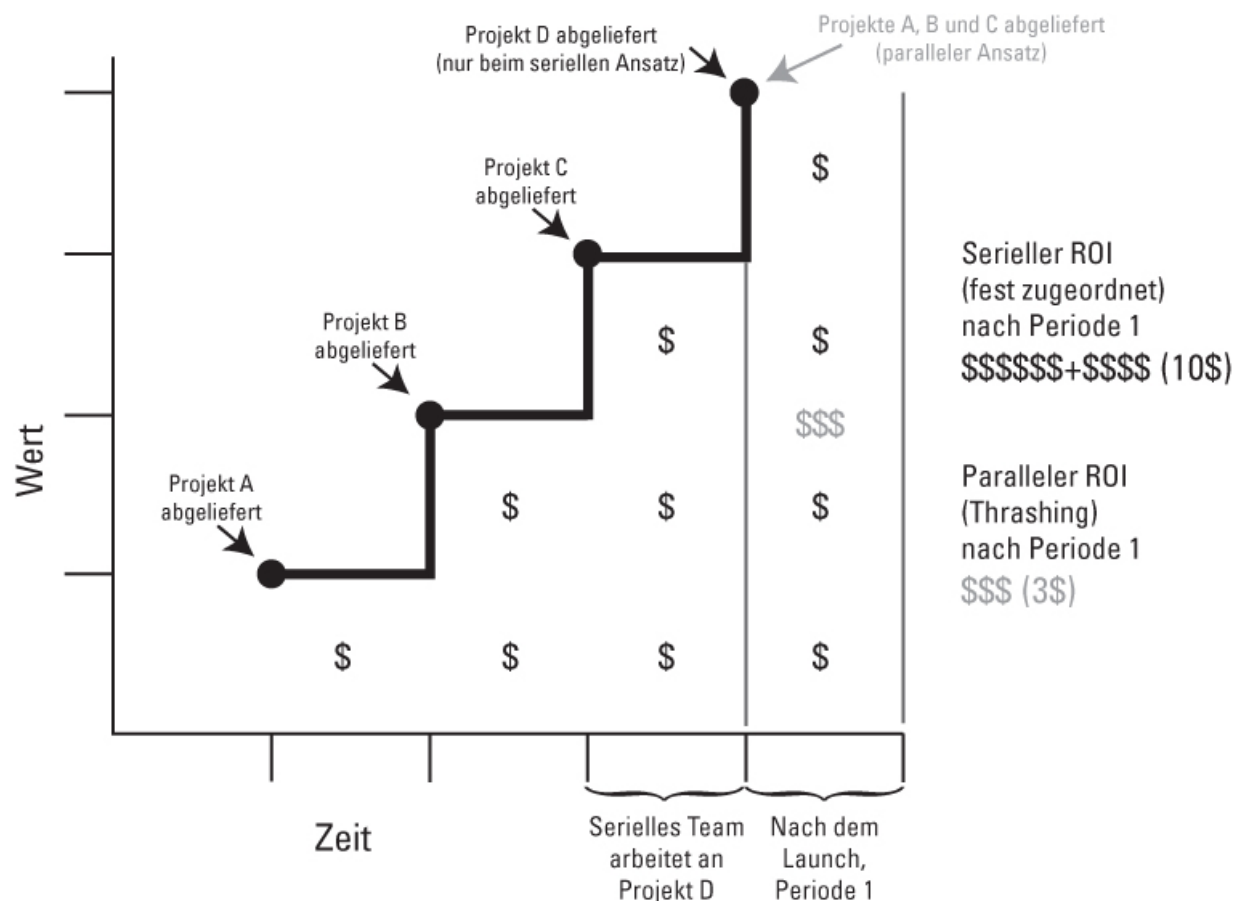


Abbildung 13.1: Vergleich der finanziellen Kosten für Verspätungen als Folge von Thrashing statt des Einsatzes von Teams

In diesem Beispiel befinden sich in einem Portfolio drei Projekte, die jeweils nacheinander (also seriell) bearbeitet werden. Angenommen, eine Werteinheit kann in einer Zeiteinheit

produziert werden und jedes Projekt kann eine Werteinheit (\$) produzieren, wenn es fertig ist.

Die Kosten für Thrashing des Teams belaufen sich auf einen mindestens 30 Prozent größeren Zeitaufwand, um jedes Projekt abzuschließen – sagen wir, 33 Prozent für dieses Szenario. Bei drei Projekten (99 Prozent) entspricht der Mehraufwand in etwa der Länge eines ganzen Projekts, wenn man die Teams zwischen Projekten wechseln lässt.

Geht man nur eine Zeiteinheit nach der Bereitstellung der seriellen Projekte, ist der ROI der parallelen Projekte am Ende dieses Zeitraums \$\$\$ (3\$ - ein \$ für jedes eingesetzte Projekt). Der Return on Investment der seriellen Projekte beträgt \$\$\$\$\$+\$\$\$\$ (10\$, der Return on Investment ist also mehr als das Dreifache).

Setzen Sie Ihre Mitarbeiter nicht mehr parallel und gleichzeitig in mehreren Projekten ein. Führen Sie ein Projekt nach dem anderen durch, und zwar jedes Projekt durch ein festes Team. Wenn Sie Teams an einem Projekt arbeiten lassen, werden die Projekte schneller fertig und Sie erhalten einen früheren, höheren ROI.

Wenn die Führungskräfte beginnen, Prioritäten zu setzen, um so das Thrashing zu eliminieren, stellt die Feedbackschleife in Form der Sprint-Reviews sicher, dass die Interessen der Stakeholder (einschließlich der Führungskräfte) direkt mit der Arbeit des Scrum-Teams in Einklang gebracht werden. Weil die Führungskräfte eindeutig kommunizieren, was höchste Priorität besitzt, können sie sicherstellen, dass sie bekommen, was sie wollen, und Entwickler vermeiden, dass nur diejenigen gehört werden, die am lautesten schreien.

Ziele werden aufeinander ausgerichtet, Strategien und Taktiken werden priorisiert. Ständige Kommunikation bedeutet, dass Probleme und Hindernisse frühzeitig angesprochen werden; hierdurch lassen sich Lösungen entwickeln oder die Richtung kann geändert werden. Kreative Innovation und Experimente werden gefördert, ohne Angst vor unnötiger Verschwendung.



Unterschätzen Sie nicht die Kosten, die entstehen, wenn der Wert später ausgeliefert wird. Eine Lieferverzögerung entspricht der Zahlung eines hohen Zinssatzes über einen langen Zeitraum, da die Organisation nicht in der Lage ist, das Portfolio zu priorisieren. Führen Sie diese ernsthaften und möglicherweise schwierigen Gespräche im Voraus, um in der Zukunft einen hohen Schuldendienst zu vermeiden.

Lean Startup

Die Gründung einer eigenen Startup-Firma ist kein kleines Unterfangen. Sie stehen vor all den üblichen Herausforderungen im Zusammenhang mit der Gründung eines Unternehmens, einschließlich der Verwendung persönlicher Mittel zur Finanzierung der Bemühungen und der Planung für den Erfolg. Startups haben gewisse Portfoliovorteile gegenüber größeren etablierten Unternehmen und ein enormes Wertsteigerungspotenzial. Ihre geringe Größe erhöht ihre Agilität. Entscheidungen können schnell und ohne das Gewicht der Unternehmensbürokratie getroffen werden. Dennoch ist es für das unternehmerische Überleben von entscheidender Bedeutung, frühzeitig schwierige Entscheidungen zu treffen und schneller als alle anderen ein minimal funktionsfähiges Produkt ausliefern zu können.

Lean Startup ist Scrum für Startups.

Bei der Wasserfallmethode beginnen viele Startups in der Regel mit einer Idee. Anschließend wenden sie Zeit und Geld auf, um aus dieser Idee einen formalen mehrjährige Business-Plan zu entwickeln, und hoffen, dass ist, was der Kunde will. Oftmals scheitern Startups, bevor sie dem Kunden das Produkt zeigen können. Aktuelle Daten zeigen, dass 80 Prozent der neuen Unternehmen innerhalb der ersten 18 Monate scheitern.



In gewisser Hinsicht ist eine Ausfallrate von 80 Prozent schrecklich. Andererseits werden all diese potenziellen Konkurrenten von der Flut der Ineffizienz erfasst. Mit Lean Startup können neue Unternehmen ihre Vorteile ausbauen und unnötige Fallstricke vermeiden.

In traditionellen Projekt-Frameworks nimmt der Kunde oft an der anfänglichen Planung und Priorisierung nicht teil, wodurch die Möglichkeit der Zusammenarbeit und der Anpassung des Projekts verpasst wird. Es gibt jedoch keine Einheitslösungen und der Benutzer könnte unzufrieden bleiben und sich gegen das Produkt entscheiden.

Bei Lean Startup wird von Anfang an der gleiche Ansatz von Überprüfung und Anpassung wie beim Scrum-Framework angewendet. Die Feedbackschleife ist entscheidend. Sprint-Planung, Sprint-Reviews und Sprint-Retrospektiven ermöglichen eine ständige Kontrolle und Anpassung auf der Startup-Ebene.



Mit Lean Startup müssen Sie nicht Monate oder Jahre auf wertvolle Prototypen warten. Durch den Priorisierungsprozess der größten Werte und der höchsten Risiken haben Sie Produkte, die Sie innerhalb weniger Wochen Stakeholdern und Kunden in die Hand geben können. Das Feedback, das Sie erhalten, können Sie sofort für zukünftige Iterationen nutzen.

Sie können auch zentrale, aber oft übersehene Fragen wie die folgenden ansprechen:

- ✓ Soll dieses Produkt erstellt werden?
- ✓ Funktioniert das Produkt auf seiner grundlegendsten Ebene?
- ✓ Ist das Produkt wirtschaftlich sinnvoll?

Bevor Sie große Geldmengen ausgeben, können grundlegende Fragen und Antworten ans Licht gebracht werden. »Früh

scheitern bedeutet billig scheitern« ist auch eine Form des Erfolgs.



Pivoting (Umschwenken) ist Teil der Lean-Startup-Welt. Sie schwenken um, wenn Sie Ihren Kurs überprüfen und ändern und dann auf der Grundlage der Informationen, die Sie aus der Feedbackschleife gewonnen haben, Anpassungen vornehmen.

Der Priorisierungsprozess wird von den Realitäten gesteuert, nicht von der Hoffnung. Bei der Entwicklung von Features und der Gewinnung von Feedback werden die Daten sofort in neue Iterationen (Sprints) und Anforderungen eingearbeitet.



Das Lean-Startup-Modell verwendet den Bauen-Messen-Lernen-Kreislauf, der im Wesentlichen dem Scrum-Ansatz entspricht:

- ✓ **Bauen:** Im Sprint-Zyklus wird das Produktinkrement erstellt, das das Release-Ziel unterstützt (minimal funktionsfähiges Produkt, MVP = minimum viable product).
- ✓ **Messen:** Feedback erhält man sowohl im Sprint-Review als auch in der Sprint-Retrospektive.
- ✓ **Lernen:** Durch Kundenfeedback werden neue Anforderungen in das Produkt-Backlog aufgenommen und der Aktionsplan für die in der Retrospektive entwickelten Verbesserungen initiiert.

Dropbox

Dropbox, eine Anwendung zum Speichern und Freigeben von Dateien, war einst ein Produkt, das die Leute für überflüssig hielten. Das Unternehmen nutzte das MVP-Konzept, um die Menschen bereits in einem frühen Stadium der Entwicklung dazu zu bringen, sein Produkt zu nutzen. Wie verkauft man etwas, von dem die Leute nicht wissen, dass sie es brauchen? Henry Ford sagte einmal: »Wenn ich die Leute gefragt hätte,

was sie wollen, hätten sie gesagt: schnellere Pferde.« Lean Startup ist ein hervorragendes agiles Werkzeug, um unternehmerische Innovationen zu ermöglichen.

Da die Entwicklung von Dropbox umfangreiche Integrationen mit Computerplattformen und Betriebssystemen erforderte, war die Entwicklung eines vollständigen Prototyps angesichts der Komplexität des Engineerings nicht möglich. Dropbox löste dieses Dilemma, indem es ein Video für technologisch versierte Early Adopters erstellte, das zeigte, warum sie das Produkt brauchten und wie einfach es zu bedienen war. Das Unternehmen wusste, dass es so früh wie möglich Feedback brauchte und dass dieses Feedback von den Leuten kommen muss, die den Entwicklern das beste Feedback geben. Der Lean-Startup-Ansatz ermöglichte Dropbox, den Traffic frühzeitig zu steigern; die Warteliste wuchs von 5.000 auf 75.000 Interessenten an. Buchstäblich über Nacht hatte das Unternehmen die Bestätigung, dass sein Produkt das war, was der Markt wollte. Das Video war das erste MVP und begeisterte das Publikum für das nächste MVP, bei dem es um die am höchsten bewerteten Funktionen zum Speichern und Teilen von Dateien ging.

Das Dropbox MVP war eine Gelegenheit, den Kunden etwas zu zeigen, von dem sie nicht wussten, dass sie es brauchten, und die Hypothese über die Marktbedürfnisse schnell und kostengünstig zu validieren oder zu verwerfen.

Skalierung von Scrum für große Portfolios

Scrum wurde im Wesentlichen für unterteilte Projekte (Projekte, die auf erreichbare Ziele heruntergebrochen wurden) konzipiert, die schnell gestartet und abgeschlossen werden konnten. Ein Scrum-Team besteht aus etwa sechs Entwicklern, einem Scrum Master und einem Product Owner. Es gibt eine natürliche Grenze,

was ein Team dieser Größe erreichen kann. Manche Projekte sind jedoch so groß, dass sie mehr als ein Standard-Scrum-Team benötigen.

Die Skalierung von Scrum ist nötig, wenn mehrere Scrum-Teams an einem Projekt oder an einem Portfolio mit Projekten arbeiten, die miteinander zusammenhängen. Ein Beispiel ist Microsoft Office. Ein Team kann an Word, ein anderes an Excel und ein drittes an PowerPoint gearbeitet haben. Alle Teams mussten jedoch integriert werden und effektiv zusammenarbeiten, damit Office als Paket verkauft werden konnte. Viele Projekte sind so groß, dass sie mehrere Teams und Projekte erfordern. Scrum ist in der Lage, auf diese Anforderungen zu reagieren.



Wenn mehrere Teams an einem einzigen Release-Paket arbeiten, ist ein Scrum-Modell auf Unternehmensebene erforderlich, um eine effektive Koordination, Kommunikation, Integration und Beseitigung von Hindernissen zu gewährleisten.

Eine Übersicht über vertikalen Schnitt

Der *vertikale Schnitt*, auch vertikale Aufteilung oder *vertical slicing* genannt, ist eine Form zur Skalierung von Scrum. Hierbei werden mehrere Scrum-Teams gebildet, um das Programmziel zu erreichen. Jedes Scrum-Team arbeitet in synchronisierten Sprints gleicher Länge an einem separaten Teil oder Modul des Gesamtproduktes, und diese Module werden nach jedem Sprint von einem Integrations-Scrum-Team integriert. Das Integrations-Scrum-Team liegt um einen Sprint hinter den Entwicklungs-Scrum-Teams zurück und ist ein eigenes Scrum-Team mit eigenen engagierten Entwicklern sowie einem Product Owner und Scrum Master.

[Abbildung 13.2](#) zeigt, wie sich die Funktionen der einzelnen Teams in den Rückstand des Integrationsteams für die Koordination auf Architektur- und Systemebene einfügen.

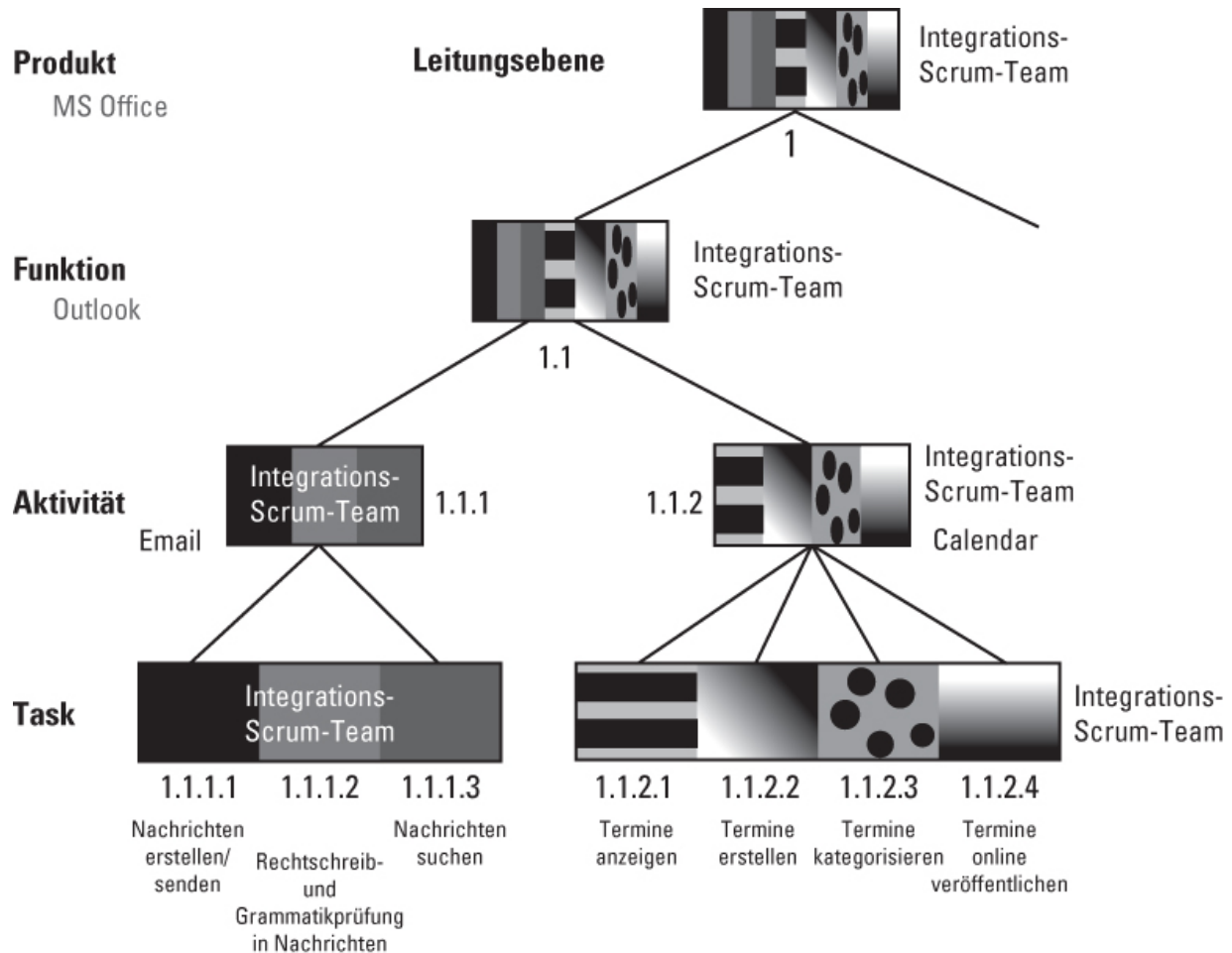


Abbildung 13.2: Ein Enterprise-Scrum-Team mit einem Integrationsteam

Das Integrations-Scrum-Team übernimmt alle Entwicklungsarbeiten auf Systemebene, die für die Integration der von den Teams produzierten Funktionen erforderlich sind. Außerdem kümmert es sich um die architektonische Kontrolle, damit die Arbeit der einzelnen Scrum-Teams vereinheitlicht wird.

Hier die Details mit Microsoft Office als Beispiel:

1. Ein Scrum-Team entwickelt die Funktionalität für das Komponieren/Senden von Nachrichten (Anforderungs-ID 1.1.1.1.1).

2. Ein anderes Team entwickelt die Funktionalität für Grammatik-/Rechtschreibprüfung in Nachrichten (1.1.1.1.2).
3. Ein drittes Scrum-Team entwickelt die Funktionalität für die Suche nach Nachrichten (1.1.1.1.3).
4. Das Integrations-Scrum-Team übernimmt die Entwicklungsarbeit, um die Funktionalität der ersten drei Teams in ein Paket zu integrieren, das das E-Mail-Integrationsteam in das E-Mail-Modul integrieren kann.
5. Das Outlook-Integrationsteam integriert E-Mail, Kalender, Kontakte und andere Module in ein Outlook-Paket, das in die gesamte Microsoft Office-Suite integriert werden kann.

In diesem Beispiel arbeiten Integrationsteams als separate Scrum-Teams, die für jede Rolle eigene Teammitglieder haben.

Scrum of Scrums

Das Scrum-of-Scrums-Modell ermöglicht eine effektive Integration, Koordination und Zusammenarbeit zwischen Scrum-Teams durch vertikalen Schnitt. Fast alle Skalierungs-Frameworks, die wir Ihnen in diesem Kapitel zeigen, verwenden Scrum of Scrums, um die tägliche Koordination zwischen den Scrum-Teams zu ermöglichen.

[Abbildung 13.3](#) verdeutlicht, dass sich jede Rolle jedes Teams täglich mit den Leuten der gleichen Rolle der anderen Teams koordiniert, um Prioritäten, Abhängigkeiten und Hindernisse, die das größere Programmteam beeinträchtigt, zu besprechen. Das Scrum of Scrums für jede Rolle wird von der Person auf Integrationsebene der jeweiligen Rolle unterstützt. Sorgfältige Integrations- und Release-Anstrengungen begründen ein konsistentes und ordnungsgemäßes Scrum-of-Scrums-Modell.

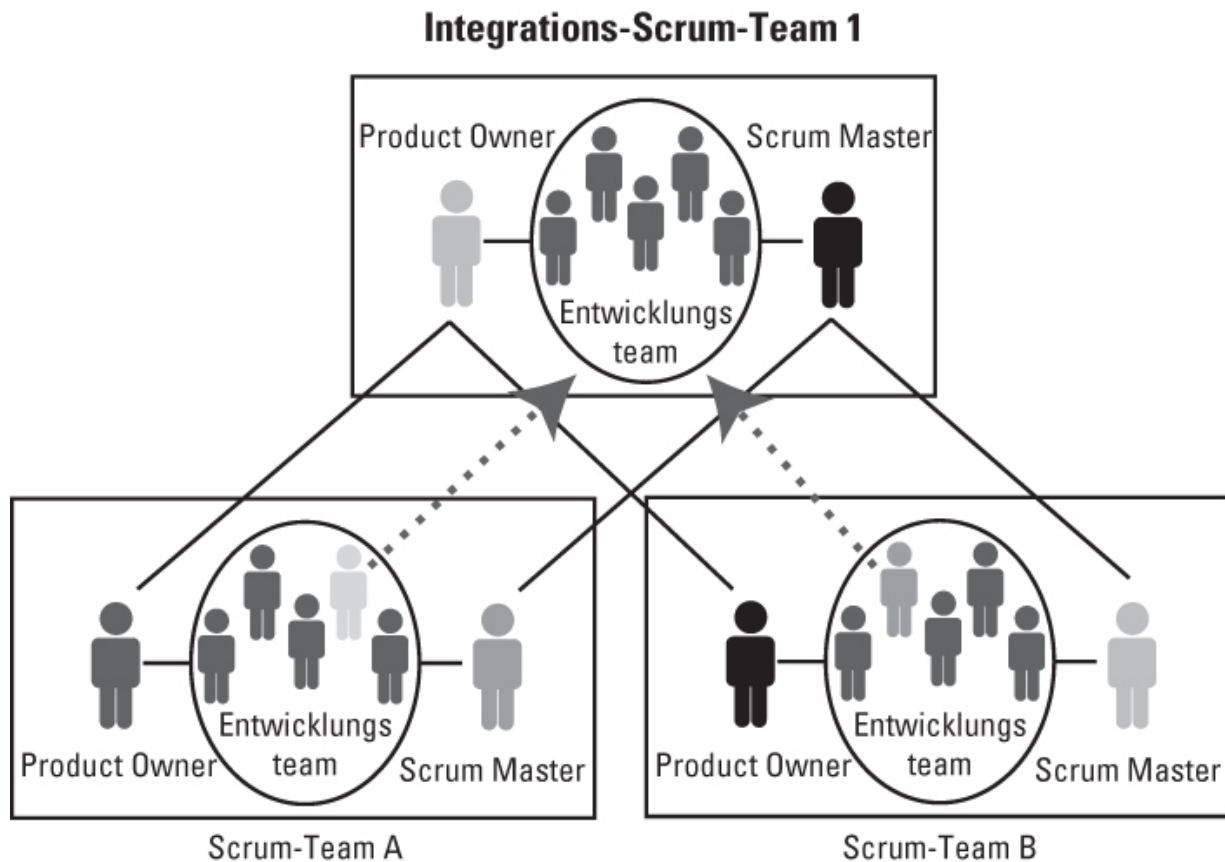


Abbildung 13.3: Scrum of Scrums zur Koordination von Scrum-Teams

Die einzelnen Scrum-Teams halten täglich ungefähr zur gleichen Zeit an verschiedenen Orten ihr Daily Scrum ab. Nach dieses Daily Scrums finden die nachfolgend beschriebenen Scrum-of-Scrums-Meetings statt.

Scrum of Scrums der Product Owner

Jeden Tag, direkt nach den Daily Scrums der einzelnen Scrum-Teams, treffen sich die Product Owner aller Teams mit dem Product Owner des Integrationsteams für maximal 15 Minuten. Sie besprechen die fertiggestellten Anforderungen und nehmen aufgrund der im Daily Scrum der einzelnen Scrum-Teams erkannten Gegebenheiten Anpassungen vor. Jeder Product Owner bringt die folgenden Themen ein:

- ✓ Die Geschäftsanforderungen, die jeder seit dem letzten Treffen abgenommen oder abgelehnt hat

- ✓ Die Anforderungen, die abgenommen sein sollten, wenn sie sich das nächste Mal treffen
- ✓ Welche Anforderungen auf Hindernisse stoßen und deren Lösung Hilfe durch andere Teams benötigt (beispielsweise: »John, wir können die Anforderung 123 erst dann erstellen, nachdem du Anforderung xyz aus deinem aktuellen Sprint-Backlog fertiggestellt hast.«)

Der Product Owner des Integrationsteams fällt die teamübergreifenden Prioritätsentscheidungen, die nötig sind, um sicherzustellen, dass die Hindernisse während des Daily Scrum of Scrums behandelt werden.

Scrum of Scrums der Entwicklungsteams

Jeden Tag, direkt nach den Daily Scrums der einzelnen Scrum-Teams, nimmt ein Mitglied des Entwicklungsteams jedes Scrum-Teams am Daily Scrum des Integrationsteams teil (dies ist das Scrum of Scrums für Entwickler) und diskutiert mit den Mitgliedern des Integrationsentwicklungsteams die folgenden Punkte:

- ✓ Was ihr Team seit dem letzten Treffen erreicht hat
- ✓ Was ihr Team zwischen jetzt und dem nächsten Treffen erreichen will
- ✓ Technische Probleme, bei denen sie Hilfe benötigen
- ✓ Technische Richtungsentscheidungen, die das Team getroffen hat und die jeder kennen sollte, um potenziellen Problemen vorzubeugen



Ziehen Sie in Erwägung, dass die Mitglieder der Entwicklungsteams der einzelnen Scrum-Teams, die am Scrum of Scrums (Daily Scrum des Integrationsteams) teilnehmen, rotieren, entweder täglich oder für jeden Sprint. Sie können so sicherstellen, dass jeder über den Integrationsaufwand für das Portfolio auf dem Laufenden bleibt.

Scrum of Scrums der Scrum Master

Die Scrum Master jedes Scrum-Teams treffen sich ebenfalls mit dem Scrum Master des Integrationsteams für maximal 15 Minuten und besprechen die Hindernisse und Blockaden, mit denen jedes Teams zu kämpfen hat. Jeder Scrum Master bringt die folgenden Themen ein:

- ✓ Die Hindernisse auf der Ebene der einzelnen Teams, die seit dem letzten Treffen gelöst wurden und wie sie gelöst wurden, für den Fall, dass andere Scrum Master auf das gleiche Problem stoßen
- ✓ Neue Hindernisse, die seit dem letzten Treffen aufgetaucht sind sowie alte Hindernisse, die noch nicht beseitigt wurden
- ✓ Bei welchen Hindernissen sie zur Behebung Hilfe benötigen
- ✓ Potenzielle Hindernisse, die alle kennen sollten

Der Scrum Master des Integrationsteams stellt anschließend sicher, dass die eskalierten Hindernisse nach dem täglichen Scrum of Scrums angegangen und behoben werden.

Beim Modell des vertikalen Schnitts existiert weiterhin ein einzelner Produkt-Backlog. Diejenigen Anforderungen, die untergliedert und an ein Entwicklungs-Scrum-Team übergeben wurden, werden mit dem betreffenden Team gekennzeichnet. Bei diesem Modell können Sie immer das gesamte Programm sehen und schnell die Teile Ihres eigenen Teams herausfiltern.

Wem gehört die Architektur?

Ihre Organisation sollte vorhandene Architekturstandards, Code-Standards und Stilrichtlinien besitzen. Auf diese Weise muss nicht jedes Team das Rad neu erfinden. Dennoch lautet eine häufig gestellte Frage: »Wer ist bei einem vertikal aufgeteilten Programm für die Architektur verantwortlich « Die Antwort lautet, dass dies davon abhängt, welche Module von der Entscheidung betroffen sind.

Nehmen Sie eine architektonische Entscheidung, die getroffen werden muss und die nur Modul A betrifft. Das Entwicklungsteam für Modul A trifft diese Entscheidung. Falls diese Entscheidung mehrere Teams beeinflussen würde, dann würde das Entwicklungsteams auf der Integrationsebene, in dem sich Vertreter aller betroffenen Teams befinden, diese Entscheidung treffen. Diese Integrationsebene kann eine oder auch vier Ebenen höher sein.

Nehmen wir [Abbildung 13.1](#) als Beispiel. Eine Architekturentscheidung, die zwei der E-Mail-Modul-Teams (1.1.1.2 und 1.1.1.3) betrifft, würde vom E-Mail-Integrationsteam getroffen (1.1.1). Eine Entscheidung, die Einfluss auf das E-Mail-Modul-Team »Nachrichten suchen« (1.1.1.3) und das Integrationsteam für den Kalender (1.1.2) hat, würde vom Outlook-Integrationsteam getroffen (1.1). Der vertikale Schnitt ist ein einfacher Weg, um die Autonomie jedes Scrum-Teams, wertvolle Funktionalität innerhalb eines größeren Programmkontextes zu liefern, zu bewahren. Gleichzeitig trägt der vertikale Schnitt effektiv dazu bei, dass Teams zeitnahe und wichtige Gespräche über Beschränkungen und den Fortschritt führen können.

Scrum at Scale

Der Scrum-at-Scale-Ansatz für zwei oder Hunderte von Scrum-Teams, die zusammenarbeiten, ist eine Art einfaches Scrum-of-

Scrums-Modell für Scrum Master und Product Owner, die die Kommunikation, die Beseitigung von Hindernisse, die Prioritäten, die Ausarbeitung der Anforderung und die Planung koordinieren. Der Einsatz eines Scrum-of-Scrums-Modells für die Scrum-Master- und die Product-Owner-Rolle ist die Methode, wie sich die tägliche Synchronisation zwischen Teams bei Programmen unterschiedlicher Größe vollzieht.

Den Scrum Master skalieren

Entsprechend dem Modell des vertikalen Schnitts bei Scrum of Scrums gruppiert Scrum at Scale fünf Scrum Master in ein Scrum Master Scrum of Scrums. Es entspricht dem Daily Scrum für einzelne Scrum-Teams, jedoch hat es die Aufgabe, Hindernisse ans Tageslicht zu bringen und zu beseitigen. Da Scrum at Scale den Umfang eines Scrum of Scrums auf fünf Scrum Master von fünf einzelnen Scrum-Teams einengt, wird auch die Komplexität bei der Kommunikation reduziert. Ein Scrum Master Scrum of Scrums koordiniert als Release-Team die Release-Aktivitäten. Dieses Modell ist in [Abbildung 13.4](#) dargestellt.

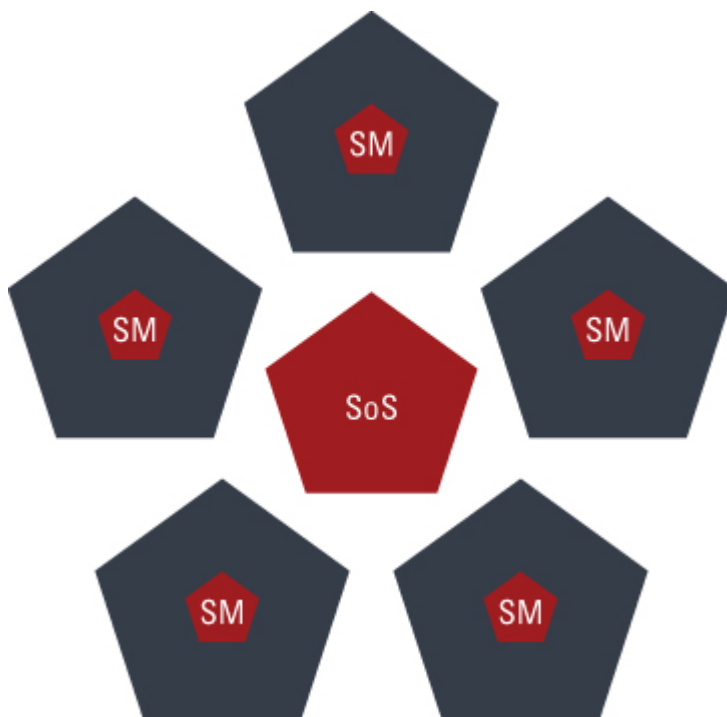


Abbildung 13.4: Scrum-of-Scrums-Modell in Scrum at Scale für fünf Teams
Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

[Abbildung 13.5](#) zeigt das Scrum-of-Scrums-Modell für bis zu 25 Teams.

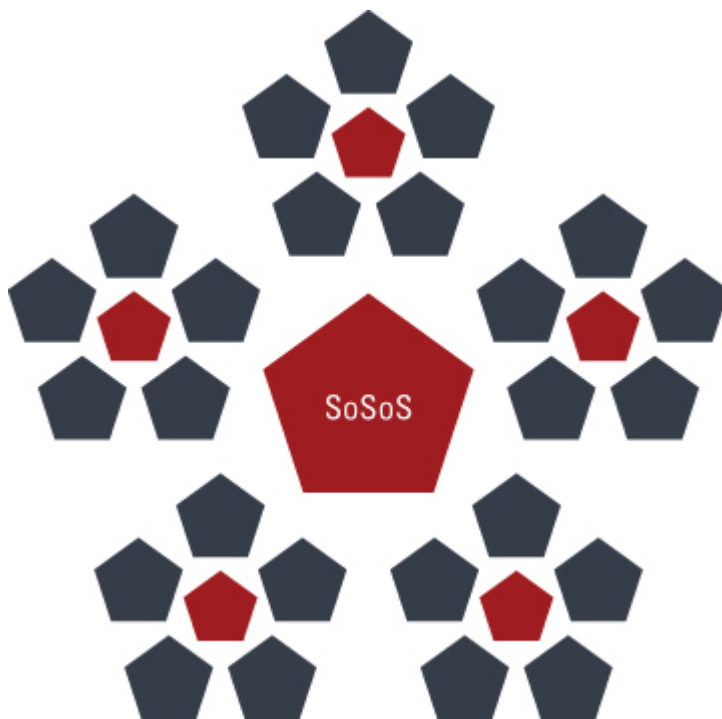


Abbildung 13.5: Scrum-of-Scrums-of-Scrums-Modell in Scrum at Scale für 25 Teams

Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

Falls ein Projekt aus mehr als 25 Teams besteht, unterstützt ein Executive Action Team (EAT) ein Scrum of Scrums of Scrums auf einer dritten Ebene, um Hindernisse und Blockaden in der Organisation zu beseitigen, die die Scrum-of-Scrums-Gruppen selbst nicht lösen können.

[Abbildung 13.6](#) verdeutlicht die dritte Ebene des Scrum-of-Scrums-of-Scrums-Modells aus Agile at Scale mit einem Executive Action Team (EAT).

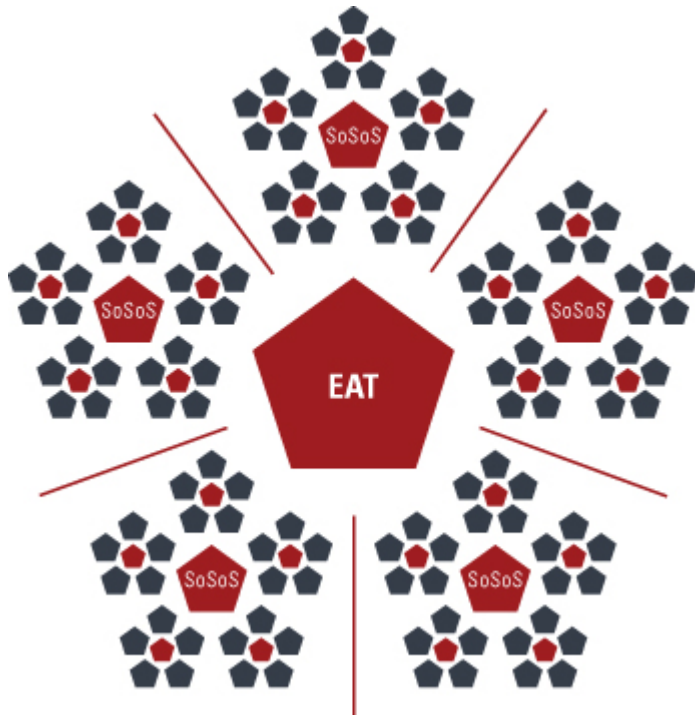


Abbildung 13.6: Dritte Ebene des Scrum-of-Scrums-of-Scrums-Modells aus Agile at Scale mit einem Executive Action Team (EAT)
Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

Den Product Owner skalieren

Die Product Owner organisieren sich auf eine ähnliche und abgestimmte Weise, nur dass dies nicht Scrum of Scrums, sondern Meta-Scrums genannt wird. Ein Meta-Scrum mit einer Ebene bringt fünf Product Owner in Meta-Scrum-Meetings zusammen, in denen Prioritäten präzisiert und geplant werden. Jedes Meta-Scrum hat einen Chief Product Owner (CPO), der den größeren Zusammenhang von Produktvision und Produkt-Backlog im Auge behält und die Koordination der Product Owner im Meta-Scrum unterstützt. [Abbildung 13.7](#) zeigt das Meta-Scrum für Product Owner aus Scrum at Scale.

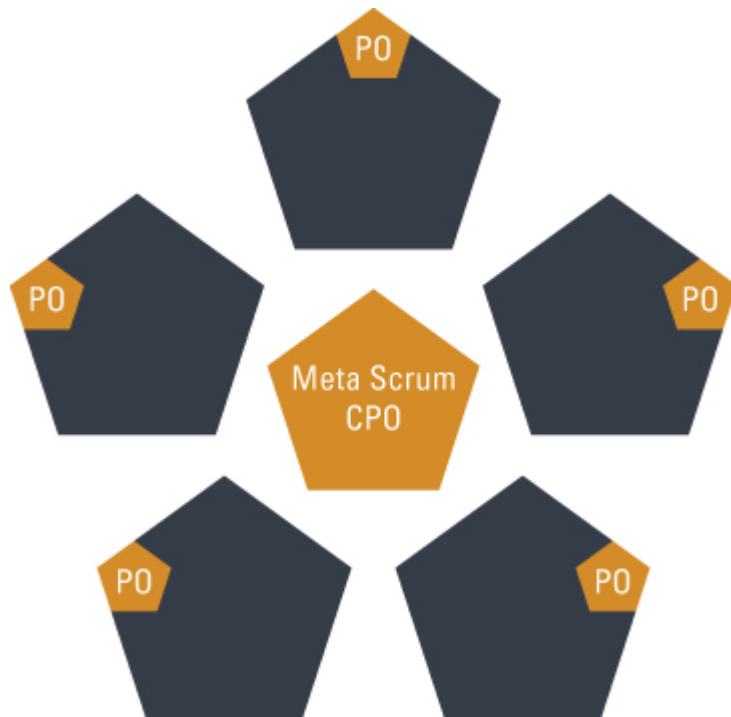


Abbildung 13.7: Meta-Scrum für Product Owner in Scrum at Scale
Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

Bei den Meta-Scrums der zweiten und dritten Ebene entspricht die Anordnung der weiter vorne beim Scrum Master Scrum of Scrums of Scrums gezeigten. Ein Executive Meta Scrum (EMS) unterstützt die Meta-Scrums, indem es die unternehmensweite Vision besitzt und kommuniziert, von den Meta-Scrums technische Prioritätsrückmeldungen entgegennimmt und die allgemeinen Prioritätsentscheidungen für das Programm fällt. [Abbildung 13.8](#) zeigt das Meta-Scrums-Modell der dritten Stufe aus Agile at Scale mit Executive Meta Scrum (EMS).



Abbildung 13.8: Dritte Ebene des Meta-Scrum-Modells aus Agile at Scale mit einem Executive Meta Scrum (EMS)

Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

Abbildung 13.9 zeigt die auf einander ausgerichtete Gruppierung der dritten Ebene des Scrum-of-Scrums-of-Scrums- und Meta-Scrum-Modells aus Agile at Scale mit Executive Action Team (EAT) und Executive Meta Scrum (EMS).

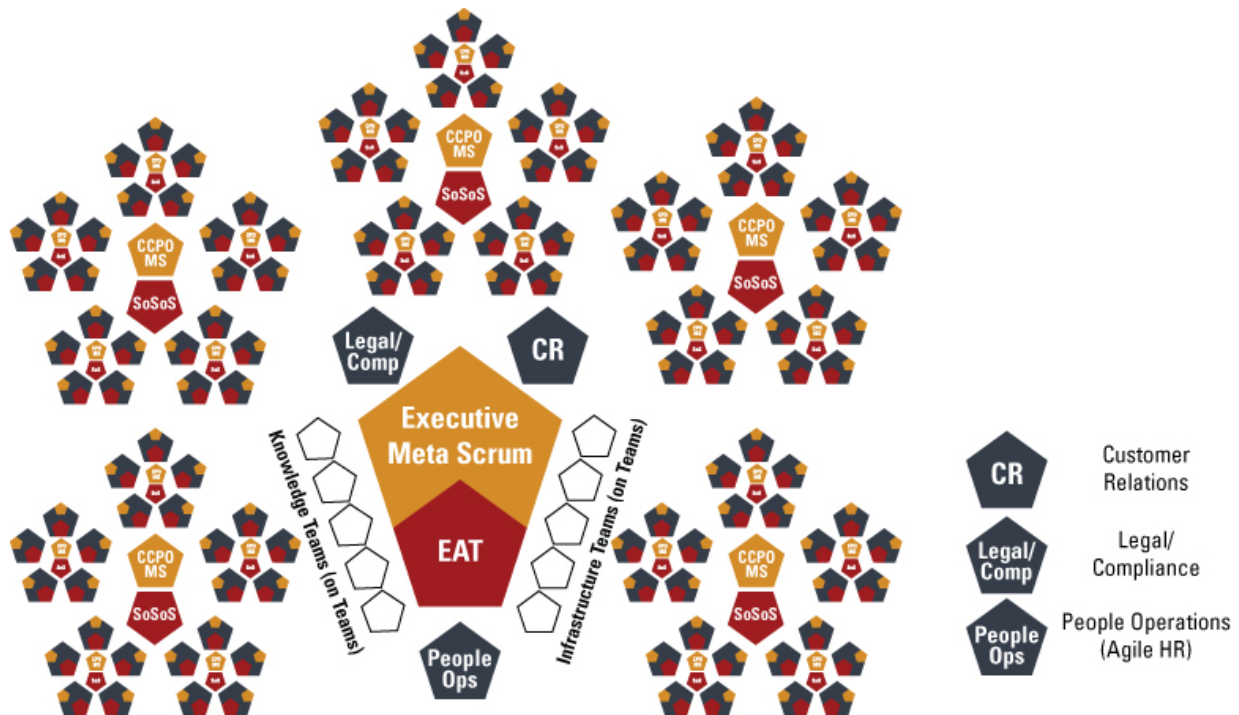


Abbildung 13.9: Ausrichtung der dritten Ebene des Scrum-of-Scrums-of-Scrums- und Meta-Scrum-Modells aus Agile at Scale

Mit freundlicher Genehmigung von Jeff Sutherland & Scrum, Inc.

Ein Meta-Scrum sollte ein Synchronisationstreffen sein, an dem alle Beteiligten teilnehmen. Alle Stakeholder auf CPO-Ebene sollten anwesend sein, um eine unternehmensweite Abstimmung und die Unterstützung der Priorisierung des CPO-Produktrückstands bei jedem Sprint zu gewährleisten.

Synchronisation in einer Stunde am Tag

In einer Stunde am Tag, oder auch kürzer, kann ein ganzes Unternehmen die Prioritäten des Tages abstimmen und effektiv die Beseitigung von Hindernissen koordinieren. Nehmen Sie beispielsweise an, dass um 8 Uhr morgens die einzelnen Scrum-Teams getrennt ihr Daily Scrum abhalten. Um 8:45 Uhr halten die Scrum Master ihr Scrum of Scrums und die Product Owner das Meta-Scrum-Meeting der ersten Ebene ab. Um 9 Uhr treffen sich die Scrum Master im Scrum of Scrums of Scrums und die Product Owner treffen sich im Meta-Scrum der zweiten Ebene. Schließlich

trifft sich um 9:15 Uhr das Scrum of Scrums of Scrums der Scrum Master mit dem Executive Action Team und die Meta-Scrum-Vertreter der Product Owner treffen sich mit dem Executive Meta Scrum.



Eine vollständige Beschreibung von Scrum@Scale finden Sie unter www.scrumatscale.com/scrum-at-scale-guide/.

Das Schöne an Scrum ist, dass es flexibel und skalierbar ist. Scrum at Scale ist der einfache Weg, die Autonomie jedes Scrum-Teams in einem größeren Programmkontext zu erhalten, aber es ist nur eines von vielen verfügbaren Skalierungsmodellen. Die folgenden Abschnitte beschreiben weitere Modelle zur Verwaltung von Projekten unterschiedlicher Größe und Portfolioebenen.

Scaled Agile Framework (SAFe)

Scaled Agile Framework (SAFe) ist ein Framework, um Scrum und agile Prinzipien über mehrere Teams und deren Projekte auf Portfolioebene zu skalieren. Der große Gesamtüberblick zu SAFe ist in [Abbildung 13.10](#) dargestellt.

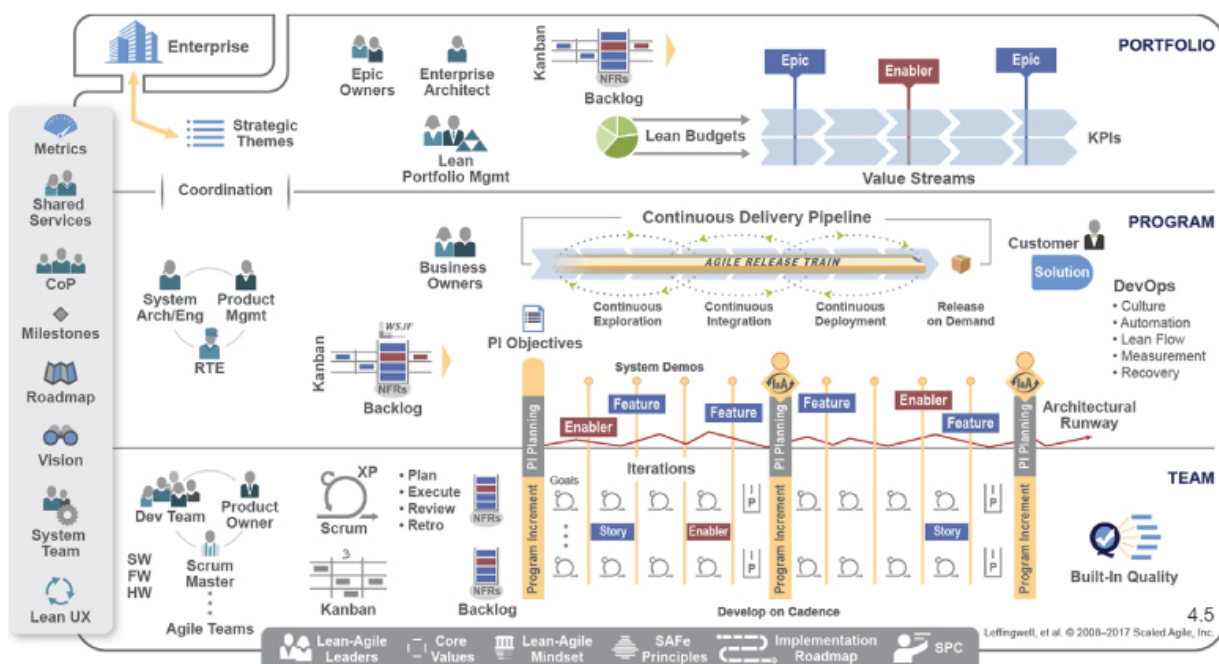


Abbildung 13.10: SAFe im Überblick

Obwohl es Unterschiede zum vertikalen Schnitt gibt, gibt es auch viele Gemeinsamkeiten, zum Beispiel die folgenden:

- ✓ Entwicklung findet in Scrum-Teams statt.
- ✓ Die Sprintdauer und Kadenz der Teams sind aufeinander abgestimmt.
- ✓ Scrum of Scrums werden zur Koordination auf der Programmebene verwendet.

SAFe kennt drei vorgeschriebene Ebenen der Integration und Koordination: Portfolio, Programm und Team.

Portfolioebene

Auf dieser Ebene werden die Vision und die Roadmap für das gesamte Portfolio erstellt. Es werden strategische Themes entwickelt, um die Vision zu unterstützen. Budget, Geschäftsziele und Einhaltung der Enterprise-Architektur werden hier gemanagt. Das Portfolio ist in Wertströmen organisiert, die die Organisation an den Werten ausrichtet, die geliefert werden sollen.

Drei Rollen auf der Portfolioebene steuern Portfolio-Entscheidungen:

- ✓ **Enterprise Architect:** Der Enterprise Architect gibt eine allgemeine technische Vision vor und steuert den ganzheitlichen technologischen Ansatz aller Programme durch kontinuierliches Feedback, Zusammenarbeit sowie anpassungsfähige Konstruktionsentwürfe und Praktiken.
- ✓ **Lean-agile Führungskräfte:** Ausbilder und Coaches in der Organisation helfen Teams, bessere Produkte durch die Werte, Prinzipien und Praktiken der schlanken und agilen Softwareentwicklung zu entwickeln. Diese Rolle kann einem Scrum Master oder einem Scrum Mentor ähnlich sein, wie in [Kapitel 2](#) beschrieben.

- ✓ **Epic Owner:** Epics werden in [Kapitel 3](#) eingeführt, jedoch verwendet SAFe eine andere Feature-Epic-Beziehung. In SAFe sind Epics die größten Anforderungen. Sie werden in Features unterteilt, die dann wiederum in User Stories zergliedert werden. Üblicherweise steuern Epic Owner eine oder zwei Epics, angefangen von ihrer Identifizierung über die Analyse bis hin zu Entscheidungen über die Umsetzung mit Entwicklungsteams.

Das Portfolio-Backlog von SAFe besteht sowohl aus Geschäfts- als auch aus Architektur-Epics. Enterprise Architect, Lean-agile Leader und Epic Owner verwalten diesen Portfolio-Backlog mit Hilfe von Kanban.

Programmebene

Innerhalb des Portfolios hat jedes Programm eine Reihe von Programm-Epen aus dem Portfolio-Backlog, die sich im Besitz eines Epic Owners dieses Programms befinden und von ihm gesteuert werden. Eine Vision und eine Roadmap für das Programm werden erstellt, um die Portfoliovision und das Backlog zu unterstützen.

Auf dieser Ebene findet das Release- und Produktmanagement statt. Zum Einsatz kommt das Modell des »Agile Release Train« (ART), bei dem es sich um ein Team agiler Teams handelt, das inkrementelle Releases liefert. Der Zug verlässt den Bahnhof nach einem zuverlässigen Zeitplan mit Möglichkeiten für das Release; jedes Programm kann, wenn es fertiggestellt ist, den Zug mit dem Release beladen. Wenn Sie ein Release verpassen, können Sie das nächste nehmen. Das ART-Modell stellt eine feste Kadenz zur Verfügung, an die sich die Teams anpassen und mit der sie sich synchronisieren können.

Auf dieser Integrationsebene laufen die Arbeiten der einzelnen Teams unter der Leitung von gemeinsamen Experten wie DevOps, User Experience, Systemarchitektur und Release Train Engineering zusammen.

Das Artefakt mit den Zielen zu den Produktinkrementen enthält Zusammenfassungen der geschäftlichen und technischen Ziele des Produkts für die kommende Version. Während der Release-Planung überprüft das Team die Vision und die Ziele, erstellt User Stories für sie und plant die für die Veröffentlichung notwendige Entwicklung. Dieser Plan wird zum Ziel des Teams für dieses Produktinkrement.

SAFe benötigt am Ende jeder Iteration zwei funktionierende Demos, die die erstellte Software oder das erstellte Produkt zeigen. Die Team-Demo stellt das Produkt dem Scrum-Team vor. Zielgruppe der System-Demo für das Produkt sind die Stakeholder auf Programmebene.

Die Release-Zusammenfassung enthält, was in jedem Release funktioniert. Die Herstellung eines funktionierenden Produktes ist das Ziel jedes Sprints oder jedes Produktinkrements.

Teamebene

Programme bestehen aus einer bestimmten Anzahl von agilen Teams, die jeweils nur in einem Programm arbeiten können. Dieser Prozess funktioniert wie die erste Schicht der Integration bei Scrum of Scrums. Agile Teams sind im Wesentlichen Scrum-Teams mit jeweils einem Entwicklungsteam mit Testern, einem Product Owner und einem Scrum Master. Die Teams arbeiten mit den anderen Teams im Programm in der gleichen Kadenz und ihre Team-Backlogs unterstützen das Programm-Backlog und sind an diesem ausgerichtet.

Vorteile des SAFe-Modells

Portfoliomanager mögen das SAFe-Modell, weil es so klar und anschaulich ist. Für größere Organisationen bietet es außerdem eine Struktur für die Einbindung des mittleren Managements in Scrum-Teams. SAFe integriert agile Techniken wie testgetriebene Entwicklung und kontinuierliche Integration.

Testgetriebene Entwicklung und kontinuierliche Integration

Testgetriebene Entwicklung (test-driven development, TDD) ist ein agiler Softwareentwicklungsprozess, bei dem ein automatisierter Komponententest geschrieben wird, bevor die Programmierung beginnt. Der Entwickler führt den Test durch, beobachtet ihn und entwickelt genug Code, um den Test ohne Rücksicht auf die Effizienz zu bestehen. Dann refaktoriert der Entwickler den Code, um so viel Coding wie möglich zu entfernen, wobei der Code weiterhin den automatisierten Komponententest bestehen muss. Dieser Prozess wird manchmal als Rot-Grün-Refaktor-Zyklus bezeichnet: Der Test schlägt fehl, der Test wird erfolgreich durchlaufen, der Code ist sauber. TDD minimiert drastisch die Verschwendung von Code, maximiert die Abdeckung von Unit-Tests und ermöglicht die nahtlose Integration der Arbeit des Entwicklers in den Hauptteil der Arbeit durch kontinuierliche Integration.

Kontinuierliche Integration (continuous integration, CI) ist eine Praxis, die Code-Entwickler verwenden, wenn sie neuen Code in ein gemeinsames Produkt integrieren. Jedes Mal, wenn Code hinzugefügt wird, wird er automatisch getestet, um eine nahtlose Integration zu gewährleisten.

Codequalität

Ein weiterer Vorteil des SAFe-Modells ist, dass die Codequalität einer der Kernwerte ist. Die Fähigkeit eines jeden Unternehmens, schnell qualitativ hochwertige Funktionalität zu erstellen, erfordert zuverlässigen Code. Die Erstellung von Codequalität besteht aus drei Teilen:

- ✓ Agile Architektur-Abstracts helfen Unternehmen bei der Lösung komplexer Probleme, die auf der Natur ihrer großen Projektanforderungen beruhen. Diese Abstracts sind bewährte Lösungen für festgelegte Probleme, die die Bedeutung der Software-Architektur in jedem Großprojekt unterstreichen.

- ✓ CI auf Team- und Systemebene beinhaltet Code aus vielen Entwicklungsteams.
- ✓ Der Code wird in jedem Schritt getestet, um sicherzustellen, dass die Software bei jedem Release funktioniert.

Jede Iteration liefert innerhalb eines strengen Zeitfensters funktionierende Software.

So wie sich Technologie und Kundenbedürfnisse ändern, so ändern sich auch die Produkte, die zur Erfüllung dieser Bedürfnisse erforderlich sind. In einem SAFe-Modell wird das Refactoring nicht nur durchgeführt, um den aktuellen Geschäftswert zu sichern, sondern auch den zukünftigen Wert.

Sicherer Übergang vom Wasserfall

Ein Big-Data-Unternehmen mit Niederlassungen in den USA und Indien hat den Übergang von Wasserfall zu SAFe vollzogen. Die Herausforderungen des Marktes bestanden für das Unternehmen unter anderem in einem verstärkten Wettbewerb, sich ändernden Technologien und langen Vorlaufzeiten, um neue Produkte auf den Markt zu bringen.

Der Wasserfallprozess des Unternehmens war so umfangreich dokumentiert, dass 25 bis 40 Dokumente von bis zu zehn Stakeholdern unterzeichnet werden mussten, um genehmigt zu werden. Mehrere laufende Projekte erzeugten Thrashing-Probleme, geringe Sichtbarkeit und niedrige Priorisierung. Die Firma strauchelte unter ihrem eigenen bürokratischen Gewicht.

Innerhalb von 22 Monaten nach der Einführung von SAFe erzielte die Organisation folgende Ergebnisse:

- ✓ Umgruppierung von mehr als 500 Mitarbeitern in mehr als 60 agile Teams
- ✓ Verkürzung der Markteinführungszeit um 27 Wochen

- ✓ Verkürzung der Fehlerbehebungszeit von Monaten auf Stunden
- ✓ Steigerung der Kundenzufriedenheit und der Mitarbeitermoral

Zunächst zögerte das Unternehmen, das Scrum-Framework einzuführen. Es war der Meinung, dass Scrum nur für kleine Unternehmen geeignet und nicht den Bedürfnissen von viel größeren Unternehmen mit multinationaler Präsenz entsprechen würde. SAFe lieferte eine skalierte Scrum-Lösung, die gut zum Unternehmen passte.

Das Unternehmen begann mit ein paar Pilot-Teams, die Scrum einsetzten, und nach der Beseitigung einiger organisatorischer Probleme erzielte es Erfolg. Es wurden agile und Scrum-Coaches eingesetzt, um das Scrum-Framework in die größere Organisation zu integrieren. Die Firma begann, mit dem Scrum-Prozess sich selbst zu überprüfen und sich anzupassen. Das Unternehmen startete mit drei Scrum-Teams als Pilotteams, fügte ein paar weitere hinzu und ließ schließlich viele Scrum-Teams gleichzeitig arbeiten. Anstatt alles auf einmal zu ändern, skalierte das Unternehmen Scrum mit SAFe inkrementell.

Aufgrund des Erfolges mit den vorangegangenen Schritten führte das Unternehmen einen kompletten SAFe-Rollout durch. Mehr Experten wurden eingestellt, interne Scrum-Schulungen wurden durchgeführt und die obere Managementebene wurde auf dem Laufenden gehalten.

Das Unternehmen ging seine Skalierungsprobleme wie folgt an:

- ✓ Es wurden Unternehmensbereiche geschaffen, um Product Owner, Management und Entwicklung zu unterstützen.
- ✓ Arbeitsnischen wurden zu agilen Pods umgestaltet, was die Kollokation und Zusammenarbeit im Team erleichterte.

- ✓ Es wurden Release Trains eingerichtet, die die fortlaufende, portfolio-übergreifende Produktfreigabe erleichterte.

Large-Scale Scrum (LeSS)

Eine weitere Möglichkeit, das Scrum-Framework in umfangreiche Projekte einzubinden, ist Large-Scale Scrum (LeSS).

Mit LeSS wurden Produkte mit lediglich vier bis fünf kleinen Teams entwickelt, bis hin zu Teams, die 1.500 Personen umfassten und sich über ein halbes Dutzend Länder erstreckten. Anders ausgedrückt: Wie groß soll »groß« für Sie sein? LeSS kann Scrum nach oben oder unten skalieren und funktioniert daher in vielen Umgebungen.

Grundlage von LeSS ist das Systemdenken, bei dem ein System als eine organisierte Einheit (in diesem Fall ein Produktbereich) aus zusammenhängenden und voneinander abhängigen Teilen (Teams) betrachtet wird. Die Macher von LeSS sehen ausdrücklich, dass die Optimierung des Ganzen nicht aus der Optimierung der Teile eines Ganzen resultiert. Die Optimierung von Teilen optimiert nur die Teile, sodass LeSS die Zusammenarbeit zwischen mehreren Teams, auch über verschiedene Standorte hinweg, optimiert.

LeSS kennt nur wenige Regeln und zwei Frameworks: LeSS und LeSS Huge. Der Unterschied liegt in der Größe der beteiligten Teams.

LeSS, das kleinere Framework

[Abbildung 13.11](#) zeigt das grundlegende LeSS-Framework. Die Anzahl der Entwicklungsteams variiert zwischen zwei und acht. Ein Product Owner kümmert sich um bis zu acht Teams und jeder Scrum Master bedient bis zu drei Teams.

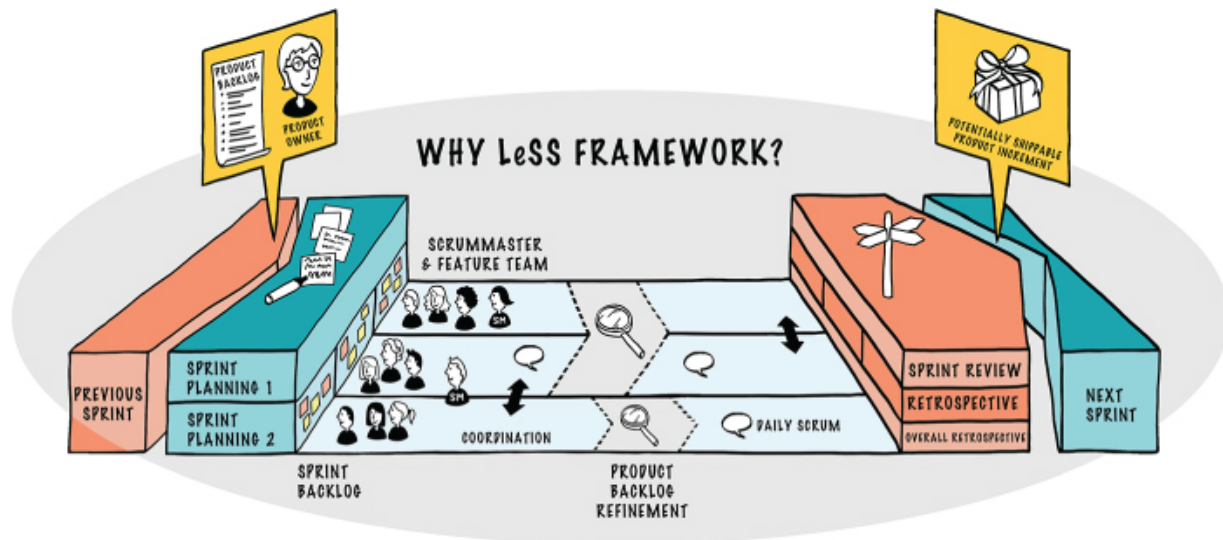


Abbildung 13.11: Das Basisframework von Less
 Mit freundlicher Genehmigung von Craig Larman und Bas Vodde

Im LeSS-Framework gibt es einen Product Owner und ein Produkt-Backlog für das gesamte lieferbare Produkt. Der Product Owner sollte nicht allein an der Ausarbeitung des Produkt-Backlogs arbeiten; er wird von mehreren Entwicklungsteams unterstützt, die direkt mit Kunden/Anwendern und anderen Stakeholdern zusammenarbeiten. Jede Priorisierung erfolgt durch den Product Owner, jedoch kann die Abklärung direkt zwischen den Teams und den Kunden/Nutzern und anderen Stakeholdern erfolgen.

Während ein Großteil von LeSS dem Ein-Team-Scrum-Framework treu bleibt, sind die folgenden Unterschiede wichtig:

- ✓ Die Sprint-Planung ist in zwei Teile untergliedert: [Teil 1](#) ist für alle Teams gleich und [Teil 2](#) gilt für jedes Team gesondert.
- ✓ Die Sprint-Planung ([Teil 1](#)) ist auf eine Stunde pro Woche beschränkt. Obwohl nicht alle Entwickler teilnehmen müssen, werden sie nicht entmutigt, dies zu tun. Es müssten mindestens zwei Mitglieder je pro Sprint-Team zusammen mit dem Product Owner teilnehmen. Die Vertreter der Teams kehren zu ihren Teams zurück und teilen die Informationen mit diesen.

- ✓ Es finden unabhängige Sprint-Planungstreffen ([Teil 2](#)) sowie Daily Scrums statt. Die Mitglieder der verschiedenen Teams können an den Meetings der anderen Teams teilnehmen, um den Informationsaustausch zu erleichtern.
- ✓ Wie die teamübergreifende Koordination stattfindet, wird von den Teams entschieden. Eine dezentrale und informelle Koordination wird einer zentralen Koordination vorgezogen. Der Schwerpunkt liegt auf informellen Netzwerken mit teamübergreifenden Gesprächen, Komponenten-Mentoren, Reisenden (Spezialisten, die sich bei Bedarf einem Team anschließen), Scouts und offenen Räumen.
- ✓ Die Backlog-Verfeinerung erfolgt für das gesamten Produkt-Backlog mit Vertretern der einzelnen Entwicklungsteams und des Product Owners. Die Verfeinerung des individuellen Team-Backlogs findet ebenfalls auf der Ebene des einzelnen Teams statt, aber die Verfeinerung des Multi-Team-Backlogs findet bei jedem Sprint statt und ist die wichtigste Praxis in LeSS.
- ✓ Sprint-Reviews werden mit Vertretern jedes Teams und des Product Owners durchgeführt.
- ✓ Zusätzlich zu den einzelnen Team-Retrospektiven finden auch Sprint-Retrospektiven statt. Scrum Master, Product Owner, Vertreter von Entwicklungsteams und Manager überprüfen und passen das Gesamtsystem des Produkts an, beispielsweise Prozesse, Tools und Kommunikation.

LeSS Huge-Framework

Mit LeSS Huge sind der Größe des gesamten Projektteams keine Grenzen gesetzt. Ein paar tausend Menschen könnten an einem Projekt arbeiten.

Die Scrum-Teams werden um größere Bereiche von Kundenanforderungen gruppiert, die Anforderungsbereiche (requirement areas) genannt werden. Diese Gruppierung ähnelt der Gruppe von Teams, die beim vertikalen Schnitt unter einem Integrationsteam zusammenarbeiten.

Für jeden Bereich gibt es einen Area Product Owner und zwischen vier und acht Scrum-Teams (das Minimum von vier Scrum-Teams in jedem Bereich verhindert zu viel lokale Optimierungen und Komplexität). Ein übergreifender Product Owner arbeitet mit mehreren Area Product Ownern; diese bilden zusammen ein Product-Owner-Team für das Projekt. [Abbildung 13.12](#) stellt das Prinzip von LeSS Huge dar.

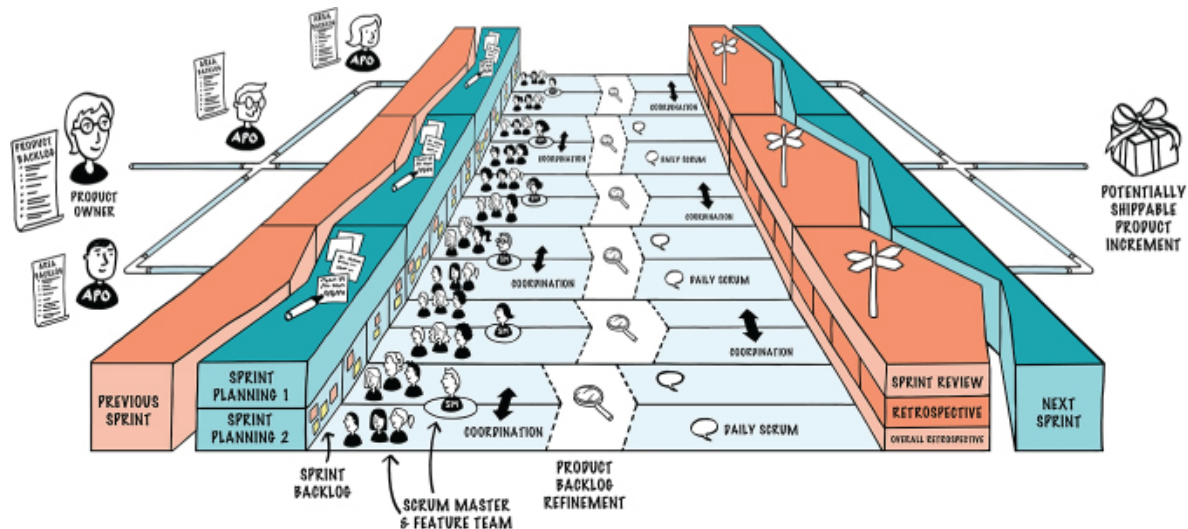


Abbildung 13.12: Das LeSS-Huge-Framework

Mit freundlicher Genehmigung von Craig Larman und Bas Vodde

So wie in Scrum mit einem Team und auch wie im Basismodell von LeSS gibt es ein Produkt-Backlog, eine Definition of Done, ein potenziell auslieferbares Produktinkrement, einen Area Product Owner sowie eine Sprint-Kadenz für alle Teams. LeSS Huge ist einfach nur ein Stapel mehrerer LeSS-Basismodell-Implementierungen für jeden Anforderungsbereich. Alle Anforderungsbereiche verwenden LeSS und alle Anforderungsbereiche zusammen ergeben LeSS Huge. Einige Unterschiede sind:

- ✓ Vor dem Sprint-Planungs-Meeting findet ein Planungstreffen der Product Owner statt.
- ✓ Es werden Meetings auf Bereichsebene hinzugefügt. Sprint-Planung, Sprint-Review und Sprint-Retrospektive werden auf

Bereichsebene durchgeführt und die Verfeinerung des Produkt-Backlogs findet ebenfalls auf Bereichsebene statt.

- ✓ Es werden Sprint-Reviews und Retrospektiven mit allen Teams durchgeführt. Dieses Review koordiniert die gesamte Arbeit und den Prozess über den gesamten Produktprogrammbereich hinweg.

LeSS ermöglicht die Implementierung von Scrum und Skalierung in einer Weise, die weitgehend den Agilen Prinzipien entspricht. Einige Elemente des Scrum-Frameworks werden durch empirisches Lernen, kurze Feedbackschleifen, Selbstorganisation und effektive Zusammenarbeit und Koordination unterstützt.

Führungsinstrumente gibt es in LeSS auch für gute Entscheidungen, die den ROI maximieren, den Kunden einen Mehrwert bieten und zufriedene, nachhaltige Teams schaffen.

J.P. Morgan und LeSS

Groß im Sinne von J.P. Morgan bedeutet, dass mehr als 3.000 Mitarbeiter an mehreren internationalen Standorten arbeiten. Ein Gruppenleiter entschied sich für die Umstellung auf LeSS. Es begann mit kleinen Pilotprojekten. Die Ergebnisse motivierten die Firma, Scrum in größerem Umfang einzuführen.

Das Unternehmen startete mit einem Unternehmensbereich – Wertpapiere – als Pilotprojekt und arbeitete an Lösungen, um einen kundenorientierten Ansatz zu schaffen. Die Kunden brauchten eher Komponentenfamilien als einzelne Komponenten. Diese kundenorientierte Herangehensweise vereinfachte die Planung und Koordination enorm, da Komponentengruppen und keine einzelnen Komponenten geplant wurden.

Ein Problem war die Auswahl der neuen Product Owner, da unter dem Wasserfallansatz die Businessabteilung nicht an der Entwicklung beteiligt war. Das Unternehmen löste dieses

Problem, indem es freiwillige Teilnehmer sowohl aus dem operativen Bereich als auch aus Forschung und Entwicklung auswählte.

J.P. Morgan bildete Anforderungsbereiche. Jedes Team erhielt ein Projekt und ein Product Owner wurde ausgewählt. Die Teams wurden ermutigt, funktionsübergreifende Funktionen zu entwickeln, um Engpässe zu reduzieren und das Risiko zu minimieren, diese eine Person zu verlieren, die als einzige eine Aufgabe erledigen kann. Die Personalabteilung eliminierte Berufsbezeichnungen wie Business Analyst und Tester. Sogar Teamleiter hatten den allgemeinen Titel des Entwicklers.

Am Ende des ersten Rollouts von LeSS im Wertpapierbereich wurden Effektivität, Kundenzufriedenheit und Mitarbeitermoral gesteigert. LeSS hat dazu beigetragen, Scrum auf diese Betriebsgröße zu skalieren und wird auch heute noch eingesetzt.

Kapitel 14

Personal und Finanzen

IN DIESEM KAPITEL

Eine Kultur schaffen, die exzellente Scrum-Team-Kandidaten anzieht

Talent für Scrum entwickeln

Fließender Übergang zu Scrum mit vorhandenen und neuen Mitarbeitern

Die Vorteile der inkrementellen Finanzierung erkennen

Den Budgetierungsprozess verbessern

Wenn der Wind der Veränderung weht, bauen die einen Mauern, die anderen Windmühlen.

Chinesisches Sprichwort

Die Babyboomer gehen in hohem Tempo in den Ruhestand, neue Mitarbeitergenerationen nehmen deren Platz ein, und das Personalwesen (Human Resources, HR) hat einen hohen organisatorischen Wert erlangt. Mittelverschwendung als Folge von gescheiterten Projekten deutet darauf hin, dass es einen intelligenteren Weg zur Finanzierung von Projekten geben muss. Wie in so vielen anderen Unternehmensbereichen sind auch beim Personalmanagement alte Methoden zur Zielerreichung überholt.

Unternehmen, die ihre HR- und Finanzprobleme erkennen und dann zu deren Lösung das Scrum-Framework anwenden können, sind immer einen Schritt voraus. Es gibt viele Möglichkeiten, das menschliche Potenzial und die finanzielle Effektivität durch Scrum zu steigern.

In diesem Kapitel führen wir Sie durch die Herausforderungen und Lösungen in diesen kritischen Organisationsbereichen.

Personal und Scrum

In einer Umfrage der Society for Human Resource Management (SHRM) lassen sich zwei zentrale Herausforderungen an die Welt der Personalarbeit von heute erkennen: die Bindung und Belohnung der besten Mitarbeiter und die Entwicklung zukünftiger Führungskräfte. Im Grunde genommen handelt es sich bei diesen beiden Interessen um das gleiche Anliegen. Ihre besten Mitarbeiter werden oft zu Ihren zukünftigen Führungskräften.

Als die Forscher die Organisationen fragten, was ihre größte Investitions Herausforderung sei, lautete die erste Antwort: »Humankapital beschaffen und Humankapitalinvestitionen optimieren.«



Wir ermutigen die Organisationen, mit denen wir zusammenarbeiten, Menschen als Menschen oder Talente zu bezeichnen und nicht als *Kapital* und *Ressourcen*. Menschen sind keine Waren. Sie verfügen über individuelle Fähigkeiten, Erfahrungen und Innovationen. Der Begriff *Ressourcen* ist, wenn er sich auf Menschen bezieht, ein Relikt. Sehen Sie Ihre Mitarbeiter als Menschen und betrachten Sie ihre Kreativität und Talente als den unersetzlichen Wert, den sie Ihrem Unternehmen bringen.

Dennoch ist die Aussage der SHRM-Umfrage nach wie vor wichtig. Es werden verschiedene Arten von Leitern benötigt. Wir brauchen Führungskräfte, die die im Kapitel über das Portfoliomanagement besprochenen harten Product-Owner-Entscheidungen treffen (siehe [Kapitel 13](#)). Wir brauchen dienende Leiter, die auf kooperative Weise arbeiten, um Teams wie zum Beispiel Scrum Master zu befähigen (siehe [Kapitel 2](#)). Diese Menschen unterstützen und befähigen selbstorganisierende Teams, die besten Lösungen zu finden. Sie schätzen iterative Ansätze zur Herstellung von Produkten, die auf schnellem und

regelmäßigem Kundenfeedback basieren, und sie wissen, wie sie dieses Feedback schnell umsetzen und darauf reagieren können.

In der Vergangenheit haben Technologieunternehmen Führungskräfte belohnt, die die Krise gut gemeistert haben. Im Gegenzug förderten diese Führer andere, die die Krise mit einem Gefühl der Dringlichkeit bewältigten. Obwohl diese Feuerwehrleute sicherlich mehr als einmal die Lage gerettet haben, behaupten wir, dass dies zu einer unausgewogenen Verschiebung hin zur Krisenbewältigung und nicht zur Krisenvermeidung geführt hat.

Wir brauchen Führungskräfte im Bereich Entwicklung, die sich dadurch auszeichnen, dass sie neue Fähigkeiten erlernen und andere bei ihrem Wachstum begleiten. Wir brauchen kooperative Führungskräfte, die im dynamischen Umfeld von Scrum arbeiten können. Wir brauchen Menschen, die komplementäre Führungsstile schätzen. Egomane müssen sich nicht bewerben. Unternehmen brauchen funktionsübergreifende Scrum-Teams, um wettbewerbsfähig zu bleiben. Mit anderen Worten, wir brauchen Führungskräfte, die den Wert der Veränderung annehmen.

Die Reduzierung von risikoreichen Single Points of Failure spart Geld aus all den Gründen, die wir bisher im Buch erörtert haben. Teams, die zusammenarbeiten, um ein Problem zu lösen, bringen Qualitätsprodukte schneller auf den Markt als Teams, die traditionelle Methoden verwenden.

Die richtige Kultur schaffen

In [Kapitel 4](#) erörtern wir, was die Mitarbeiter am meisten motiviert: nicht Geld, sondern die Autonomie und das Vertrauen, die für die Arbeit notwendig sind, die Möglichkeiten zu wachsen und ein Gefühl der Sinnhaftigkeit. Die Menschen wollen, dass das, was sie tun, Sinn stiftend ist. Außerdem wollen sie mit anderen Menschen zusammenarbeiten, die sich weiterentwickeln und sich dafür engagieren, was sie tun.

Die Früchte der Selbstorganisation und des Selbstmanagements sind das Entstehen einer Organisationskultur, die die Besten und Klügsten anzieht und hält. Wie wir in [Kapitel 1](#) erläutern, sind dies die agilen Prinzipien 5, 8, 11 und 12.



Die Entwicklung der richtigen Kultur hat viele Facetten. Das Wichtigste, was wir von einem effektiven Scrum-Teammitglied erwarten, ist Vielseitigkeit. Geben Sie uns jemanden, der intellektuell neugierig ist und das hat, was wir eine *beitragende/mitwirkende Persönlichkeit* nennen, und wir werden mit dieser Person erfolgreich sein. Wir können technische Fertigkeiten vermitteln. Primadonnen, die denken, dass diese dünne Scheibe der Entwicklung das Einzige ist, was ihrer Zeit würdig ist, werden das ganze Team beflecken. Die wichtigste Aufgabe in einem Scrum-Team ist die Aufgabe, die für die Auslieferung des Produkts notwendig ist. Manchmal ist es die Programmierung, manchmal die Qualitätssicherung, manchmal die Dokumentation. Was auch immer diese Aufgabe ist, es ist die höchste und kritischste Aufgabe des gesamten Projekts.

Entscheidend für die Schaffung einer attraktiven Organisationskultur ist die Haltung der Führungskräfte: Welche Qualitäten sie begrüßen sowie fördern und wie sie in ihre Mitarbeiter investieren. Wie ein Magnet ziehen Führungskräfte Menschen an, die sich von der Kultur angezogen fühlen, die sie bieten.

Eine breite Organisationskultur ist wichtig, aber dies gilt auch für taktische Teamfähigkeiten. Entscheidend ist die funktionsübergreifende Zusammenarbeit innerhalb von Teams, die die Fähigkeit zum Swarming fördert. Entscheidend sind Eignung und Fähigkeiten und nicht die Titel. Ein funktionsübergreifendes, selbstorganisierendes Team ist die perfekte Umgebung für die Entwicklung von Fähigkeiten. Wenn die Mitarbeiter wachsen, können sie sich besser für das Ziel und den Zweck der Organisation engagieren.



Die Technologie ändert sich zu schnell, um darin verhaftet zu bleiben, nach Spezialisten zu suchen, die sich nur mit einer einzelnen Technologie auskennen. Die technischen Standardverfahren von heute können völlig anders sein als die, die morgen gebraucht werden. Die Spezialität von morgen ist die Fähigkeit, schnell zu lernen und sich anzupassen. Sie brauchen Teammitglieder, die das erkennen.

Durch die Betonung der Fähigkeiten anstatt der Titel entsteht eine Kultur, in der man nicht für jede Fähigkeit eine Person einstellen muss. Expertise (nicht Spezialisierung) ist immer erforderlich. Man braucht nicht auf jedem Platz einen Spezialisten.

Situative Führung ist wichtig. Wenn Jim Erfahrung mit .NET hat, werden Sie sich wahrscheinlich an ihn wenden, wenn Sie mit einem .NET-Problem konfrontiert sind. Carol hat vielleicht Erfahrung in der Qualitätssicherung. Wenn es um die Qualitätssicherung geht, können Sie sich an Carol wenden. Sam mag gerade von einer .NET-Konferenz zurückgekommen sein, und obwohl Jim vielleicht stark in .NET ist, hat Sam vielleicht etwas gelernt, was Jim nicht weiß.

Die folgenden Abschnitte beschreiben zwei Möglichkeiten, HR und Scrum zu betrachten.

HR und bestehende Organisationsstrukturen

In diesem Abschnitt werden Möglichkeiten zur Organisation und zum Management der vorhandenen Mitarbeiter innerhalb einer Scrum-Struktur untersucht.

Anreize schaffen

Erzwungene Ranglisten und Anreizstrukturen, die auf Wettbewerb basieren, fördern die Konkurrenz zwischen den Mitarbeitern. Scrum ist teamorientiert. Das Team erhält Lob und Verbesserungsvorschläge. Daher sollten Incentives auch auf

Teamebene erfolgen. Wenn das Team erfolgreich ist, bekommt es den Preis.

Wenn eine Fußballmannschaft den Super Bowl gewinnt, bekommt jede Person im Team einen Ring, egal ob sie an diesem Tag gespielt hat oder nicht. Wenn ein Entwicklungsteam ein gutes Jahr hat, könnte jeder im Team einen Bonus von 15 Prozent bekommen, nicht nur ein paar ausgewählte Leute. Abhängig von ihrem Dienstalter können 15 Prozent signifikant sein.

Bei großen Teams kann es sein, dass diese Praxis nicht funktioniert. Aber bei Scrum bestehen die Entwicklungsteams aus drei bis neun Mitgliedern, sodass es für ein oder zwei Teammitglieder schwierig ist, sehr lange unbemerkt zu bleiben. Ein selbstorganisierendes, selbstverwaltendes Team fordert den Beitrag jedes Teammitglieds, damit am Ende jeder den Preis verdient.

Vergütung

Wenn ein Team funktionsübergreifend arbeitet und keine Titel oder Spezialisten vorhanden sind, wie wird dann der Status festgelegt? Sie erzeugen Status durch Wissen und dadurch, dass Sie der Ansprechpartner für ein bestimmtes Thema sind. Jemand, der im Team Schwerstarbeit leistet, hat eine Menge Aufgaben in der Spalte *Erledigt* des Taskboards stehen, und andere Leute wollen mit diesen Personen im Paar arbeiten oder an Aufgaben im Shadowing arbeiten.

Das Dienstalter und die relative Vergütung werden durch eine Kombination aus Qualifikationstiefe und -breite festgelegt. Jedes Mitglied eines Entwicklungsteams hat mindestens eine Fähigkeit. Wenn Sie eine Fertigkeit besitzen, dann gehören Sie zur Rangstufe 1. Rang Gleichgültig ob Sie eine Einstiegskraft oder eine Führungskraft sind, mit einer Fertigkeit gehören Sie zur Rangstufe 1.

Wenn Sie zwei Fertigkeiten beherrschen, dann gehören Sie zur Rangstufe 2. Sie können entweder eine Einstiegskraft mit Rangstufe 2 sein oder eine Führungskraft mit Rangstufe 2. In

beiden Fällen gehören Sie zur Rangstufe 2. Mit drei Fähigkeiten gehören Sie zur Rangstufe 3 und so weiter.

Mit jeder neuen Fertigkeit ändert sich die Rangstufe des Entwicklers. Während das Team durch Shadowing, Swarming, Arbeiten in Paaren und Wissensaustausch reift, sollte jedes Teammitglied seine Fähigkeiten erweitern (crossfunktional werden). Wenn Teammitglieder ein höheres Niveau erreichen (zum Beispiel Rangstufe 2 und Rangstufe 3), steigen ihre Gehälter entsprechend an.

Diese Situation ist eine echte Win-Win-Situation, denn eine Organisation kann Teammitgliedern mehr bezahlen *und* Geld sparen, indem sie Teams von funktionsübergreifenden Mitarbeitern hat, anstatt eine Armee von Scharfschützen, die nur eine Sache beherrschen.



Indem Sie Entwickler dazu anregen, mehr Fähigkeiten zu erlernen und damit ihren Status zu erhöhen, geben Sie ihnen die Möglichkeit, ihren eigenen Ausbildungs- und Karrierepfad zu gestalten. Oft sind die Fähigkeiten, die sie entwickeln, diejenigen, die sie am meisten interessieren oder für die sie die größte Neigung haben. Sie können an neue Fähigkeiten herangeführt werden, die sie vorher nie ausprobiert hätten, für die sie aber eine Begabung haben.

Teammitglieder haben einen Anreiz, mehr Fähigkeiten zu erwerben, und das Unternehmen profitiert, indem es weniger Menschen höhere Gehälter zahlt und Scrum-Teams entwickelt, die effektiver und effizienter sind.

Leistungsschwache Teammitglieder

Wenn jemand in einem selbstorganisierenden, selbstverwaltenden Team nicht sein Gewicht in die Waagschale wirft, deckt Scrum diese Tatsache schnell auf, damit das Team das Problem beheben kann. In [Kapitel 4](#) erörtern wir den Hawthorne-Effekt, der zeigt, dass sich die Leistung eines Arbeiters verbessert, wenn er beobachtet wird. Sichtbarkeit und Leistung stehen zueinander in direkter Proportionalität. Scrum

verwendet auch Informationsverteiler wie Taskboards und Burndown-Diagramme, die die Sichtbarkeit und Performance erhöhen.



Ein Kunde hatte mehrere Flachbildfernseher in der Lobby seines Gebäudes. Die Bildschirme wurden in Quadranten aufgeteilt und zeigten den Burndown des Tages von vier Projekten gleichzeitig an, wobei alle 20 Sekunden neue Projekte angezeigt wurden. Jede Person, die diese Lobby betrat (einschließlich Lieferanten und Kunden), konnte den Status jedes Projekts sehen, an dem das Unternehmen arbeitete – eine Situation mit hoher Sichtbarkeit für jedes Scrum-Team und jedes Teammitglied.

Mangelnde Transparenz in Bezug auf Ziele und Erwartungen kann als Ursache für unzureichende Leistung ebenfalls eine Rolle spielen. Wenn eine Person nicht genau weiß, wohin sie unterwegs ist, ist es verständlich, dass sie vom Kurs abkommt.

In Scrum lässt sich ungenügende Leistung leicht in Burndown-Charts erkennen, in denen die täglichen Aufgaben als erledigt oder nicht erledigt gekennzeichnet sind. Der neue Code wird jede Nacht im Rahmen der kontinuierlichen Integration durch automatisierte Tests automatisch getestet. Mit dieser Art der Sichtbarkeit, bei der das Team als Einheit verantwortlich ist, lassen sich Fehler leicht finden und beheben.

In einem traditionellen Modell kann ein Entwickler, der jeden Tag drei Stunden lang auf Facebook rumhängen möchte, dies ganz einfach tun. Er kann dem Projektleiter sagen, dass er 4 Stunden braucht, um eine 45-minütige Aufgabe zu erledigen. Wenn der Projektleiter sagt, dass er die Arbeit in kürzerer Zeit erledigen könnte, könnte seine Antwort lauten: »Wirklich? Zeig mir, wie.« Da der Projektmanager wahrscheinlich nicht über die notwendigen Programmierkenntnisse verfügt, um die Aufgabe vollständig zu verstehen, besteht seine einzige Möglichkeit darin, den Entwickler mit allem, was er tun möchte, weitermachen zu lassen. Der Rest des Teams ist der Meinung, dass, solange sie

ihre Arbeit machen, das Problem dieser Entwickler und der Projektmanager ist.

Bei Scrum werden die Teams jedoch gemeinsam und als Einheit zur Verantwortung gezogen. Wenn ein Entwickler drei Stunden auf Facebook verbringt, werden seine Teamkollegen ihn dafür verantwortlich machen, dass er seinen Beitrag nicht geleistet hat.

Hier sind einige andere Möglichkeiten, wie Scrum Performanceprobleme aufdecken kann:

- ✓ Bevor es zur Eskalation eines Problems kommt, können die Teammitglieder im Daily Scrum vorschlagen, dass sich ein problematischer Entwickler an einer Aufgabe beteiligt. Dieser Vorschlag ist die Idee des Teams und kann schon ausreichend sein, um das Problem zu lösen.
- ✓ Sprint-Retrospektiven sind ausgezeichnete Momente, um Lücken in den Fähigkeiten und möglichen Lösungen zu identifizieren. Ursachen von unzureichender Leistung können aufgedeckt werden, wie etwa Teamdynamik, Umfeld und Missverständnisse.
- ✓ Im Sprint-Review kann das Team die Person, die Schwierigkeiten hat, ermutigen, sich an der Demo zu beteiligen. In Vorbereitung auf die Demo können Dinge ans Licht kommen, die Anregungen für die Retrospektive geben.
- ✓ Wenn alle Entwickler wissen, dass eine Person die Last nicht tragen kann, aber es ihnen unangenehm ist, das Problem selbst anzusprechen, wird ein guter Scrum Master die Interaktionen erleichtern, die nötig sind, um das Problem anzugehen.

HR und Scrum bei der Einstellung

Wenn Sie organisatorische Änderungen vornehmen und Scrum-Praktiken anwenden, werden Sie neue Mitarbeiter nach anderen Kriterien als bisher suchen und einstellen. Im Folgenden finden Sie einige Dinge, die Sie beachten müssen, wenn Sie Personen für die verschiedenen Scrum-Rollen einstellen:

✓ Entwickler

- Entfernen Sie Titel aus den Stellenbeschreibungen und nehmen Sie die gewünschten Fähigkeiten als Grundlage für Ihre Suche. Sie filtern Ihre Suchergebnisse nicht nur nach den Schlüsselwörtern, die die Kandidaten in ihren Profilen angeben, sondern auch nach Kandidaten, die mehr daran interessiert sind, ihre Titel als ihre Fähigkeiten zu bewerben.
- Suchen Sie nach Kandidaten, die neugierig sind und den Wunsch haben, sich funktionsübergreifend weiterzubilden, das sind diejenigen, die ihre Fähigkeit bewiesen haben, außerhalb ihrer Komfortzonen zu arbeiten.
- Suchen Sie nach Generalisten, die über breite Fähigkeiten und Erfahrung verfügen und in der Lage sind, zwischen diesen Fähigkeiten hin und her zu schalten.
- Finden Sie Menschen, die nach Möglichkeiten suchen, mit einem Team zu arbeiten. Selbst die introvertierteste Seele arbeitet gerne mit respektvollen Teamkollegen.

✓ Product Owner

- Achten Sie auf das Hauptmerkmal der Entschlossenheit.
- Finden Sie Leute, die mit Entwicklern und Stakeholdern zusammengearbeitet haben.
- Suchen Sie die nach sozialen Kompetenzen im Bereich Kommunikation und Proaktivität. Eine effektive und rechtzeitige Klärung ist entscheidend für den Erfolg eines Scrum-Teams.

✓ Scrum Master

- Sie suchen nach sozialen Kompetenzen wie beispielsweise Konfliktmoderation und nachweisbarer Wirkung auf die Leistungen früherer Teams.

- Suchen Sie nach jemandem, der das Team beschützen, Einfühlungsvermögen zeigen und als dienender Leiter Mentor des Teams sein kann.
- Schreiben Sie die Stelle nicht für einen Projektmanager mit der Hoffnung aus, diese Person in einen Scrum Master zu verwandeln. Die Rolle des Projektmanagers existiert bei Scrum nicht und sie passt nicht in dieses Konzept.

Unabhängig von der Position, die Sie zu besetzen versuchen, suchen Sie nach Leuten, die eine positive Erfahrung mit Scrum und/oder Agilität haben. Überprüfen Sie regelmäßig die Stellenbeschreibungen und aktualisieren Sie sie mit der entsprechenden Scrum-Rollenterminologie und den Agilen Prinzipien, um Kandidaten zu gewinnen, die es vorziehen, mit Scrum zu arbeiten.

Erfolgskontrolle

Nachdem Sie Ihre neuen Scrum-fokussierten Mitarbeiter eingestellt haben, können Leistungsbeurteilungen teambasiert durchgeführt werden. Belohnen Sie Menschen nach ihrem Beitrag zum Team, indem Sie den Schwerpunkt vom Einzelnen auf das Team verlagern.

Leistungsbeurteilungen auf Teamebene sind bei Scrum eine Selbstverständlichkeit. Angesichts der hohen Sichtbarkeit, die diesem Framework innewohnt, erfolgt die tägliche Bewertung durch Peers, und die Teamleistung ist konsistent und greifbar.

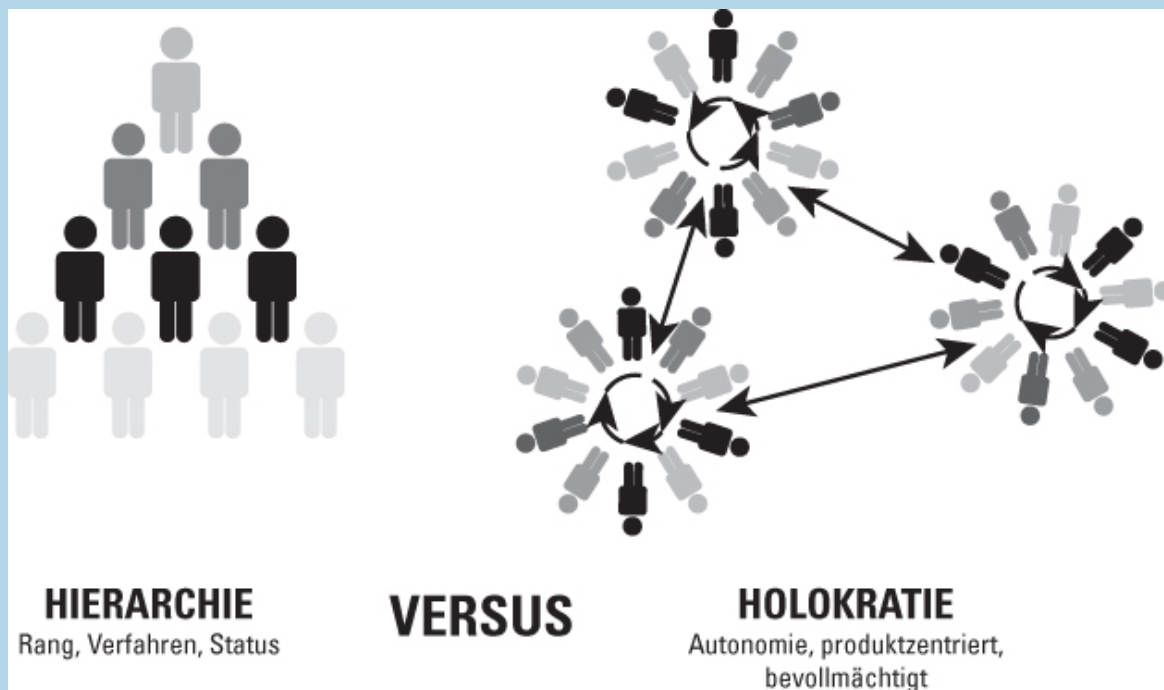
Holokratie bei Zappos

Holokratie ist ein System der organisatorischen Steuerung, das aus selbstorganisierenden Teams und nicht aus traditionellen Hierarchien besteht. Man kann es sich als eine Hierarchie von Kreisen vorstellen, in der jeder Kreis ein selbstorganisierendes Team ist, das nach demokratischen Prinzipien geführt wird. Zappos war die größte Organisation

(damals 1.500 Mitarbeiter), die das Ideal der Holokratie mit Erfolg einsetzte.

John Bunch, einer der führenden Kräfte des Transitionsprozesses bei Zappos, sagte: »Eines der Kernprinzipien besteht darin, dass Menschen persönliche Verantwortung für ihre Arbeit übernehmen. Holokratie ist nicht gleichbedeutend mit Führungslosigkeit. Es gibt sicherlich Menschen, die einen größeren Anwendungsbereich für die Organisation haben als andere. Was Holokratie macht, ist, die Führung auf die verschiedenen Rollen aufzuteilen. Von jedem wird erwartet, dass er in seiner eigenen Rolle führt und ein Unternehmer ist, und Holokratie befähigt ihn dazu. «

Die folgende Abbildung stellt das Modell der Holokratie visuell dar.



Im Idealfall, sagt Bunch, ist Holokratie »politikfrei, entwickelt sich schnell weiter, um den Zweck der Organisation zu definieren und zu betreiben, und reagiert auf Markt- und Realbedingungen in Echtzeit. Es geht darum, eine Struktur zu

schaffen, in der die Menschen die Flexibilität besitzen, dem nachzugehen, wo ihre Leidenschaften liegen.«

Holokratie ist das Scrum-Konzept selbstorganisierender, selbstverwaltender, funktionsübergreifender Teams, die auf der Organisationsebene eingesetzt werden.

Formale jährliche Überprüfungen einzelner Mitarbeiter sind zum größten Teil Artefakte traditioneller Projektmanagement-Frameworks. Sie haben sich aus mehreren Gründen als unwirksam erwiesen:

- ✓ Die Menschen brauchen und wollen regelmäßiges Feedback und nicht nur eine Überprüfung am Ende eines jeden Jahres. Regelmäßiges Coaching ist der Schlüssel dazu.
- ✓ Es ist schwierig, die Leistung eines ganzen Jahres mit all seinen Nuancen und situativen Faktoren in einem Review zu beurteilen. Die Überprüfung wird naturgemäß unvollständig sein.
- ✓ Diejenigen mit schlechten Leistungen sollten früh gecoacht und betreut werden, nicht erst nach einem Jahr.
- ✓ Positives, konstruktives Feedback ist viel effektiver als das Beurteilungsformat eines jährlichen Reviews.

Der logischste und einfachste Weg, um Feedback zu geben, das den Teammitgliedern hilft, sich zu verbessern, besteht darin, dass die Kollegen jeder Person ein 360-Grad-Feedback geben (siehe [Abbildung 14.1](#)). In Scrum-Teams sind dies Teammitglieder, Stakeholder und Kunden. Statt einer formellen Leistungsbeurteilung ist eine 360-Grad-Bewertung ein Feedback-Tool. Jedes Mitglied des Teams kann die Wirkung seiner Arbeit auf jeden anderen Beteiligten nachvollziehen. Diese ganzheitliche Sichtweise zeigt, wie sich ein Mitarbeiter im gesamten Arbeitsumfeld verhält.



Abbildung 14.1: Das 360-Grad-Review zur Leistungsbeurteilung

Verwenden Sie 360-Grad-Feedback zur Leistungsbewertung, um die Vergütung festzulegen und (noch wichtiger) zur Unterstützung der Teammitglieder bei der Überprüfung, Anpassung und Verbesserung. Wenn Sie sie nur zur Erfüllung von HR-Anforderungen oder zur Ermittlung von Leistungssteigerungen einsetzen, verpassen Sie die volle Leistungsmöglichkeiten von 360-Grad-Reviews.

Einige Vorteile eines 360-Grad-Reviews sind:

- ✓ Die Betrachtung der Leistung eines Mitarbeiters aus verschiedenen Blickwinkeln ermöglicht ein besseres Verständnis dafür, welche Fähigkeiten verbessert werden müssen und welche herausragend sind.
- ✓ Das Wachstum hängt von häufigem, qualitativ hochwertigem Feedback ab.
- ✓ Ansatzpunkte zur Verbesserung der Fähigkeiten können identifiziert werden.

- ✓ Es kann ein Ausgangswert für das Messen der Verbesserung festgelegt werden.
- ✓ Persönliche blinde Flecken im Verhalten eines Mitarbeiters können identifiziert werden.

Einstellungen mit Scrum managen

Scrum beeinflusst, wie Sie Talente einstellen und entwickeln, um die Leistung von Scrum zu optimieren. Sie können Scrum auch für Einstellungs- und Onboarding-Projekte verwenden.

Ein globaler Anbieter von biopharmazeutischen und medizintechnischen Lösungen nutzte Scrum für einen großen Rekrutierungsschub, der 21 neue Mitarbeiter erforderte, die geschult und einsatzbereit waren. Es wurden Entwickler, Datenbankadministratoren, Business Analysten und technische Redakteure benötigt. Vom Beginn des Rekrutierungsprozesses bis zum Projektstart hatte das Einstellungsteam nur sieben Wochen Zeit, um diese Personen zu finden, einzustellen und zu schulen.

Die ersten Herausforderungen waren umfangreich:

- ✓ Derzeit gibt es keinen Onboarding-Prozess.
- ✓ Es gab kein formelles Neuanstellungsprogramm.
- ✓ Der derzeitige Rekrutierungsprozess war ineffizient.
- ✓ Es gab keine eigene Schulungsabteilung oder -inhalte.
- ✓ Neue Geräte wie Workstations, Telefone und Computer wurden benötigt.

Das Unternehmen nutzte das Scrum-Framework für das Management dieses Projekts integriert. Zu Beginn versammelte der Product Owner (der Leiter der Personalabteilung) alle einstellenden Manager, den IT-Manager und die HR-Mitarbeiter zu einem eintägigen Planungsgespräch. Sie identifizierten alles, was sie tun mussten, und wer jeden Teil erledigen würde, wie etwa

- ✓ Sammlung und Erstellung von Inhalten für Unternehmens-, Produkt- und Branchenübersichten, agile Entwicklungs- und Softwareübersichten sowie IT- und HR-Orientierungen,
- ✓ Erstellung eines einwöchigen Schulungsprogramm vor Ort unter Verwendung des Handbuchs,
- ✓ neue Gebäude in Betrieb nehmen,
- ✓ Einkauf der Geräte und Netzwerk- und Software-Tools,
- ✓ Rekrutieren der interne Trainer (Fach- und Führungskräfte),
- ✓ aktuelles Material sammeln und vereinheitlichen, das in verschiedenen Formen existiert.

Nach diesem einzigen Tag der Planung konnten sie Roadmap-Features und Produkt-Backlog-Epics erstellen für

- ✓ Schulungshandbuch,
- ✓ Beschaffung von IT-Geräten,
- ✓ Beschaffung von IT-Software, Lizenzen und Berechtigungsnachweisen,
- ✓ Einrichtung der neuen Büros,
- ✓ Prozess zur termingerechten Einstellung neuer Mitarbeiter.

Für die Rekrutierung der neuen Mitarbeiter identifizierten sie auch den Bedarf an

- ✓ einem Portal und Verfahren zur Prüfung von Lebensläufen,
- ✓ einem Interviewplan,
- ✓ Standardtests und Fragen für die Einstellungsgespräche.

Die Manager benutzten einwöchige Sprints. Jede Woche demonstrierten alle den Fortschritt an ihrem Teil des Handbuchs und nahmen das Feedback der anderen entgegen. Die IT zeigte die Fortschritte bei den Anlagen und Ausstattungsanforderungen. Sie erstellen das Sprintziel für die nächste Woche und gingen wieder an die Arbeit.

Die wöchentlichen Schnittstellen zur IT waren wichtig. Mehrmals wurden Hindernisse vor Ort identifiziert und konnten behoben werden, damit die IT fristgerecht fertig werden konnte.

Die einstellenden Manager gaben sich gegenseitig Feedback zu den Inhalten der Schulungsunterlagen. Auf diese Weise war am Ende keine umfassende Bearbeitung mehr nötig. Als die Schulung begann, wussten sie, dass ihr Inhalt auf dem Weg geprüft und angepasst worden und bereits sehr gut ausgearbeitet war, selbst wenn sie nicht die Zeit gehabt hatten, das gesamte Handbuch zu lesen.

Jede Woche fanden Sprint-Retrospektiven statt: Deren Ergebnis waren oft Pläne zur Verbesserung der Koordination und Probleme mit dem Anbieter des Einstellungsportals. Probleme wurden rechtzeitig behoben, da ihnen wöchentlich Aufmerksamkeit geschenkt wurde. Außerdem wurde jemandem die Aufgabe übertragen, sich im Rahmen des Follow-up-Aktionsplans mit allen Fragen zu befassen.

In diesen organisatorischen Rahmen wurde der Einstellungsprozess integriert. Eine Onboarding-Firma stellte ein Portal für die Verwaltung der Kandidaten zur Verfügung. Dieses Portal machte jede Phase des Einstellungsprozesses sichtbar und half bei der Raumplanung und dabei, die Kandidaten von einer Interviewphase zur nächsten zu bringen.

Als Ergebnis des koordinierten Scrum-Frameworks erhielten alle Neueinstellungen die gleiche Ausbildung und besaßen umfassendes Verständnis ihrer Rolle.

Nach dieser ersten Einstellungswelle wurde das System inspiziert und angepasst und verbessert. Insgesamt bedeutete der Scrum innewohnende Überprüfungs- und Anpassungszyklus, kombiniert mit dem Swarming zur Erstellung jedes einzelnen Features, einen bemerkenswerten Erfolg in einem potenziell alptraumhaften Einstellungsprojekt.

Finanzen

Das Ziel von Unternehmen ist es, Geld zu verdienen, um Produkte oder Dienstleistungen anzubieten und zu verbessern. Deshalb steht die Finanzierung im Mittelpunkt eines jeden erfolgreichen Projekts. Zunehmender Wettbewerb und knappe Budgets erfordern harte, kluge Finanzierungsentscheidungen.

Milliarden von Dollar werden jedes Jahr für Projekte verschwendet, die entlang ihrer langwierigen und schmerzhaften Entwicklung scheiterten oder starben. Hier nur einige Beispiele:

- ✓ Ford Motor Co. hat nach der Bereitstellung ein Einkaufssystem aufgegeben, was zu einem Verlust von 400 Millionen Dollar führte.
- ✓ J Sainsbury PLC (UK) hat ein Supply-Chain-Management-System nach der Einführung aufgegeben, was zu einem Verlust von 527 Millionen Dollar führte.
- ✓ Hewlett-Packard erlitt Verluste in Höhe von 160 Millionen Dollar aufgrund von Problemen mit einem ERP-System (Enterprise Resource Planning).
- ✓ AT&T Wireless hatte Probleme mit einem Upgrade des Kundenmanagementsystems, was zu einem Verlust von 100 Millionen Dollar führte.
- ✓ McDonald's annullierte sein innovatives Informationsbeschaffungssystem, nachdem es dafür bereits 170 Millionen Dollar ausgegeben hatte.

Dies sind nur einige Beispiele für vermeidbare finanzielle Verschwendung.

Eine von KPMG durchgeführte Umfrage zum globalen IT-Projektmanagement ergab einige überraschende Ergebnisse. Als die Befragten nach ihren Erfahrungen bei der Wertschöpfung in Projekten gefragt wurden, ergab sich dieses Bild:

- ✓ 49 Prozent gaben an, in den letzten 12 Monaten mindestens mit einem Projekt gescheitert zu sein.
- ✓ 2 Prozent erreichten den angestrebten Nutzen bei jedem Projekt in den gleichen 12 Monaten.
- ✓ 86 Prozent gaben an, dass sie bezogen auf ihr gesamtes Projektportfolio bis zu 25 Prozent der angestrebten Vorteile verloren hätten.

Das meiste davon war wahrscheinlich vermeidbar.

Inkrementelle Finanzierung

Es überrascht nicht, dass Unternehmen ein erstes Ergebnis sehen wollen, bevor sie sich entscheiden, ob sie ein Projekt weiter finanzieren wollen. Mit Scrum werden bei jedem Sprint Produktinkremente produziert und für jeden Release-Zyklus verpackt, sodass eine inkrementelle Finanzierung problemlos möglich ist.



Die *inkrementelle Finanzierung* ist eine finanziell gesteuerte Methode der Projektförderung. Sie konzentriert sich auf die Maximierung der Rendite durch die Bereitstellung von sequenzierten, portionierten, vom Kunden bewerteten Funktionen, um den Nettobarwert eines Projekts zu maximieren – den Barwert der zukünftigen eingehenden Cashflows abzüglich des Kaufpreises und aller zukünftigen ausgehenden Cashflows.

Was früher in Unternehmen geschah, war, dass das Geschäftsteam vor einem Finanzierungsausschuss mit einem Vorschlag und ROI-Zahlen (Return on Investment) stand, die die Teammitglieder für notwendig hielten, um ihr Projekt vollständig zu finanzieren. Wenn sie erfolgreich waren, gab das Team jeden Cent aus und lieferte dann Ergebnisse oder bat um mehr Mittel. Nur wenige Leute schauten sich die ROI-Zahlen von vorherigen Monaten oder Jahren an, weil die Firma doch nicht in der Lage war, dieses Geld irgendwie zurückzubekommen. Versunkene

Kosten bedeuteten, dass die Unternehmen um jeden Preis vorankommen mussten.

In Scrum geht der Product Owner mit seinem Vision Statement und seiner monetären Produkt-Roadmap vor ein Förderkomitee. Er könnte sagen: »Ich brauche 3 Millionen Dollar, um all diese Funktionen zu implementieren.« Das Förderkomitee kann antworten: »Okay, wir stellen 3 Millionen Dollar für Ihr Projekt bereit. Aber zuerst geben wir Ihnen 500.000 Dollar. Zeigen Sie uns, dass Sie den für Ihr erstes Release versprochenen ROI einhalten können, und wir geben Ihnen mehr Geld. Wenn Sie nicht in der Lage sind, ein wenig zu liefern, dann sind Sie auch nicht in der Lage, viel zu liefern.«

Viele Unternehmen ziehen es vor, ihre Projekte mit oder ohne begleitende Scrum-Praktiken zu finanzieren. Die inkrementelle Finanzierung wird für Folgendes verwendet:

- ✓ Risiken verringern. Stakeholder und Product Owner können den prognostizierten ROI mit minimalen Kosten testen, um zu sehen, ob sie ihn erreichen.
- ✓ Kosten reduzieren. Eine minimale Investition wird am Anfang verwendet.
- ✓ Renditen maximieren. Frühere Monetarisierung führt zu höheren Umsätzen.



Eine weitere Möglichkeit, die Teamfinanzierung zu betrachten, ist das MVP (minimum viable product), das wir in den [Kapiteln 5](#) und [13](#) vorstellen. Sie sollten in der Lage sein, das MVP zu monetarisieren (expliziter ROI). Wenn das MVP den prognostizierten ROI realisiert, finanzieren Sie das nächste MVP.

Die inkrementelle Finanzierung bietet Product Ownern und Stakeholdern bei jedem Release die Möglichkeit, ihren ROI zu überprüfen. Je nach Größe und Komplexität der Probleme können

die Beteiligten bei jedem Release in Korrekturen investieren oder das Projekt beenden, bevor mehr Geld verschwendet wird.

Es werden früher Umsätze gemacht. Das Ziel ist nicht nur weniger Vergeudung, sondern eine frühere Rendite – und eine frühere Rendite bedeutet eine höhere Rendite. Der Wert wird dem Kunden früher geliefert und sein Buy-In steigt. Die Kontrolle von Risiko und Kosten ist wichtig, ebenso wie die Wertschöpfung. Wenn Sie Möglichkeiten erkennen können, wie Sie früher bestimmte Funktionen zur bereitstellen können, mit denen sich Umsätze erzielen lassen, sollten Sie diese zunächst ausklammern und ausliefern, damit der Kunde noch vor Ende des gesamten Projekts Umsätze erzielen kann. Wenn der Umsatz eintrifft, werden die anfänglichen Kosten früher ausgeglichen, und über die Lebensdauer des Produkts sind die Gesamterlöse und der Gewinn höher.

Historisch gesehen wird der ROI eines Projekts prognostiziert und die Finanzierung erfolgt auf der Grundlage dieser Projektion. Projekte werden selten abgebrochen, da Tests und Kundenfeedback erst zum Ende des Entwicklungszyklus stattfinden. Selbst wenn Probleme entdeckt werden, ist es schwierig, Geld zurückzunehmen, nachdem es zugewiesen wurde. Es überrascht nicht, dass die ROI-Projektionen manchmal überhöht sind, um Mittel zu erhalten.

Mit einer inkrementellen Finanzierung wird der ROI auf eine ehrliche Grundlage gestellt, da er bei jedem Release nachgewiesen werden muss. Wenn das Inkrement nicht geliefert werden kann, wird kein Geld mehr investiert.

Diese geringeren Kosten des Scheiterns ermöglichen es Entwicklern, sich auf ihre Kreativität zu besinnen und sogar eine unternehmerische Kultur zu bilden. Frische Ideen können getestet werden und wenn der ROI erreicht ist, können sie fortgesetzt werden. Unternehmenskulturen verändern sich, da Vertrauen durch greifbare gelieferte Ergebnisse aufgebaut und der innerbetriebliche Geist gehegt wird.

Fehlschläge gibt es nicht mehr; sie werden durch konsequentes Lernen mit Blick auf den nächsten Erfolg ersetzt, wenn die Kosten niedrig sind und Innovationen gefördert werden.

Selbstorganisierende Teams haben die Freiheit und Motivation, ihre eigenen besten Designs und Lösungen zu entwickeln.

In den folgenden Abschnitten werden Standard-Rechnungslegungspraktiken erläutert, die diesem inkrementellen Finanzierungsansatz entsprechen.

Kreativkultur

Intrapreneurship bedeutet, sich innerhalb einer großen Organisation wie ein Unternehmer zu verhalten. Diese Art des Denkens hilft Unternehmen, innovativ zu sein und Wettbewerbsvorteile zu erlangen. Organisationen, die ihre Mitarbeiter ermutigen, neue Ideen zu formulieren und Dinge außerhalb ihrer zugewiesenen Projekte auszuprobieren, haben einige der nützlichsten Produkte hervorgebracht, darunter Lockheed Martins ADP, GORE-TEX[®] Stoff, Googles Gmail, 3Ms Post-its (ein persönlicher Favorit), Sonys PlayStation und Facebooks »Like«-Button.

Statement of Position (SOP)

Das American Institute of Certified Public Accountants veröffentlicht Stellungnahmen (SOPs) zu Fragen der Rechnungslegung. SOP 98-1 definiert den Punkt in einer internen Softwareentwicklung, an dem Projektkosten als Aufwand erfasst (das heißt aktiviert) werden:

- ✓ Aufwendungen in der Vorplanungsphase werden in der Periode als Aufwand erfasst, in der sie anfallen.
- ✓ Kosten, die während der Entwicklung anfallen, können aktiviert oder aufgeteilt werden, um in den Gesamtpreis des Software-Assets einbezogen und dann als Kosten während der Lebensdauer des Produkts abgeschrieben zu werden.

Scrum ermöglicht es, nach der Planung und während des gesamten Projekts mehr Kosten zu aktivieren, da Planung und Gemeinkosten geringer sind und die Wertschöpfungsaktivitäten höher. Der größte Teil eines Sprints wird für wertschöpfende Aktivitäten verwendet, während ein traditionelles Modell genauso viel Zeit für Planung und Analyse aufwendet wie für die Entwicklungsphase.

Scrum ermöglicht es, dass die Kosten für die Geschäftstätigkeit inkrementell anfallen, anstatt das gesamte Projekt im Voraus zu verbuchen. Der Gewinn kann viel früher realisiert werden, wenn Sie die Anfangskosten über die gesamte Projektlaufzeit verteilen. Je schneller Sie dem Kunden einen Mehrwert liefern, desto schneller kommt der Umsatz. Dieser Kapitalisierungsansatz bedeutet, dass der Gewinn eher früher als später realisiert wird.

Scrum und Budgets

Traditionelle Budgetierungsverfahren, insbesondere in Personengesellschaften, beinhalten die Erstellung eines Budgets für das gesamte Geschäftsjahr. Dieses Budget soll alle Ausgaben und Einnahmen für dieses Jahr erfassen.

Das Management verpflichtet sich auf der Grundlage dieser geschätzten Budgetprognose. Die Prognose wird außerdem verwendet, um die tatsächlichen Ergebnisse zu bewerten und um Boni oder andere Belohnungen festzulegen. Wenn ein Manager oder eine Produktgruppe mehr einnimmt als geschätzt oder weniger ausgibt als geschätzt, wird er oder sie belohnt. Wenn der Manager oder die Gruppe die Umsatzziele nicht erreicht oder die geschätzten Ausgaben überschreitet, müssen Fragen beantwortet werden, aber die Bewertung der Über- oder Unterperformance erfolgt durch Reflexion der Vergangenheit; es handelt sich um einen reaktiven Prozess.

Organisationen sollten ein Jahresbudget als das akzeptieren, was es ist: eine Schätzung. Die Schätzung für das nächste Quartal sollte das tatsächliche Ziel sein; das Team sollte hieran seine Ziele messen. Das Team versucht, das Budget für dieses eine

Quartal einzuhalten oder zu überschreiten, und nicht das Budget für das gesamte Jahr.

Nachfolgende Quartale werden weiterhin als Schätzungen betrachtet und als solche behandelt. Die Teams können das, was sie im laufenden Quartal gelernt haben, in das nächste Quartal einfließen lassen (Überprüfung und Anpassung im Finanzbereich).

So geht die Wall Street mit Firmenschätzungen um. Die Unternehmen veröffentlichen ihre Schätzungen für das laufende Quartal und legen diese für die Zukunft fest. Diese zukünftigen Schätzungen haben jedoch weniger Gewicht als die aktuelle Schätzung.

Kapitel 15

Business Development – Geschäftsfeldentwicklung

IN DIESEM KAPITEL

Die Marketing-Agilität mit Scrum verbessern
Den Umsatz steigern

Man muss sich nicht verändern. Überleben ist keine Pflicht.

W. Edwards Deming

Geschäftsfeldentwicklung, oft auch als Vertrieb und Marketing bezeichnet, ist für das Wachstum und die Gesundheit fast jeder Organisation von grundlegender Bedeutung. Schließlich ist es wichtig zu verbreiten, was Sie tun, um neue Kunden zu finden und dies dann in Verkäufe umwandeln, um so den Motor Ihres Unternehmens am Laufen zu halten.

Was ist der Unterschied zwischen Marketing und Vertrieb?

Das *Marketing* liefert Informationen, um potenzielle Kunden zu finden, diese dazu zu bewegen, Produkte zu erwerben und so den Verkaufsprozess anzustoßen. Das Marketing ändert sich, da neue Blickwinkel und Methoden entwickelt werden, um Ihr Produkt oder Ihre Dienstleistung so vielen Menschen wie möglich zugänglich zu machen. Marketing ist der Himmel über dem Trichter des Vertriebs.

Verkauf beziehungsweise *Vertrieb* ist ein kontinuierliches Gespräch zwischen zwei oder mehr Personen. Der Vertrieb versteht die Bedürfnisse und Wünsche dieser Person oder

Organisation und erklärt, wie Ihr Produkt oder Ihre Dienstleistung diese Bedürfnisse und Wünsche erfüllt.

In diesem Kapitel erörtern wir, wie Scrum für die entscheidenden Rollen von Marketing und Vertrieb funktioniert.

Scrum und Marketing

Im Rahmen des Wasserfall-Projektmanagements ist es schwierig zu wissen, ob das richtige Produkt gebaut wird. Auf dem Weg dorthin wurde wenig bis gar kein Benutzer-Feedback eingeholt, sodass jeder nur raten kann, ob das Produkt ein großer Erfolg wird.

Hier ist der Haken: Organisationen haben traditionell einen festen jährlichen Marketingplan (ähnlich dem jährlichen Budgetierungsprozess, den wir in [Kapitel 14](#) besprechen). Firmen kommen mit einem Produkt heraus, ohne eine klare Vorstellung davon zu haben, ob es ein Erfolg wird, jedoch wird das Marketing oft ein Jahr im Voraus geplant.

Zudem ändern sich die Märkte häufig. Angesichts des rasanten technologischen Fortschritts kommt es heute noch häufiger zu Veränderungen. Selbst für Produkte, die geprüft wurden und sich als das erwiesen haben, was der Kunde wünscht, kann es notwendig sein, dass die Unternehmen ihre Marketingperspektive im Laufe der Zeit ändern.

Manchmal wissen Kunden nicht einmal, was sie wollen, bis sie ein Produkt in die Hand bekommen und es benutzen. Trotz der besten Absichten des Unternehmens bleibt es ein Rätsel, wie und was zu vermarkten ist.

Außerdem kann es eine Firma mit Millionen Kunden zu tun haben, die unterschiedliche Geschmäcker und Wünsche haben. Betrachten Sie, wie viele unterschiedliche Arten und Modelle Autos, Kleidung und technische Geräte in einem beliebigen Jahr verkauft werden. Jedes Jahr kommen neue Modelle heraus. Die Marketing-Spezialisten wollen den individuellen Geschmack jedes Kunden befriedigen.

Obwohl die Zukunft nicht vorhersehbar ist, ist die Vorhersage genau das, was das traditionelle Marketing von den Menschen verlangt.

Entwicklung des Marketings

Wie die meisten Branchen und Geschäftsfunktionen hat sich auch das Marketing weiterentwickelt. Anna Kennedy, Autorin von *Business Development For Dummies* (John Wiley & Sons, Inc.), berichtet, dass die aktuellen Änderungen im Marketing die folgenden sind:

- ✓ Interessenten sind in der Lage, sich selbst Informationen zu beschaffen. Dies hat zur Folge, dass bereits zwei Drittel des Kaufvorgangs abgeschlossen sind, bevor der Käufer direkten Kontakt mit dem Unternehmen hat, welches das Produkt anbietet.
- ✓ Das Marketing hat sich darauf verlagert, für die Generierung von Leads verantwortlich zu sein. Jeder Chief Marketing Officer, der das nicht erreicht, wird bald arbeitslos sein.
- ✓ Die Technologie ermöglicht es den Marketing-Spezialisten zu beurteilen, wo im Kaufprozess (Buyer's Journey) sich die Interessenten befinden, und sie individuell zu fördern.
- ✓ Die sozialen Medien bieten ein Forum für Gespräche zwischen Marken und deren Influencern, Interessenten und Kunden.

Angesichts dieser Trends lautet die Frage: Wie kann Scrum den Marketing-Fachleuten helfen, Veränderungen anzunehmen und Agilität zu entwickeln, um auf diese sich verändernde Welt zu reagieren?

Vielleicht können Kunden selbst mehr recherchieren, bevor sie in Kontakt mit einem Verkäufer kommen, was aus Sicht des Verkäufers eine Herausforderung sein kann. Ein Marketing-Scrum-Team, das seine Marketingmaterialien nachverfolgbar macht, kann sein Marketing jedoch mit Hilfe von Analyse-Tools überprüfen und an das sofortige Feedback anpassen.

In den letzten Jahren hat sich das Marketing zu einem Hauptabnehmer von Big Data entwickelt. Die Fähigkeit, Marktdaten und Trends zu konsolidieren, zu analysieren und darauf zu reagieren, macht Scrum zu einem idealen Rahmen für das Marketing.

Der Zugriff auf Feedback (Aktionen potenzieller Kunden und Interaktion mit Marketingmaterialien) ermöglicht es einem Scrum-Team, häufiger und schneller als bisher zu reagieren und Anpassungen vorzunehmen. Für dieses Feedback muss der Kunde nicht selbst an einem Sprint-Review teilnehmen. Ein Product Owner hat dank Marketing-Analyse-Technologien Zugriff auf dieses Feedback in Echtzeit.

Scrum und Soziale Medien

Die Marketing-Abteilung ist für das Management der Social-Media-Images ihrer Marken verantwortlich. Die Geschwindigkeit, mit der Posts viral werden, ist erstaunlich und erfordert eine sofortige taktische Reaktion. Ein Beispiel für diese Dynamik sind die Präsidentschaftswahlen 2016 in den USA. Die Kampagnen nutzten Social Media, um Trends zu verfolgen und darauf zu reagieren.

Eine beliebte globale Wohltätigkeitsorganisation sah sich kürzlich mit einer Krise in den sozialen Medien konfrontiert und konnte aufgrund ihrer Investitionen in Marketingexpertise und schnelle Einsatzstrategien effektiv reagieren. Über die Wohltätigkeitsorganisation wurden öffentlich Falschaussagen verbreitet. Innerhalb weniger Stunden gefährdete die sich rasant verbreitende Information die Fähigkeit der Hilfsorganisation, in der Welt, in der sie tätig war, Gutes zu tun, und die Spenden nahmen ab.

Ebenfalls innerhalb weniger Stunden begannen die Markenverantwortlichen der Wohltätigkeitsorganisation mit der Produktion von Inhalten, um die Falschinformationen zu korrigieren. Lediglich in den sozialen Medien zu reagieren, war jedoch nicht genug. Die Markenmanager arbeiteten mit allen Websites und Entwicklern der Organisation zusammen, um

sicherzustellen, dass die offizielle Position der Organisation klar war und aus den neu erstellten Inhalten hervorging. Da die Website-Entwickler Teil von Scrum-Teams waren, wurden die neuen Inhalte zur höchsten Priorität und innerhalb weniger Tage auf den verschiedenen Plattformen bereitgestellt. Wenn die Änderungen einen Wasserfall-Entwicklungs- und Testprozess durchlaufen hätten, hätte die Geschichte vielleicht ein unglückliches Ende gehabt. Stattdessen wurde die Botschaft der Organisation klarer und ein negatives Ereignis wurde in ein positives Ergebnis umgewandelt. Diese Geschichte wiederholt sich fast täglich, und zwar sowohl in kleinen als auch in den Fortune-500-Unternehmen.

Der agile Vorteil

In einer der jährlichen »Agile Advantage«-Umfragen befragte CMG Partners, ein Beratungsunternehmen für strategisches Marketing, Chief Marketing Officers (CMOs), Marketingleiter und agile Experten und kam zu dem Schluss, dass langfristige, hoch entwickelte Marketingpläne der Vergangenheit angehören. Schnelligkeit und Agilität sind heute wichtige Werkzeuge. Werbetreibende müssen in der Lage sein, auf veränderte wirtschaftliche Rahmenbedingungen und veränderte Kundenwünsche und -bedürfnisse zu reagieren.

Die Umfrage ergab drei wesentliche Vorteile der Nutzung des agilen Frameworks in Marketingkampagnen und -projekten:

- ✓ **Besseres Geschäftsergebnis:** Diese Leistungssteigerung erfolgte in Form einer schnelleren Markteinführung sowie einer höheren Priorisierung und Produktivität.
- ✓ **Größere Mitarbeiterzufriedenheit:** Selbstorganisierende und selbstverwaltende Teams, die transparent arbeiten und mit Entscheidungsbefugnis ausgestattet sind, machen die Menschen zufriedener und glücklicher.
- ✓ **Erhöhte Anpassungsfähigkeit:** Als Markenzeichen von Agilität und Scrum stand die Anpassungsfähigkeit im

Vordergrund, als Marketingteams agile Frameworks einführen. Im Wasserfall fehlt es an Anpassungsfähigkeit.

Es ist klar, dass der Einsatz von Agilität und Scrum im Marketing eine bessere Zusammenarbeit zwischen dem Endverbraucher und den Marketing-Spezialisten ermöglicht hat. Die Produktmarke kann sich dann in Zusammenarbeit mit dem Kunden entwickeln, anstatt zu hoffen, dass es schon irgendwie passt.

Die Kosten werden gesenkt, mit dem neuen, fein abgestimmten Ansatz werden weniger Marketingpläne scheitern. Mit dieser Steigerung der Erfolgsquote steigt auch die Kunden- und Mitarbeiterzufriedenheit.

Scrum im Marketing

Bevor wir konkrete Beispiele für Scrum im Marketing erörtern, schauen wir uns einige konkrete Wege an, wie Scrum eingesetzt werden kann. Wie der jährliche Budgetplan, der in [Kapitel 14](#) besprochen wird, ist der jährliche Marketingplan eine Annahme, die im Laufe des Jahres verfeinert wird, wenn die Organisation lernt und sich anpasst.

Mit Scrum können Marketingpläne monatlich, wöchentlich und sogar täglich erstellt werden. Nach jeder Iteration werden basierend auf dem Feedback von Sprint-Reviews und Retrospektiven Überprüfungen und Anpassungen durchgeführt. Marketingpläne können angepasst werden, Veränderungen können als Fortschritt betrachtet werden, und angepasste Marketingmaßnahmen entsprechen besser den sich ändernden Bedürfnissen der Kunden.

Neue Marketingstrategien können mit realen Nutzern getestet werden, bevor sie für ein breiteres Publikum freigegeben werden. In gleicher Weise können Marken im Rahmen von Sprints getestet werden, was enorme Kosten einspart, da das Branding gemeinsam mit dem Kunden entwickelt und nicht am Ende präsentiert wird.

Agilität im Marketing

Es gibt eine formelle Bewegung zur Anwendung agiler Prinzipien im Marketing und eine Reihe von anerkannten agilen Marketingprinzipien. Das so genannte Agile Marketing-Manifest wurde unter <http://agilemarketingmanifesto.org> veröffentlicht. Das Manifest besagt:

Wir entdecken bessere Wege, um durch neue Marketingansätze Mehrwert für unsere Kunden und unsere Organisationen zu schaffen. Durch diese Arbeit haben wir zu schätzen gelernt:

1. Validiertes Lernen anstelle von Meinungen und Konventionen
2. Kundenfokussierte Zusammenarbeit anstelle von Silos und Hierarchien
3. Anpassungsfähige und sich immer wiederholende Kampagnen anstelle von Big-Bang Kampagnen
4. Den Kunden und seine Bedürfnisse im Prozess besser kennenlernen anstatt statische Prognosen über ihn aufzustellen
5. Flexible statt starrer Planung
6. Auf Veränderungen reagieren, statt starr einen Plan zu verfolgen
7. Durchführung vieler kleine Experimente anstatt nur einiger weniger großer Kampagnen

Scrum in Aktion für das Marketing

Scrum wird heute mit erstaunlichem Erfolg im Marketing eingesetzt. Scrum passt in die wilde und chaotische Welt der sich ändernden Kundenbedürfnisse und Technologien. In den folgenden Abschnitten diskutieren wir drei Beispiele für Scrum und Marketing.

CafePress

CafePress verkauft Geschenke und Waren, die ein Kunde individuell gestalten kann, wie zum Beispiel Tassen mit Bildern und/oder Memes.

In der Scrum-Implementierung von CafePress lässt sich erkennen, wie vielseitig sich der Scrum-Ansatz anpassen lässt. Die rechtlichen Aspekte von anpassbaren Produkten erforderten eine enge Zusammenarbeit mit der Rechtsabteilung, um eine schnelle Reaktion auf rechtliche Anfragen und Bedenken zu ermöglichen. Das Marketingteam musste sich darüber im Klaren sein, was es sagen konnte und was nicht, um unnötige rechtliche Probleme zu vermeiden.

Die Rechtsabteilung wurde integraler Bestandteil des Marketing-Scrum-Teams von CafePress und sorgte dafür, dass das Marketing innerhalb von 20 Minuten alle Fragen beantworten konnte.

Die Strategie von CafePress, die Rechts- und die Marketingabteilung zusammenzubringen, ist ein gutes Beispiel für die Vorteile, die sich aus der Ausrichtung des Geschäfts auf die Entwicklung ergeben. Manchmal scheinen die Ziele der Abteilungen im Widerspruch zu stehen, aber mit Scrum sind sie direkt darauf ausgerichtet, ein besseres Produkt herzustellen und Kunden zufriedenzustellen.

Xerox

Xerox ist ein bekannter Name, der normalerweise mit Büro-Fotokopieren in Verbindung gebracht wird. Im Laufe der Jahre hat das Unternehmen auch andere Produkte und Dienstleistungen

wie IT, Dokumentenmanagement, Software und Support entwickelt.



Der ursprüngliche Fotokopiervorgang zur Herstellung von Papierkopien von Dokumenten und visuellen Bildern wird als Xerographie (Elektrofotografie) bezeichnet. Elektrostatische Aufladungen werden auf einem lichtempfindlichen Lichtrezeptor verwendet, um Pulver auf Papier zu übertragen und so die Bilder zu kopieren. Hitze besiegelt dann den Deal, buchstäblich. Wer hat die Xerographie erfunden? Eine Firma namens Xerox.

Das globale interaktive Marketing-Team von Xerox hat Scrum in sein Geschäfts-Framework integriert, um die Nachfrage von Kunden und Marketingteams auf der ganzen Welt zu vereinfachen und zu priorisieren. Das Team wusste, dass es auf den sich ändernden Markt und die sich ändernden Bedürfnisse der Kunden reagieren musste, und zwar genauso schnell wie kleinere Wettbewerber, die schneller reagieren und sich schneller anpassen konnten. Scrum bot Xerox den Rahmen, um sich selbst zu organisieren, zu iterieren, zu inspizieren, anzupassen und häufiger zu liefern.

Ähnliche Probleme gab es im Portfoliomanagement (siehe [Kapitel 13](#)). Projekte wurden nicht priorisiert, also wurden die Entwickler von Projekt zu Projekt verschoben und der lauteste Stakeholder erhielt sein Produkt zuerst. Als neue Initiativen aufkamen, konnten die Teams nicht schnell genug reagieren, um potenzielle Vorteile zu nutzen. Die Teams verschlissen Mitarbeiter, indem sie gleichzeitig an mehreren Projekten arbeiten mussten, oder es wurden Projekte abgebrochen, damit die Mitarbeiter neuen Projekten zugeordnet werden konnten.

Mit Scrum segmentierte Xerox die Art der Arbeit in drei Kanäle: Neuentwicklung, kreative Projekte (zum Beispiel Grafiken) und schnelle Reaktion (Serviceprobleme, deren Lösung acht Stunden oder weniger dauert). Kurze Sprintzyklen und der Überprüfungs- und Anpassungsprozess ermöglichten es den Teammitgliedern,

sich in jeden Sprint auf jeweils eine Warteschlange und nicht auf alle zu konzentrieren.

Das Xerox-Marketing-Team hat wurde zwar weiterhin in verschiedenen Sprints woanders eingesetzt, aber zumindest nicht mehr innerhalb der Sprints. (In [Kapitel 13](#) finden Sie mehr für Informationen darüber, warum Thrashing unerwünscht ist.) Der Fokus bei jedem Sprint ermöglichte es dem Team, Inkremente über die drei Warteschlangen hinweg zu liefern und die wichtigsten zu priorisieren. Die Markteinführung wurde beschleunigt, da das Team vor Abschluss eines Projekts nicht abgelenkt wurde. Auch die Mitarbeitermoral stieg und die Kundenzufriedenheit verbesserte sich, weil die Kunden schneller und besser bekamen, was sie wollten.

Scrum für den Vertrieb

Der Vertrieb ist keine Insel, sondern Teil eines Geschäftsfeldentwicklungszyklus. Dieser beginnt damit zu definieren, was der Markt braucht und welches Produkt und welcher Service diesen Bedarf erfüllt. Dann geht der Zyklus durch Marketing, Verkauf, Lieferung und Auswertung, um den Kunden für Wiederholungsverkäufe und Lobbyarbeit zu gewinnen.

Im klassischen Projektmanagement ist der Vertrieb die klassische, individualistische Rolle. Individuelle Quoten müssen erfüllt (und idealerweise überschritten) werden und die Vergütung jedes Einzelnen basiert auf den Ergebnissen der Verkaufsquoten. Verkäufer der Woche, des Monats und des Jahres sind begehrte Auszeichnungen. Der Wettbewerb ist hart und in jeder Organisation steigen ein paar Stars an die Spitze. Im Vertrieb bezieht sich das Wort Team typischerweise auf die Mitarbeiter der Abteilung, nicht auf die Abteilung, die als einheitliches Team für ein gemeinsames Ziel wie bei Scrum arbeitet. Um Scrum auf den Vertrieb anzuwenden, müssen Sie herausfinden, wie Sie die Verkäufer dazu bringen können, in einer Abteilung als Team zusammenzuarbeiten und nicht als Einzelpersonen zu arbeiten.

Scrum nimmt den traditionellen Vertriebsansatz auf und stellt ihn in mehrfacher Hinsicht auf den Kopf:

- ✓ Scrum ist von Natur aus auf den Erfolg des Teams ausgerichtet. Sie können sich eine Verkaufsmannschaft als eine Gruppe Verkäufer vorstellen, die nach Kunden suchen und die gleichen Rollen innehaben. Ein Scrum-Verkaufsteam ist jedoch viel mehr. Wie ein funktionsübergreifendes Produktentwicklungsteam, bestehend aus Designern, Ingenieuren und Testern, benötigt ein Vertriebsteam Experten und Mitarbeiter aus den Bereichen Marketing, Vertriebsförderung, Verkaufstechniken, Vertragswesen, Recht und Finanzen. Das gesamte Vertriebsteam hat Ziele. Wenn das Team seine Ziele nicht erreicht, können die Mitglieder gemeinsam nach Lösungen suchen.
- ✓ Die Belohnungen sind teamorientiert. Ein traditionelles Verkaufsteam (alle Verkäufer) wird vom Wettbewerb angetrieben, wobei alle in Silos arbeiten. Eine Teamstruktur bedeutet, dass alle Teammitglieder um einen Lead ausschwärmen, um fertig zu werden (den Verkauf abzuschließen).
- ✓ Bei Scrum geht es um Ergebnisse des Prozesses, mit dem diese Ergebnisse erzielt werden. Es konzentriert sich auf realitätsnahes Überprüfen und Anpassen. Sie müssen den Prozess überprüfen und für jeden Kunden anpassen.
- ✓ Die Ergebnisse zählen. Burndown- und Burnup-Diagramme (die wir in [Kapitel 5](#) besprechen), können im Verkauf verwendet werden. Anstatt zu verfolgen, wie viel getan wurde, verfolgt der Verkauf, wie viel in diesem Sprint noch zu tun ist (Burndown). Die Verfolgung der kumulativen Verkäufe (Burnup) zeigt in Echtzeit den Fortschritt auf dem Weg zum Sprint-Verkaufsziel.
- ✓ Daily Scrums im Stehen funktionieren gut, weil niemand lange Meetings will. Werden Hindernisse außerhalb des Teams entdeckt, kann eine externe Person zum Daily Scrum

eingeladen werden und an einer After-Party teilnehmen, in der die Hindernisse entfernt werden (siehe [Kapitel 6](#)).

Microchip

Microchip, ein führendes Halbleiterunternehmen, hat seine Vergütungsstruktur von einem traditionellen, variablen Vergütungsplan auf einen Plan umgestellt, der eine kleine, auf der Leistung des Unternehmens und der Einheit basierende Variable enthält. Die Mitarbeiter erhielten einen Bonus, der sich an der Leistung ihres unmittelbaren Teams sowie an der Leistung des gesamten Unternehmens orientierte. Das Ergebnis war ein verbessertes Wachstum und eine höhere Rentabilität, größeres Engagement der Mitarbeiter und eine Fluktuation nahe null.

Anders ausgedrückt wurden die besten Verkäufer gefunden, um mit anderen im Verkaufsteam zu interagieren und die besten Gesamtergebnisse zu erzielen. Kollektive Fähigkeiten und sogar Crowdsourcing wurden als Methoden genannt, mit denen sich das Team hervortat. Führende Manager nutzten Team-Brainstorming mit ihren Mitarbeitern und liehen sich sogar Ideen aus anderen Bereichen des Unternehmens aus.

Die Scrum-Lösung

Der Verkaufsprozess der Umwandlung von Leads in abgeschlossene Verkäufe wird auch als Sales-Pipeline oder Verkaufstrichter bezeichnet. [Abbildung 15.1](#) zeigt einen einfachen Verkaufstrichter.



Abbildung 15.1: Ein klassischer Verkaufstrichter, der den Verkaufsprozess von Anfang bis Ende darstellt.

Ein Vertriebsleiter einer agilen Organisation implementierte ein Taskboard mit Backlog-Einträgen. Die Spalten umfassten die Stufen des Verkaufstrichters, wie etwa Leads/Anfragen, Verkaufsanruf und Follow-Up. Das Verkaufsteam und das Management hatten sofort ein Bild davon, wo sich das Team im Projekt befand. Die Umsatzschätzungen wurden überprüft und an die Realität angepasst. [Abbildung 15.2](#) zeigt ein Beispiel für den Verkaufstrichter als Taskboard.

RELEASE-ZIEL:

SPRINT-ZIEL:

RELEASE-DATUM:

SPRINT-REVIEW:

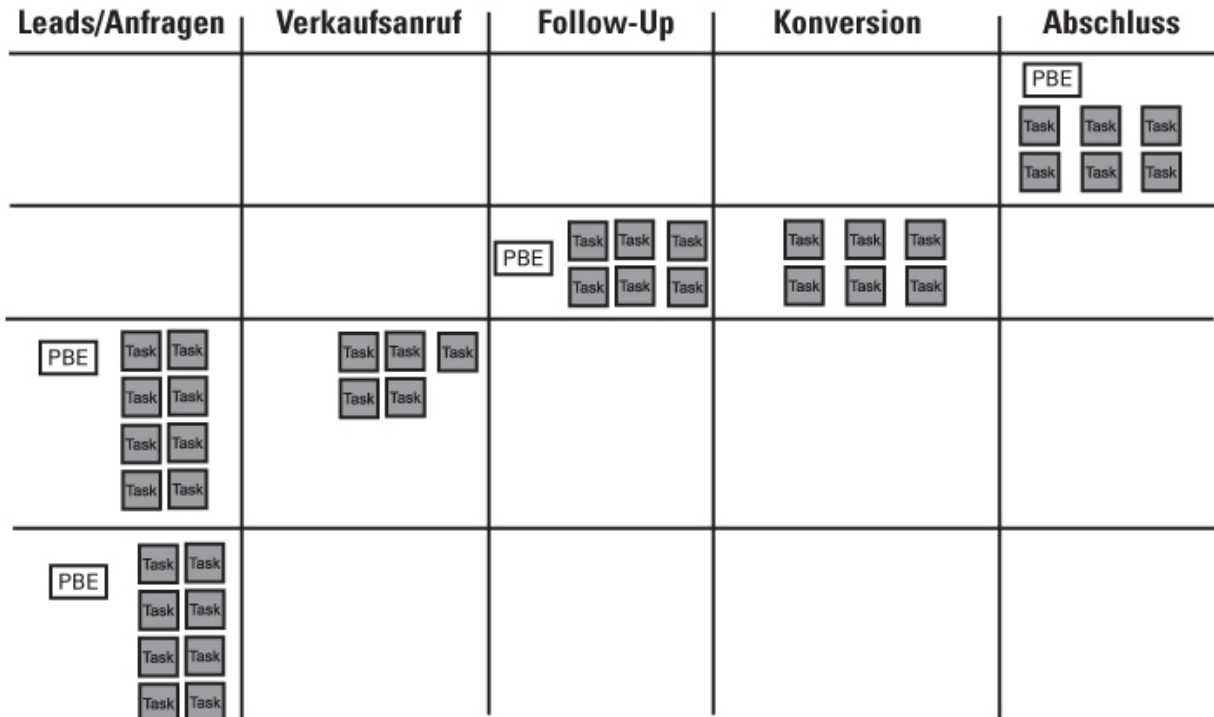
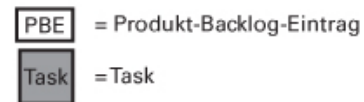


Abbildung 15.2: Verkaufs-Taskboard mit Spalten, die die Verkaufstrichterstufen darstellen

Ein sichtbarer Trichter in Form eines Taskboards ist sehr wertvoll. Der Verkaufstrichter in [Abbildung 15.1](#) wird wie folgt zum Taskboard in [Abbildung 15.2](#):

- ✓ **Leads:** Jede Zeile (Swimlane) stellt einen Lead dar. Leads kommen von einer Vielzahl von Quellen durch eine Vielzahl von Methoden, aber sie alle landen hier und werden diskutiert, überprüft und priorisiert, bevor der Verkaufsanruf eingeleitet wird. Etwas qualifizierte Leads in kaufbereite Leads zu verwandeln, wird zu einer Marketingfunktion. Scrum kann helfen, die traditionelle Kluft zwischen Marketing und Vertrieb durch Sichtbarkeit und Koordination des Vertriebsteams zu überbrücken.

- ✓ **Verkaufsanruf:** Ein erstes Verkaufsgespräch wird durchgeführt, um den Lead weiter zu qualifizieren (das heißt, den Bedarf des Interessenten zu definieren und das Angebot als Lösung zu positionieren). Diese Phase umfasst verschiedene Aufgaben, die bei der Sprint-Planung in die Spalte Leads aufgenommen wurden.
- ✓ **Follow-up:** Der Lead hat reagiert und sein Interesse an der vorgeschlagenen Lösung zum Ausdruck gebracht. Das Team koordiniert, terminiert und führt eine Reihe von Aufgaben aus.
- ✓ **Konversion:** Hindernisse und Hemmnisse für den Abschluss der Verkäufe werden identifiziert und das Team schwärmt aus, um sie zu überwinden.
- ✓ **Verkaufsabschluss:** Der Auftrag geht ein. Das Team dokumentiert eindeutig, wie es den Auftrag erhalten hat.

All dies geschieht effektiv im Team. Schauen Sie sich die Fähigkeiten an, die für einen Verkaufsabschluss erforderlich sind, einschließlich Marketing, Verkaufsförderung, Vertriebstechnik, Sachkenntnis, Verträge, Recht und Finanzen.

Im Idealfall wollen Sie funktionsübergreifende Teammitglieder, die alles können. Es kommt jedoch selten vor, dass eine Person alles gleich gut kann. Ein Team von Leuten zu haben, die alles gut können, ist noch unwahrscheinlicher. Aber ein funktionsübergreifendes Team von Mitgliedern, die sich selbst organisieren, Pläne entwerfen und um einen Interessenten, eine Veranstaltung oder einen Vertrag ausschwärmen können, führt zu besseren Ergebnissen als eine Person, die versucht, alles alleine zu tun.

Da sich Scrum auf die kontinuierliche Kommunikation zwischen allen Beteiligten konzentriert, hat ein Scrum-Vertriebsteam ein klareres Verständnis dafür, was potenzielle Kunden brauchen und wollen. So kann das Team seine Vertriebsideen so präsentieren, dass sie sich direkt auf diese Interessenten auswirken.

Der Scrum-Verkaufsprozess

Erfolgreiche Verkäufer sind außergewöhnliche Zuhörer und haben ein ausgeprägtes Beobachtungsvermögen. Ein Ziel ist es, die Probleme der Interessenten zu entdecken und dann zu aufzuzeigen, wie das Produkt oder die Dienstleistung diese Probleme löst. Um diese Verbindung herzustellen, muss ein Verkäufer eine vertrauensvolle Beziehung aufbauen. Menschen kaufen Dinge von Menschen, die sie kennen und denen sie vertrauen.

Beim Verkaufen mit Scrum geht es darum, durch Teamarbeit Vertrauen zu gewinnen. Zu den unterstützenden Akteuren können Presales-Dienste, Business Development, Kundenbetreuer, Außendiensttechniker, Installateure, Innendienstmitarbeiter und Servicemitarbeiter gehören. Jede Kombination dieser Personen kann ein Scrum-Team und einen Schwarm bilden, um die Kommunikation zu synchronisieren und die Bemühungen zu unterstützen, den Kunden dazu zu bringen, einen Vertrag zu unterzeichnen. Die Schwarmaktivitäten können sich um Folgendes drehen:

- ✓ Vorbereitung auf eine Messe
- ✓ Nachbereitung von Messekontakten
- ✓ Entwicklung von zeitgebundenen Vorschlägen zu möglichen Lösungen für ein Verkaufsargument, das den Wettbewerb übertrifft
- ✓ Risikobehaftete Verkäufe durch Brainstorming und Priorisierung von offenen Punkten



Verkaufszyklen sind erfolgreicher, wenn ein potenzieller Verbraucher oder ein Unternehmen zu einem Produkt oder einer Lösung hingezogen wird, anstatt auf einen rein ausgehenden Ansatz mit ungebetenen Anrufen oder Vertreterbesuchen zu reagieren. Pull-Strategien beinhalten Content-Marketing, Webinare, Messen und Marketing in den sozialen Medien. Die Unterschiede zwischen dem Push- und dem Pull-Modell werden in [Kapitel 5](#) besprochen.

Wesentlich für den Scrum-Verkaufsprozess ist, dass Menschen miteinander umgehen; außerdem folgt er dem gesunden Menschenverstand. Er spiegelt die Roadmap to Value wie folgt wider:

1. Vision

Beispiel: Steigerung des Bruttoumsatzes in Dallas um 20 Prozent in diesem Geschäftsjahr durch Steigerung der Social-Media-Präsenz und Analytik. Follow-up in Form eines persönlichen Anrufs oder einer E-Mail an alle, die Kontaktinformationen bereitstellten.

(Das in [Kapitel 2](#) vorgestellte Visionsmodell dient der Produktentwicklung. Für Vertriebs- und andere Nicht-Produktentwicklungsprojekte ist es sinnvoll, dieses Modell an der Realität auszurichten.)

2. Produkt-Roadmap (grober Überblick) und Produkt-Backlog (aufgeschlüsselt)

Beispiel: Um die Vision Wirklichkeit werden zu lassen, sind allgemeine und untergliederte Änderungslisten erforderlich, wie zum Beispiel

- Entwicklung der Vertriebsstrategie
- Beschaffung von Werkzeugen
- Entwicklung zusätzlicher Materialien
- Vertriebsprozessentwicklung und -optimierung
- Aufgaben durchführen, die für bestimmte Abschlüsse notwendig sind
- Zuweisung der wichtigsten (detaillierten) Social Media-Sites an ein Teammitglied
- Finden eines Tools zur Nachverfolgung von Verkaufstrichtern, die das Team verwenden wird
- Definition von Berichtsprozessen wie zum Beispiel Analyse-Workflows

3. Release-Planung

Beispiel: Priorisieren Sie die Backlog-Einträge nach Monat und/oder Quartal.

4. Sprint-Planung

Beispiel: Legen Sie als Team Aufgaben fest und teilen Sie die Backlog-Einträge auf.

5. Sprint

Beispiel: Tägliche Meetings im Stehen zur Koordination des täglichen Swarming und zur Identifizierung von Hindernissen.

6. Sprint-Review

Beispiel: Zeigen Sie den Stakeholdern (etwa dem Vizepräsidenten Verkauf), dass der Sprint abgeschlossen ist (zum Beispiel eine betriebsbereite Facebook-Seite oder Mailinglisten mit früheren Kunden).

7. Sprint-Retrospektive

Beispiel: Überprüfen Sie, was gut funktioniert hat, und erleichtern Sie den Wissenstransfer. Mitarbeiter des Kundendienstes/Installateure lernen zum Beispiel, wie neue Kunden gewonnen werden, was ihnen hilft, die Unternehmenskommunikation, die zum Verkauf führt, fortzusetzen.

iSense Prowareness

iSense Prowareness ist ein Informations- und Computertechnologie-Beratungsunternehmen (ICT) mit Sitz in den Niederlanden, das Scrum in seinen Verkaufsprozessen implementiert hat. Die Probleme, mit denen das Unternehmen konfrontiert war, waren vielfältig und basierten auf einem großen Mangel an Transparenz:

- ✓ Das System verfolgte den individuellen Fortschritt, aber nur der Manager betrachtete ihn. Die Anzahl und Größe jedes Geschäftsabschlusses waren die einzigen Daten, die geteilt wurden, und der Fortschritt jedes Vertreters wurde allein anhand dieser Metriken gemessen. Jede Woche

schätzte jeder Einzelne seine Zahlen für diese Woche und berichtete am Ende der Woche, was er tatsächlich erreicht hatte.

- ✓ Das Unternehmen hatte keine Möglichkeit zu erfahren, wie sich individuelle Anstrengungen auf den gesamten Teamumsatz auswirkten.
- ✓ Es gab keine Prozessindikatoren. Die einzige Metrik war die verkaufte Menge. Es konnte zwar die Anzahl der Anrufe verfolgt werden, jedoch war die Abschlussrate nicht bekannt. Das Management hatte nur begrenzte Möglichkeiten, ihrem Verkaufsteam zu helfen, da sie nicht wussten, was und wo die einzelnen Hindernisse waren. Der einzige Input war, härter zu arbeiten, da sie keine Möglichkeit hatten zu bestimmen, was »intelligenter arbeiten« bedeutet.
- ✓ Die Vertriebsmitarbeiter arbeiteten hart daran, ihre Zahlen zu verbessern, aber es wurde keine strukturierte Zeit für die Reflektion oder die Einrichtung eines Systems zur Verbesserung aufgewendet.
- ✓ Vor diesem ganz typischen Verkaufshintergrund begann iSense mit der Implementierung von Scrum. Man beauftragte externe Berater, bei der Umstellung zu helfen, und begann mit folgenden Schritten: Wöchentliche Sprints, die von Montag bis Freitag dauerten.
- ✓ Vierteljährliche Verkaufsziele, die 13 Sprints pro »Release« beinhalteten.
- ✓ Jede Woche passten die Verkäufer das aus den Daily Scrums und Sprint-Reviews und Retrospektiven Gelernte an.
- ✓ Ein Scrum-Taskboard mit einem Burndown-Diagramm und den erhaltenen Aufträgen wurde erstellt.
- ✓ Täglich wurden vor dem Taskboard Daily Scrums durchgeführt.

Nach Abschluss der Scrum-Integration verdoppelte sich der Umsatz, wovon die Hälfte direkt auf Scrum entfiel – ein hervorragender Return on Investment für den Übergang zu Scrum.

Interessanterweise gehörte zu den wichtigen Lehren, die man daraus gezogen hatte, dass nun beurteilt werden konnte, welche Verkaufstaktiken funktionierten und welche nicht zum Gesamtwert beitrugen. Die Transparenz von Scrum machte deutlich, dass es nicht zur Wertschöpfung der Organisation beiträgt, wenn sie einen erhaltenen Auftrag nicht erfüllen kann. Während die Vertriebsmannschaft durch diesen teambasierten Ansatz ihre Trefferquote erhöhte, war ihr Pool an Dienstleistern nicht größer geworden, um die Nachfrage zu befriedigen. Die Vertriebsleiter erkannten einen Weg, wie sie den Personalvermittlern helfen konnten, die Nachfrage durch befristete Verträge zu befriedigen, was ihnen mehr Zeit verschaffte, diese Positionen dauerhaft zu besetzen.

Das soeben vorgestellte Team führte Telefonwerbung zu Veranstaltungen durch, bei denen die Produkte hervorgehoben wurden. Auf der Veranstaltung werden die Teilnehmer von kalten in warme Leads verwandelt, und dann führt der Nachkontakt dazu, dass sich einige dieser Leads in heiße Leads und letztendlich in Verkäufe verwandeln. Durch die Analyse des Prozesses und der Ergebnisse erkannte das Team, dass es nur 10 Prozent in tatsächlichen Umsatz umwandelte. So konnte das Team Lücken im Verkaufsprozess diskutieren und identifizieren, die Reihenfolge verfeinern und den Umsatz steigern. Es dauerte nicht lange und sie verdoppelten die Konversionsrate.

Durch die Integration von Scrum machte das iSense-Vertriebsteam eine weitere wichtige Entdeckung: Der Verkauf war erst der Anfang. Um den Kreislauf wirklich zu vollenden, musste das Unternehmen das Produkt liefern. Klingt einfach, aber durch den Prozess der Sprint-Reviews und Retrospektiven, bei denen Mitarbeiter aus verschiedenen

Teilen des Unternehmens Informationen austauschen, begann sich auch diese Lücke zu schließen, und der Kunde profitierte.

Die Vertriebsteams erkannten, dass sie sich nicht primär auf den Aufbau neuer Geschäfte konzentrieren, sondern mehr mit bestehenden Kontakten und Kunden gewinnen konnten. Die Kunden, mit denen sie bereits erfolgreiche Projekte durchgeführt hatten, konnten als Ausgangspunkt für weitere Geschäfte genutzt werden. Sie konnten Förderer von Projekten für andere Unternehmen werden.

Kapitel 16

Kundendienst

IN DIESEM KAPITEL

Kundenfeedback für Verbesserungen fördern und einbeziehen
Sparen und mehr Umsatz machen
Serviceprozess verbessern

Kommen Sie Ihren Kunden näher als jemals zuvor. So nah, dass Sie ihnen sagen, was sie brauchen, bevor sie es selbst merken.

Steve Jobs

Kundendienst ist einer der kritischsten Aspekte eines jeden Unternehmens und wohl der nächste wichtige Bereich für die Scrum-Implementierung. Positive Erfahrungen geben uns Anlass, auf ein Produkt zurückzugreifen oder Freunde und Familie darauf hinzuweisen. Doch der Kundendienst ist genau der Bereich, den viele Kunden und Auftraggeber als den problematischsten Teil der Erfahrung bezeichnen. Sie mögen das Produkt, aber sein wahrgenommener Wert nimmt bei schlechtem Service ab.

Kunden können auch intern sein. Ineffiziente interne Unterstützung und unvollständige Kommunikation führen zu Produktivitätsverlust und schlechter Moral.



Kundendienst ist gleich Verkauf. Ein glücklicher Kunde informiert Freunde und Familienmitglieder und kehrt für weitere Dienstleistungen und Produkte zurück.

Der Kundenservice ist eine gute Möglichkeit, den Umsatz und damit den Gewinn zu steigern. Diese eine Abteilung kann ein

Unternehmen auf neue Höhen treiben. Wenn der Kundenservice vernachlässigt wird, kann dies das Ansehen und das Wohlbefinden, das dem Unternehmen entgegengebracht wird, beeinträchtigen.



Wissen über die Kunden zu haben, führt zu wissensbasiertem Marketing. Die Suche nach neuen Kunden wird effektiver. Daher kann die Verwaltung von Kundenkonten als eine Erweiterung von Vertrieb und Marketing betrachtet werden.

In diesem Kapitel erforschen wir Möglichkeiten, Scrum zu nutzen, um Ihren externen und internen Kundenservice zu verbessern.

Kunden: Die wichtigsten Stakeholder

Ihre Kunden sind hervorragende Quellen für Ideen zu Produktinnovationen und -verbesserungen. Diese Ideen nicht zu beachten, führt nicht nur zu einem schlechten Ruf, sondern verhindert auch eine entscheidende Feedbackschleife, die zu Innovationen führen kann.

Internet Service Provider, Telefongesellschaften und lokale Versorgungsunternehmen gehören zu den Anbietern, die die meisten Beschwerden im Hinblick auf Kundendienst Anfragen erhalten. Wo die Anzahl der Anbieter begrenzt ist, ist der Kundenservice in der Regel (wenn auch nicht immer) schlechter als in anderen Branchen. Kunden treffen häufig auf lange Wartezeiten, frustrierenden Service und labyrinthische, automatisierte Servicemenüs. Der Druck im Hinblick auf das Betriebsergebnis hat viele Unternehmen veranlasst, das Budget für den Kundenservice zu kürzen, doch die Kunden sind die wichtigsten Stakeholder in jedem Unternehmen und müssen einen guten Service erhalten. Ohne die Kunden gibt es kein Geschäft.

Comcast ist ein Beispiel für ein Unternehmen, das Erfahrungen mit negativen Schlagzeilen hinsichtlich der Qualität des Kundendienstes gemacht hat. Fälle, in denen Kunden keine Rückerstattungen für falsch abgerechnete Gebühren erhielten, Wartezeiten bis zu dem Moment, in dem die Kundendienstabteilung schloss, und die Unmöglichkeit, Abonnements zu kündigen, sind nur einige der Beschwerden, die im Laufe der Jahre erhoben wurden.

Obwohl der Bedarf an Service enorm ist, fehlt es oft an Qualität und Millionen von Menschen sind bereit, sich die Zeit zu nehmen, andere für ihre Erfahrungen zu sensibilisieren. Kundenbeschwerden über die Abwicklung von Kundendienst Anfragen gibt es zuhauf.

Schwierige Fragen beim Kundendienst

Vielleicht ist es das größte Problem beim Kundendienst, dass Kunden oft berichten, ihre Probleme würden nicht einmal vom Kundendienstmitarbeiter gelöst. Die Kunden möchten und erwarten, dass die Servicemitarbeiter Fragen beantworten und Probleme lösen, die sie mit den Produkten des Unternehmens haben. Leider ist es allzu oft nicht möglich, den Kundendienstmitarbeitern das Wissen zu vermitteln, das sie benötigen, um diesem Anspruch der Kunden gerecht zu werden.

Der Kundendienstmitarbeiter versteht das Problem möglicherweise nicht, weil der Kunde es schlecht erklärt, oder er versteht einfach nicht die Situation, die der Kunde beschreibt.



Achtundsiebzig Prozent der Kunden gaben an, dass ihre guten Erfahrungen mit dem Kundendienst mit dem Wissen des Mitarbeiters zusammenhängen, während nur 38 Prozent angaben, dass dies auf den persönlichen Kontakt zurückzuführen sei. Wissen ist wichtig. Die Ausbildung der Mitarbeiter hat einen hohen Stellenwert. Vorgeplante Antworten fallen nicht so ins Gewicht wie fundiertes, echtes Wissen.

Die Kosten für den Verlust eines Kunden sind weit höher als der Preis des Jahresabonnements des Kunden.

Kundendienstabteilungen werden oft als Kostenstellen betrachtet, jedoch führen sie zu Ersparnissen und bringen Ihrem Unternehmen enorme Einnahmen. Im Folgenden finden Sie einige Statistiken, die zeigen, wie sich der Kundendienst auf ein Unternehmen auswirken kann, ohne dass Ihnen dies vielleicht bisher bewusst war:

- ✓ Es ist mehr als sechsmal teurer, einen neuen Kunden zu gewinnen, als einen aktuellen zu behalten.
- ✓ 78 Prozent der Kunden haben auf einen Kauf oder eine Transaktion aufgrund von vorhergehenden schlechten Erfahrungen mit dem Kundendienst verzichtet.
- ✓ Loyale Kunden sind bis zu zehnmal mehr wert als ihre ersten Einkäufe.
- ✓ Zwölf positive Erfahrungen sind nötig, um eine negative auszugleichen.
- ✓ Für jeden Kunden, der sich die Zeit nimmt, sich zu beschweren, berichten stattdessen 26 nur ihren Freunden über den mangelhaften Service.

Informationsüberflutung

In der heutigen Zeit werden zu viele Daten als Ursache für Call-Center-Versagen genannt. Zu viele Informationen können das effektive Funktionieren eines Servicemitarbeiters lähmen.

Servicemitarbeiter können durch zu viele Daten, die nur schwer verwendbar sind, überfordert werden, und in vielen Call-Centern sind diese Informationen nicht so organisiert, dass sie für die Mitarbeiter nützlich sind.

Vielleicht waren Sie mit einem Kundendienstmitarbeiter im Gespräch und wurden mit einer anderen Person verbunden, nur um festzustellen, dass Sie das Problem noch einmal erklären mussten. Häufig werden die historischen Daten eines Kunden nicht an den Mitarbeiter weitergegeben. Es ist frustrierend für den Kunden, seine Situation erneut zu beschreiben oder die ominösen Worte zu hören: »Ich kann hier keine Aufzeichnungen über Ihren Beschwerde finden.«



Die richtigen Informationen an die richtigen Leute zu bringen, ist Teil des Problems. Sechzig Prozent der Kundendienstzentren geben an, dass sie nicht einmal in der Lage sind, bestimmte Kundeninformationen (beispielsweise Teile der Kundenhistorie) an die Kundendienstmitarbeiter weiterzugeben. Hinzu kommt, dass mehr als 30 Prozent der Mitarbeiter keine Kundenzufriedenheitsdaten erfassen und aufzeichnen.

Manchmal werden Daten nicht zwischen den Unternehmensbereichen konsolidiert oder sind schwer zu finden. Unternehmen sind Experten im Sammeln von Informationen, aber die rechtzeitige und effektive Verbreitung dieses Wissens ist ein Problem.

Gebräuchliche Kundenservice-Metriken

Bei der Beurteilung von Serviceanfragen misst das Management oft die Effizienz anstatt die Kundenbedürfnisse und die Zufriedenheit der Kunden. In diesem Sektor werden in Prozent die folgenden Metriken gemessen:

- ✓ 79 Prozent sammeln die durchschnittliche Bearbeitungszeit.

- ✓ 75 Prozent messen die Abbruchrate.
- ✓ 71 Prozent messen die durchschnittliche Antwortgeschwindigkeit.
- ✓ Weniger als 50 Prozent messen tatsächlich die Kundenzufriedenheit.

Scrum und Kundenservice

Die meisten der im vorhergehenden Abschnitt aufgeführten allgemeinen Probleme lassen sich durch Scrum verbessern. Diese Probleme scheinen komplex zu sein, aber sie sind nicht unlösbar. Scrum bietet Lösungen.

Überprüfen und Anpassen durch Feedback

Ein Grundprinzip von Scrum ist die Feedbackschleife (siehe [Kapitel 7](#)), die die Überprüfung und Anpassung erleichtert. Im Kundenservice erhält man täglich, stündlich und sogar minutengenau Feedback von den Kunden. Dieses Feedback bezieht sich sowohl auf das Produkt als auch auf den Kundenservice selbst. Diese Feedback-Struktur ist perfekt für einen anrufbezogenen Sprintzyklus.

In jedem Sprint-Event können sich die Kundendienst-Teams fragen, ob die vorhandenen Daten das sind, was sie brauchen, um den besten Kundenservice zu bieten. Teams müssen wissen, ob die vorhandenen Werkzeuge dem Kunden am besten dienen und ob das vorhandene Wissen ausreicht, um die meisten Probleme zu lösen. Sie können die Roadmap to Value nutzen, um diese Probleme für das Kundenservice-Team anzugehen:

- ✓ **Vision:** Die Vision eines Anrufs beim Kundendienst ist einfach: dem Kunden einen qualitativ hochwertigen Support zu

bieten, der damit endet, dem Kunden ein gutes Gefühl über das Unternehmen zu vermitteln.

- ✓ **Sprint-Planung:** Bei jedem Anruf können konkrete Schritte zur Problemlösung überprüft und angepasst werden. Während des Anrufs kann der Mitarbeiter dem Kunden bestimmte Fragen stellen, um zu prüfen, ob Fortschritte bei der Lösung des jeweiligen Problems gemacht werden. Wenn die Anruferdaten zeigen, dass der Kunde früher länger hat warten müssen, kann der Mitarbeiter den Kunden kurz fragen: »Glauben Sie, dass Ihre Wartezeit heute zumutbar war?« Das Sammeln von Ja- und Nein-Antworten kann bestätigen, ob Fortschritte bei bestehenden Prozessen erzielt werden oder ob Änderungen erforderlich sind, um das Problem zu lösen.
- ✓ **Sprint-Review:** Die Mitglieder des Service-Teams finden es möglicherweise nicht effizient, sich nach jedem Anruf zum Review zu versammeln. Die Überprüfung und Anpassung am Ende jeder Schicht gibt dem Team und den Teams der nächsten Schicht jedoch Feedback in Echtzeit. Hier kann das Team die aus dem Sprint gesammelten Informationen aufnehmen, um sich auf zukünftige Anrufe vorzubereiten. Sie reflektieren, was sie aus der vorherigen Schicht gelernt haben, und legen fest, welche Fragen sie beim nächsten Sprint stellen sollen.
- ✓ **Sprint-Retrospektive:** Am Ende einer Schicht kann das Team untersuchen, ob es bessere Wege gibt, von Kunden Informationen zu erhalten. Suggestieren beispielsweise die aktuellen Fragen eine bestimmte Antwort der Kunden oder sind sie wirklich objektiv? Wie gut haben sie ihre Ergebnisse bewertet? Wie gut wurden sie dokumentiert? Welche Aspekte ihres Prozesses sollten sie auf Verbesserungen hin überprüfen?
- ✓ **Daily Scrums im Team:** Wenn Service-Teams eintägige Sprints durchführen, wird das Daily Scrum natürlich automatisch auch zum Sprint-Review und zur Sprint-Planung. Wenn Sprints länger als eine Schicht oder ein Tag sind, können individuelle Probleme und Hindernisse aufgeworfen

werden, die das umfassendere Kundenservice-Team in einem Daily Scrum lösen kann. Wenn es einem Kundendienstmitarbeiter schwer fällt, die Fragen zu stellen, mit denen die Anrufe untersucht werden, kann möglicherweise ein anderer Service-Mitarbeiter anbieten, mit ihm gemeinsam ein paar der Anrufe durchzuführen (Shadowing oder Paarprogrammierung).

Viele Service-Center haben genügend Mitarbeiter, sodass aus diesen mehrere Teams gebildet werden können. In diesem Fall würde es sich anbieten, den vertikalen Schnitt oder ein anderes skaliertes Scrum-Modell zu implementieren (siehe [Kapitel 13](#)), um die Koordination und Integration von Aktivitäten und das Schaffen von gemeinsamem Wissen zu ermöglichen. Ein Scrum of Scrums, das jeden Tag stattfindet, wäre der natürliche Ort, um Erfahrungen auszutauschen und die Bemühungen aller Teams schnell und zielgerichtet zu koordinieren.

Ein Kundenservice-Product-Owner wäre ein wichtiger Stakeholder in einem Scrum-Team, das die Produkte entwickelt, die der Product Owner unterstützt.

Kundenservice-Produkt-Backlog

Die Schulung sachkundiger Kundendienstmitarbeiter ist der wichtigste Faktor für die Schaffung zufriedener Kunden. Diese Ausbildung ist keine einmalige Veranstaltung. Neue Produkte und Dienstleistungen erfordern aktualisierte Weiterbildungen. Außerdem sind Auffrischungen im Umgang mit Beschwerden und Missverständnissen erforderlich.

Indem sie es den Kunden äußerst einfach machen, Feedback zu geben, können Unternehmen einen Produkt-Backlog erstellen, das Probleme enthält, die gelöst werden müssen. Diese Probleme können priorisiert und aufgrund ihrer Priorität angegangen werden.



Für Neuproduktentwicklungsteams ist die Release-Planung (siehe [Kapitel 5](#)) der Zeitpunkt, zu dem der Kundendienst die

Schulung von Service-Mitarbeitern über das neue Produkt bereits während der Entwicklung vorbereiten kann (unter Verwendung der bevorstehenden Überprüfungen und Anpassungen, damit der Schulungsplan auch wirklich fundiert ist).

Falls das Kundendienst-Scrum-Team im Rahmen seiner Kontakte mit den Kunden einen wiederkehrenden Bedarf erkennt, kann das Finden einer Lösung in das Produkt-Backlog eingefügt und mit einer hohen Priorität versehen werden; diese Schritte helfen dabei, den Implementierungszyklus so kurz wie möglich zu gestalten. Kundenbedürfnisse lassen sich in Echtzeit erkennen und adressieren. Die Priorisierung dieser Bedürfnisse und das Swarming für das Erstellen der Lösung hilft dem Kundenservice, das Produkt-Backlog zu bewältigen.



Der Kundenservice ist ein wichtiger Stakeholder für Produktentwicklungsteams. Das Feedback dieser Mitarbeiter sollte als ebenso zuverlässig angesehen werden (wenn nicht sogar zuverlässiger) als das der Abteilung Marktforschung. Wenn Kundendienstmitarbeiter nicht als integrale Stakeholder in Produktentwicklungs-Scrum-Teams vertreten sind, können sich die Product Owner nicht sicher sein, dass sie wissen, was die Kunden wollen.

Definition of Done für den Kundenservice

Die Definition of Done für Kundenservice-Teams sollte klar umreißen, was es für einen Kunden bedeutet, mit dem Serviceniveau zufrieden zu sein. Die Definition of Done sollte mehr als nur die Anrufdauer und die Abbruchstatistik beinhalten.

Um auf die Roadmap to Value zurückzukommen, sollte eine Kundendienstabteilung eine Vision haben, die spezifisch darlegt, was Kundenbedürfnisse sind, wie das Team diese Bedürfnisse erfüllt und wie die Rolle des Kundendienstes die Organisation im

Markt differenziert und an die Unternehmensstrategie anknüpft.
(Mehr zu Visionen finden Sie in [Kapitel 2.](#))

Die Vision ist der Rahmen, um zu definieren, was es bedeutet, »fertig« zu sein, wenn jeweils einem Kunden ein qualitativ hochwertiger Kundendienst angeboten wurde. Die Definition of Done des Kundendienstes könnte etwa so aussehen:

Bei jedem Kunden gelingt es mir, unsere Vision [Aussage dazu, wie die Ansprüche gedeckt werden und wie sich das Unternehmen unterscheidet] zu verwirklichen, indem ich

- ✓ das Problem des Kunden vollständig löse,
- ✓ die Produktdokumentation um das Problem und die Lösung ergänze,
- ✓ den Verlauf der Kundenkontakte gründlich überprüfe, bevor eine Lösung angeboten wird,
- ✓ die Kontakthistorie aktualisiere,
- ✓ in den Wissensdatenbanken (ABC und XYZ) nach Antworten suche,
- ✓ fachliche Kompetenz von Kollegen und/oder meinem Vorgesetzten einhole,
- ✓ cool bleibe,
- ✓ dem Kunden helfe, die Ruhe zu bewahren, indem [Methoden aufführen, die von der Organisation angewendet werden],
- ✓ Zusagen einhalte, die ich dem Kunden gegeben habe.

Die Kundendienstmitarbeiter müssen verstehen, was von ihnen erwartet wird, um so klar zu vermitteln, wie ein Erfolg aussieht.



Berücksichtigen Sie die Definition of Done, wenn Sie den Kundendienstmitarbeitern das Erforderliche zur Verfügung stellen, um ein Produkt oder einen Service zu unterstützen. Gleichgültig, ob dieses Material in Form einer Dokumentation, einer zusätzlichen Schulung oder einer anderen Maßnahme vorliegt, klären Sie es und kommunizieren Sie es dann.

Blick nach innen

Jeder, der in einem großen Unternehmen gearbeitet hat, kennt wahrscheinlich das in diesem Kapitel beschriebene Maß an Frustration über den Kundenservice. Wir haben hier externe und interne Kunden zusammengefasst, weil sich viele der Dynamiken ähneln. Im Folgenden finden Sie einige häufige Beschwerden von externen und internen Kunden:

- ✓ Die Unternehmen wachsen zu schnell.
- ✓ Die Budgets werden gekürzt.
- ✓ Helpdesk-Mitarbeiter sind nicht ausreichend geschult.
- ✓ Unternehmen haben eine kurzfristige Vision.
- ✓ Die Feedbackschleifen funktionieren schlecht.

Ineffizienter Support verursacht Produktivitätsverluste und Frustration, die ihrerseits zu starker Fluktuation, Terminüberschreitungen, Umsatzverlusten und frustrierten externen Kunden führen können. Es ist nicht länger akzeptabel, Ausreden wie »Das System ist ausgefallen« oder »Ich kann das Support-Ticket nicht finden« zu benutzen. Scrum bietet einen idealen Rahmen für die Bewältigung dieser Probleme, indem es Arbeit sichtbar macht und Wissenssilos durch Zusammenarbeit und Kommunikation im Team auflöst.

Wir haben Helpdesk-Teams geschult und hierbei Scrum und eintägige Sprints verwendet. Die Transformation des Teams war erstaunlich. Innerhalb von ein paar Monaten hat sich die

Einschätzung des Teams innerhalb des Unternehmens von schlecht zu ausgezeichnet geändert. Eine Gruppe beschrieb die Veränderung folgendermaßen:

- ✓ Jeden Tag besser auf die Prioritäten konzentrieren
- ✓ Besseres Gesamtbild der Auslastung
- ✓ Deutlichere Muster (einige Probleme verursachten den größten Teil der Arbeitsbelastung)
- ✓ Mehr Transparenz gegenüber Kunden und Management

Durch die Sichtbarmachung der Arbeit und die Anwendung gemeinsamen Wissens konnte das Team einige der zugrunde liegenden Problemursachen beseitigen und erhielt Unterstützung vom Management für zukünftige Initiativen.



Der Informationsaustausch ist ein wichtiger Bestandteil der wachsenden Leistungsfähigkeit. Stellen Sie sicher, dass das Umfeld den Wissensaustausch innerhalb des Teams und zwischen den Abteilungen belohnt. Achten Sie auf Kommentare wie »Sie ist die Einzige, die das kann« und »Das kann ich nicht«.

Überprüfen und Anpassen in der Praxis

Früher hatte E-Mails im Vergleich zu Anrufen beim Call-Center eine geringere Priorität, sodass Beschwerden über fehlende Antworten auf schriftliche Anfragen häufig anzutreffen sind. Diese Priorität ist im Rahmen der Agilen Prinzipien sinnvoll.

Kundendienstprofis verstehen den Wert von Kommunikation in Form von Sprache im Vergleich zu schriftlicher Kommunikation. Dennoch scheuen sich die Kunden vor Anrufen, es sei denn, sie haben den Eindruck, dass sie nicht lange darauf warten müssen, live mit jemanden sprechen zu können. Anderenfalls weichen sie auf die Optionen E-Mail und Live-Chat aus.

Ein Kundenservice-Center, das wirklich überprüft und anpasst, sucht nach operativen und prozeduralen Möglichkeiten, um den schriftlichen Kundendienst erfolgreicher und integraler zu gestalten. Schließlich kann ein Kundendienstmitarbeiter mehrere Chats gleichzeitig führen, aber immer nur einen Anruf auf einmal. Mindestens eine Studie behauptet, dass ein Kundendienstmitarbeiter bis zu sechs Chats statt eines Anrufes bearbeiten kann.



Prompt beantwortete Sprachanrufe, die kompetent gehandhabt werden, sind im Hinblick auf den Kundendienst besser. Chat wiederum ist besser als E-Mail, da der Kunde so sofort eine Antwort erhält.

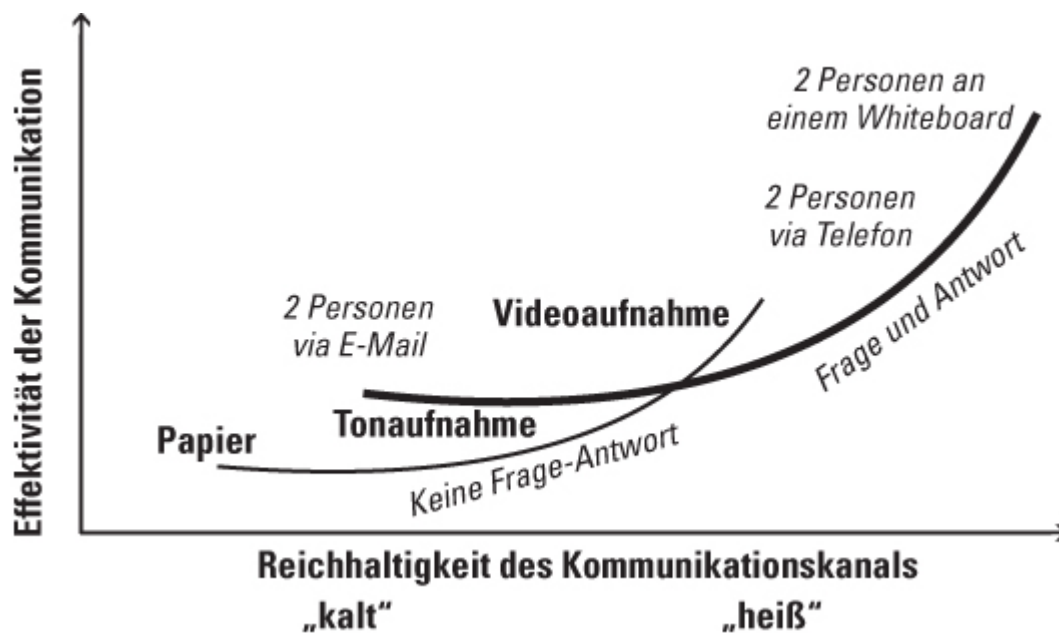


Abbildung 16.1: Alistair Cockburns Kommunikationsmodi. Copyright © Humans and Technology, Inc.



Wie wir in [Kapitel 4](#) erwähnen, hat Dr. Albert Mehrabian, emeritierter Professor für Psychologie an der UCLA, bewiesen, dass 55 Prozent der Bedeutung durch Körpersprache und Mimik vermittelt wird; 38 Prozent der Bedeutung paralinguistisch, das heißt durch die Art, wie wir

sprechen; und 7 Prozent der Bedeutung ist in den tatsächlich gesprochenen Worten enthalten.

Die Bedeutung der Qualität der Kommunikation kann nicht hoch genug eingeschätzt werden. Die persönliche Interaktion ermöglicht Fragenstellen und Klärung in Echtzeit, die durch Körpersprache und nonverbale Signale unterstützt wird.

Mangelnde Interaktivität führt zu Missverständnissen.

Missverständnisse führen zu Mängeln, und nichts ist schädlicher für den Zeitplan und das Budget und damit für die Rentabilität als Fehler – ob es sich nun um Produktfehler oder Mängel im Kundenservice handelt, die Kosten sind erheblich.

Weniger Missverständnisse sind nicht der einzige Vorteil. Das gleiche Gespräch, das tagelange Hin- und Her-Antworten erfordert, kann in fünf Minuten erledigt sein, wenn man aufsteht und über den Flur zu einem anderen Büro oder einer anderen Arbeitsnische geht. Beim Kundendienst dürfen Einschränkungen lediglich dazu führen, dass ein Kundendienstmitarbeiter sein Headset aufsetzt sowie eine Nummer wählt, um ein persönliches Gespräch zu führen. Persönliche Gespräche sind viel effektiver als das Hin und Her in Form von Chats oder E-Mails.

Angesichts der Bedeutung der Kommunikation in Form von Gesprächen sollte Ihre Definition of Done für qualitativ hochwertigen Kundendienst um Kommunikation in Form eines persönlichen Gesprächs erweitert werden. Zum Beispiel könnten Sie die Definition of Done, die wir weiter vorne beschrieben haben, ändern, indem Sie etwa das Folgende hinzufügen:

- ✓ Einen E-Mail-Kunden nach zwei Durchgängen hin und her anrufen, um weitere Missverständnisse zu vermeiden.
- ✓ Kundenanfrage, wann immer möglich, per Telefon statt per E-Mail weiterverfolgen

Scrum im Einsatz im Kundenservice

Viele Unternehmen verwenden Scrum, haben es aber nicht auf ihre Kundendienstabteilungen angewandt. Diese Firmen könnten ihren Kundenservice und ihr Ergebnis erheblich verbessern, würden sie Scrum in allen Bereichen einsetzen. Wenn eine Organisation bereits Scrum einsetzt, sollte sie über genügend Erfahrung und Wissen verfügen, um diese mit anderen Abteilungen zu teilen.

Bisher haben wir in diesem Kapitel verschiedene Möglichkeiten zur Anpassung von Scrum durch Events, die Definition of Done und das Produkt-Backlog erörtert. Integraler Bestandteil dieser Transformation ist sicherzustellen, dass bei jedem Sprint-Review für Produktentwicklungs-Scrum-Teams ein Kundenbetreuer anwesend ist. Transparenz und Bewusstsein entstehen durch die Integration von Kundenservice und Produktentwicklung.

Scrum-Teams aus Mitarbeitern des Kundendienstes können während des Sprints Schulungen und andere Aktivitäten zur Prozessverbesserung planen. Zusätzlich können Service-Mitarbeiter aus den Sprint-Reviews der Produktentwicklung lernen und diese Informationen mit ihren Teams teilen.

L.A. Water and Power

Das Versorgungsunternehmen Los Angeles Department of Water and Power hat das Scrum-Framework implementiert, ohne es Scrum zu nennen. Ziel war es, die Kundenzufriedenheit im Rahmen des Solar Incentive Programms zu erhöhen. Das Programm bietet Anreize, um die Kosten für die Installation einer Solaranlage auf privaten oder gewerblichen Gebäuden auszugleichen. Die Abteilung wollte Verspätungen reduzieren, das Kundenerlebnis insgesamt optimieren und die Transparenz erhöhen.

Um die Transparenz zu erhöhen, implementierte die Abteilung zwei Dashboards, die so genannten Mayor's Dashboards, um den Kunden zu helfen, über Änderungen und Verbesserungen auf dem Laufenden zu bleiben. Ein Dashboard enthielt wöchentliche Aktualisierungen der Metriken, einschließlich des

Status der Rabattprüfungen. Das zweite Dashboard zeigte das 100-Megawatt-Programm und Einspeisevergütungen (Feed-in Tariff, FiT), mit dem die Stadt Solarstrom von Dritten bezogen hat. Die Zeit und der Prozess für die Bearbeitung von Verträgen über Einspeisevergütungen wurde angezeigt und beide Dashboards zeigten Probleme, Lösungen und Antwortzeiten an. Hierdurch stieg die Transparenz enorm an.

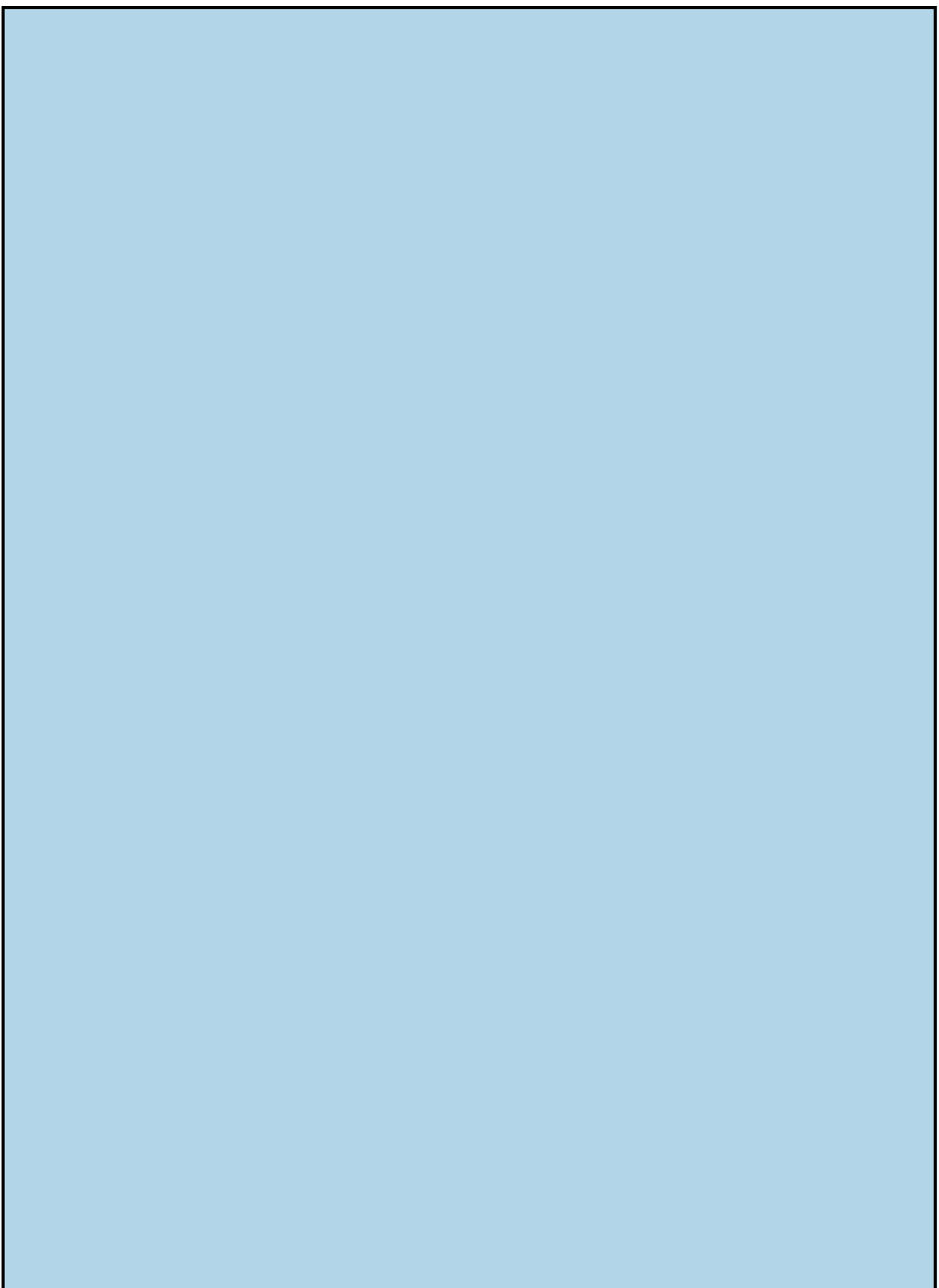
Diese Sprint-Backlogs wurden allen Stakeholdern und Kunden zur Verfügung gestellt. Darüber hinaus prüfte und passte die Abteilung ihren Prozess konsequent an, um den Prozess der Antragsprüfung zu rationalisieren. Zu den Ergebnissen dieser Überprüfung und Anpassung gehörten die Aufstockung des Personals für die Bearbeitung von Anträgen und Hotline-Service, die Beseitigung von Abhängigkeiten zwischen Rabattzahlung und Aktivieren des Service, die Verkürzung der Überprüfungszeit von Mietverträgen durch ein Formular zur Einhaltung von Mietverträgen und die Automatisierung der routinemäßigen E-Mail-Kommunikation mit Kunden und Auftragnehmern.

Die wichtigsten Verbesserungen strafften die bestehenden Systeme und erhöhten die Übersichtlichkeit. Durch die Verbesserung des Kundenerlebnisses konnte die Abteilung den Kundenservice deutlich verbessern. Sie zielte auf die Ursachen von Problemen ab und machte sich daran, sie mit Scrum-Prinzipien zu beheben.

Teil V

Scrum im Alltag





IN DIESEM TEIL ...

Entwickeln Sie erfüllende Beziehungen.

Planen Sie persönliche und Familienprojekte ohne Stress.

Erreichen Sie finanzielle Ziele.

Maximieren Sie Bildungschancen.

Kapitel 17

Partnersuche und Familienleben

IN DIESEM KAPITEL

Den richtigen Partner finden
Familienbeziehungen verbessern
Größere Projekte zu Hause anpacken

Ich will nicht perfekt sein. Ich will es wert sein.

Unbekannt

Es ist leicht zu sagen, dass sich die Welt verändert hat, und das hat sie auch. Scrum kann jedem Unternehmen dabei helfen, zu wachsen und sich an die sich verändernden Gegebenheiten der Welt anzupassen.

In den nächsten Kapiteln besprechen wir einige Bereiche Ihres Lebens, bei denen Sie vielleicht nicht daran gedacht haben, Scrum zu implementieren. Es mag vielleicht emotionslos erscheinen, über Ihr Privatleben in Scrum-Begriffen zu sprechen. Jedoch beinhaltet auch unser privates Leben die Priorisierung und die Notwendigkeit kritischer Entscheidungen auf der Basis unvollkommener Informationen und mit einer unbekannten Zukunft – genau wie das Geschäftsleben.

Bitte beachten Sie beim Lesen dieses und des folgenden Kapitel die folgenden Hinweise:

- ✓ Im Geschäftsleben bezieht sich das Produkt-Backlog auf ein in Entwicklung befindliches Produkt – ein Software- oder

Hardwareprodukt, eine Dienstleistung oder ein anderes lieferbares Produkt. In der Familie und den persönlichen Bemühungen kann das Produkt ein Ferien- oder Hochzeitsplan, eine Beziehung, eine Ausbildung oder die Planung der Altersversorgung sein.

- ✓ Wenn wir im Geschäftsleben von Kunden sprechen, dann meinen wir damit die Menschen, die das Produkt benutzen werden. In persönlichen und familiären Bemühungen können Sie selbst, Ihre Familie, Ihr Lehrer oder Ihr zukünftiges Ich im Ruhestand der Kunde sein.

Lassen Sie sich von der Terminologie nicht abschrecken. Es funktioniert.

Zusätzlich zu den Vorteilen in der Geschäftswelt kann Scrum Ihnen helfen, persönliche Beziehungen aufzubauen und sich Zeit zu nehmen, die Dinge zu tun, die Sie mögen. Die meisten von uns werden sagen, dass unsere Familie oder die bessere Hälfte das Wichtigste in unserem Leben ist, und Scrum kann Ihnen helfen, diese wichtigen Menschen zu priorisieren.

Partnersuche mit Scrum

Menschen haben einen sehr realen Drang, nach Liebe und Beziehungen zu suchen. Auch wenn die ideale Beziehung für jeden anders ist, sucht fast jeder eine liebevolle, langfristige Beziehung. Aber die Wahrheit ist, dass Dating zwar Spaß machen und abenteuerlich sein kann, es aber schwierig werden kann, ein dauerhaftes, liebevolles, gesundes Verhältnis zu finden.

Die Einstellung zu Beziehungen hat sich verändert. In modernen Beziehungen suchen die Menschen nach ehrlichen Verbindungen. In dauerhaften Beziehungen geht es nicht mehr darum, aufgrund von gesellschaftlichen Erwartungen verheiratet zu sein oder dafür zu sorgen, dass man nicht alleine alt wird. Menschen suchen erfüllende, befriedigende Beziehungen, die auf Kommunikation und Verständnis basieren.

Obwohl die Technologie eine große Rolle bei der Eröffnung des Dialogs in der Partnersuche gespielt hat und neue Möglichkeiten bietet, potenzielle Partner zu finden, hat sie nicht alle Herausforderungen beseitigt. Trotz dieser technologischen Fortschritte und Bequemlichkeiten kann die Partnersuche ein schwieriges Thema sein, über das man diskutieren oder sogar nachdenken sollte.

Wir wollen Ihnen zeigen, wie Scrum den Umgang mit den Herausforderungen beim Dating einfacher machen und dass Scrum ein Framework für die Schritte bieten kann, die Sie in Ihrem Streben nach Glück nie für möglich gehalten hätten. Hier sind einige dieser Herausforderungen beim Dating:

- ✓ **Zeit:** Im Leben geht es darum, wie man die endliche Zeit, die man jeden Tag hat, verbringt. Singles haben Karrieren, Weiterbildung, Nebenjobs, persönliche Ziele, Hobbys, Freunde und andere Aktivitäten. Wer Single ist, für den ist es einfach, länger auf der Arbeit zu bleiben oder den zusätzlichen Nebenjob oder das Projekt zu übernehmen. Obwohl viele Menschen eine Beziehung wollen, haben sie ihr Leben nicht so strukturiert, dass sie Zeit für eine Beziehung haben. Um Raum für die Partnersuche zu schaffen, muss man Prioritäten setzen.
- ✓ **Kommunikation:** Bei der modernen Partnersuche findet die Kommunikation meist via E-Mail oder Sofortnachrichten statt. Die technologischen Fortschritte, die das Herstellen von Verbindungen erleichtern, können jedoch auch Beziehungen behindern, da sie die für die Minimierung von Missverständnissen und die Lösung von Konflikten so wichtige persönliche Interaktionen ersetzen. Kein Wunder, dass E-Mail und Sofortnachrichten es schwer machen, jemanden auf einer tiefen Ebene kennenzulernen, und es gleichzeitig leichter machen, sich zu streiten.
- ✓ **Treffen:** Die richtige Person zu finden, ist schwierig, und meistens fühlt es sich so an, als ob man einfach Glück hatte, jedoch geht es meistens um etwas viel Einfacheres. Die

meisten Beziehungen, die nicht funktionieren, scheitern aus einem oder zwei einfachen Gründen: Anziehungskraft und Chemie oder Persönlichkeit und Werte.

Es ist ein Problem, jemanden mit passender Chemie zu finden, dessen Persönlichkeit und Werte zum eigenen Leben passen. Diese beiden Kategorien vereinfachen eine Vielzahl von Faktoren, die die Kompatibilität beeinflussen, etwa Finanzen, Egoismus und Intimität. All diese Faktoren tragen zu erfolgreichen Beziehungen bei.

Sie können sich diese Probleme ansehen und denken, dass sie nichts mit Scrum zu tun haben, weil die Partnersuche eben einfach schwierig ist! Jedoch ist genau dies der Grund, warum Scrum auch bei der Suche nach dem richtigen Partner funktioniert. Bei Scrum geht es darum, die Dinge zu vereinfachen und dann an der Realisierung der Vision zu arbeiten. Hier sind einige Möglichkeiten, wie Sie das Scrum-Framework auf romantische Beziehungen anwenden können:

- ✓ Auf Veränderungen (wie Akzeptanz, Ablehnung, Liebe, Herzschmerz, Verbindung und Verlegenheit) reagieren
- ✓ Früh und schnell scheitern (so wenig Zeit wie möglich für jemanden verschwenden, der nicht kompatibel ist)
- ✓ Überprüfen und Anpassen (die perfekte Ergänzung für jemand anderen werden und neu definieren, wer für einen selbst die perfekte Ergänzung ist)

Fast jeder hat schon einmal an Liebeskummer gelitten (bis auf die wenigen Glücklichen, die beim ersten Mal auf die Liebe ihres Lebens treffen). Für manche Menschen führt Herzschmerz zu Bitterkeit oder Introvertiertheit mit dem Ziel, sich vor weiterem Herzeleid zu schützen. Jede weitere Schicht, mit der Sie versuchen, sich zu schützen, hindert Sie jedoch gleichzeitig daran, dem Ziel der Liebe näher zu kommen. Liebe ist ein offensives Spiel, kein defensives. Es ist einleuchtend, dass Überprüfen und Anpassen auf der Grundlage häufiger

empirischer Nachweise die Schlüssel sind, um glücklich zu werden.

Die folgenden Abschnitte betrachten, wie Sie Schwierigkeiten beim modernen Dating überwinden und wie Sie sich auf die Suche nach der Liebe Ihres Lebens mit Scrum konzentrieren können.

Die Vision festlegen

Die Roadmap to Value (oder in diesem Fall die Roadmap to Love) beginnt mit dem Festlegen einer Vision (siehe [Kapitel 2](#)). Wie in geschäftlichen Dingen brauchen Sie eine Vision, die Sie zum Ziel führt.

Dieser Bereich hat sich in den letzten Jahren dramatisch verändert. Früher bestand der einzige Zweck jeder romantischen Beziehung darin, zu heiraten und Kinder zu bekommen. Auch wenn diese Vision für manche Menschen ideal ist, ist sie für andere nicht ideal. Definieren Sie, was Sie suchen, wenn Sie eine Beziehung suchen. Jemand, der alle seine Interaktionen durch die Vision »Ich will einfach nur Spaß mit jemandem haben, zu dem ich mich hingezogen fühle« definiert, wird sich einem Date auf eine radikal andere Weise nähern als jemand mit der Vision »Ich will jemanden heiraten, den ich als meinen besten Freund und Seelenverwandten betrachte«.

Denken Sie daran, dass sich Visionen ändern können, wenn sich Ihre Lebensumstände und -phasen ändern. Vielleicht möchten Sie gerne heiraten, aber der Gedanke mag so beängstigend sein, dass Sie sich mit der Vision, Spaß zu haben, auf den Weg begeben haben. In dieser Situation werden Sie wahrscheinlich nicht so bald einen Ehepartner finden. Sie können viel Spaß haben, aber Spaß wird nicht unbedingt zu einer langfristigen Bindung oder Heirat führen, bis Sie Ihre Vision anpassen.



Vision Statements sind zukunftsgerichtete interne Aussagen. Sie beschreiben, wie Sie sich selbst in der Zukunft sehen.

Spaß zu haben, ist kein schlechtes Ziel, aber es ist eine schlechte Vision, wenn man einen anderen Endzustand anstrebt. Visionen werden nicht sofort erfüllt, und das Ziel, einen Lebenspartner zu haben, bedeutet nicht, dass die nächste Person, mit der man ein Date hat, diese Person sein wird. Aber zu wissen, was Sie auf lange Sicht suchen, wird Ihre Interaktionen während der Partnersuche verändern.

Dating in Schichten

Beziehungen entwickeln sich. Sie sollten wissen und klar definieren, was jede Phase einer Beziehung für Sie bedeutet. Wenn Sie die Vision haben, einen Lebenspartner zu finden, werden Sie eine Reihe von Schritten unternehmen, die Sie in Richtung dieses Ziels führen. In Scrum sind diese Schritte Ihre Roadmap, und jeder wichtige Meilenstein ist ein Release-Ziel. Hier sind einige Beispiele für Release-Ziele auf einer Roadmap, die auf die Vision hinarbeitet, einen Lebenspartner zu finden:

- ✓ Bereiten Sie sich auf die Dating-Welt vor (aktualisieren Sie Ihre Garderobe, beginnen Sie mit einem Fitness-Training, ändern Sie Ihre Frisur oder kaufen Sie Weingläser).
- ✓ Erstellen Sie eine kurze Liste von Eigenschaften Ihres idealen Partners und unterteilen Sie diese in solche, die für Sie ein Muss sind, und solche, die Sie gerne hätten.
- ✓ Beginnen Sie, sich mit zu Ihnen passenden Menschen zu treffen.
- ✓ Treten Sie in eine Beziehung ein (wie von Ihnen definiert).
- ✓ Binden Sie sich langfristig (Dating exklusiv nur mit einer Person, Vereinbarung zu Monogamie, Verlobung oder persönliches Äquivalent).
- ✓ Verflechten Sie Ihre Leben miteinander (zusammenleben, größere gemeinsame Einkäufe tätigen, heiraten und/oder Kinder bekommen).

Jedes Release-Ziel erfordert, dass Sie sich an Aufgaben und Aktivitäten beteiligen, mit denen Sie sich von Stufe zu Stufe bewegen. Das erste Release-Ziel allein kann einige Zeit in Anspruch nehmen.

Wenn Sie definieren können, wie jede Phase der Partnersuche aussieht, können Sie bestimmen, auf welches Release-Ziel Sie hinarbeiten. Verwenden Sie eine Definition of Done. So könnte beispielsweise die Definition of Done für die Bereitschaft, einen Partner zu suchen, darin bestehen, dass Sie Einladungen zu Single-Partys annehmen (und dann auch wirklich dort erscheinen), auf denen Sie neue Leute kennenlernen können sowie das Erstellen von Profilen auf seriösen Dating-Websites. Für jede Ebene einer Beziehung sollten Sie eine klare Definition of Done erstellen.



Eine Definition of Done für die Nutzung von Dating-Websites könnte sein, mindestens eine Person (idealerweise mehr) persönlich zu treffen, falls Ihr Ziel darin besteht, voranzukommen und jemanden für eine persönliche Beziehung zu finden.

Eine Definition of Done ermöglicht eine klare Kommunikation und Grenzziehung für Ihr Dating-Leben. Der Übergang von einem Date zu einer Beziehung ist kompliziert, aber Sie können ihn etwas weniger verwirrend gestalten, indem Sie für sich selbst den Unterschied zwischen einem Date und einer Beziehung klären. Wenn Sie jemanden mehrmals getroffen haben und sich aus dem Date ein Kontakt entwickelt, der alle Merkmale Ihrer Definition of Done für eine Beziehung aufweist, dann sollten Sie überprüfen, ob Sie sich nun in einer Beziehung befinden. Das Beste am Einsatz von Scrum für die Partnersuche ist, dass Sie nicht zufällig in einer Beziehung landen, sondern dass Sie ein aktiver Teilnehmer sind, der sich ständig auf sein Ziel zubewegt.



Nicht jeder in der Dating-Welt verwendet für die verschiedenen Phasen einer Beziehung dieselbe Definition of Done. Nutzen Sie die offene Kommunikation, um Vertrauen aufzubauen. Während Sie sich von Release-Ziel zu Release-Ziel bewegen, müssen Sie gegenüber denjenigen, mit denen Sie sich treffen, offen darüber sein, wonach Sie gerade suchen und wie Sie die verschiedenen Phasen einer Beziehung definieren.

Freundschaften und Scrum entdecken

Mit Ihrer Roadmap haben Sie den Rahmen für Ihr Backlog der Anforderungen: die Aktivitäten, die Sie planen und an denen Sie teilnehmen werden, um Ihre Release-Ziele und Visionen zu erreichen, beginnend mit Ihrem ersten Release-Ziel.

Ein Teil Ihres Produkt-Backlogs für die Vorbereitung auf die Dating-Welt können Aktivitäten und Hobbys sein, die Ihnen Spaß machen. Während Sie die Dinge tun, die Ihnen Spaß machen, kann es wertvoll sein, die Zeit damit zu verbringen, Bekanntschaften aufzubauen und Freunde zu finden. In dieser Phase geht es vor allem darum, sich selbst zu entdecken. Durch Menschen, die Sie während der Zeit kennenlernen, in der Sie Dinge tun, die Ihnen Spaß machen, entstehen neue Beziehungen, durch die Sie anderen Personen begegnen, mit denen Sie in der Zukunft möglicherweise gerne ein Date hätten. Außerdem ist es so möglich, eventuelle Dating-Partner kennenzulernen, ohne den Druck, dies sofort als Date zu definieren.



Nutzen Sie vorhandene Bekanntschaften und Freundschaften, um Ihren Produkt-Backlog zu identifizieren und um zu verfeinern, wonach Sie suchen.

Manchmal reicht es aus, auszugehen, um zu einem Date zu kommen. Dates können sich oft wie von selbst ergeben, wenn Sie sich Leuten treffen, mit denen Sie gemeinsame Interessen haben,

und mit ihnen dann Zeit zu verbringen. Beziehungen entwickeln sich in Phasen. Auch wenn ein Unterschied zwischen Freundschaften und Bekanntschaften und Dates zu zweit besteht, kann sich aus Ersterem das Letztere ergeben.

Vergessen Sie nicht, zu kommunizieren. Wenn sich zu jemanden eine Beziehung entwickelt, mit dem Sie bereits auf freundschaftlicher Basis Zeit verbracht haben, kann es sein, dass Sie es versäumen, über die Veränderung in der Beziehung zu kommunizieren. Um Ihre Fähigkeit zu stärken, ein gutes Timing in potenziell unangenehmen Situationen zu haben, verwenden Sie Ihre eigene Definition of Done, um den Unterschied zwischen der mit jemandem als Freund verbrachten Zeit, einem Date sowie einer festen Beziehung zu identifizieren. Die Planung dieser Diskussion kann ein Beispiel für einen Eintrag im Dating-Backlog sein.

Seien Sie vorsichtig, wenn Sie mit jemandem auf eine Art und Weise Zeit verbringen, die Elemente von Intimität und Merkmale einer festen Beziehung aufweist – Sie diesen Kontakt jedoch lediglich als Freundschaft identifizieren. Falls Sie Probleme damit haben, dies klar zu kommunizieren, nutzen Sie dies als Gelegenheit zur Überprüfung und Anpassung. Verfolgen Sie weiterhin die Release-Ziele, angefangen von den Vorbereitungen für das Dating über das Dating selbst hin bis zu einer Beziehung. Schlechte Kommunikation ist für beide Seiten schlecht und führt oft zu der Art von Dating-Drama, dass die meisten Menschen zu vermeiden versuchen. Verwenden Sie empirische Beweise, um zu verfeinern, welche Art von Beziehung Sie suchen, und verwenden Sie beispielsweise die folgenden Dating-Retrospektiven, um sicherzustellen, dass Sie sich auf Ihr Ziel zubewegen:

- ✓ Was läuft gut?
- ✓ Was wollen wir ändern?
- ✓ Wie können wir diese Veränderung spürbar umsetzen?

Dating mit Scrum

Zu den Vorteilen von Scrum gehören das strukturierte Erreichen eines Ziels, empirische Evidenz, der Faktor der Überprüfung und Anpassung und die Zeitersparnis. Wenn Ihr Leben ausgefüllt ist und Sie akzeptieren können, dass Sie einen Lebenspartner haben möchten, dann müssen Sie auch akzeptieren, dass Sie für diese Person Platz schaffen müssen. Viele Leute haben die Ausrede gehört (oder benutzt), dass sie gerne eine Beziehung finden würden, aber keine Zeit dafür haben.

Die gute Nachricht ist, dass Ihr Leben in Ihrer Hand liegt. In den frühen Phasen beim Dating müssen Sie sich Zeit für die Dates nehmen. In den späteren müssen Sie Platz dafür machen, dass die Beziehung wachsen kann. Beziehungen entstehen nicht in Petrischalen. Es gibt keinen schnellen Weg, um von dem Punkt, an dem Sie keinen Platz für eine Beziehung haben, zu einer zu einer wunderbaren Work-Life-Love-Balance zu gelangen.

Stellen Sie sich Ihr Leben wie ein Produkt-Backlog vor. Falls die Priorität für Ihre Liebes-Roadmap und die Backlog-Einträge immer sehr niedrig ist, werden Sie sie immer zugunsten von etwas anderem beiseiteschieben. Wenn Sie sich entscheiden, dass Liebe und Partnersuche eine hohe Priorität haben, schaffen Sie Raum dafür.

Die Verfolgung Ihrer Roadmap to Value und die Verwendung von Scrum für das Dating umfasst kurze Sprintzyklen, Überprüfung und Anpassung sowie Verbesserung für zukünftige Sprints.

Bei Scrum ist jedes Date ein Sprint; daher arbeiten Sie in kurzen Sprintzyklen, die Sie zu Ihrem Freigabeziel führen sollen. Da Sie vielleicht nicht jeden Tag der Woche ein Date haben, gibt Ihnen dieser Zeitplan genügend Zeit zwischen den Sprints, um die Änderungen und Verbesserungen für den nächsten Sprint zu reflektieren (zu überprüfen) und zu planen (anzupassen). Die Zeit zwischen den Sprints (den Dates) bietet Ihnen die Gelegenheit zur Verfeinerung Ihres Backlogs und zur Verbesserung des Prozesses.

Der Überprüfungsteil des Überprüfungs- und Anpassungs-Ansatzes konzentriert sich darauf, wie Sie sich selbst und Ihren

Datingpartner sehen. Sie können die folgenden Dinge untersuchen:

- ✓ Hat diese Person die Eigenschaften, die ich suche?
- ✓ Habe ich Spaß mit dieser Person?
- ✓ Erfüllt diese Person meine unverzichtbaren Wünsche?
- ✓ Wie verhalte ich mich, wenn ich in der Nähe dieser Person bin?
- ✓ Was sind meine Gedanken und Gefühle über diese Person, wenn wir nicht zusammen sind?

Der Anpassungsteil liegt bei Ihnen. Sie können anpassen, wonach Sie suchen (zum Beispiel Ihre Vision) oder Sie können anpassen, mit wem Sie sich treffen (zum Beispiel mit diesen Personen keinen weiteren Sprint durchführen). Möglicherweise müssen Sie Ihre Interaktion anpassen und einen weiteren Sprint mit derselben Person auf der Grundlage Ihrer Bewertung versuchen. Wenn Sie etwas ausprobieren wollen, was mehr Spaß machen kann, probieren Sie ein Date mit etwas unbeschwerteren Aktivitäten oder unternehmen Sie gemeinsam etwas, das Ihnen Spaß macht, und schauen Sie, wie sich diese Anpassungen auf Ihre Gefühle für die andere Person auswirken.

Der beste Teil bei der Verwendung von Scrum für die Partnersuche ist der Prozess der Überprüfung und Anpassung. Sie und Ihr Partner können sich eine eigene Version von dem machen, was *wir* bedeutet – oder auch nicht. Wie in geschäftlichen Dingen führt auch hier die Fähigkeit, etwas aufgrund der frühen empirischen Erkenntnis, dass es nicht zu Ihrer Vision passt, abzukürzen und zu beenden, nicht nur zur Zeit- und Geldersparnis, sondern es wird so auch Enttäuschung auf beiden Seiten vermieden.

Als Team gewinnen

Leider ist die moderne Dating-Welt voller Beispiele dafür, dass nur Spielchen gespielt werden und die Leute sich gegeneinander

ausspielen. Diese Mentalität beim Dating verkompliziert Interaktionen und verhindert echte Kommunikation. Bei Scrum planen Sie als Team, führen den Plan als Team aus und sind als Team gemeinsam für Erfolg oder Misserfolg verantwortlich. Selbst wenn Sie das Gefühl haben, alleine das Team zu bilden, wenn es um die Partnersuche geht, denken Sie daran, dass Sie einer in einem Zweierteam sind. Das Beste, was Sie für sich selbst tun können, ist, das Konzept von Scrum anzunehmen und sich gemeinsam mit Ihrem Date als Team zu betrachten, für das Team zu arbeiten, um erfolgreich sein zu sein, anstatt sich an einem Wettbewerb mit Ihrem Datingpartner zu beteiligen. Mit Scrum können Sie sich über den Unsinn erheben und es vermeiden, Spielchen zu spielen.

Ehrliche Kommunikation ist für jedes Team und vor allem für jede erfolgreiche Beziehung notwendig. Eine häufige Falle beim modernen Dating besteht darin, Dates mit vorgefertigten Fragen oder Aktionen zu testen. Scrum vereinfacht die Kommunikation. Scrum für das Dating zu verwenden, bedeutet, echte Fragen zu stellen und die Antworten der anderen Person als Chance zu sehen, ihn oder sie im Hinblick auf das Ziel dieses Sprints besser kennenzulernen. Um eine offene und ehrliche Kommunikation zu erreichen, beginnen Sie das Gespräch damit, dass Sie Ihre eigenen Ansichten teilen und Ihren Datingpartner nach seinen Gedanken fragen. Während Sie die Dates – Ihre Sprints – planen, planen Sie Fragen, die echt sind und ermutigen Sie ehrliche Antworten, um so festzustellen, ob diese Person gut zu Ihnen passt.



Um als Team zu arbeiten, vermeiden Sie es, Ihren Datingpartner in eine Situation zu bringen, in der Sie ein bestimmtes Verhalten erwarten, aber der Partner einfach nicht wissen kann, was das Ziel ist. Die Personen, die Sie treffen, werden oft versuchen, so zu antworten oder sich so zu verhalten, wie es ihrer Ansicht nach von ihnen erwartet wird – wobei sie oft das Ziel verfehlen.

Hier sind einige Beispiele für Verhaltenstests, die verzerrte Ergebnisse bringen:

- ✓ Sie schlagen eine Dating-Aktivität vor, die Sie für extrem (oder unbequem) halten, nur um zu sehen, wie Ihr Date reagiert, und treffen ein Urteil, das ausschließlich auf dieser Reaktion basiert.
- ✓ Eine Frau, die glaubt, dass ein Gentleman für ein Date bezahlen sollte, bietet an, die Rechnung zu teilen, aber wenn er ihr Angebot annimmt, ist sie nicht mehr interessiert.
- ✓ Ein Mann, der glaubt, dass körperliche Intimität bis zu einem bestimmten Zeitpunkt in einer Beziehung warten sollte (eine von ihm definierte, aber nicht geteilte Zeitmarke), strebt danach, körperlich intim zu sein. Wenn die Frau körperliche Intimität akzeptiert, ist der Mann nicht mehr interessiert.

Diese Beispiele enthalten verschiedene Geschlechternormen, aber es gibt unterschiedliche Arten dieser Tests. Stellen Sie offene Fragen ohne verborgene Absicht und teilen Sie Ihre eigenen Gedanken und Ansichten mit dem Partner

Fokussierung versus Multitasking

Scrum ist ein Werkzeug zur Fokussierung. Scrum ist unter anderem deshalb so erfolgreich, weil Herausforderungen in kleinere und kürzere Ziele zerlegt werden, sodass Sie nicht mehr alles auf einmal tun müssen. Anstatt alle Projekte schlecht zu erledigen, hilft Ihnen Scrum, sich darauf zu konzentrieren, ein Projekt nach dem anderen mit einer höheren Qualität zu erledigen, als wenn Sie permanent zwischen mehreren Projekten hin- und herschalten und sich so unnötig verschleißten würden.

Das folgende Zitat fasst eine aktuelle Studie der Stanford University zusammen: »Multitasking in Meetings *und anderen sozialen Umgebungen* zeugt von einem geringen Selbst- und Sozialbewusstsein, zwei Fähigkeiten emotionaler Intelligenz, die für den Erfolg am Arbeitsplatz entscheidend sind« (kursive Hervorhebungen ergänzt).

Die Studie untersuchte, wie Multitasking die Effektivität im Arbeitsumfeld senkt – aber emotionale Intelligenz ist eine Lebenskompetenz. Es wird schwer sein, Liebe zu finden, wenn Sie Ihre Selbst- und Sozialkompetenz verringern.

Wenn Sie gleichzeitig zwei oder mehr Personen daten, werden Ihr Geist und Ihre Emotionen durch dieses Multitasking stärker abgelenkt, und Sie sind weniger darauf fokussiert herauszufinden, ob jemand gut zu Ihnen passt. Multitasking beim Dating kann auch den Versuch bedeuten, durch mehrere Phasen einer Beziehung zu hetzen.

Die moderne Dating-Praxis versucht, allen nahezulegen, sich mit so vielen Menschen wie möglich zu treffen und sich dann so lange wie möglich alle Optionen offen zu halten. Der Versuch, das Risiko zu mindern, dass eine Beziehung nicht funktioniert, führt fast immer zu dem Risiko, dass zahlreiche oder alle Beziehungen nicht funktionieren.

Sich mit mehreren Personen zu verabreden, kann jedoch in den frühen Phasen eine gute Sache sein, um Ihre Chancen zu erhöhen, den Partner zu finden, nach dem Sie suchen. Wenn Ihre Vision darin besteht, den Partner zu finden, mit dem Sie Ihr Leben verbringen wollen, wird es zu dieser Vision nicht gut passen, wenn Sie viele Beziehungen zur gleichen Zeit haben (jedoch kann dies gut zu der Vision »Ich will Spaß haben« passen).



Schränken Sie Ihren Pool an möglichen Terminen ein, indem Sie Work-in-progress-Limits verwenden (siehe [Kapitel 5](#) und [6](#)).

Die Roadmap to Value führt Sie durch das Date (also den Sprint) zur Ziellinie (Sprint-Review und Retrospektive). Nach dem Ende jedes Dates nehmen Sie sich Zeit, das Date selbst zu überprüfen (das ist das Sprint-Review). Fragen Sie sich, ob Sie die Person mochten und ob Sie das Zusammensein mit dieser Person als angenehm erlebt haben. Dann gehen Sie zur Retrospektive über, um mögliche Prozessänderungen vorzunehmen, beispielsweise die Art und Weise, wie Sie das Date geplant haben, die Fragen,

die Sie während des Dates gestellt haben und die Taktiken oder Werkzeuge, mit denen Sie bewertet haben, ob Ihr Datingpartner gut zu Ihnen passt. Ihre Definition of Done sollte spezifisch und klar genug sein, um einfach zu überprüfen, ob Ihre Sprints erfolgreich waren und um Ihr Backlog für den nächsten Sprint anzupassen.

Wenn eine potenzielle Beziehung endet oder nicht zum nächsten Release übergeht, kann ein anderer Produkt-Backlog-Eintrag (potenzieller Partner) seinen Platz einnehmen. Wie im Geschäftsleben ist das Umschwenken nicht schlecht, solange man früh genug umschwenkt, um Investitionen zu minimieren (in Bezug auf Zeit, emotionalen Einsatz sowie Geld), und so lange dies dazu führt, dass man seiner Vision näherkommt.

Planung Ihrer Hochzeit mit Scrum

Wenn Sie die Liebe Ihres Lebens schon gefunden haben und bereits verlobt sind, herzlichen Glückwunsch! Hochzeiten gehören zu den größten Feiern des Lebens. Sie sollen Familien und Freunde zusammenbringen und den Beginn des Lebens als Paar symbolisieren. Hochzeitstage sind voll von vielen positiven Erfahrungen, aber die Zeit vor diesem großen Tag kann zu einer Belastung für das Paar werden, das diesen Tag zu planen versucht.

Verwenden Sie Scrum für das, wofür es entwickelt wurde: die Ausführung eines wichtigen Projekts. Eine Hochzeit ist das ultimative Projekt. In der Hochzeitsplanung haben Sie einen festen Termin und (idealerweise) ein festes Budget. Es ist notwendig, Prioritäten zu setzen und Entscheidungen zu treffen, damit die Hochzeit so wird, wie Sie es sich vorgestellt haben. Die erfolgreiche Planung einer Hochzeit mit Scrum umfasst die folgenden Komponenten:

- ✓ Eine Vision festlegen. Sprechen Sie über die idealen und unverzichtbaren Elemente Ihrer Hochzeit und das Erlebnis, das Sie mitnehmen möchten.

- ✓ Einen Zeitplan (Hochzeitstag) und Budgetrestriktionen definieren.
- ✓ Hochzeitsvoraussetzungen in einer Roadmap festlegen und priorisieren (zum Beispiel Veranstaltungsort, Einladungen und Kuchen).
- ✓ Release-Ziele festlegen (beispielsweise Veranstaltungsort gebucht, beim Standesamt angemeldet, Freunde und Familie benachrichtigt, Dekorationen angebracht, Sitzplätze festgelegt, Fotoshootings eingeplant).
- ✓ Den ersten Sprint organisieren, um die Backlog-Einträge mit der höchsten Priorität anzugehen; diese Einträge in einzelne Aufgaben aufteilen und Swarming nutzen, um die Aufgaben fertigzustellen.
- ✓ Sprints ausführen und aller neuen Anforderungen, die in den einzelnen Sprints entdeckt wurden, neu priorisieren.
- ✓ Im Sprint-Review überprüfen und anpassen (nicht nur Prioritäten, sondern auch Produkt-Backlog-Einträge wie beispielsweise das verbleibende Budget) und neue Entscheidungen treffen, wie es weitergehen soll.

Wenn Sie das Hochzeits-Backlog untersuchen, können Sie sofort erkennen, welche Prioritäten gesetzt werden müssen. An manchen Veranstaltungsorten ist es beispielsweise möglich, einen externen Caterer zu beauftragen, an anderen ist es erforderlich, dass Sie auf einen bestimmten Caterer zurückgreifen. Außerdem gelten für manche Veranstaltungsorte strenge Zeitvorgaben. In Anbetracht der Tatsache, dass sich der Ort auf andere Backlog-Einträge auswirken kann, betrachten Sie ihn als einen Eintrag mit hohem Risiko (wenn nicht sogar als den Posten mit dem höchsten Risiko), den Sie in einem frühen Sprint angehen sollten. Nachdem dieser Punkt erledigt ist, kann dies Änderungen an Ihrem bestehenden Backlog nach sich ziehen. Sie können Entscheidungen über Posten treffen, die ein großes Budget erfordern, die Sie früh priorisiert haben, um zu wissen, wie hoch das verbleibende Budget für die anderen Einträge ist.

Gegen Ende der Hochzeitsplanung haben die meisten Paare noch lange Listen mit kleinen Dingen zu tun. Verwenden Sie ein Taskboard oder eine Kanban-Tafel, um kleine Aufgaben voranzubringen und zu visualisieren, was noch zu tun ist.

Hochzeiten sind teuer. Wie wir es bereits in Bezug auf die Release-Planung ([Kapitel 5](#)), das Portfoliomanagement ([Kapitel 13](#)) und die Finanzierung von Projekten ([Kapitel 14](#)) erörtert haben, stellen Sie mit Hilfe der inkrementellen Finanzierung und der Erledigung der Anforderungen mit den höchsten Prioritäten zunächst sicher, dass Sie die kritischsten Funktionen finanzieren, selbst wenn das Geld ausgeht, bevor Sie das gesamte Backlog fertigstellen. Wenn die Location und das Essen bereits gesichert sind, kann es keine große Sache sein, wenn ein paar Kleinigkeiten von der Liste gestrichen werden müssen.

Familien und Scrum

Ausgelastet scheint das Wort zu sein, dass Familien überall am besten definiert. Durch mehrere Arbeitsebenen und schulische Verpflichtungen ist es schwierig, Zeit zu finden, sich wirklich als Familie zusammenzufinden. Wenn Sie sich nicht entspannen, indem Sie mit Ihrer Familie Zeit verbringen, bleibt Ihr Stressniveau hoch und verringert Ihre Effektivität in vielen Lebensbereichen, einschließlich der Arbeit. Mit der Familie zusammen zu sein und sich zu erholen, sind wichtige Elemente für das seelische und emotionale Wohlbefinden.

Familien können in einen Kreislauf von schnelllebiger Kommunikation über Kurznachrichten und lange To-do-Listen geraten. Jedes Familienmitglied neigt dazu, an seiner eigenen Liste zu arbeiten. Die Familienmitglieder versäumen es, die Unterstützung zu maximieren, die sie sich gegenseitig geben könnten, weil es an Kommunikation und Koordination mangelt.

Die Herausforderungen, vor denen Familien stehen, sind so vielfältig wie die Familien selbst, aber die folgenden Fragen

können für die meisten Familien gelten und mit Scrum angegangen werden:

- ✓ **Widersprüchliche Familienprioritäten:** Ein Familienmitglied will Urlaub machen, ein anderes will ein Auto kaufen oder studieren, ein anderes will das Haus renovieren. Wie in der Wirtschaft können nicht alle Ziele gleichzeitig erreicht werden. Wie in erfolgreichen Unternehmen sind Planung und koordinierte Entscheidungsfindung entscheidend für die Zielerreichung.
- ✓ **Kommunikation:** Kurze Textnachrichten. Das Gespräch wurde durch einen Anruf von der Arbeit abgebrochen. Gesichter, die in mobilen Geräten vergraben sind und Fotos in sozialen Netzwerken austauschen. Kommunikation ist für Familien lebenswichtiger als für Arbeitsbeziehungen, aber Erschöpfung und Geschäftigkeit verhindern echte Kommunikation.
- ✓ **Widersprüchliche Zeitpläne:** Es ist ein Luxus für Familienmitglieder geworden, zur gleichen Zeit am selben Ort zu sein. Scrum kann Ihnen helfen zu verfolgen, wer an welcher Aktivität beteiligt ist und einen Zeitplan zu erstellen, um gemeinsam Zeit zu verbringen.
- ✓ **Persönliche Verantwortung und Verantwortlichkeit:** Jede Familie bewältigt den Druck des täglichen Lebens, wie zum Beispiel Hausarbeiten, Putzen, Besorgungen und das Bezahlen von Rechnungen. Oft fühlt sich ein Elternteil nicht ausreichend unterstützt oder überfordert oder ein Kind fühlt sich zu viel oder zu wenig verantwortlich. Für Alleinerziehende wird dieses Problem noch verschärft. Scrum kann Ihnen helfen, Verantwortlichkeiten zwischen den Familienmitgliedern auszugleichen.

Mit den Herausforderungen des modernen Lebens, das an den Bindungen zwischen den Familienmitgliedern zerrt, ist es wichtig, innovative Wege zu finden, um miteinander verbunden und geerdet zu bleiben. Stabilität und gesunde Beziehungen innerhalb

der Familie beeinflussen das, was man außerhalb der Familie tut, angefangen bei der Schule über die Karriere bis hin zu Freundschaften.

In den folgenden Abschnitten zeigen wir Ihnen konkrete Beispiele für die Umsetzung von Scrum im Alltag, indem wir die Kommunikation vereinfachen, eine Familienvision nutzen, ein Familien-Backlog priorisieren, Entscheidungen im Team treffen und die Verantwortung durch Sichtbarkeit und Eigenverantwortung erhöhen.

Familienstrategie und Projektvisionen festlegen

Die größte Herausforderung bei Familienentscheidungen sind widersprüchliche Prioritäten. In alter Zeit hatten Familien häufig Mottos oder Sprüche, die ihren Charakter bestimmten, und wir denken, dass es an der Zeit ist, diese Tradition in moderner Form zurückzubringen. Wäre es nicht schön, Konflikte innerhalb und außerhalb des Hauses anzugehen, indem man erkennt, dass das Verhalten nicht zur Familienstrategie passt?

Eine klar definierte Familienstrategie oder Mission kann Verhalten, Werte und Struktur liefern und die Familie in all ihren Entscheidungen leiten. Familienstrategien funktionieren besonders gut, wenn alle Familienmitglieder im Besitz der Strategieerklärung sind und sie nicht von den Eltern diktiert wird.

Wie wir in [Kapitel 2](#) erörtern, beginnt die Roadmap to Value mit der Vision. Eine Vision für ein Projekt muss die Gesamtstrategie oder Mission unterstützen oder es kommt zu einer Trennung zwischen der Projektleitung und der Gesamtstrategie des Unternehmens. Für Familien ist die Situation nicht anders. Wenn Familien Projekte planen, beginnen sie mit einer Vision des Projektergebnisses, die sich direkt in die Strategie der Familie integriert.

Planung und Prioritätensetzung

Die meisten Familien haben oft nur den aktuellen Tag oder sogar nur die nächsten Stunden im Blick. Aber die Realisierung von Zielen erfordert Planung und Prioritätsmanagement. Scrum bietet einen Rahmen für den Fortschritt auf dem Weg zum Ziel.

Da eine Familie ihre Strategie definiert und über Projekte und Aktivitäten entscheidet, die die Strategie ausführen, werden die Projekte und Aktivitäten klarer und einfacher zu priorisieren und zu planen. Der iterative Ansatz von Scrum bei der Planung, Durchführung, Überprüfung und Anpassung führt Familien zum Erfolg.

Projektplanung

Familien legen eine Strategie fest (falls noch nicht geschehen) und definieren dann die Vision (siehe [Kapitel 2](#)) für das Projektziel. Jedes Familienmitglied kann Ideen liefern, um die Projektvision zu verwirklichen, die zur Roadmap für dieses Familienprojekt wird. Die Familie schätzt den Aufwand und die Komplexität jeder Idee und ordnet die Ideen nach Aufwand, Komplexität, Projektwert und Risiken. Die Roadmap ist im Besitz des Familienteams.



[Kapitel 4](#) beschreibt Methoden zur Erstellung von Schätzungen. Familienschätzungen müssen vermutlich nicht die Fibonacci-Sequenz verwenden. Die Schätzung ist einfach eine Möglichkeit für die Familie, sich ein Bild von der relativen Anstrengung zu machen, eine Angelegenheit zu vollenden, sodass sie entsprechend planen können. Die Verwendung von Affinitätsschätzungen mit dem vertrauteren Konzept der T-Shirt-Größen kann schon ausreichend sein.

Release-Planung

Familien können die Releases für die jeweiligen »Produkte mit den minimalen Anforderungen und Eigenschaften« (MVP, minimal viable product) (siehe [Kapitel 5](#)) planen, die letztendlich zum Abschluss des Projektes führen, und jeweils eines der Produkte planen, nachdem das vorherige abgeschlossen ist. Wenn Sie beispielsweise eine Geburtstagsfeier planen, kann das erste

Release die Auswahl und Reservierung eines Veranstaltungsortes sein. Wenn Sie den Ort nicht kennen, sind Ihnen nicht alle Details darüber bekannt, was Sie im Rahmen der Geburtstagsfeier tun können. Wenn Sie den Veranstaltungsort kennen, können Sie Einladungen verschicken, einen Caterer buchen und Dekorationen auswählen.

Verwenden Sie immer eine Definition of Done, wenn Sie die einzelnen Sprints und Releases in Angriff nehmen. Wenn ein Eintrag im Familien-Produkt-Backlog so vage ist, dass niemand weiß, wann er als erledigt angesehen werden kann, dann ist der Eintrag noch nicht ausgearbeitet genug, um in einem Sprint ausgeführt zu werden. Vage oder ergebnisoffene Backlog-Einträge, die es in einen Sprint schaffen, verweilen oft lange Zeit in der Spalte *In Arbeit*. Wenn sich in Ihrem Familien-Produkt-Backlog Einträge wie die folgenden befinden, wissen Sie, dass diese untergliedert und verfeinert werden müssen und dass Sie definieren müssen, wie sie erledigt werden:

- ✓ Das Haus putzen. (Dieser Eintrag kann je nach Art der Reinigung und den zu reinigenden Teilen des Hauses mehrere Tage in Anspruch nehmen.)
- ✓ Geld für einen Ausflug nach Yellowstone sparen. (Dieser Eintrag sagt nichts darüber aus, wie viel Geld benötigt wird und auf welche Art und Weise das benötigte Geld verdient oder gespart werden kann.)

Sprint-Planung

Wenn man das langfristige Ziel (Vision), den Weg dorthin (Roadmap) und die großen Schritte, die nötig sind, um dorthin zu gelangen (Releases), kennt, ist es einfach, die Arbeit in Sprints aufzuteilen. Sprints von einwöchiger Dauer sind für Familien und Unternehmen ein natürlicher Rhythmus. Für die meisten Menschen sind Arbeits- und Schulpläne vorhersehbar; ein Tag der Woche dient normalerweise als Start- oder Endpunkt mit einem logischen Ort zum Zurücksetzen, Planen und Überprüfen. Selbst

dann, wenn Sie die Alltagsroutine nicht regelmäßig unterbrechen, kann die Familie eine entsprechende Zäsur einrichten.

Die Sprint-Planung für eine Familie muss nicht lange dauern. Mit einem Taskboard kann eine Familie schnell die Einträge aus dem Backlog identifizieren, die während der Woche erledigt werden können. Jedes Familienmitglied kann erkennen, wie es bei jedem Backlog-Eintrag helfen kann. Die Mitglieder verschieben die Elemente aus dem Backlog in die *To-Do*-Spalte und sprechen über die Aufgaben, die für jedes Element im Sprint-Backlog erforderlich sind. Dann untersuchen sie, wann in der Woche sie in der Lage sein werden, diese Aufgaben zu erfüllen. (In [Kapitel 5](#) finden Sie weitere Informationen zur Sprint-Planung.)

Alle Familienmitglieder freuen sich darüber, dass sie über den Plan einig sind. Jeder kann sich in der kommenden Woche den übernommenen Aufgaben widmen. Weiter hinten in diesem Kapitel wird beschrieben, wie diese Art der Planung in bestehende oder natürliche Familieninteraktionen implementiert werden kann.



Sichtbarkeit und Transparenz sind für Familien genauso wichtig wie für Unternehmen. Erstellen Sie ein Familien-Taskboard, so wie Sie es auch in einer Firma tun würden. Eine tatsächlich greifbare Memotafel ist das Beste. Kinder lieben es, einen Aufkleber, einen Magneten, einen Haftzettel oder eine Karteikarte von einer Spalte in die andere zu bewegen, genauso wie Erwachsene. Familien können auch virtuelle Taskboards verwenden. Finden Sie heraus, was für Ihre Familie am sinnvollsten ist, aber stellen Sie sicher, dass Ihr Board leicht zugänglich und sichtbar ist, da sich die Familienmitglieder jeden Tag darauf beziehen und es aktualisieren.

Daily Scrums

Es braucht nicht viel Zeit, den Familienmitgliedern jeden Morgen oder Abend die Möglichkeit zu geben zu sagen, was sie getan haben, woran sie als Nächstes arbeiten werden und welche Hilfe

sie von anderen Familienmitgliedern benötigen, um diese Aufgaben zu erfüllen – aber es kann die Erfolgsaussichten für diesen Tag erheblich verbessern.



Verwenden Sie Swarming-Techniken, um Einträge im Sprint-Backlog in die Spalte *Fertig* zu verschieben. Als gute Möglichkeit, um Führung und Unterstützung zu erlernen, können größere Kinder einem jüngeren Geschwisterkind ermöglichen, eine neue Fähigkeit zu erlernen. Als gute Möglichkeit, die Einheit zu stärken und Kooperation und Zusammenarbeit zu lehren, können die Familienmitglieder gemeinsam an Backlog-Einträgen arbeiten.

Scrum-Teams sind funktionsübergreifend. Verwenden Sie zu Hause das Konzept des Shadowing und der Paarprogrammierung, indem Sie Familienmitglieder neue Fähigkeiten vermitteln lassen, wenn es angebracht ist, damit die Verantwortlichkeiten geteilt werden können. Wenn eine Person für das gesamte Kochen verantwortlich ist, kochen Sie gemeinsam. Für das Kochprojekt kann eines der Kinder die Rolle des Product Owners übernehmen, indem es die Aufgaben für die anderen Familienmitglieder priorisiert, anordnet und dann seine Entscheidungsfähigkeiten übt, indem es die getane Arbeit abnimmt oder ablehnt. Eltern können ihre Kinder während des gesamten Prozesses begleiten.

Diese Projekte sind nicht nur Gelegenheiten zur Bindung, sondern bieten auch die Möglichkeit, lebenswichtige Fähigkeiten weiterzugeben, die es jedem Familienmitglied ermöglichen, bestimmte Aufgaben zu übernehmen.



Thrashing tritt sowohl in Familien als auch in Unternehmen auf. Familien sind wie große Organisationen, da sie häufig viele Projekte gleichzeitig durchführen. (In [Kapitel 14](#) erhalten Sie weitere Informationen über die Auswirkungen des Versuchs, zu viel auf einmal zu tun. Dort wird auch die Hierarchie des Thrashing erörtert.) Die Planung eines Familienurlaubs, eines Familientreffens und eines

Renovierungsprojekts, während gleichzeitig die Vorbereitung dafür getroffen werden, dass eines der Kinder zu studieren beginnt, sind zu viele Projekte für eine Familie, um sie noch gleichzeitig planen zu können. Wie in der Wirtschaft muss die Familienleitung entscheiden, welche Projekte die höchste Priorität haben, um ihren Erfolg zu sichern. Vielleicht verspüren Sie den Druck, alle Projekte gleichzeitig durchführen zu müssen, aber das müssen Sie nicht. Da eine Familie die Planungsstruktur von Scrum verwendet, um zu bestimmen, welche Projekte sie übernehmen soll, bewertet sie den Aufwand und die Komplexität jedes Projekts. Dann kann sie bestimmen, welche Ressourcen ihr zur Verfügung stehen, und kann die Projekte dann priorisieren und ordnen. Der allgemeine Überblick ermöglicht es den Familienmitgliedern, besser fundierte Entscheidungen zu treffen.

Thrashing führt dazu, dass die Durchführung dieser Projekte mindestens 30 Prozent mehr Zeit in Anspruch nimmt. Immer nur an einem Projekt zu arbeiten, führt schneller zum Ziel als Multitasking. In [Kapitel 13](#) finden Sie weitere Details zu diesem Thema.

Mit Scrum kommunizieren

Scrum bietet Möglichkeiten, die kritische Kommunikation zu verbessern, auch wenn die Zeit knapp bemessen ist. Scrum verwendet eine Vielzahl von Tools für die Kommunikation, von denen eines die Priorisierung der Kommunikation von Angesicht zu Angesicht ist, wann immer dies möglich ist. In einem Scrum-Team werden wichtige Entscheidungen nie per Sofortnachricht oder E-Mail getroffen. Diese eine einfache Änderung kann die Dynamik in einer Familie dramatisch beleben. Eltern wissen, dass das Verständnis dramatisch verbessert werden kann, wenn der entsprechende Tonfall und die Körpersprache vorhanden sind!

Legen Sie die mobilen Geräte in Ihrem Zuhause zur Seite und sprechen Sie direkt mit anderen Familienmitgliedern. Oder machen Sie es zu einer Hausregel: Wenn ein Gespräch

stattfindet, sollte kein Teilnehmer fernsehen, ein Videospiel spielen, an einem Computer arbeiten oder ein mobiles Gerät benutzen. Das Mantra einer Familie könnte lauten: »Schau nach oben, wenn wir miteinander sprechen.«

Persönliche Kommunikation funktioniert nur, wenn beide Personen an einem fokussierten Gespräch teilnehmen. Reduzieren Sie Missverständnisse und die Dauer von Gesprächen, indem Sie die Gewohnheit des Gesprächs-Multitasking hinter sich lassen.

Um die Kommunikation von Angesicht zu Angesicht zu nutzen, können einige bestehende Familienaktivitäten ein natürliches Forum für Sprint-Planung, Daily Scrums, Sprint-Reviews und Retrospektiven bieten. Eine dieser Aktivitäten ist die Familienmahlzeit.

Der Autor und Sprecher Bruce Feiler hielt einen TED-Vortrag zum Thema »Agile Programmierung – für Ihre Familie«

(www.ted.com/talks/bruce_feiler_agile_programming_for_your_family/transcript), in dem er die dramatischen Veränderungen beschreibt, die seine Familie in ein paar Minuten pro Tag mittels des Modells des Daily Scrum erzielt hat.

Denken Sie an den Nutzen, den Ihre Familie aus einem Daily Scrum in einem einfachen 15-Minuten-Zeitfenster ziehen kann! Im Folgenden finden Sie einige Beispiele für die Vorteile, die Ihre Familie daraus ziehen kann, sich die Zeit zu nehmen, ohne Ablenkung zu kommunizieren:

- ✓ Von jedem Mitglied der Familie hören, was während seines Tages passiert ist
- ✓ Einander die Pläne oder Ziele für den nächsten Tag mitteilen
- ✓ Die aktuellen Herausforderungen jedes Familienmitglieds kennenlernen
- ✓ Die Möglichkeit, Probleme im Team zu lösen oder sich gegenseitig bei der Bewältigung von Herausforderungen zu unterstützen

✓ Täglich Erfolge feiern

Obwohl das Familienessen ein guter Zeitpunkt für ein Familientreffen ist, können diese wichtigen 15 Minuten zu jeder Tageszeit stattfinden. Ein Daily Scrum beim Frühstück, beim Abendessen oder sogar vor einer bestimmten Fernsehsendung wird funktionieren, solange das Zeitfenster konsistent und auf 15 Minuten begrenzt ist. Daily Scrums sind zu kurz, um alle Probleme des täglichen Lebens zu lösen, aber sie erfüllen die Aufgabe, Kommunikationswege zu öffnen und Vertrauen und Unterstützung aufzubauen. So ist bei größeren Herausforderungen das Fundament der gegenseitigen Unterstützung bereits etabliert.

Überprüfung und Anpassung für Familien

Sprint-Reviews und Retrospektiven sind für Familien ebenso wichtig wie für Unternehmen. Für vielbeschäftigte Familien kann es sinnvoll sein, die Leistungen zu überprüfen (Sprint-Review) und dann die verwendeten Prozesse und Werkzeuge sowie die Kommunikations-, Beziehungs- und Disziplintechniken (Sprint-Retrospektive) während derselben Meetingzeit kurz vor der Planung einer neuen Woche zu überprüfen (Sprint-Planung).

Sprint-Reviews können Ergebnisse der Schulwoche, Fortschritte bei der Planung der nächsten Familienferien und Erfolge bei Sport, Musik und anderen Aktivitäten beinhalten. Sprint-Reviews sind auch die Zeiten, Ereignisse zu überprüfen, die für die Zukunft festgelegt werden. So können Backlog-Einträge identifiziert werden, die dabei helfen, diese Ereignisse vorzubereiten. Es kann sich dabei um eine Geburtstagsfeier handeln oder darum, das Haus für den mehrtägigen Besuch entfernter Verwandter vorzubereiten.

Für diese wöchentlichen Familientreffen folgen die nachstehenden Fragen dem Sprint-Retrospektive-Modell von Scrum:

- ✓ Was hat in dieser Woche gut funktioniert?
- ✓ Was hat nicht gut funktioniert?
- ✓ Woran können wir in der nächsten Woche arbeiten?

Sie werden vielleicht vom Feedback der Familienmitglieder überrascht sein. Während des Familientreffens sollte sich jeder für zwei oder mehr Verbesserungen entscheiden, an denen er in der folgenden Woche arbeiten will. Diese Vereinbarung wird in der agilen Welt als Teamvereinbarung bezeichnet.

Die Familie kann auch wählen, welche Konsequenzen sich ergeben, wenn Regeln verletzt werden oder wenn ein Familienmitglied seinen Verpflichtungen nicht nachkommt. Auch hier gilt, dass sich das Team (die Familie) in einem Scrum-Modell nicht auf das traditionelle Modell, bei dem die Konsequenzen auf das Handeln folgen, einigt, sondern auf die Lektionen, die gelernt wurden, und wie man sie verbessern kann, bevor man das neue Sprintziel annimmt. Das Team stimmt auch über die Vorteile oder Belohnungen der Erreichung neuer Ziele überein. Die Belohnung für das Erreichen des Familienziels ist ein wunderbarer Motivator.

Tägliche Pflichten können Spaß machen

Es wäre wunderbar, wenn die Alltagspflichten und Aufgaben einfach ohne Schwierigkeiten erledigt werden könnten, anstatt in Streitigkeiten zu eskalieren. Viele Eltern entdecken, wie sie ihre Kinder dazu bringen, sich selbst zu motivieren, indem sie persönliche Taskboards oder Kanban-Tafeln verwenden.

Der Mensch hat ein angeborenes Bedürfnis nach dem Gefühl, etwas erreicht zu haben. Die meisten Erwachsenen legen To-do-Listen an und mögen die Dinge, die einfach und schnell zu erledigen sind, sodass sie von der Liste gestrichen werden können. Auch Kinder sehnen sich danach, etwas vollbracht, etwas geleistet zu haben. Sie müssen wissen, dass ihre Beiträge wichtig sind. Die Teilnahme am Haushalt ist absolut notwendig, damit Kinder wissen, dass sie geschätzt werden. Die Wahl

altersgerechter Möglichkeiten für Kinder, sich an der Haushaltsführung zu beteiligen, macht die Familie zu einer Einheit. Taskboards machen es jedem leicht zu sehen, was es heißt, erfolgreich zu sein.



Wenn Kinder mit größeren Aufgaben zu kämpfen haben, behandeln Sie die größere Aufgabe als User Story und helfen Sie mit, indem Sie sie in wichtige Aufgaben unterteilen. Wenn die Kinder Mühe haben, »Mein Kinderzimmer aufräumen« zu bewerkstelligen, machen Sie »Mein Kinderzimmer aufräumen« zur User Story, die dann aus kleineren Aufgaben besteht wie beispielsweise »Bücher aufräumen«, »Schmutzige Wäsche wegräumen«, »Bett machen«, »Fenster putzen«, »Spielzeug wegräumen« und »Bücherregale abstauben«. Im Taskboard können die Kinder dann in der Spalte *In Arbeit* die User Story »Mein Kinderzimmer aufräumen« zusammen mit anderen kleineren Aufgaben stehen haben, bis alle Aufgaben erledigt sind. Sie können um Hilfe bei größeren, schwierigeren Aufgaben bitten, bis sie lernen, wie sie diese erledigen können.



Wie im Geschäftsleben sollten die Anforderungen an ein Familien-Taskboard (User Stories) diejenigen sein, die innerhalb eines Sprints erfüllt werden können, und die Aufgaben sollten diejenigen sein, die innerhalb eines Tages oder eines kürzeren Zeitraums erledigt werden können. Für jüngere Kinder sollten die Aufgaben eine Stunde oder weniger dauern, damit sie sie erfolgreich erledigen können.

Taskboard für die Zeit vor dem Schlafengehen

Blogger Chris Scott von der Agile School

(<http://theagileschool.blogspot.co.uk>) hat ein kurzes Video seiner 5-jährigen Tochter gepostet. Sie bereitet sich darin mithilfe eines einfachen Taskboards auf das Schlafengehen vor. Jede Aufgabe wird auf einem Blatt Papier mit einem Bild

dargestellt, das ihr hilft, sich an die Aufgabe zu erinnern. Auf Anregung ihres Vaters schaut sie sich die To-do-Spalte an und wählt die nächste Aufgabe aus, die durch ein Foto eines Welpen (das »Füße waschen« symbolisiert) dargestellt wird, und verschiebt sie in die Spalte *In Arbeit*. Dann wendet sie sich an ihren Vater und sagt ihm, dass es Zeit ist, ihre Füße zu waschen. Dieses Video ist ein großartiges Beispiel dafür, dass auch die kleinsten Kinder die Grundlagen von Scrum verstehen und Spaß daran haben, Verantwortung zu übernehmen. Wir kennen viele Familien, die Schwierigkeiten damit haben, kleine Kinder ins Bett zu bringen – das Taskboard hilft einem kleinen Kind, sich praktisch selbst ins Bett zu bringen.

Kapitel 18

Scrum für Lebensziele

IN DIESEM KAPITEL

Finanzielle Sicherheit aufbauen
Gewichtsabnahmeziele erreichen
Ausgeglichenheit und Wohlbefinden pflegen
Familienurlaub planen
Erfolgreich in Schule und Studium

Wenn Sie Luftschlösser gebaut haben, dann muss Ihre Arbeit nicht unbedingt vergebens gewesen sein; genau dort gehören sie hin. Nun sollten Sie die Fundamente dafür errichten.

Henry David Thoreau, Walden

Mittlerweile sollte deutlich geworden sein, dass sich Scrum universell anwenden lässt. Scrum ist so einfach, dass es auf jedes geschäftliche oder persönliche Ziel angewandt werden kann, und anpassungsfähig genug, dass es mit jedem wichtigen Projekt funktioniert, das Sie angehen möchten.

Jeder hat Ziele im Leben, die sich vielleicht als zu groß und zu weit weg anfühlen. Das Beste am Scrum ist, dass das, was einen zu erdrücken scheint, in überschaubare Teile zerlegt werden kann.

Der Weg in den Ruhestand

Es gehört zur Realität des Lebens, dass eine Zeit kommen wird, in der Sie nicht mehr Vollzeit arbeiten können. Der Ruhestand kann als der Zeitpunkt definiert werden, an dem Ihr passives Einkommen – das Einkommen, für das Sie nicht aktiv arbeiten – ausreicht, um Ihre Ausgaben zu decken. Wenn Sie den Ruhestand so definieren, wird es zum Ziel in jedem Alter, darauf hinzuarbeiten, und Ihr Produkt-Backlog enthält dann Einträge und Release-Ziele, die das Erreichen der Vision der finanziellen Freiheit unterstützen. Je weiter der Ruhestand entfernt ist, desto schwieriger kann es sein, sich auf die Dinge zu konzentrieren, die Sie heute tun können, um diese wichtigen finanziellen Ziele zu erreichen, da auch aktuell finanzielle Bedürfnisse auftauchen.

Eine große Herausforderung für jüngere Generationen besteht darin, mit den steigenden Lebenshaltungskosten fertig zu werden, während die Löhne nicht im gleichen Maße steigen. Mit Scrum können Sie sich dieser Herausforderung und den Fortschritten in Richtung finanzielle Sicherheit in jedem Jahr stellen.

Der beste Weg, um ein langfristiges Zukunftsziel in die Gegenwart zu bringen, ist, die Roadmap to Value zu nutzen, um Ihre Ruhestandsvision zu entwickeln, Release-Ziele und Ihren Produkt-Backlog zu erstellen und Sprints zu nutzen, um auf die einzelnen Ziele hinzuarbeiten.

Für den Notfall sparen

Ihr erstes Release-Ziel in Richtung der Vision der finanziellen Unabhängigkeit sollte das Sparen für finanzielle Notfälle beinhalten. Notfälle sind vorprogrammiert und können mit erheblichen finanziellen Kosten verbunden sein.

Gesundheitskosten, ein Arbeitsplatzverlust, Reparaturen an Ihrem Haus oder Auto, und Scheidung oder andere Familienangelegenheiten, die zu unbezahlten Freistellungen führen, können finanzielle Belastungen verursachen. Die einzige Möglichkeit, Ihre zukünftige finanzielle Sicherheit zu schützen, ist, jetzt für Notfälle zu planen, indem Sie einen Sparfonds aufbauen, der Notfälle abdeckt, zu Zinseinnahmen führt und bei dem Sie

keine Gebühren zahlen müssen, falls Sie für einen Notfall Geld entnehmen müssen.

Ein Notfallfonds ist eines der Release-Ziele in Ihrer Vision zur Erlangung finanzieller Sicherheit. Sprints sollten Möglichkeiten zur Anpassung Ihres Budgets beinhalten, wobei Sie sich auf das Beiseitelegen von Geld konzentrieren sollten.

Das häufigste finanzielle Ziel beim Notfallsparen besteht darin, den Gegenwert der Lebenshaltungskosten für sechs Monate auf der hohen Kante zu haben. Um dieses Release-Ziel zu unterstützen, legen Sie die Sprintdauer fest (die nicht länger als einen Monat dauern sollte) und bearbeiten in jedem Sprint die Backlog-Einträge, die den Aufbau eines Sparplans unterstützen.



Ihr Einkommen abzüglich der Ausgaben entspricht einem Überschuss. Überschüsse, die im Laufe der Zeit Zinsen generieren, führen zu Vermögen. Die Gleichung lautet:

$$\text{Überschuss} \times \text{Zins} \times \text{Zeit} = \text{Vermögen}$$

Wenn Sie am Ende jedes Lohnzyklus wenig oder keinen Überschuss haben, müssen Sie Ihr Budget überprüfen und Ihr Backlog mit den Prioritäten neu ordnen, um Ihr Einkommen zu erhöhen, Ihre Ausgaben zu senken oder beides. Denken Sie an dieses Motto: Leben Sie frei, bevor Sie gut leben. Es mag schön sein, teure Dinge zu besitzen, solange diese nicht Ihre finanzielle Freiheit beeinträchtigen oder diese Ihre zukünftige Sicherheit kostet.



Beim Aufbauen Ihrer Ersparnisse sollten Sie dem Drang widerstehen, die Notfallmittel für nicht dringende Dinge zu nutzen. Dies ist der sicherste Weg, Ihren Fortschritt auf dem Weg zum Ziel zu boykottieren. Erstellen Sie ein separates Sparkonto für Konsumausgaben (beispielsweise Reisen, ein neues Paar Skier oder ein neues Auto), um die Verwendung der Notfallmittel für diese Zwecke zu vermeiden. Die

Anerkennung des Bedarfs an Vergnügungskäufen ist ein wichtiger Schritt für einen nachhaltigen Sparplan. Der Schlüssel liegt darin, kurzfristige Spaßkäufe gegen langfristige Sicherheitsziele abzuwägen.

Ihre Sprints sollten sich konsequent auf Möglichkeiten konzentrieren, Überschüsse zu erzielen, die dann gespart werden können. Untersuchen Sie die Ergebnisse der einzelnen Sprints, damit Sie diese schnell prüfen, anpassen und ändern können. Bleiben Sie beispielsweise nicht Sprint für Sprint in passiven Einkommensbemühungen verhaftet, mit denen Sie Ihr Ziel nicht erreichen können. Sprint- und Releaseziele und Ihre Definition of Done sollten Ihnen dabei helfen, Backlog-Einträge zu definieren, die spezifisch sind und die klar angeben, was Erfolg bedeutet. Haben Sie keine Angst vor Fehlschlägen und machen Sie mit dem aus Ihren Fehlern Gelernten weiter. Lean Startup (siehe [Kapitel 13](#)) kann für Sie hilfreich sein, wenn Sie sich für den unternehmerischen Ansatz zur Steigerung Ihres Einkommens entscheiden.

Planen für den Ruhestand

Nachdem Sie ein Sicherheitsnetz haben, sollten Sie den Fokus verschieben und für den Ruhestand sparen. Dieser Teil der Zielfortschritte bleibt bestehen, auch wenn Sie zu anderen Release-Zielen übergehen. Ihr Sprintzyklus beinhaltet die Prüfung Ihrer Optionen, um den größten Return on Investment zu erzielen. Am Ende des ersten Sprints sollten Sie, wenn die Arbeitgeberleistungen entsprechende Mittel enthalten, Ihren Produkt-Backlog mit Einträgen aktualisieren, mit denen Sie die bestmögliche Übereinstimmung erzielen.

Bei der Durchführung der Sprints, in denen Sie für den Ruhestand sparen, sollte Sie sich darauf konzentrieren, wie Sie sicher den maximalen Return on Investment erzielen können. Wenn Sie weiter vom Ruhestand entfernt sind, scheuen Sie sich nicht, in den ersten Jahren, in denen Sie sparen, aggressiver mit Ihren Investitionen umzugehen. Sie können online nach Informationen suchen, die Ihnen helfen, festzustellen, wie viel Sie sparen

sollten, um ein Gleichgewicht zwischen Ihrer Zukunftsvision und Ihrem aktuellen Budget zu entwickeln. Denken Sie bei der Nutzung dieser Instrumente daran, dass wir den Ruhestand als den Punkt definieren, an dem Sie aufhören können, aktiv zu arbeiten, weil Ihr passives Einkommen ausreicht, um Ihre Ausgaben zu decken. Obwohl die meisten Online-Tools und Altersberatungsrechner ein Alter von 65 Jahren oder älter vorschlagen, wird Ihr Rentenalter tatsächlich von Ihnen und Ihrer Vision bestimmt.

Sicherung der finanziellen Freiheit

Nachdem Sie die beiden Release-Ziele Notfallsparen und aktives Sparen für den Ruhestand erreicht haben, ist Ihr nächstes Release-Ziel der Aufbau von Vermögenswerten. Wir verwenden einfache Definitionen für Vermögenswerte und Verbindlichkeiten. Ein Vermögenswert ist alles, was Ihnen Geld bringt, und eine Verbindlichkeit ist alles, was Sie Geld kostet. Der Aufbau von Vermögenswerten ist eine persönliche Entscheidung und kann mehrere Sprints erfordern, damit Sie forschen und Ihr Wissen erweitern können, um dann Ihre Vision davon zu unterstützen, wohin die Reise gehen soll. Sie können einen Spike (agiler Begriff für Forschung oder Risikobewertung) als Eintrag im Produkt-Backlog verwenden, um ein Problem zu erforschen und zu analysieren, damit Sie eine Frage beantworten können.

Beispiele für frühe Sprintziele, die Sie erwägen könnten, sind unter anderem:

- ✓ Budget strukturieren, um eine bestimmte Anzahl von Indexfondskäufen pro Monat zu ermöglichen und zu automatisieren.
- ✓ Möglichkeiten finden, das Einkommen um 5 Prozent zu steigern, indem Sie etwas tun, was Ihnen Spaß macht.

Ihre Sprintziele werden dann auf die Ebene der Anforderungen heruntergebrochen, wie beispielsweise:

- ✓ Ein Budget erstellen.
- ✓ Den automatischen Kauf von Indexfonds einrichten.
- ✓ Eine Website für Ihr Online-Geschäft entwickeln.

Ein weiteres Beispiel für ein Sprintziel ist es, festzustellen, ob der Kauf eines Hauses einen Vermögenswert oder eine Verbindlichkeit schaffen würde. Die Produkt-Backlog-Einträge beinhalten Schritte und Aktivitäten, um herauszufinden, ob der Kauf eines Hauses die beste Verwendung Ihres Geldes darstellt:

- ✓ Eine Analyse durchführen, ob das Haus einen ausreichenden Return on Investment bringt.
- ✓ Untersuchen, ob ein anderer Ort oder eine andere Investition eine höhere Rendite bringen würde.

Ein großes, geräumiges Haus kann nett sein, aber es erhöht gleichzeitig Ihre Verbindlichkeiten, falls Sie es nicht leicht mit Gewinn verkaufen können. Recherchieren Sie sorgfältig und verwenden Sie frühe Sprints, um das Release-Ziel des Aufbaus von Vermögenswerten zu überprüfen und anzupassen. Beim Vermögensaufbau geht es darum, passives Einkommen aufzubauen, das Ihre Vision vom Ruhestand unterstützt.



Häuser beziehungsweise Eigentumswohnungen sind Verbindlichkeiten (sie kosten Sie Geld), bis sie ein Einkommen durch eine Miete generieren oder mit Gewinn verkauft werden. Ihr Hauptwohnsitz ist wahrscheinlich kein Vermögenswert; Sie müssen irgendwo wohnen.

Wenn Sie Scrum für den Aufbau finanzieller Freiheit verwenden, verwenden Sie auf die gleiche Weise wie bei einem Softwareprojekt empirische Nachweise und den Überprüfungs- und Anpassungsprozess. Sie können sehen, ob Sie sich spürbar auf das Ziel zubewegen und bestimmen, wie Sie das Ziel schneller erreichen können. Sie können erkennen, welche Hindernisse Ihnen im Weg stehen und einen Plan für deren Beseitigung entwickeln.

Ein perfektes Beispiel für den Aufbau von Vermögenswerten und die Beseitigung von Hindernissen ist die akademische Ausbildung. Wenn Ihr Bildungsstatus Sie daran hindert, sich finanziell weiterzuentwickeln, prüfen Sie das potenzielle Vermögensergebnis gegen die Verbindlichkeit. Einige Fragen, die Sie beachten sollten, sind:

- ✓ Wie viel wird die Ausbildung, die ich in Betracht ziehe, kosten (tatsächliche Kosten plus Alternativkosten)?
- ✓ Wie viel Einkommen wird die Ausbildung bringen, wenn ich fertig bin?
- ✓ Sind die Einnahmen, die ich nach der Ausbildung erzielen kann, größer als die Kosten für die Ausbildung?

Wie beim Kauf eines Hauses sollten Sie einen klaren finanziellen Vermögenswert für die zusätzliche Ausbildung erzielen können, bevor Sie mit ihr beginnen.

Eine Grube graben

Eine aktuelle Studie der Federal Reserve, der US-amerikanischen Notenbank (www.federal-reserve.gov/releases/g19/current/default.htm) zeigt, dass College-Absolventen des Jahrgangs 2016 ihr Studium mit durchschnittlichen Studienschulden in Höhe von 37.172 US-Dollar abgeschlossen haben. Diese Absolventen, die jetzt 20 bis 30 Jahre alt sind, müssen je Monat durchschnittlich 350 US-Dollar zurückzahlen.

Insgesamt belaufen sich die Darlehen an US-Studenten auf 45 Billionen US-Dollar, verteilt auf mehr als 44 Millionen Kreditnehmer. Laut Gallup (<http://news.gallup.com/businessjournal/188984/americans-big-debt-burden-growing-not-evenly-distributed.aspx>), »ist die Situation für ein Drittel der Millennium-Generation (35%), die Studentenschulden haben, in der Tat entsetzlich«. Darüber hinaus haben 64 Prozent der Millennium-Generation

Verbraucherschulden. Wenn die Studiendarlehen hinzugerechnet werden, dann betragen die durchschnittlichen Schulden der Millennium-Generation 40.000 US-Dollar.

Schuldentilgung

Ein weiterer Aspekt Ihrer finanziellen Unabhängigkeit kann die Schuldenfreiheit sein. Scrum kann eine effektive Strategie bieten, um Schulden zu tilgen, indem die Leistungen in kleine, messbare Schritte wie die folgenden aufgeteilt werden:

- ✓ Behandeln Sie die Tilgung eines Darlehens oder Ihrer Kreditkartenschulden wie ein Release-Ziel.
- ✓ Bezahlen Sie mehr als die minimale Rate zurück, damit die Gesamtverschuldung sinkt.
- ✓ Schauen Sie sich rückblickend den Fortschritt an und feiern Sie eine Party, wenn Sie Ihr Ziel erreicht haben.

Fitness- und Gewichtsziele erreichen

Wenn Fitness ein Ziel für Sie ist, ist der Einsatz von Scrum zur Erreichung Ihrer Vision einer der besten Wege zum Erfolg. Viele Menschen kämpfen im Hinblick auf Gewichtsverlust und Fitness mit dem sogenannten Jo-Jo-Effekt. Viele Leute beginnen mit einem Fitness- und Diätprogramm, aber oft rutschen sie nach Erreichen des Ergebnisses langsam aber sicher dahin zurück, wo sie angefangen haben.

Einer der Gründe, warum dies passiert, ist die Fokussierung. Der Nachteil bei der Gewichtsabnahme ist, dass der Fokus oft auf einer extremen, reglementierten Situation liegt. Wenn Sie dann sozusagen rückfällig werden, bricht Ihre hart erarbeitete Arbeit zusammen. Scrum ermöglicht Fokussierung, jedoch liegt diese Fokussierung auf kleinen, messbaren und erreichbaren

Abschnitten. Mit anderen Worten, bei Scrum geht es darum, Schritte zu Ihrem Ziel zu unternehmen und es nachhaltig zu erreichen, und nicht nur auf eine extreme Achterbahn zu springen und dann auszubrennen.

Arbeiten Sie die Roadmap wie bei jedem anderen Großprojekt durch. Im Folgenden finden Sie einige Beispiele für die Anwendung der Roadmap to Value für Gewichtsabnahme und Fitness, die Sie auf Ihre eigenen Visionen und Ziele zuschneiden können:

- ✓ **Legen Sie Ihre Vision fest.** Sie möchten zu Ihrer körperlichen Form wie in Studienzeiten zurückkehren; 85 kg, Jeansgröße 32, 1.500 Meter in 7 Minuten laufen und 90 kg Bankdrücken.
- ✓ **Erstellen Sie eine Produkt-Roadmap.** Anfängliche Einträge in der Roadmap- können Dinge sein wie: 10 Pfund abnehmen (dieses Element kann mehrere Male vorkommen, weil inkrementelle Verbesserung das Ziel ist), 750 Meter laufen, ohne anzuhalten, oder die Senkung Ihres Blutdrucks.
- ✓ **Erstellen Sie ein Produkt-Backlog.** Dieses Backlog könnte Einträge wie die Verwendung anderer Kochrezepte, die Mitgliedschaft in einem Fitness-Studio und einen Ernährungs- und Bewegungsplan enthalten.
- ✓ **Legen Sie Ihr erstes Release-Ziel fest.** In zwei Monaten können Sie 3 Pfund verloren haben und in der Lage sein, 1.500 Meter zu laufen.
- ✓ **Sprintlängen bestimmen.** Ein Sprint kann beispielsweise eine Woche dauern.
- ✓ **Wählen Sie aus, was im ersten Sprint geschehen soll.** Sie können sich dafür entscheiden, 50 Prozent weniger Erfrischungsgetränke (die mit dem hohen Zuckergehalt) zu trinken, nur drei Mal pro Woche Dessert zu essen, einmal pro Woche 1.500 Meter zu gehen sowie mindestens einmal wöchentlich an Aerobic-Aktivitäten teilzunehmen.

Überprüfen Sie am Ende jedes Sprints Ihre Fortschritte in Richtung der Ziele, aktualisieren Sie Ihren Produkt-Backlog mit dem, was Sie während des Sprints gelernt haben, und passen Sie die nächsten Schritte an, damit diese an Ihrem Release-Ziel ausgerichtet sind.

Auch nach nur einem Sprint sollten Sie die Sprint-Retrospektive als Möglichkeit zur Überprüfung und Anpassung nutzen. Stellen Sie sich in der Sprint-Retrospektive die folgenden drei Fragen:

- ✓ **Was lief gut?** Sie könnten beispielsweise Folgendes formulieren: »Die Kochwebsite, die ich benutzt habe, hat gute Rezepte. Ich sollte sie weiter benutzen. Die mobile App zur Kalorienverfolgung ist einfach zu bedienen. Meine Familie unterstützt mich sehr.«
- ✓ **Was will ich ändern?** Sie könnten beispielsweise Folgendes formulieren: »Ich mag es nicht, an Tagen, an denen ich Süßigkeiten esse, Kardiotraining zu machen. Workouts am späten Abend zu machen, fällt mir schwer. Die Gruppe, mit der ich zu Mittag essen, macht es mir schwer, bei meinen neuen Gesundheitszielen zu bleiben, was entmutigend ist.«
- ✓ **Wie kann ich diese Veränderung erreichen?** Sie könnten beispielsweise Folgendes formulieren: »Ich werde feste Wochentage für Süßigkeiten festlegen, die keine Kardiotage sind.« Sie könnten im folgenden Sprint versuchen, die Workouts morgens zu machen, um zu sehen, ob es Ihnen dann leichter fällt. Sie können das gemeinsame Mittagessen mit der Gruppe auf einmal wöchentlich reduzieren, es ganz sein lassen oder sich eine neue Gruppe suchen.

Führen Sie Ihren nächsten Sprint aus, indem Sie sowohl das, was Sie verbessern wollen, als auch die neuen Einträge aus Ihrem Backlog einbeziehen. Überprüfen Sie nach zwei Monaten das gesamte Release und untersuchen Sie, ob Sie Ihr Ziel erreicht haben, und legen Sie Ihr nächstes Release-Ziel fest.

Entscheidend bei Verwendung von Scrum, um Ihrem Ziel der Gewichtsabnahme näherzukommen, ist die Erkenntnis, dass

jeder Schritt ein kleiner, aber wirklich inkrementeller Schritt ist. Es ist zu keinem Zeitpunkt ein Misserfolg, das Ziel nicht zu erreichen, sondern stattdessen eine Gelegenheit, einen neuen Weg zu finden, um voranzukommen. Auch wenn Sie in schlechte Gewohnheiten zurückfallen, ist es keine großartige Anstrengung, wieder auf Kurs zu kommen, weil die Sprintzyklen so kurz sind.

Verwenden Sie ein gut sichtbares Taskboard für die Elemente, die den spezifischen Status »Fertig« haben. Um das vorhergehende Beispiel fortzusetzen, könnten Sie drei Workout-Aufgaben haben, die Sie immer dann, wenn Sie sie erledigt haben, in die Spalte *Fertig* verschieben. Eine visuelle Darstellung der erledigten Übungen hilft Ihnen auch, Chancen (Übungen, die Ihnen Spaß machen) und Hindernisse (Übungen, die Sie vermeiden) zu erkennen und Ideen für Ihren nächsten Sprint zu entwickeln.



Ein Beispiel für einen sichtbaren und realisierbaren Plan, der vielen Erfolg gebracht hat, ist ein Plan für Anfänger zur Teilnahme an einem Halbmarathon (www.halhidon.com/training/51131/Half-Marathon-Novice-1-Training-Program). Das Ziel für die einzelnen Sonntage während des Trainings sind klar und inkrementell.

Das Leben im Gleichgewicht halten

Eine große Herausforderung im Leben ist das Management des emotionalen und mentalen Wohlbefindens. Das Leben kann eine Achterbahnfahrt sein und es liegt in Ihrer Verantwortung, einen Weg zu finden, die Volatilität zu minimieren und den Genuss und die Erfüllung zu maximieren.

Psychische und emotionale Gesundheit ist ein umfassendes Thema, das von einer vorübergehenden Zunahme des Lebensstresses bis hin zu diagnostizierbaren psychischen Problemen reicht. Scrum ist ein Framework für die Adressierung

von Zielen und die Priorisierung von Themen, kein Ersatz für psychische oder emotionale Gesundheitsdienste. Sie werden sich wahrscheinlich mit jemandem zusammentun wollen, dem Sie vertrauen, um kontinuierliche Unterstützung und Begleitung zu erhalten, während Sie Ihre Vision, Ihre Roadmap und Ihr Backlog aufbauen und die Scrum-Ereignisse durcharbeiten. Gehen Sie diesen Weg nicht alleine.

Wenn das Leben einen überwältigt, kann dies unterschiedliche Ursachen haben (einige davon werden in [Kapitel 17](#) wie auch in den bisherigen Beispielen in diesem Kapitel behandelt). Die Bausteine unseres Lebens werden im Laufe der Zeit bewegt und verschoben. Bei Stress kann jeder an seine Belastungsgrenze kommen. Für eine Person könnte eine schlecht verlaufende Trennung der Lebensbaustein sein, der die anderen Stressfaktoren des Lebens als zu viel erscheinen lässt. Für eine andere Person ist die finanzielle Not oder der Verlust eines Arbeitsplatzes der Baustein, der zusammen mit anderen Lebensfaktoren das Gefühl entstehen lässt, dass alles einfach zu viel ist.

Wenn Stressfaktoren lästig werden, verwenden Sie Scrum, um Lösungen zu finden. Hier sind einige Fragen, die Sie sich stellen sollten, wenn Sie beginnen, Ihre Vision, Ihre Roadmap und Ihr Backlog zu definieren:

- ✓ Was sind meine Stressoren?
- ✓ Warum verursachen sie gerade jetzt so viel Stress?
- ✓ Welcher Stressor überwältigt mich?

Wenn Sie sich diese Fragen stellen, können Sie die Quelle der Probleme und den Faktor identifizieren, der derzeit den größten Stress verursacht. Von hier aus beginnen Sie, die Roadmap to Value zu durchlaufen.

Mit dem derzeit höchsten Stressfaktor als Priorität erstellen Sie eine Vision davon, was Sie in Ihrem Leben sehen möchten. Ab diesem Punkt gliedern Sie Ihr erstes Release auf und

identifizieren dann die Sprintziele und Aufgaben, die Sie durchführen können, um dieses Release zu erreichen.

Wie in den Sprints bei den anderen Lebenszielen, denen Sie nachstreben, kehren Sie konsequent zur Überprüfung und Anpassung zurück. Sie fragen sich vielleicht: »Ist das immer noch der größte Stressfaktor und der, auf den ich mich konzentrieren muss, damit ich ausgeglichener werden?« Wenn ja, arbeiten Sie weiter, bis Sie das Release-Ziel erreicht haben. Wenn nicht ($TK + AK > W$; siehe [Kapitel 5](#)), beenden Sie dieses Release und setzen Sie sich ein neues Release-Ziel.

Während Sie sich von Sprint zu Sprint bewegen und ein Thema nach dem anderen angehen, werden Sie vielleicht zu Ihrer Überraschung feststellen, dass Sie weniger Stress verspüren, aber dass ein anderer Backlog-Eintrag jetzt eine höhere Priorität besitzt. Dann ist es an der Zeit, sich auf dieses Element zu konzentrieren, um sich geistig und emotional stabiler zu fühlen.



Das Hin und Her zwischen Projekten verlängert nicht nur die benötigte Zeit um mindestens 30 Prozent, sondern es kann auch erdrückend sein. Im Hinblick auf Ihr mentales und emotionales Wohlbefinden ist es am besten, wenn Sie sich immer nur auf eine Sache konzentrieren.

Alltagsstress reduzieren

John ist Anfang 30; er kämpft mit dem Gefühl der Überforderung und weiß nicht, wie er mit den vielen Frustrationen in seinem Leben umgehen soll. Zuerst untersucht er seine Stressoren und entdeckt Folgendes:

- ✓ Die Arbeit ist sehr stressig und findet in einem negativen Umfeld statt; außerdem wird sie nicht gut bezahlt, was finanziellen Stress verursacht.
- ✓ Seine jetzige Wohnung ist ein stressiger, unsicherer Ort zum Leben; gleichzeitig kann er nicht umziehen, da er mit seiner Arbeit nicht gut verdient.

- ✓ Er fühlt sich oft allein und obwohl er eine Beziehung will, hat er bis jetzt nicht die richtige Person gefunden, und hat das Gefühl, nicht für eine Beziehung bereit zu sein. Seine Freunde und Familienmitglieder erleben ihn als distanziert, weil er bei der Arbeit und zu Hause gestresst ist.

John entscheidet, dass seine schlechte Arbeitssituation der schlimmste dieser Faktoren ist (Prioritätsentscheidung). Er schreibt eine Vision, die widerspiegelt, was er will: »Ich will einen Job, bei dem ich mindestens 80.000 Dollar im Jahr verdiene, mit allen Zusatzleistungen und festen Arbeitszeiten, und nicht die langen, unplanbaren Arbeitstage, mit denen ich derzeit konfrontiert bin.« (Diese Vision ist klar und konkret.)

Sein erstes Release-Ziel ist vielleicht die Suche nach neuen Jobs. In seinem ersten Sprint kann er beispielsweise je Abend eine Stunde mit der Aktualisierung seines Lebenslaufs und der Bewerbung für fünf neue Jobs verbringen. Außerdem kann der Sprint Aufgaben wie die erneute Kontaktaufnahme mit Bekannten, die Suche nach potenziellen Jobs oder die Suche nach Stellen mit einem Gehalt ab 80.000 US-Dollar und mit bestimmten Arbeitszeiten enthalten.

Nach mehreren Sprints, in denen er sich auf neue Jobs konzentriert, geht John in die Vorstellungsphase einschließlich Bewerbungsgesprächen. Nach jedem Sprint (das heißt, nach jedem Vorstellungsgespräch), führt er ein Review durch und verwendet den Ansatz von Überprüfung und Anpassung, um seine Jobsuche anzupassen oder um zu überprüfen, welcher Stressfaktor als nächstes den Fokus erhält.

Zu jedem beliebigen Zeitpunkt während eines Sprints könnte John entscheiden, dass er seinen Stress genug reduziert und das Release-Ziel erreicht hat und er sich auf einen neuen Stressfaktor konzentrieren kann. Entscheidend bei der Verwendung von Scrum zur Aufrechterhaltung und Wiedererlangung des mentalen und emotionalen Gleichgewichts ist, dass hierbei das Konzept von messbaren Schritten, mit denen das Sprintziel erreicht werden kann, mit

dem Sprint-Review kombiniert wird. Im Sprint-Review können diese Schritte überprüft und – falls nötig – angepasst werden, oder man kann aus dem Produkt-Backlog einen anderen Stressfaktor auswählen, der dann als nächster angegangen wird.

Während Sie die Roadmap abarbeiten, verwenden Sie die folgenden Fragen, um Ihre Vision, Release-Ziele und Sprint-Ziele zu definieren:

- ✓ **Vision:** Wie soll meine Situation letztlich aussehen?
- ✓ **Release-Ziel:** Welcher kleinere Meilenstein würde mich zu dieser Vision führen?
- ✓ **Sprintziele:** Welche Einträge aus dem Produkt-Backlog wähle ich aus, um zuerst daran zu arbeiten, damit ich *jetzt und sofort* Schritte zur Erreichung dieses Release-Ziels unternehme?

Reiseplanung

Ferien bieten hervorragende Möglichkeiten zur Entspannung, zum Erkunden und zur Teambildung. Auch mit Geschäftsreisen können ähnliche Ziele erreicht werden. Allzu oft jedoch kämpfen Teams, Familien und Freunde mit Frustrationen im Hinblick auf die Finanzierung und damit, Zeit für das Reisen finden. Auch die Meinungen darüber, was zu tun ist und wie man sich entspannt, können auseinandergehen. Auch die Umstände der Reise können sich jederzeit ändern, und zwar sowohl vor als auch während der Reise. Scrum ist ein wichtiges Planungstool, das perfekt für diese Art von Projekten geeignet ist. Denken Sie darüber nach: Sie haben ein genaues Datum und oft ein festes Budget, alles andere ist eine Frage der Prioritätensetzung. Anstatt dass eine Person alle Entscheidungen trifft und hofft, dass alle die getroffenen Entscheidungen mögen, führt die Zusammenarbeit im Team zum Engagement und zur Zufriedenheit von allen. Beispielsweise könnten alle zusammenarbeiten, um einen Konsens über die folgenden Dinge zu erreichen:

- ✓ Eine Vision für die Reise oder den Urlaub erstellen.
- ✓ Das Datum festlegen.
- ✓ Das Budget bestimmen.
- ✓ Anhand der Vorschläge der Stakeholder (zum Beispiel Familienmitglieder und Reisebegleiter) ein Urlaub-Backlog erstellen.
- ✓ Die Einträge im Backlog priorisieren, um das bestmögliche Ergebnis zu erzielen.

Bei Familienreisen kann und sollte jedes Familienmitglied an der Entscheidung beteiligt sein, wie ein idealer Urlaub für alle aussieht. Beachten Sie, dass die Vision für die diesjährige Reise nicht die gleichen idealen Qualitäten wie die Reise des vergangenen oder des folgenden Jahres enthalten muss, wenn der Urlaub zu einer Familiengewohnheit wird. Das ist genau der Punkt. Die Überprüfung des Plans als Familie gibt Aufschluss darüber, welche Schwerpunkte man im Rahmen des Budgets setzt.

Wie bei jedem Scrum-Projekt ist das Urlaubsbudget der Faktor, den Sie zuerst festlegen wollen. Wenn die Familienmitglieder auf Aktivitäten fokussiert sind, die teurer sind, als das Budget es hergibt, kann die Familie andere, günstigere Orte untersuchen. Wenn ein bestimmter Ort für die Familie oberste Priorität hat, kann der Urlaub mit günstigeren Optionen an diesem idealen Ort geplant werden.

Scheuen Sie sich nicht, Kinder in die Diskussion der Prioritäten gemäß der Budgetierung einzubeziehen. Auch wenn sie vielleicht nicht die Verantwortung dafür haben, das Geld für den Urlaub bei der Bank anzusparen oder das Gesamtbudget festzulegen, sollten sie als Team an einer Kompromissentscheidung beteiligt sein. Sie können beispielsweise sagen: »Wir können auf Hawaii schnorcheln oder Seilrutschen, aber nicht beides. Was möchtet ihr gern tun?« Es ist wichtig, den Kindern beizubringen, dass das Setzen finanzieller Prioritäten und das Einhalten eines Budgets

wichtige Fähigkeiten im Familienleben sind. Scrum blüht auf, wenn alle Beteiligten Eigentümer der Entscheidungen sind.



Statt einer einfachen Abstimmung, in der jedes Mitglied nur Ja oder Nein sagt, versuchen Sie es mit Fist of Five (siehe [Kapitel 4](#)), gefolgt vom Dot Voting. Mit der Methode Fist of Five geben Sie Ihre Zustimmung zu einer Idee an. Einen Finger hochzuhalten, bedeutet völligen Widerstand gegen die Idee, während fünf Finger bedeuten, dass es eine großartige Idee ist. Um eine Entscheidung zu treffen, sollte jeder in der Familie mindestens drei Finger oben haben, was bedeutet, dass er den Vorschlag zwar nicht liebt, aber auch nicht hasst und bereit ist, die Entscheidung zu unterstützen. Die Verwendung von Fist of Five erlaubt es, zuerst einen Pool von Optionen einzugrenzen, die niemand in der Familie hasst. Anschließend verwenden Sie das Dot Voting für eine endgültige Entscheidung. Beim Dot Voting haben alle Familienmitglieder jeweils fünf Stimmen. Für jede der Optionen, für die sie stimmen wollen, vergeben Sie Punkte. Sie können alle Punkte auf die gleiche oder jeweils einen Punkt auf fünf verschiedene Optionen setzen und hiermit Ihre Präferenz für bestimmte Entscheidungen deutlich machen.

Einträge aus dem Produkt-Backlog müssen von allen Familienmitgliedern ausgeführt werden. Reservierungen müssen vorgenommen werden. Gleichzeitig müssen fortlaufend Anpassungen vorgenommen werden, die sich aus der Verfügbarkeit oder dem Preis ergeben oder vielleicht aufgrund des erwarteten Wetters. Dieser Ansatz ermöglicht es, dass die wesentlichen Positionen mit höherem Risiko frühzeitig bearbeitet werden können, wodurch gleichzeitig das Risiko abnimmt, je mehr sich der Urlaub nähert.

Da die Termine für den Urlaub näher rücken, werden weniger große Entscheidungen getroffen und bei der Urlaubsplanung stehen Aufgaben im Vordergrund, die vor der Abreise erledigt werden müssen. Behalten Sie den Kalender im Auge und führen Sie kurze Sprints durch, um die verbleibenden Einträge zu

erledigen. Gliedern Sie die Einträge im Produkt-Backlog, damit die Einträge leicht in die Spalte *Fertig* verschoben werden können. Das Einkaufen der passenden Kleidung und das Einpacken für die Reise können nicht in der gleichen Woche erfolgen, ohne beispielsweise Stress und Chaos zu verursachen. Fügen Sie das Einkaufen in einen Sprint ein, der weit vor der Reise liegt, sodass alle Gegenstände verfügbar sind, wenn es ans Einpacken geht. Packen Sie beispielsweise in einen früheren Sprint testweise alles Benötigte ein. Sie können so früh erkennen, was fehlt und die fehlenden Dinge noch einkaufen. Dies ist eine einfache Methode, um am Abend vor Ihrer Abreise unangenehme Überraschungen zu vermeiden.

Urlaubsplanung mit Scrum

Reisen – sowohl geschäftliche als auch private – können unvorhersehbar sein. Einer unserer Trainer plante vor einigen Jahren mit Scrum eine Geburtstagsreise auf Hawaii. Er nutzte einige Last-Minute-Reiseangebote (weniger als zwei Wochen im Voraus) und hatte nicht viel Zeit, die Reise im Detail zu planen. Er und seine Frau waren noch nie auf Hawaii, also fragte er Freunde und Familie nach Ideen, was sie dort tun und wo sie essen sollten. Mit Gepäck, Flugtickets, Hotel- und Autoreservierung und einer Liste von möglichen Aktivitäten (Roadmap), die auf das Budget abgestimmt waren, stiegen sie in das Flugzeug. Die Vision war: einfach entspannen, ohne Druck und ohne sich selbst zu verpflichten, etwas Bestimmtes zu unternehmen oder zu besichtigen.

Während des Flugs konnten sie wieder zu Atem kommen und widmeten sich dann ihrem Backlog mit den Aktivitäten und Restaurants. Sie planten den ersten Tag und schätzten, wie viel des Budgets dies kosten würde. Dann genossen sie den Rest des Fluges.

Zu Beginn des nächsten Tages hakten sie auf ihrer Liste die Dinge ab, die sie getan hatten, und identifizierten weitere

Dinge, die interessant klangen. Eine der Aktivitäten war die Reservierung einer Tour, also planten sie diese für drei Tage.

Jeden Tag wiederholten sie den gleichen Vorgang: die Liste erweitern, abgeschlossene Punkte streichen und die Priorität entsprechend ihren Interessen anpassen. Einige Einträge erreichten nie den Anfang der Liste oder wurden verworfen, da ihr Interesse hieran abnahm. Der Prozess fühlte sich kaum wie eine Planung an.

Sie hatten das Glück, den Urlaub an einem beliebigen Tag beenden zu können (was keine Voraussetzung für den Einsatz von Scrum ist). Nach acht Tagen statt der geplanten zehn Tage hatten sie genug erlebt; ihre Vision für die Reise war Wirklichkeit geworden; sie hatten die geplanten Dinge getan und sie waren in den Restaurants gewesen, in denen sie essen wollten. Zufrieden kehrten sie nach Hause zurück (TK + AK > W).

Schule und Studium

Lernen ist ein notwendiger Teil des Lebens, von der Geburt bis zum Tod. Der Physiker Albert Einstein sagte: »Wenn man aufhört zu lernen, stirbt man.« Die menschliche Neugier treibt uns zu dem Leben, das wir führen wollen. Die Grundschule sowie die Sekundarstufen I und II bieten für Schüler ab dem Alter von 6 Jahren (für einige Schüler sogar ab einem früheren Zeitpunkt, falls sie die Vorschule oder den Kindergarten besuchen) bis zur 12. Klasse strukturierte Lernformate. Das formale Lernen geht oft über die Sekundarstufe hinaus, in Form einer Berufsausbildung oder eines Studiums.

Nicht nur in den Vereinigten Staaten soll das Schulsystem grundlegende Kompetenzen für Kinder vereinheitlichen und ihnen helfen, potenzielle Interessengebiete zu identifizieren. Wie wir in [Kapitel 10](#) untersuchen, kann Scrum Schulen auf der Verwaltungsebene bei organisatorischen Fragen und Lehrern auf

der Klassenebene bei der Unterrichtsgestaltung helfen. Obwohl Schulen und Lehrer daran arbeiten, die Gegebenheiten auf diesen Ebenen zu verbessern, können auch die Schüler selbst Scrum nutzen, um ihre Erfahrungen mit ihrer eigenen Ausbildung zu verbessern.

Schüler und Studenten können die Roadmap to Value nutzen, um ihre Ziele zu erreichen. Sie entwickeln eine Vision, generieren ein Backlog von Elementen aus einer allgemeinen Roadmap, identifizieren ein erstes Release, führen kurze Sprints für dieses Release durch und verwenden ein Sprint-Review, um ihr Produkt (zum Beispiel Hausaufgaben) für den nächsten Sprint zu überprüfen und anzupassen. Sie führen eine Sprint-Retrospektive durch, um ihre Prozesse und Werkzeuge zu überprüfen und anzupassen, damit sie den Fortschritt ihrer Ausbildung optimieren können. Dieser Prozess ermöglicht es den Schülern und Studenten in jedem Schritt, ihren Produkt-Backlog zu verfeinern oder ihre Unterrichtseinheiten nach Bedarf anzupassen.

Frühes Lernen

Wenn Sie ein Elternteil oder ein Lehrer sind, sehen Sie, dass Kinder auf die eine oder andere Art mit dem Lernen Schwierigkeiten haben. Abhängig von ihrem Alter kann es für Kinder mühsam sein, wenn sie etwas Schwieriges lernen sollen, ohne ein Gefühl dafür zu haben, warum dies für ihre Zukunft wichtig sein kann. Wenn sie keinen Zusammenhang zwischen ihrem Lernen und ihren Zielen erkennen, ist es schwer, sich durch die auftretenden Schwierigkeiten zu kämpfen. Auch kleine Kinder können Scrum benutzen. Scrum besteht einfach aus drei Rollen, drei Artefakten und fünf Ereignissen. Es befähigt Kinder und es macht Spaß, das eigene Backlog abzuarbeiten. In diesem Abschnitt geht es um Schülerinnen und Schüler ab dem Kindergarten (ca. 4 Jahre alt) bis zur 8. Klasse (ca. 14 Jahre alt).

Jungen Lernenden zu helfen, eine Vision zu erkennen und diese mit ihrem eigenen Leben in Beziehung zu setzen, ist der Schlüssel, um sie dazu zu bringen, sich zu engagieren. Helfen Sie ihnen, die Dinge, die sie interessieren und die sie gerne tun, mit

dem in Beziehung zu setzen, was sie lernen. Wir haben einen Freund, dessen Sechsjährige morgens aufwacht und alle Zutaten zum Backen eines Kuchens herausholt. Aber dieselbe Tochter ist frustriert, wenn sie Zahlen und einfaches Rechnen lernen soll. Ihre Mutter hat diese Chance genutzt und ihrer Tochter vermittelt, dass auch beim Backen eines Kuchens Zahlen benötigt werden. Nach dieser kleinen Lektion war ihre Tochter viel eher bereit, sich anzustrengen und die Zahlen zu lernen.

Junge Lernende können von der Verwendung von Taskboards profitieren, um ihre unterschiedlichen Hausaufgaben zu bearbeiten, da die Taskboards leicht anpassbar sind, wenn sich die Hausaufgaben ändern und der Schüler älter wird.

Wir sprechen in fast allen Kapiteln, vor allem in den [Kapiteln 6, 13](#) und [15](#), über Thrashing. Multitasking reduziert Qualität und Effektivität und Scrum reduziert diese Fehlerquote, indem es sich auf jeweils ein Projekt konzentriert. Schüler aller Altersgruppen müssen gleichzeitig Inhalte vieler Unterrichtsfächer lernen. Der Schlüssel zur Nutzung von Scrum für Schüler sind kurze Sprints zu bestimmten Themen oder Projekten. Möglicherweise dauern Sprints mehrere Tage oder sogar einen einzigen Tag, damit sich die Schüler auf ein Thema konzentrieren können.

Unabhängig von der Sprintlänge können die Schüler die Pomodoro-Technik (siehe [Kapitel 2](#)) nutzen, um leichter zwischen verschiedenen Aufgaben zu wechseln. Die Pomodoro-Technik schafft einen natürlichen Ruhepunkt und bietet eine einfache Möglichkeit zum Aufgabenwechsel. Mit der Pomodoro-Methode kann der Schüler leicht erkennen, ob die Aufgabe nach einer Pause fortgesetzt werden muss oder ob es Zeit für eine neue Aufgabe ist.

Es ist wichtig, Kindern frühzeitig beizubringen, wie sie überprüfen und anpassen können. Einer unserer Trainer hat eine Tochter in der dritten Klasse. Sie kam nach Hause und zeigte ihrem Vater ihren Rechtschreibtest für diese Woche. Sie war normalerweise perfekt in der Rechtschreibung. Als er sah, dass sie vier Wörter falsch schrieb, war er überrascht. Er räumte zunächst ein, dass

sie immer noch eine relativ gute Punktzahl hatte, fragte aber, warum sie mehr Fehler als sonst gemacht hatte. Nachdem sie versucht hatte, andere Schüler zu beschuldigen, diese hätten sie abgelenkt, gab sie zu, dass sie sich durch ihre Tests gehetzt hatte und nicht so viel geübt hatte, wie sie es jeden Tag zwischen dem Vortest und dem Abschlusstest hätte tun können.

Anstatt seiner Tochter Vorwürfe zu machen, dass sie etwas falsch gemacht habe, fragte er sie, ob sie eine Idee habe, wie sie es beim Test in der nächsten Woche besser machen und wie sie es vermeiden könne, dies einfach schnell zu erledigen. Sie erinnerte sich, dass sie zu Beginn des Jahres jeden Tag der Woche kurze Übungstests gemacht hatte, nicht nur am Anfang und am Abend vor dem Test. Sie waren sich einig, dass es gut wäre, dies noch einmal zu versuchen, und sie setzte ihren neuen Plan in der folgenden Woche um. Sie konnte aus ihren eigenen Erfahrungen lernen und selbst Lösungen zur Verbesserung entwickeln.

Von der Schule ins Studium

Schüler auf der High School, die in etwa der Sekundarstufe II entspricht, haben eine Reihe von Anforderungen, die auf ihre Vision vom Leben nach der High School ausgerichtet sind. Bis sie die High-School-Zulassung erreicht haben, konzentrieren sich die meisten Schüler darauf, gut von einer Jahrgangsstufe zu nächsten zu kommen. Ab der High School beziehungsweise der Sekundarstufe 2 verschiebt sich der Fokus. Die Schüler werden im Kurssystem (Wahlfächer) und nicht mehr im Klassenverband unterrichtet und sie beginnen damit, sich auf ihre eigenen Visionen zu konzentrieren. Die Schüler können die Zügel selbst in die Hand nehmen, selbst die Richtung bestimmen und unterschiedliche Vision Statements entwickeln. Dies kann der Wunsch sein, nach dem Abschluss der High School beziehungsweise Sekundarstufe 2 an einem College oder Universität zu studieren, bis zum Wunsch, sofort nach dem Abschluss zu arbeiten und alle möglichen Varianten, die dazwischen liegen. Der beste Weg, diese Vision zu verwirklichen, ist, sie jedes Jahr als Zwischenschritt zu nutzen.

In diesem Abschnitt konzentrieren wir uns auf das beliebte Ziel, ins College zu kommen und zu studieren. Wenn ein Schüler die Vision hat, ein College zu besuchen, muss ein spezifischer Backlog, der sich an den Anforderungen des Colleges orientiert, aufgestellt werden. In dieser Liste sollten alle nicht-wissenschaftlichen Arbeiten enthalten sein, die durchgeführt werden müssen. Die Suche nach dem geeigneten College oder der besten Universität dauert beispielsweise länger, als ein Schüler es erwartet, und sollte so früh wie möglich durchgeführt werden, damit zukünftige Sprints Aktivitäten beinhalten können, die auf die Erfüllung der Zulassungsbedingungen zugeschnitten sind. Taskboards sind ein effektives Werkzeug, um bei wichtigen Produkt-Backlog-Einträgen voranzukommen.

Die Prioritätensetzung ist für High-School-Schüler entscheidend. Nur wenige Schüler können jedes akademische Fach belegen, jede Sportart ausüben, bei allen Prüfungen perfekt abschneiden und an jeder Aktivität teilnehmen – und sie können diese Dinge definitiv nicht gleichzeitig tun. Durch die richtige Prioritätensetzung kann ein Schüler Elemente weglassen, die seine Vision nicht direkt unterstützen und sich auf Release-Ziele konzentrieren, die sowohl persönlich als auch für den Einstieg in das richtige College wichtig sind. Die Verwendung dieser Release-Ziele bei der Arbeit an seiner Vision hilft ihm, Meilensteine zu setzen. Das Vorhandensein von Release-Zielen erlaubt es ihm, die unterteilten Anforderungen in Sprints zu organisieren.

So könnte diese Aufteilung für einen High-School-Schüler aussehen:

- ✓ **Vision:** Für ein vierjähriges Studium zugelassen werden.
- ✓ **Release-Ziel, das die Vision unterstützt:** Beim Scholastic Aptitude Test (SAT) mehr als 2.200 Punkte erzielen. (A. d. Ü.: SAT ist ein standardisierter Test in den USA, mit dem die Eignung für ein Studium überprüft wird.)
- ✓ **Sprint-Ziel, das das Release-Ziel unterstützt:** Bei fünf praktischen SAT-Tests aus dem Bereich Mathematik mehr als

720 Punkte erzielen.

- ✓ **Anforderungsebene, die den Sprint unterstützt:** Einen SAT-Tutor finden, in dieser Woche drei Stunden das Buch zur Vorbereitung auf SAT studieren, Fokussierung auf die mathematischen Konzepte, den SAT-Tutor nach den Inhalten zum Bereich Geometrie befragen.

Die Erwartungen sind hoch und von den Studierenden wird erwartet, dass sie sich in einer großen Anzahl von Fächern auskennen. Auch der Wettbewerb um den Einzug ins College und den Erhalt von Stipendien ist extrem hoch. Die Schüler finden verschiedene Konstellationen von Lehrern und Klassen vor, sodass die Arbeitsbelastung von Schüler zu Schüler und von Semester zu Semester variiert.

Der Sprint-Zyklus aus Überprüfung und Anpassung bietet den Rahmen, den die Schüler benötigen, um sich von Jahrgangsstufe zu Jahrgangsstufe an unterschiedliche Belastungen und Situationen anzupassen, während sie ihre Roadmap abarbeiten.

Im Jahr 2014 verfasste Alexis Wiggins einen Gast-Blogbeitrag, in dem sie die Herausforderungen, die ihr auffielen, detailliert beschrieb, als sie zuerst einen Schüler der zehnten und dann einen Schüler der zwölften Jahrgangsstufe für einen Tag begleitete. Für sie waren vor allem die ständige Anstrengung, den ganzen Tag zu sitzen, und die Passivität, die vom Lernen erwartet wird, von Bedeutung. Durch die Integration von Scrum in das häusliche Lernen kann ein Schüler auf eine andere Art und Weise Verantwortung für das Lernen übernehmen. Während die Schüler keinen Einfluss auf den Verlauf des Unterrichts in der Schule haben, können sie zu Hause neben dem Sitzen und Auswendiglernen nach alternativen Wegen suchen. Dies ist auch der Ort, an dem das Produkt-Backlog den Fortschritt des Schülers maßgeblich beeinflussen kann. Wenn außerschulische Aktivitäten ein wichtiger Faktor in Ihrer Vision sind, versuchen Sie, eine zu wählen, die praktische Tätigkeiten wie beispielsweise sportliche oder künstlerische Aktivitäten ermöglicht, und damit verschiedene Methoden des Lernens in Richtung Zielfortschritt zu nutzen. Wenn

aus irgendeinem Grund körperliche außerschulische Aktivitäten nicht geeignet oder möglich sind, versuchen Sie, aktive Pausen einzulegen, wenn Sie zum Lernen die Pomodoro-Technik verwenden.

Persönliche Bildungsvision

Die Vision einer Schülerin oder eines Schülers kann sich im Laufe der Schullaufbahn ändern. Nehmen Sie die folgende Vision, die eine Schülerin zu Beginn ihrer High-School-Zeit aufgestellt hat, und beachten Sie, wie sie sich entwickelt, während sie sich auf dem Weg dorthin Überprüfungen und Anpassungen vornimmt. Zu Beginn der neunten Jahrgangsstufe weiß sie, dass sie die High-School mit einem GPA-Notendurchschnitt von mindestens 3,8 Punkten (auf einer Skala von 1 bis 5 entspricht 5,0 dem besten Notendurchschnitt) abschließen will, um sich für den Besuch einer der acht Elite-Hochschulen im Nordosten der USA, der sogenannten Ivy League Colleges, zu qualifizieren.

Vielleicht hat sie nicht hinreichend recherchiert, welche Zulassungsanforderungen diese acht Elite-Hochschulen stellen. Vielleicht weiß sie nicht, dass mehr als nur ein bestimmter Notendurchschnitt erforderlich ist, um zum Studium zugelassen zu werden. Vielleicht weiß sie auch nicht, welche Laufbahn sie nach dem Abschluss der High-School einschlagen will. Obwohl viele Dinge noch ungeklärt sind, ist diese Vision dennoch ein guter Anfang.

Am Ende der neunten Jahrgangsstufe weiß sie, dass ein GPA-Notendurchschnitt von 3,8 wahrscheinlich nicht ausreichen wird. Sie hat gelernt, welche Testergebnisse sie für die Zulassung benötigt werden, welche außerschulischen Aktivitäten erforderlich sind und dass auch ehrenamtliches Engagement bei der Zuteilung eines Studienplatzes berücksichtigt wird. Sie hat an mehreren sozialwissenschaftlichen Kursen teilgenommen, die ihr deutlicher machten, in welche Richtung ihre berufliche Zukunft

gehen soll. Zu Beginn der zehnten Jahrgangsstufe hat sie eine aktualisierte Vision: Sie will sich nach der High School für Yale, Stanford oder Harvard qualifizieren.

Ihre Roadmap und ihr Backlog beinhalten Dinge wie »einen GPA-Notendurchschnitt von mindestens 4,2 Punkten, mindestens 30 Punkte bei meinem ACT (ein weiterer standardisierter Test zur Messung der Studierfähigkeit) und 2.200 Punkte bei meinem SAT, genug Geld sparen, um die Hälfte einer Reise nach Südamerika (Mama und Papa werden die andere Hälfte bezahlen) für humanitäre Arbeit vor dem Ende meines Juniorjahres bezahlen zu können«.

Am Ende eines jeden Jahres wird sich die Vision und die Roadmap dieser Schülerin wahrscheinlich weiterentwickeln, weil sie mehr weiß als zu Beginn. Regelmäßige Überprüfungen und Anpassungen in logischen Abständen ermöglichen es ihr, sich von vertrauenswürdigen Eltern und Beratern beraten zu lassen, um fundierte Entscheidungen zu treffen.

Im Studium weiterkommen

Das College ist die perfekte Gelegenheit für einen Studenten, Scrum zu benutzen. Für jeden Studenten, dessen Vision, die ein guter Job nach dem Studium ist, hilft die Implementierung von Scrum-Praktiken bei der Vorbereitung auf die Geschäftswelt.

Das Curriculum der einzelnen Lehrveranstaltungen wird von den jeweiligen Dozenten festgelegt, jedoch ist der gesamte Lehrinhalt den Studenten selbst überlassen. Zu entscheiden, was man als Hauptfach studieren will, ist für jeden Studenten eine gewaltige Aufgabe, unabhängig davon, ob diese Entscheidung vor oder während des Studiums getroffen werden. Das Konzept von Überprüfung und Anpassung ist für Studenten weiterhin auf der Ebene des Sprints wichtig; das Ziel besteht darin, dass Lernen zu verbessern. Im Studium müssen diese Aufgaben jedoch auf die Release-, wenn nicht sogar auf die Ebene der Vision gebracht werden. Ein Student kann zum Beispiel am Ende eines

Semesters (Release-Ebene) feststellen, dass Biologie kein Fach ist, das er weiterstudieren möchte, und dass ein anderes Fach zu einem vorrangigen Interesse geworden ist. Während sich seine Interessen entwickeln, passt er seine Vision und Roadmap so an, dass sie die Grenzen und die Richtung für jedes Release (Semester) und jeden Sprint (Woche) vorgeben.

Das Studium ist außerdem ein guter Zeitpunkt, um innerhalb eines Scrum-Frameworks Teambildung zu betreiben.

Hochschulgruppenprojekte stellen eine gute Gelegenheit dar, um Scrum für die Planung, Überprüfung, Koordination sowie Erstellung von Teamvereinbarungen und Reaktion auf Veränderungen zu nutzen. Die Durchführung von Scrum-Ereignissen und die Verwendung eines Taskboards geben einen sofortigen Überblick darüber, wer an welchem Teil des Projekts arbeitet und wie das Projekt voranschreitet. Die Studenten sollten die Gruppenmeetings so kurz wie möglich halten, indem sie Daily Scrums verwenden. Wenn die Gruppe ein Hindernis erkennt oder für ein sich verzögerndes Sprint-Backlog-Element die Technik des Swarming einsetzen muss, ermöglicht diese Sichtbarkeit, dass das Projekt schneller fortschreitet.

Darüber hinaus können die Studierenden den erfolgreichen Abschluss von Projekten auf ihren eigenen Lernstil zuschneiden. Projekte im Rahmen des Studiums werden häufig durch eine Definition of Done (die vom Dozenten angegebene Rubrik) definiert, jedoch bleibt deren Erledigung den Studenten überlassen.



Teambildungs- und Paarprogrammiertechniken kommen im Studium zum Einsatz. Die Studenten müssen sich nicht in einem Gruppenprojekt befinden, um gegenseitig die erledigte Arbeit zu überprüfen, bevor sie fertiggestellt ist. Sie können Sprint-Reviews verwenden, um vor der endgültigen Einreichung (das Release an den Kunden, in diesem Fall den Dozenten) Feedback zu erhalten.

Teilmengen der Release- und der Sprint-Ziele ähneln dem, was in der Zeit vor dem Beginn des Studiums getan wurde. Im Umfeld der Hochschule kann zu den Zielen auch der Aufbau von den Beziehungen zu den Professoren gehören, die auf Stipendien hinweisen, bei der Jobvermittlung helfen oder die Suche nach spezifischen Praktikumsplätzen unterstützen können, um so praktische Berufserfahrungen zu erlangen. Die Studenten können eine gute Arbeitsbeziehung zu den Professoren aufbauen, indem sie deren Feedback berücksichtigen und es zur Verbesserung ihrer eigenen Arbeit nutzen.



Die Anzahl der für den Abschluss anrechenbaren Semesterwochenstunden (Credit Hours) kann wie die Velocity (siehe [Kapitel 4](#)) nur einen Teil der Geschichte erzählen. Nur weil ein Student im letzten Semester 12 Semesterwochenstunden abgeleistet hat, bedeutet das nicht, dass auch im nächsten Semester 12 Semesterwochenstunden die richtige Zahl sind. Möglicherweise erfordert eine Lehrveranstaltung im vergangenen Semester, für die drei Credit Hours angerechnet werden, nicht den gleichen Aufwand außerhalb der Lehrveranstaltung wie eine völlig andere Lehrveranstaltung im nächsten Semester, für die ebenfalls drei Credit Hours gutgeschrieben werden. Um im nächsten Semester ein Praktikum absolvieren zu können, kann es erforderlich sein, an weniger Lehrveranstaltungen als im letzten Semester teilzunehmen, was dann zwangsläufig zu weniger Credit Hours führt. Vielleicht kann ein Student im nächsten Semester an mehr Online-Veranstaltungen teilnehmen als im letzten Semester.

Sprintzyklus für einen Studenten

Bei Scrum für Studenten beträgt die Sprintdauer jeweils eine Woche. Die Studenten überprüfen die Roadmap und konzentrieren sich auf das aktuelle Release, indem sie Backlog-Einträge für den Sprint auswählen, die in Aktivitäten

unterteilt sind, die an einem Tag oder kürzer durchgeführt werden können. Wenn für die Lehrveranstaltung B beispielsweise 30 Seiten gelesen werden müssen, kann die Gesamtanzahl auf 10 Seiten pro Tage für drei der sieben Tage des Sprints aufgeteilt werden. Am Ende des ersten Tags weiß der Student, ob 10 Seiten die richtige Menge Betrag für die restlichen Tage sind. Eine weitere Anforderung für den Sprint könnte sein, ein bestimmtes Thema für eine anstehende schriftliche Hausarbeit zu recherchieren.

Einige dieser Aktivitäten müssen möglicherweise zeitlich begrenzt werden. Die Recherche kann zum Beispiel in zwei Stunden oder weniger durchgeführt werden. Aufgrund von Arbeits- und Stundenplänen stehen dem Schüler möglicherweise nur 60 Minuten zum Lesen zur Verfügung. Anhand der zur Verfügung stehenden Zeit und der noch zu erledigenden Arbeit muss der Schüler jeden Tag entscheiden, wie er diese Zeit am besten nutzen kann.

Auf der Roadmap-, Release-, Sprint- und Tagesebene priorisieren und sortieren die Studenten die Arbeiten, die sie erledigen müssen, und passen sie bei Bedarf an.

Am Ende eines jeden Sprints vergleicht der Student die fertiggestellten Arbeiten mit dem Ziel und passt die verbleibenden Produkt-Backlog-Einträge entsprechend an. Außerdem wird der Prozess überprüft und es werden Hindernisse identifiziert, die ihn davon abhalten, seine Ziele zu erreichen. Vielleicht dauert das Lesen zu lange und er recherchiert Möglichkeiten zur Verbesserung der Geschwindigkeit, wie beispielsweise die Teilnahme an einem Schnellesekurs. Vielleicht wird am College ein Workshop zum Thema Notizen erstellen angeboten oder es wird ein Tutor für ein bestimmtes Fach benötigt. Diese Überprüfung der Ziele und Prozesse zur Erreichung dieser Ziele findet jede Woche statt und kleinere Korrekturen und Anpassungen können bei geringem Arbeitsaufwand vorgenommen werden – dies geht sehr schnell, im Gegensatz zu dem typischen Pauken, das

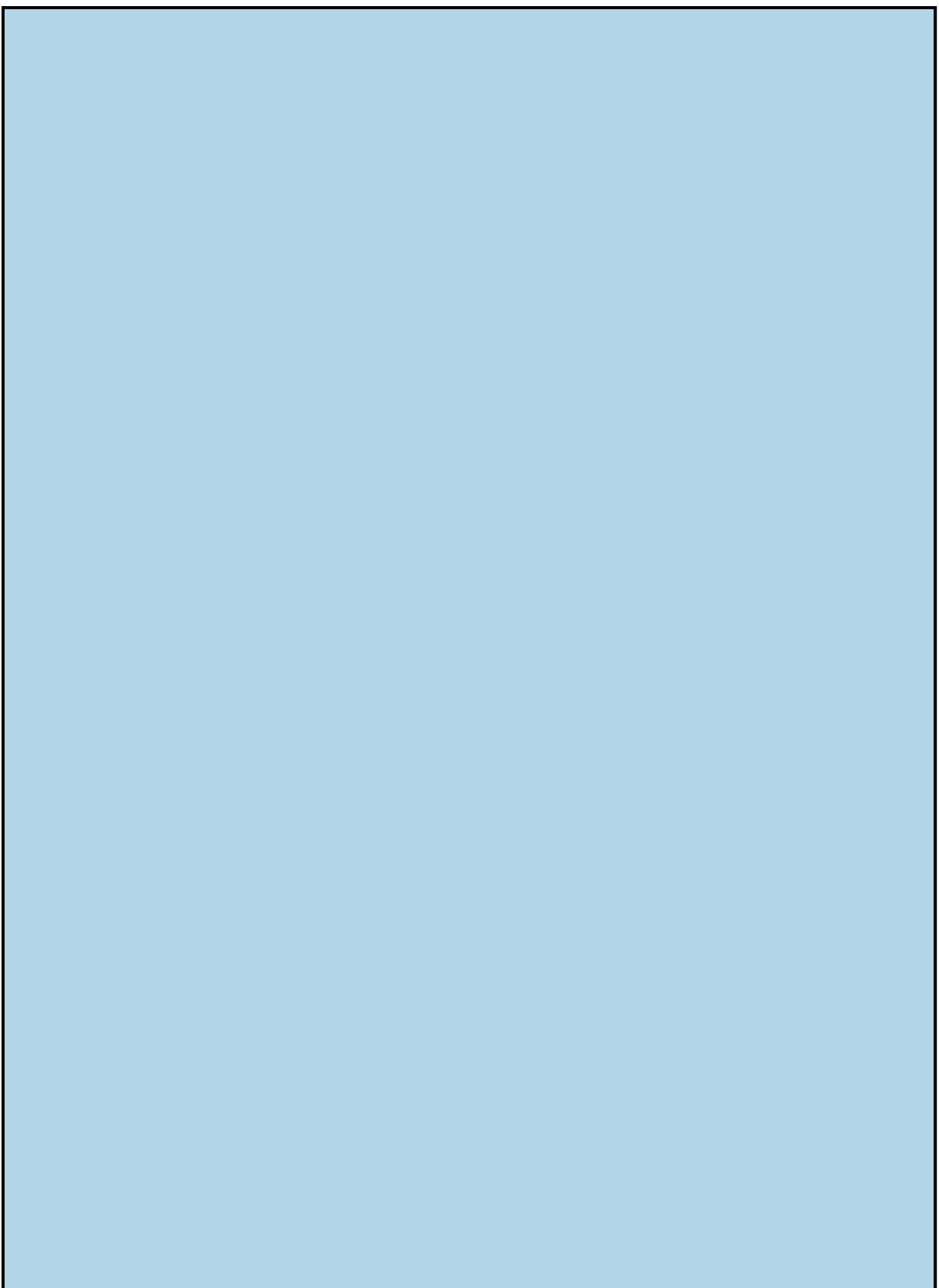
erforderlich ist, um die durch ständiges Prokrastinieren entstandenen Wissenslücken zu füllen.

Teil VI

Der Top-Ten-Teil

Der
Top-Ten-
Teil





IN DIESEM TEIL ...

Steigen Sie erfolgreich auf Scrum um.

Vermeiden Sie Stolperfallen in Scrum.

Erfreuen Sie sich an den Vorteilen von Scrum.

Messen Sie die richtigen Dinge.

Beteiligen Sie sich an der Scrum-Community.

Kapitel 19

Zehn Schritte für die Einführung von Scrum

IN DIESEM KAPITEL

Nutzen aus erfolgreichen Implementierungen ziehen
Für den Erfolg planen

In diesem Buch betonen wir, dass sich Scrum sehr von traditionellem Projektmanagement unterscheidet. Eine Organisation vom Wasserfall zu Scrum zu bringen, ist eine wichtige Veränderung. Aufgrund unserer Erfahrung bei unterschiedlichen Arten von Unternehmen, die wir bei diesen Veränderungen begleiteten, haben wir die folgenden wichtigen Schritte identifiziert, um eine agilere Organisation zu werden und Scrum erfolgreich umzusetzen.

Durchführung eines Audits

Bevor Sie planen, wohin Sie gehen wollen, müssen Sie wissen, wo Sie gerade stehen. Ein Third-Party-Audit Ihrer aktuellen Prozesse, Methoden, Praktiken und Strukturen sagt Ihnen nicht nur einige Dinge, die Sie nicht darüber wissen, wie Sie derzeit arbeiten, sondern bestätigt auch einige Dinge, die Sie bereits wissen. Außerdem erhalten Sie zusätzlichen Einsichten, die Sie vielleicht nicht berücksichtigt haben.

Eine *Implementierungsstrategie* ist das Ergebnis eines Audits und eines Plans, der skizziert, wie Ihr Unternehmen von dort, wo es heute steht, zu Scrum übergehen wird. Eine gründliche

Umsetzung von Scrum beinhaltet Analysen und Empfehlungen für:

- ✓ **Aktuelle Prozesse:** Beispiele dafür sind, wie Ihr Unternehmen heute Projekte durchführt, was es gut macht und was seine Herausforderungen sind.
- ✓ **Zukünftige Prozesse:** Beispiele sind, wie Ihr Unternehmen von Scrum profitieren kann, welche zusätzlichen agilen Praktiken Scrum in Ihrer Organisation verbessern können, welche wichtigen Änderungen Ihre Organisation vornehmen muss, welche organisatorischen und personellen Auswirkungen diese Herausforderungen haben und wie Ihr transformiertes Unternehmen aus Team- und Prozessperspektive aussehen wird.
- ✓ **Schritt-für-Schritt-Plan:** Beispiele sind, wie Sie von bestehenden Prozessen zu Scrum wechseln, was sich sofort und was sich in sechs Monaten oder einem Jahr oder länger ändern wird. Dieser Plan sollte eine Roadmap von Schritten sein, die das Unternehmen zu einem nachhaltigen Zustand der Scrum-Reife führen.
- ✓ **Vorteile:** Beispiele sind die Vorteile, die Scrum für die Menschen und Gruppen in Ihrer Organisation und für die Organisation als Ganzes bietet. Das Wissen um die Vorteile wird eine wertvolle Ressource sein, wenn Sie beim Senior Management und in Ihrer Organisation für den Übergang werben.
- ✓ **Mögliche Herausforderungen:** Beispiele sind die schwierigsten Veränderungen, die Abteilungen oder Personen, die die größten Probleme mit Scrum haben werden, deren Machtbereich gestört wird, mögliche Barrieren und wie Sie diese Herausforderungen meistern werden.
- ✓ **Erfolgsfaktoren:** Beispiele sind organisatorische Faktoren, die Ihnen bei der Umstellung auf Scrum helfen, wie sich das Unternehmen zu einem neuen Ansatz bekennt und welche Personen oder Abteilungen die Vorkämpfer für Scrum sind.

Ihre Umsetzungsstrategie dient als Fahrplan für eine erfolgreiche Transformation.

Talente identifizieren und rekrutieren

Auch wenn die eindeutigen Scrum-Rollen immer mehr zum Gemeingut werden, wissen nicht alle Recruiter, wie man Leute findet, die sie besetzen können. Menschen, die wissen, was sie tun, müssen mit Personalvermittlern zusammenarbeiten, damit diese das nötige Scrum-Wissen erlangen, um die Fabelhaften von den Fakern unterscheiden zu können.

Funktionsübergreifende Entwickler lernen weiterhin neue Fähigkeiten und erkennen die Notwendigkeit, ihre Fähigkeiten zu erweitern, anstatt ihr Wissen auf einen einzigen Bereich zu beschränken. Dieses Talent ist vorhanden. Damit Ihr Übergang zum Scrum effektiv ist, müssen Sie die richtigen Leute auf den richtigen Plätzen haben.

Sie sollten nach Personen suchen, die bereits über Zertifizierungen und Schulungen wie Certified Scrum Master (CSM), Certified Scrum Product Owner (CSPO) und Certified Scrum Developer (CSD) verfügen. Wenn Sie einen Scrum Coach suchen, suchen Sie einen Certified Scrum Professional (CSP-SM, CSP-PO) oder einen Certified Scrum Coach (CEC oder CSC). Die Zertifizierungen entstammen der Scrum Alliance (www.scrumalliance.org), es gibt jedoch auch andere Organisationen, die Zertifizierungen anbieten. Weitere Informationen zu Zertifizierungen finden Sie in [Kapitel 23](#).

Sicherstellung der richtigen Ausbildung

Ausbildung ist ein entscheidender Schritt bei der Einführung von Scrum. Die Kombination aus persönlicher Kommunikation mit einem Scrum-Experten und dem Durcharbeiten von Übungen mit Scrum ist der beste Weg, um Menschen dabei zu helfen, die für ein erfolgreiches Scrum-Projekt erforderlichen Fähigkeiten aufzunehmen und zu verinnerlichen.

Aus- und Weiterbildung funktionieren am besten, wenn die Mitglieder des Projektteams gemeinsam üben und lernen können. Durch die Ausbildung entwickelt sich ein gemeinsames Vokabular, das während des Übergangs verwendet werden kann. Als Scrum-Trainer und Mentoren haben wir Gespräche zwischen Projektteammitgliedern gehört, die mit »Kannst du dich erinnern, dass Mark uns zeigte, wie ...? « begannen. »Das hat funktioniert, als wir es in der Schulung gemacht haben. Lass uns das versuchen und sehen, was passiert.« Wenn das Entwicklungsteam, der Product Owner, der Scrum Master und die Projektbeteiligten am selben Kurs teilnehmen können, können sie die Lektionen auf ihre Arbeit als Team anwenden.

Auch Stakeholder profitieren von der Ausbildung. In [Kapitel 13](#) sehen Sie, wie wichtig Entscheidungen des oberen Managements für den Erfolg von Scrum sind. Einige unserer Kunden haben in diesem Training für ihre Scrum Master, Product Owners und Entwickler so viel Wert gesehen, dass sie ihr gesamtes Führungsteam und ihre Kreativen, Vertriebs-, Marketing-, Finanz- und HR-Mitarbeiter zum gleichen Training schickten, damit das gesamte Unternehmen auf der selben Wellenlänge ist und die gleiche Sprache sprechen kann.

Mobilisierung eines Übergangsteams

Identifizieren Sie ein Übergangsteam von Entscheidungsträgern in Ihrem Unternehmen, das für die Scrum-Transformation auf Organisationsebene verantwortlich sein kann. Dieses Team besteht aus Führungskräften, die die Fähigkeit und die

Einflussmöglichkeiten haben, Prozesse systematisch zu verbessern und Anforderungen und Leistungsmessungen unternehmensweit zu berichten.

Das Übergangsteam schafft Veränderungen, indem es Scrum verwendet, um Übergangsbemühungen durchzuführen, so wie das Entwicklungsteam Produktfunktionen innerhalb von Sprints erstellt. Die Umsetzungsstrategie des Audits ist die Roadmap des Transition-Teams, die aufzeigt, was für eine erfolgreiche Transformation getan werden muss. Das Transformationsteam konzentriert sich in jedem Sprint auf die Änderungen mit der höchsten Priorität und demonstriert seine Umsetzung (wenn möglich) während eines Sprints.

Wenn Sie beginnen, ist der Übergang ein guter Zeitpunkt, um einen erfahrenen Scrum-Coach oder Mentor zu engagieren, der Ihnen durch diese ersten unangenehmen Schritte hilft. Ein guter Coach kann sowohl mit den Pilotteams als auch mit dem Übergangsteam als Nachfolger des Gruppentrainings arbeiten.

[Abbildung 19.1](#) zeigt, wie die Abläufe des agilen Transition-Teams und die Abläufe des Scrum-Teams für das Pilotprojekt (siehe nächster Abschnitt) aneinander ausgerichtet sind. Hindernisse, die in der Sprint-Retrospektive des Pilotteams erkannt werden, werden zu Backlog-Einträgen für das Transition-Team; diese müssen gelöst werden, damit der Prozess des Pilotteams verbessert wird.

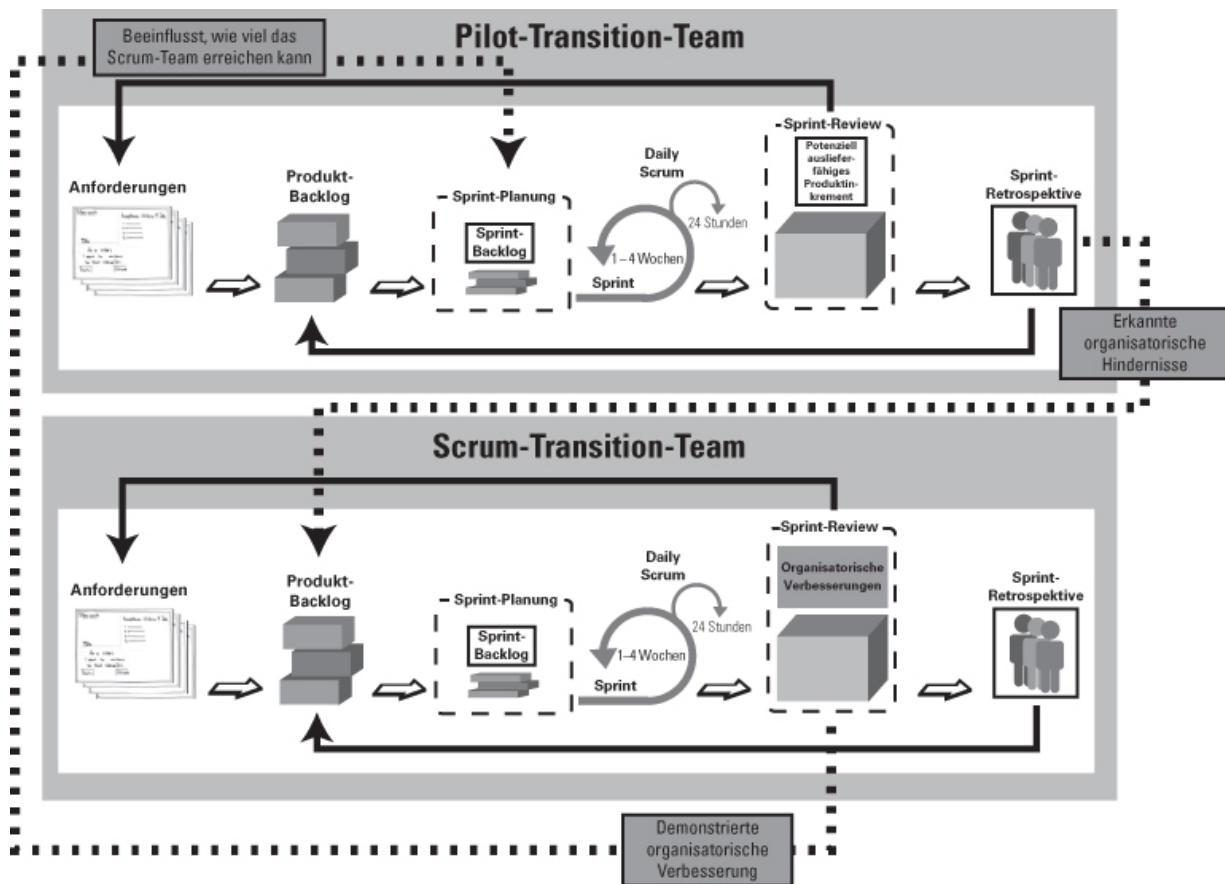


Abbildung 19.1: Aufeinander abgestimmte Abläufe des agilen Transition-Teams und des Scrum-Teams für das Pilotprojekt

Pilotprojekt identifizieren

Ihre agile Transformation mit genau einem Pilotprojekt zu starten, ist eine hervorragende Idee. Indem Sie anfangs nur ein Projekt haben, können Sie herausfinden, wie Sie agile Methoden einsetzen, und verursachen nur geringe Störungen der weiteren geschäftlichen Abläufe Ihrer Organisation. Indem Sie sich auf ein Projekt konzentrieren, können Sie einige der Macken beheben, die unabdingbar auf Veränderungen folgen.

Wenn Sie Ihr erstes agiles Projekt auswählen, suchen Sie nach einem Kandidaten mit diesen Merkmalen:

- ✓ **Angemessen wichtig:** Stellen Sie sicher, dass das Projekt, das Sie auswählen, wichtig genug ist, damit es Interesse in

Ihrer Firma findet. Vermeiden Sie jedoch, das allerwichtigste Projekt zu wählen; Sie sollten Freiraum haben, um Fehler zu machen und aus ihnen zu lernen.

- ✓ **Ausreichend sichtbar:** Ihr Pilotprojekt sollte für die Meinungsführer Ihrer Organisation sichtbar sein, aber machen Sie es nicht zu Ihrem prominentesten Element auf der Tagesordnung. Sie benötigen die Freiheit, sich an neue Prozesse anzupassen; kritische Projekte bieten diese Freiheit möglicherweise nicht, wenn Sie den ersten Versuch machen, einen neuen Ansatz anzuwenden.
- ✓ **Eindeutig und machbar:** Halten Sie nach einem Produkt mit klaren Anforderungen Ausschau und einer Business-Gruppe, die sich verpflichtet, diese Anforderungen zu definieren und zu priorisieren. Versuchen Sie, ein Projekt auszuwählen, dass ein klares Ende hat, und keines, das unendlich ausgedehnt werden kann.
- ✓ **Nicht zu groß:** Wählen Sie ein Projekt aus, dass Sie mit maximal zwei Scrum-Teams, die gleichzeitig arbeiten, fertigstellen können, um zu vermeiden, dass zu viele Teile gleichzeitig umgestellt werden.
- ✓ **Nachvollziehbar messbar:** Wählen Sie ein Projekt aus, von dem Sie wissen, dass es innerhalb der Sprints messbaren Werte produziert.



Menschen benötigen Zeit, um sich organisatorischen Veränderungen aller Art anzupassen, nicht nur agilen Transformationen. Studien haben ergeben, dass Firmen und Teams bei umfangreichen Änderungen zuerst einen Abfall der Leistung sehen, bevor Verbesserungen sichtbar werden. Das Modell für Veränderungen nach Satir, das Sie in [Abbildung 19.2](#) sehen, illustriert den Prozess der Begeisterung des Teams, das eintretende Chaos und schließlich die Anpassung an neue Prozesse.

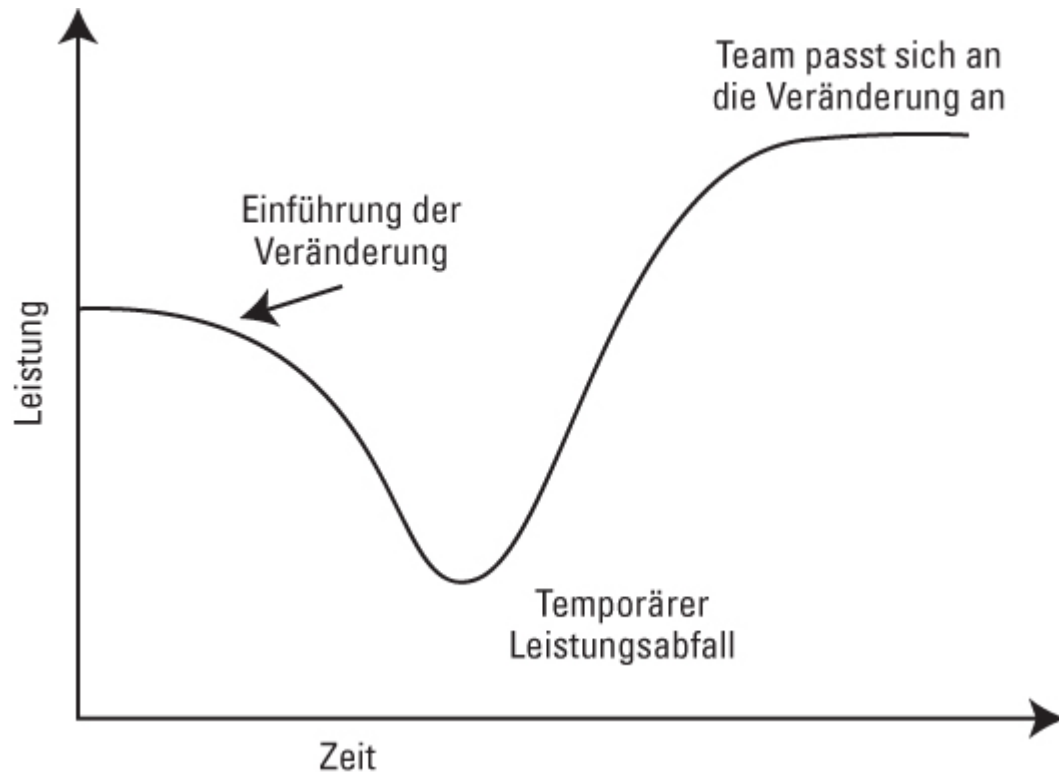


Abbildung 19.2: Modell für Veränderungen nach Satir

Mit der erfolgreichen Durchführung Ihres ersten agilen Projekts haben Sie die Grundlagen für zukünftige Erfolge geschaffen.

Maximieren der Effizienz des Umfelds

Nachdem Sie ein Pilotprojekt identifiziert haben, stellen Sie sicher, dass das Arbeitsumfeld für den Erfolg geeignet ist. Die Umsetzungsstrategie aus dem Audit wird wahrscheinlich die zu behandelnden Punkte aufzeigen, die Folgendes beinhalten können:

- ✓ **Kollokation:** Wenn Sie ein neues Team zusammenstellen, stellen Sie eines zusammen, das zusammen mit dem Product Owner und dem Scrum Master am gleichen Ort arbeiten kann ist. Auch wenn Ihr Unternehmen typischerweise mit verteilten Teams arbeitet, ist dieses Projekt eine Gelegenheit,

zurückzutreten und zu beurteilen, ob verteilte Teams optimal sind.

- ✓ **Arbeitsbereiche:** Stellen Sie sicher, dass das Pilotteam über einen Arbeitsbereich, einen Raum verfügt, der die Zusammenarbeit und die persönliche Kommunikation fördert. Teammitglieder brauchen einen Bereich, in dem sie sich nicht nur ungestört auf ihre Aufgaben konzentrieren können, sondern auch die Zusammenarbeit und das Swarming bequem und effektiv gestalten können. Arbeitsnischen mit Trennwänden sind möglicherweise nicht ideal. Große Tische mit gemeinsamen Monitoren und anderen gemeinsamen Arbeitsbereichen können besser sein. Whiteboards und zahlreiche Tools wie Haftnotizen, Karteikarten, Stifte und Marker erleichtern die Zusammenarbeit.
- ✓ **Technologie:** Für eine bessere Zusammenarbeit mit allen Stakeholdern oder anderen Projektmitarbeitern, die nicht am gleichen Ort mit dem Scrum-Team arbeiten, minimiert die Technologie den Produktivitätsverlust. Videokonferenzen, Aufzeichnungen, Bildschirmfreigaben und virtuelle Whiteboards sind Schlüsselfaktoren für die Unterstützung eines Pilot-Scrum-Teams bei der Entwicklung eines Qualitätsprodukts mit der für den Erfolg erforderlichen Effizienz. Erwägen Sie Investitionen in die Technologie, die den Anforderungen Ihres Teams entspricht, wie beispielsweise Monitore, Kameras, Software für virtuelle Besprechungsräume, Mikrofone und Headsets.

Single Points of Failure reduzieren

Ihr Pilot-Scrum-Team muss so funktionsübergreifend wie möglich sein, um erfolgreich zu sein. Identifizieren Sie Kompetenzlücken bei jedem Teammitglied, wenn für eine bestimmte Fertigkeit ein Single Point of Failure existiert. Machen Sie sofort Pläne für das Ausbildung, damit jeder im Team mehr als eine Sache tun kann

und nicht nur ein Teammitglied über die erforderlichen Fähigkeiten verfügt.

Das Training kann ein formales Training sein oder Paarbildung sowie Shadowing mit einem anderen Entwickler für einen bestimmten Zeitraum beinhalten. Das Training kann während des ersten Sprints durchgeführt werden und muss den Start des Pilotteams nicht verzögern (siehe Abschnitt »Mit dem Pilotprojekt loslegen«).



Sie werden nicht ab dem ersten Tag ein komplett funktionsübergreifendes Team haben. Aber das Ziel sollte es sein, am ersten Tag anzufangen und dabei zu überprüfen und sich anzupassen. Machen Sie die crossfunktionale Weiterbildung und Paarprogrammierung zu einer Priorität und widerstehen Sie dem Druck, auf Kosten eines funktionsübergreifenden Teams willkürliche Termine einzuhalten.

Definition of Done festlegen

Kein Zweifel, Sie werden begeistert sein, wenn das Projektteam sofort loslegt. Aber bevor Sie kopfüber ins Wasser springen, stellen Sie sicher, dass das Team die Definition of Done definiert hat. Anfangs wird die Definition nicht perfekt sein. Das Team wird sie zunächst oft überarbeiten, bis es herausfindet, was Sinn macht. Es wichtig, bei jeder Sprint-Retrospektive die Definition of Done zum Thema zu machen.

Stellen Sie sicher, dass sich das Team darüber im Klaren ist, was für jede Anforderung auf der Sprint- und Release-Ebene getan wurde. Stellen Sie sicher, dass die Definition of Done immer sichtbar ist und von allen Teammitgliedern für jede Anforderung herangezogen wird. So lassen sich viele Probleme vermeiden. In [Kapitel 4](#) finden Sie Details zur Erstellung einer Definition of Done.

Mit dem Pilotprojekt loslegen

Wenn Sie Ihr Pilotprojekt ausgewählt haben, geraten Sie nicht in die Falle, einen Plan aus einer alten Methodik zu verwenden. Verwenden Sie stattdessen Scrum und andere gängige Praktiken aus der Roadmap to Value, die wir in [Kapitel 1](#) erörtern. Diese Praktiken umfassen:

- ✓ Projektplanung
- ✓ Sich auf eine gemeinsame Vision verständigen
- ✓ Erstellung einer Roadmap mit Schätzungen



Obwohl Sie über die Produkt-Roadmap eine langfristige Sicht auf das Produkt haben, müssen Sie das Produkt oder den Projektumfang nicht im Voraus definieren, um loszulegen. Machen Sie sich keine Sorgen, dass Sie zu Beginn Ihres Projekts umfassende Anforderungen stellen müssen; fügen Sie einfach die Funktionen hinzu, die das Projektteam derzeit kennt. Sie können jederzeit nachträglich Anforderungen hinzufügen.

- ✓ Release-Planung
- ✓ Identifizierung der wichtigsten Releases für das minimal funktionsfähige Produkt (MVP)
- ✓ Planung des ersten Releases
- ✓ Sprint-Planung: Den ersten Sprint planen und an die Arbeit gehen



In [Kapitel 7](#) beschreiben wir, warum man Unsicherheit nicht wegplanen kann. Werden Sie kein Opfer der Paralyse durch Analyse; legen Sie eine Richtung fest und gehen Sie!

Während des ersten Sprints sollten Sie sich bewusst an Scrum-Events halten. Die folgenden Dinge sollten während des ersten

Sprints passieren:

- ✓ Führen Sie das Daily Scrum jeden Tag durch, auch wenn Sie das Gefühl haben, dass keine Fortschritte gemacht wurden. Identifizieren Sie Hindernisse und kommunizieren Sie diese umgehend an das Transition-Team.
- ✓ Das Entwicklungsteam muss daran denken, sich selbst als ein Team von Gleichgesinnten zu verwalten und nicht auf den Product Owner, den Scrum Master oder irgendwohin anders als auf den Sprint-Backlog zu schauen, wenn es um die Zuweisung von Aufgaben geht.
- ✓ Der Scrum Master muss das Entwicklungsteam vor externer Arbeit und Ablenkung schützen, besonders während sich andere Mitglieder der Organisation daran gewöhnen, ein eigenes Scrum-Team zu haben.
- ✓ Der Product Owner muss sich daran gewöhnen, direkt mit dem Entwicklungsteam zusammenzuarbeiten, für Fragen zur Verfügung zu stehen und fertiggestellte Anforderungen sofort zu überprüfen und zu akzeptieren.

Im ersten Sprint wird der Weg etwas steinig und holprig sein. Das ist in Ordnung, bei Empirie geht es um Lernen und Anpassung.

In [Kapitel 5](#) finden Sie weitere Details über die Planung von Releases und Sprints.

Überprüfen, anpassen, sich entwickeln und skalieren

In diesem Schritt geht es darum, das Gelernte zu neuen Höhen zu bringen. Wenn Sie die Schritte in den vorhergehenden Abschnitten befolgen, beginnen Sie mit einem soliden Fundament. Wie in [Kapitel 7](#) skizziert, wird Empirie Ihre weitere Anpassung von Scrum und Transformation zu Scrum vorantreiben.

Überprüfen und anpassen, Sprint 1

Am Ende Ihres ersten Sprints sammeln Sie Feedback und führen mit zwei wichtigen Meetings Verbesserungen durch: dem Sprint-Review und der Sprint-Retrospektive (siehe [Kapitel 6](#)).



Das Sprint-Review und die Sprint-Retrospektive sind wichtige Ereignisse. Der Product Owner sollte sicherstellen, dass die Mitglieder des Entwicklungsteams sich darüber im Klaren sind, was es bedeutet, den Stakeholdern funktionierende Software vorzuführen, und er oder sie sollte Fragen vorbereiten, um den Stakeholdern Feedback zu geben und ihnen den Wert der Teilnahme an dieser Review zu zeigen. Um die Teilnahme der Stakeholder sicherzustellen, können mehrere Einladungen und Erinnerungen nötig sein.



Der Scrum Master sollte Ressourcen wie *Agile Retrospektiven: Übungen und Praktiken, die die Motivation und Produktivität von Teams deutlich steigern* von Esther Derby und Diana Larsen (Vahlen Bookshelf) studieren, um sicherzustellen, dass er die richtige Atmosphäre schafft und eine Retrospektive ermöglicht, die das Dampfablassen und erneute Wiederkäuen verhindert, aber Erfolge feiert und Aktionspläne für Verbesserungen identifiziert.

Sich weiterentwickeln und Verbesserungen stabilisieren

Überprüfung und Anpassung erlaubt es Scrum-Teams, sich als Team weiterzuentwickeln und mit jedem Sprint zu reifen.

Agile Praktiker vergleichen den Prozess des Reifens manchmal mit *Shu-Ha-Ri*, einem Konzept für das Erlernen von Kampfsportarten. Dieser japanische Begriff lässt sich mit »erst lernen, dann loslösen und endlich übertreffen« übersetzen.

Dieses Modell beschreibt drei Phasen, in denen Menschen neue Fähigkeiten lernen:

- ✓ **In der *Shu*-Phase** üben die Schüler die neue Fähigkeit so aus, wie sie sie gelernt haben, ohne Hingabe. Es geht darum, die Fähigkeit auswendig zu lernen, damit sie zu einem Automatismus wird.

Neue Scrum-Teams können davon profitieren, es zur Gewohnheit werden zu lassen, agile Prozesse genau zu befolgen, bis sie mit den Prozessen vertraut sind. Während der Shu-Phase können Scrum-Teams eng mit einem agilen Coach oder Mentor zusammenarbeiten, damit sie die Prozesse korrekt durchführen.

- ✓ **In der *Ha*-Phase** beginnen die Schüler zu improvisieren, da sie mehr darüber wissen, wie ihre neue Fähigkeit funktioniert. Manchmal funktionieren die Improvisationen und manchmal nicht. Anhand der Erfolge und Misserfolge lernen die Schüler mehr über die Fähigkeit.

Wenn Scrum-Teams besser verstehen, wie agile Ansätze funktionieren, können sie für ihr eigenes Projekt Variationen ausprobieren. Während der Ha-Phase entdecken die Scrum-Teams, dass die Sprint-Retrospektive ein wertvolles Werkzeug ist, um darüber zu sprechen, ob die Improvisation funktionierte oder nicht. In dieser Phase können Scrum-Teams noch immer von einem agilen Mentor lernen, jedoch lernen sie auch voneinander, von anderen agilen Profis und davon, anderen agile Fähigkeiten beizubringen.

- ✓ **In der *Ri*-Phase** wendet der ehemalige Schüler die Fähigkeit natürlich an; er weiß, was geht und was nicht. Der ehemalige Schüler kann nun voller Vertrauen innovativ werden.

Mit Übung kommen die Scrum-Teams zu dem Punkt, an dem agile Prozesse einfach und vertraut sind, wie das Rad- oder Autofahren. In der Ri-Phase können Scrum-Teams die Prozesse anpassen; sie wissen, was im Sinne des Agilen Manifests und der 12 Agilen Prinzipien funktioniert.

Viral skalieren

Das erfolgreiche Beenden eines Projekts ist ein wichtiger Schritt, um eine Organisation auf Scrum umzustellen. Mit Kennzahlen, die den Erfolg Ihres Projekts und den Wert agiler Methoden nachweisen, können Sie Ihre Firma dazu bringen, neue Scrum-Projekte zu unterstützen.

Beginnen Sie folgendermaßen, um Scrum schrittweise in der gesamten Organisation zu skalieren:

- ✓ **Bilden Sie neue Teams.** Ein Scrum-Team, das eine gewisse Reife erreicht hat – die Leute, die im ersten Scrum-Projekt gearbeitet haben –, sollte nun die Expertise und den Enthusiasmus besitzen, in der Organisation zu Scrum-Botschaftern zu werden. Diese Leute können sich neuen Scrum-Projektteams anschließen und diesen Teams helfen, zu lernen und sich zu entwickeln.
- ✓ **Definieren Sie Metriken neu.** Identifizieren Sie in der gesamten Organisation mit jedem neuen Scrum-Team und jedem neuen Projekt Messgrößen für den Erfolg.
- ✓ **Skalieren Sie systematisch.** Es kann faszinierend sein, hervorragende Ergebnisse zu erzielen, jedoch erfordern firmenweite Verbesserungen signifikante Prozessveränderungen. Stellen Sie nicht schneller um, als es die Organisation verkraften kann.
- ✓ **Erkennen Sie neue Herausforderungen.** Möglicherweise haben sich bei Ihrem ersten Scrum-Projekt Hindernisse aufgetan, an die Sie beim Erstellen des ursprünglichen Implementierungsplans nicht gedacht haben. Aktualisieren Sie Ihre Strategie und den Entwicklungsplan, falls nötig.
- ✓ **Bilden Sie sich weiter.** Stellen Sie bei der Einführung neuer Prozesse sicher, dass die neuen Teammitglieder gut geschult sind, dass ein Mentor zur Verfügung steht und dass sie die Ressourcen besitzen, um effektiv Scrum-Projekte durchzuführen.

Um Ihre langfristigen Bemühungen zur Verbesserung und Reifung mit Scrum zu unterstützen, sollten Sie sich mit einem erfahrenen Scrum-Coach zusammensetzen, um die Leitungsebene und die Teams bei der Durchführung der empfohlenen Änderungen aus Ihrem Audit zu unterstützen. Beginnen Sie außerdem die Suche nach CSP-Talenten – Certified Scrum Professional (CSP auch CSP-SM und CSP-PO), Certified Scrum Coach (CTC und CEC) oder Certified Scrum Trainer (CST) – um die langfristige Transformation zu unterstützen. Etablieren Sie diese interne Rolle als Quelle für die Klärung, die laufende Schulung und die Entwicklung. Betten Sie dieses Talent ein, damit es jeweils einzelne Teams oder einzelne Mitarbeiter effektiv begleiten kann.

Kapitel 20

Zehn Klippen, die Sie umschiffen sollten

IN DIESEM KAPITEL

Häufig gemachte Fehler vermeiden
Solide Praktiken implementieren

Scrum-Teams können bei der Implementierung von Scrum häufige, aber schwerwiegende Fehler machen. Dieses Kapitel gibt einen Überblick über einige typische Probleme und Möglichkeiten für Scrum-Teams, diese zu vermeiden.



Wie Sie vielleicht feststellen, sind viele dieser Fallstricke auf mangelnde organisatorische Unterstützung, mangelnde Schulung und den Rückfall in alte Projektmanagement-Praktiken zurückzuführen. Wenn Ihr Unternehmen in externe Berater investiert, die bei der Transition zu Scrum helfen sollen, wenn es positive Veränderungen unterstützt, wenn das Projektteam geschult ist und wenn Scrum-Teams sich aktiv für die Einhaltung des Scrum-Frameworks und der agilen Werte einsetzen, haben Sie einen erfolgreichen Übergang zu agilen Verfahren.

Unechtes Scrum

Manchmal sagen Organisationen, dass sie »Scrum machen«. Sie mögen einige der Scrum-Events durchlaufen, aber sie haben sich nicht den Prinzipien der Agilität verschrieben und erstellen letztendlich Wasserfallergebnisse und Produkte unter falschen

Scrum-Labeln. Was diese Organisationen tun, wird manchmal gefaktes Scrum oder unechtes Scrum genannt und ist ein sicherer Weg, die Vorteile von Scrum zu verpassen.

Der Versuch, Scrum zusätzlich zu Wasserfallprozessen, Dokumenten und Meetings einzusetzen, ist ein weiterer Ansatz für unechtes Scrum.

Doppelarbeit agil führt zu einem schnellen Burnout des Projektteams. Wenn Sie die doppelte Arbeit machen, machen Sie kein Scrum und halten sich nicht an die agilen Prinzipien.

Lösung: Bestehen Sie darauf, Scrum zu folgen. Holen Sie sich Unterstützung durch das Management, um die Anwendung nicht agiler Prinzipien und Praktiken zu vermeiden.

Mangelnde Ausbildung

Die Investition in eine praktische Schulung bietet eine schnellere und bessere Lernumgebung als das beste Buch, Blog oder Whitepaper. Fehlende Ausbildung deutet oft auf mangelndes organisatorisches Engagement für Scrum hin.



Training und Fortbildung kann Scrum-Teams helfen, viele der Fehler in diesem Kapitel zu vermeiden.

Lösung: Bauen Sie Schulungen in Ihre Implementierungsstrategie ein. Den Teams das richtige Fundament an Fähigkeiten zu geben, ist entscheidend für den Erfolg und zu Beginn der Transition zu Scrum notwendig.

Ineffektiver Product Owner

Keine Scrum-Rolle unterscheidet sich mehr von traditionellen Rollen als die des Product Owners. Scrum-Teams brauchen einen Product Owner, der ein Experte für Geschäftsanforderungen und Prioritäten ist und gut mit dem Rest des Scrum-Teams

zusammenarbeiten kann. Ein abwesender oder unentschlossener Product Owner versenkt schnell ein Scrum-Projekt.

Lösung: Beginnen Sie das Projekt mit einer Person, die die Zeit, das Fachwissen und den Charakter hat, ein guter Product Owner zu sein. Vergewissern Sie sich, dass der Product Owner über eine ordnungsgemäße Schulung verfügt. Der Scrum Master kann helfen, den Product Owner zu coachen, und kann versuchen, Blockaden zu beseitigen, die den Product Owner daran hindern, effektiv zu sein. Wenn die Beseitigung von Hindernissen nicht funktioniert, sollte das Scrum-Team darauf bestehen, den unwirksamen Product Owner durch einen zu ersetzen, der Produktentscheidungen treffen und dem Scrum-Team helfen kann, erfolgreich zu sein.

Fehlende automatisierte Tests

Ohne automatisierte Tests kann es unmöglich sein, die Arbeit innerhalb eines Sprints vollständig abzuschließen und zu testen. Die meisten manuellen Tests stellen Verschwendung von Zeit dar, die schnelllebige Scrum-Teams nicht haben. Die Automatisierung Ihrer Regressionstests überprüft, dass Ihre bisherige Arbeit nicht durch Ihre neue Arbeit kaputtgemacht wird, und ist entscheidend, um ein Backup der technischen Schulden zu verhindern.

Lösung: Heutzutage stehen Ihnen zahlreiche kostengünstige Open-Source-Testwerkzeuge zur Verfügung. Schauen Sie sich die geeigneten Tools an und engagieren Sie sich als Entwicklungsteam für den Einsatz dieser Tools.

Fehlende Unterstützung in der Transitionsphase

Ohne die Unterstützung von Fach- und Führungskräften, die Scrum-Teams durch neue Ansätze führen können, können neue Scrum-Teams in alte Gewohnheiten zurückfallen.

Lösung: Holen Sie sich die Hilfe eines agilen Mentors – entweder aus Ihrer Organisation oder von einem Beratungsunternehmen –, der Sie beim Übergang unterstützen kann. Der Implementierungsprozess ist einfach, aber es ist schwierig, Menschen zu Verhaltensänderungen zu bewegen. Es lohnt sich, in professionelle Unterstützung der Transitionsphase durch einen erfahrenen Partner zu investieren, der sich mit Verhaltenswissenschaften und organisatorischen Veränderungen auskennt.

Unangemessene Umgebung

Wenn Scrum-Teams nicht am gleichen Ort zusammenarbeiten, verlieren sie den Vorteil der persönlichen Kommunikation. Im selben Gebäude zu sein ist nicht genug; Scrum-Teams müssen im selben Bereich des Gebäudes sitzen.

Lösung: Ergreifen Sie Maßnahmen, um das Team räumlich zusammenzubringen. Hier ein paar Beispiele:

- ✓ Wenn sich die Mitglieder eines Scrum-Teams im selben Gebäude, aber nicht im selben Bereich befinden, bringen Sie sie zusammen.
- ✓ Erwägen Sie, für das Scrum-Team einen eigenen Raum oder einen Anbau zu schaffen.
- ✓ Versuchen Sie, den Arbeitsbereich des Scrum-Teams von Ablenkern fernzuhalten, wie beispielsweise von dem Burschen, der ununterbrochen redet, oder dem Manager, der ständig um einen kleinen Gefallen bittet.
- ✓ Bevor Sie ein Projekt mit einem nicht an einem Ort arbeitenden Scrum-Team starten, tun Sie Ihr Möglichstes, um lokale Talente zu gewinnen.

Siehe [Kapitel 4](#) für weitere Informationen zur Kollokation.

Schlechte Teamauswahl

Scrum-Teammitglieder, die Scrum nicht unterstützen, die nicht gut mit anderen zusammenarbeiten oder die nicht die Fähigkeit zur Selbstverwaltung haben, können ein neues Scrum-Projekt von innen heraus sabotieren.

Lösung: Bei der Erstellung eines Scrum-Teams sollten Sie bedenken, wie gut potenzielle Teammitglieder die agilen Prinzipien umsetzen können. Der Schlüssel ist Vielseitigkeit und Lernbereitschaft.

Laxe Disziplin

Scrum-Projekte benötigen Anforderungen, Design, Entwicklung, Tests und Releases. Diese Arbeiten in Sprints zu erledigen, erfordert Disziplin.

Lösung: Bauen Sie von Anfang an die Gewohnheit ein, dass die einsatzfähige Funktionalität anhand der Definition of Done erstellt wird. Sie brauchen mehr und nicht weniger Disziplin, um funktionierende Produkte in einer kurzen Iteration zu liefern. Der Fortschritt muss konsequent und konstant sein. Um gute Gewohnheiten zu entwickeln,

- ✓ verwenden Sie das Daily Scrum, um sicherzustellen, dass während des gesamten Sprints Fortschritte erzielt werden,
- ✓ nutzen Sie die Sprint-Retrospektive als Gelegenheit, die Disziplin wiederherzustellen,
- ✓ überprüfen und verfeinern Sie regelmäßig die Definition of Done des Teams.

Fehlende Unterstützung für das Lernen

Scrum-Teams sind als Team erfolgreich und sie scheitern als Team; die Fehler einer Person für das Scheitern des Teams verantwortlich zu machen und gegenseitige Schuldzuweisungen

auszusprechen, wirkt sich negativ auf die Lernbereitschaft und die Innovation aus.

Lösung: Scrum-Teams können sich beim Projektstart verpflichten, Raum zum Lernen zu lassen und Erfolge und Misserfolge als Gruppe zu akzeptieren.

Verwässerte Prozesse

Die Verwässerung von Scrum-Rollen, Artefakten und Ereignissen mit alten Wasserfallgewohnheiten untergräbt die Vorteile agiler Prozesse, bis diese nicht mehr existieren.

Lösung: Wenn Sie Prozessänderungen vornehmen, hören Sie auf zu überlegen, ob diese Änderungen das Scrum-Framework, das Agile Manifest und die Agile Prinzipien unterstützen. Widerstehen Sie Änderungen, die nicht mit dem Manifest und den Prinzipien funktionieren. Denken Sie daran, die Verschwendung zu reduzieren, indem Sie die Menge der unnötigen, nicht geleisteten Arbeit maximieren.

Kapitel 21

Zehn Hauptvorteile von Scrum

IN DIESEM KAPITEL

Ergebnisse verbessern
Risiko reduzieren
Einfaches Reporting durchführen
Sicherstellen, dass Projekte sich lohnen

Dieses Kapitel führt zehn wichtige Vorteile auf, die Scrum Organisationen, Teams, Produkten und Einzelpersonen bietet.



Um die Vorteile von Scrum zu nutzen, müssen Sie auf die Empirie vertrauen, mehr über das Scrum-Framework herausfinden und Ihre Scrum-Implementierung kontinuierlich überprüfen und anpassen.

Bessere Qualität

Projekte dienen dazu, eine Vision oder ein Ziel zu erreichen. Scrum bietet den Rahmen für kontinuierliches Feedback und Transparenz, um sicherzustellen, dass die Qualität so hoch ist wie möglich. Wenn wir von Qualität sprechen, geht es nicht nur darum, eine Reihe von Tests zu bestehen. Das richtige Produkt zu entwickeln und es gut zu entwickeln, erfordert den Austausch mit Kunden und Fachleuten. Scrum hilft bei der Sicherung der Qualität durch die folgenden Vorgehensweisen:

- ✓ Definition und Ausarbeitung von Anforderungen just in time, damit die Kenntnis der Produkteigenschaften so relevant wie

möglich ist.

- ✓ Tägliche Tests und Rückmeldungen von Product Ownern in den Entwicklungsprozess einfließen lassen, damit das Entwicklungsteam Probleme bereits in der Anfangsphase beheben kann.
- ✓ Regelmäßige und kontinuierliche Verbesserung des Outputs des Scrum-Teams (Produkt oder Dienstleistung) durch Sprint-Reviews mit Stakeholdern.
- ✓ Durchführung von Sprint-Retrospektiven, die es dem Scrum-Team ermöglichen, teamspezifische Faktoren wie Prozesse, Tools, Beziehungen und Arbeitsumgebungen kontinuierlich zu verbessern.
- ✓ Fertigstellung der Arbeiten anhand der Definition of Done, die Entwicklung, Test, Integration und Dokumentation für jeden Backlog-Eintrag beinhaltet.
- ✓ Reduzierung des Kontextwechsels durch Fokussierung auf kleine Iterationen führt zu weniger Fehlern und ermöglicht das Testen innerhalb weniger Tage nach der Änderung.

Verkürzte Markteinführungszeit

Scrum führt nachweislich um 30 bis 40 Prozent schneller als herkömmliche Methoden zu Ergebnissen. Diese Zeitverkürzung ist auf folgende Faktoren zurückzuführen:

- ✓ Früherer Beginn der Entwicklung aufgrund der Tatsache, dass die Vorab-Dokumentationsphasen von Wasserfallprojekten (die in der Regel Monate dauern) entfallen, da ein dedizierter Product Owner in das Scrum-Team eingebunden ist, um die Anforderungen nach und nach just in time auszuarbeiten und in Echtzeit zu klären.
- ✓ Die Anforderungen mit der höchsten Priorität werden von den Anforderungen mit der niedrigsten Priorität getrennt. Die inkrementelle Lieferung von Wert für den Kunden bedeutet, dass die Anforderungen mit dem höchsten Wert und dem

höchsten Risiko (siehe [Kapitel 5](#) und [6](#)) vor den Anforderungen mit niedrigerem Wert und niedrigerem Risiko fertiggestellt werden können. Sie müssen nicht warten, bis das gesamte Projekt abgeschlossen ist, bevor Sie etwas auf den Markt bringen können. Erstellen Sie nur die Features, die verwendet werden, und bieten Sie den größten geschäftlichen Nutzen.

- ✓ Die Funktionalität wird in jedem Sprint mittels Swarming fertiggestellt. Am Ende eines jeden Sprints produzieren Scrum-Teams funktionierende Produkt- und Serviceinkremente, die ausgeliefert werden können.

Erhöhte Kapitalrendite

Die Verkürzung der Markteinführungszeit ist ein wesentlicher Grund dafür, dass Scrum-Projekte einen höheren Return on Investment (ROI) erzielen. Da Einnahmen und andere angestrebte Vorteile früher eintreffen, bedeutet eine frühere Akkumulation eine höhere Gesamtrendite im Zeitablauf, was ein Grundprinzip der Barwertberechnung ist (siehe [Kapitel 14](#)). Zusätzlich zu den Vorteilen bei der Markteinführungszeit erhöht Scrum den ROI durch die folgenden Maßnahmen:

- ✓ Regelmäßiges Feedback durch Sprint-Reviews direkt von Stakeholdern (einschließlich Kunden) ermöglicht frühzeitige Kurskorrekturen, was weniger kostspielig und zeitaufwendig ist, als wenn dies später im Prozess erfolgt.
- ✓ Weniger kostspielige Fehler durch Automatisierung und Vorabtests bedeuten weniger Arbeitsaufwand und schnellere Bereitstellung.
- ✓ Die Kosten des Scheiterns werden reduziert. Wenn ein Scrum-Projekt scheitert, scheitert es früher und schneller als ein Wasserfallprojekt.

Größere Kundenzufriedenheit

Scrum-Teams sind bestrebt, Produkte und Dienstleistungen zu produzieren, die den Kunden zufriedenstellen. Scrum ermöglicht glücklichere Projektsponsoren durch

- ✓ Zusammenarbeit mit den Kunden als Partner und Einbindung der Kunden während der gesamten Projektlaufzeit.
- ✓ Einen Product Owner, der ein Experte für Produktanforderungen und Kundenbedürfnisse ist. ([Kapitel 2](#) bis 6 behandeln die Rolle des Product Owners.)
- ✓ Aktualisierung und Priorisierung des Produkt-Backlogs, um schnell auf Änderungen reagieren zu können. (Mehr über das Produkt-Backlog erfahren Sie in [Kapitel 3](#).)
- ✓ Demonstration der Funktionsfähigkeit gegenüber internen Stakeholdern und Kunden bei jedem Sprint-Review. ([Kapitel 6](#) zeigt Ihnen, wie Sie einen Sprint-Review durchführen.)
- ✓ Schnellere und häufigere Lieferung von Produkten an Kunden mit jedem Release statt erst am Ende.
- ✓ Schrittweise Finanzierung von Projekten statt großer Vorleistungen. ([Kapitel 14](#) informiert Sie über die schrittweise Finanzierung von Projekten.)

Bessere Teammoral

Die Arbeit mit glücklichen Menschen, die Spaß an ihrer Arbeit haben, kann befriedigend und lohnend sein. Selbstmanagement legt Entscheidungen, die normalerweise von einem Manager oder der Organisation getroffen werden, in die Hände der Scrum-Teammitglieder. Scrum verbessert die Moral der Teammitglieder auf die folgende Weise:

- ✓ Als Teil eines selbstverwaltenden und selbstorganisierenden Teams können Menschen kreativ und innovativ sein und ihre Expertise wird anerkannt.
- ✓ Entwicklungsteams können ihre Teamstruktur um Menschen mit spezifischen Arbeitsstilen und Persönlichkeiten

organisieren. Die Organisation rund um den Arbeitsstil bietet diese Vorteile:

- Sie ermöglicht es den Teammitgliedern, so zu arbeiten, wie sie wollen.
 - ermutigt die Teammitglieder, ihre Fähigkeiten zu erweitern, sodass sie in Teams passen, die ihnen gefallen.
 - Sie steigert die Teamleistung, weil Menschen, die gute Arbeit leisten, gerne zusammenarbeiten und leichter aufeinander zugehen.
- ✓ Scrum-Teams können Entscheidungen treffen, die auf die Balance zwischen dem beruflichen und privaten Leben der Teammitglieder zugeschnitten sind.
 - ✓ Eine Peer-Beziehung mit einem Geschäftsvertreter (Product Owner) im selben Team bringt technische und kaufmännische Fachleute zusammen und überwindet organisatorische Barrieren.
 - ✓ Ein Scrum Master, der dem Scrum-Team dient, beseitigt Hindernisse und schirmt das Entwicklungsteam von externen Störungen ab.
 - ✓ Die Konzentration auf nachhaltige Arbeitspraktiken und Rhythmen stellt sicher, dass Menschen nicht durch Stress oder Überlastung ausbrennen.
 - ✓ Die funktionsübergreifende Arbeit ermöglicht es den Mitgliedern des Entwicklungsteams, neue Fähigkeiten zu erlernen und durch das Unterrichten anderer zu wachsen.
 - ✓ Die Förderung eines Servant-Leader-Ansatzes unterstützt Scrum-Teams bei der Selbstverwaltung und der aktiven Vermeidung klassischer Befehls- und Kontrollstrukturen.
 - ✓ Ein Umfeld der Unterstützung und des Vertrauens erhöht die allgemeine Motivation und Moral der Menschen.
 - ✓ Persönliche Gespräche helfen, Frustration, die durch Missverständnisse entsteht, zu verringern.

- ✓ Letztendlich können sich die Scrum-Teams auf Regeln einigen, wie sie vorgehen wollen, um die Arbeit zu erledigen.



Die Idee der Teamanpassung ermöglicht an Scrum-Arbeitsplätzen eine größere Vielfalt. Organisationen mit traditionellen Führungsstilen haben in der Regel monolithische Teams, in denen jeder die gleichen Regeln befolgt. Scrum-Arbeitsumgebungen sind wie eine Salatschüssel. Da Salate Zutaten mit ganz unterschiedlichen Geschmacksrichtungen haben können, die ein köstliches Gericht ergeben, können Scrum-Projekte in Teams Mitglieder mit unterschiedlichen Stärken haben, die großartige Produkte herstellen.

Bessere Zusammenarbeit und mehr Eigenverantwortung

Wenn Scrum-Teams Verantwortung für Projekte und Produkte übernehmen, können sie großartige Ergebnisse erzielen. Scrum-Teams setzen die folgenden Praktiken ein, um zusammenzuarbeiten und gemeinsam die Verantwortung für Qualität und Projektleistung zu übernehmen:

- ✓ Das Entwicklungsteam, der Product Owner und der Scrum Master arbeiten täglich eng zusammen.
- ✓ Durchführung von Sprint-Planungsmeetings, die es dem Entwicklungsteam ermöglichen, seine Arbeit anhand fundierter geschäftlicher Prioritäten zu organisieren.
- ✓ Daily Scrum-Meetings, in denen sich die Mitglieder des Entwicklungsteams um fertiggestellte Arbeiten, zukünftige Arbeiten und um Hindernisse kümmern.
- ✓ Durchführung von Sprint-Reviews, in denen der Product Owner seine Priorisierungsentscheidungen umreißt und das

Entwicklungsteam das Produkt direkt den Stakeholdern demonstrieren und es mit ihnen diskutieren kann.

- ✓ Durchführung von Sprint-Retrospektiven, die es den Mitgliedern des Scrum-Teams ermöglichen, vergangene Arbeiten zu überprüfen und bei jedem Sprint bessere Praktiken zu empfehlen.
- ✓ Arbeiten an einem gemeinsamen Ort, was eine sofortige Kommunikation und Zusammenarbeit zwischen den Mitgliedern des Entwicklungsteams, dem Product Owner und dem Scrum Master ermöglicht.
- ✓ Entscheidungen im Konsens treffen, unter Verwendung von Techniken wie Schätzpoker und Fist of Five.

In [Kapitel 4](#) erfahren Sie, wie Entwicklungsteams den Aufwand für Anforderungen schätzen, Anforderungen verfeinern und untergliedern und einen Teamkonsens erzielen. Mehr über die Sprint-Planung und die Daily Scrum-Meetings erfahren Sie in [Kapitel 5](#). Weitere Informationen über Sprint-Reviews und Retrospektiven finden Sie in [Kapitel 6](#).

Weitere relevante Metriken

Die Metriken, die Scrum-Teams verwenden, um Zeit und Kosten zu schätzen, um die Projektleistung zu messen und um Projektentscheidungen zu treffen, sind oft relevanter, sichtbarer und genauer als die Metriken traditioneller Projekte. Bei Scrum-Projekten sind Metriken relevanter, denn

- ✓ Diejenigen, die die Arbeit machen werden, und niemand sonst, geben Aufwandsabschätzungen für die Projektanforderungen ab.
- ✓ Zeitpläne und Budgets basieren auf den tatsächlichen Leistungen und Fähigkeiten jedes Entwicklungsteams.
- ✓ Mit Hilfe von relativen Schätzungen, anstatt Stunden oder Tagen, wird der geschätzte Aufwand auf das Wissen und die

Fähigkeiten eines bestimmten Entwicklungsteams zugeschnitten.

- ✓ In weniger als einer Minute pro Tag können Entwickler das Burndown-Diagramm aktualisieren und so täglich sehen, wie sich das Entwicklungsteam auf ein Sprintziel zubewegt.
- ✓ Am Ende eines jeden Sprints kann ein Product Owner die tatsächlichen Kosten des Projekts (TK) plus die Opportunitätskosten/Alternativkosten zukünftiger Projekte (AK) mit dem Wert vergleichen, den das aktuelle Projekt zurückgibt (W). So weiß er, wann ein Projekt beendet und ein neues begonnen werden muss. Sie müssen nicht bis zum Ende eines Projekts warten, um zu wissen, was sein Wert ist. (In [Kapitel 5](#) finden Sie weitere Informationen zur Formel $TK + AK > W$, die als Trigger für das Bestimmen des Projektendes dient.)



Vielleicht ist Ihnen aufgefallen, dass die Velocity in dieser Liste fehlt. *Velocity* (ein Maß für die Entwicklungsgeschwindigkeit, wie in [Kapitel 4](#) beschrieben) ist eine Tatsache und kein Ziel. Es ist eine Metrik, aber nur für das jeweilige Scrum-Team. Scrum-Teams können diesen Input nutzen, um zu bestimmen, wie viel Arbeit sie in zukünftigen Sprints leisten können, jedoch funktioniert dies nur, wenn es auf ein einzelnes Team zugeschnitten ist. Die Velocity von Team A hat keinen Einfluss auf die Velocity von Team B. Außerdem ist die Velocity für Messungen und Trends hervorragend geeignet, jedoch funktioniert sie nicht als Kontrollmechanismus. Der Versuch, ein Entwicklungsteam dazu zu bringen, eine bestimmte Geschwindigkeitszahl zu erreichen, stört nur die Teamleistung und verhindert das Selbstmanagement.

Wenn Sie mehr über die relative Schätzung erfahren möchten, lesen Sie [Kapitel 4](#). [Kapitel 22](#) beschreibt zehn Schlüsselkennzahlen für Scrum-Projekte.

Verbesserte Sichtbarkeit und Transparenz des Fortschritts

Bei Scrum-Projekten hat jedes Mitglied des Projektteams (einschließlich des Scrum-Teams und der Stakeholder) die Möglichkeit, jederzeit zu erfahren, wie das Projekt verläuft. Transparenz und Sichtbarkeit machen Scrum zu einem Expositionsmodell, das dem Projektteam hilft, Probleme genau zu identifizieren und den Verlauf des Projekts besser vorherzusagen. Scrum-Projekte können einen hohen Grad an Transparenz über den Fortschritt bieten, indem sie:

- ✓ Großen Wert legen auf eine offene, ehrliche Kommunikation zwischen dem Scrum-Team, Stakeholdern, Kunden und allen anderen Personen innerhalb einer Organisation, die über ein Projekt informiert werden möchten.
- ✓ Daily Scrums, die einen täglichen Einblick in den unmittelbaren Fortschritt des Entwicklungsteams und in die Hindernisse geben.
- ✓ Daily Scrums am Taskboard, die es Entwicklern ermöglichen, sich selbst zu organisieren und die Aufgaben mit der höchsten Priorität für den Tag zu identifizieren.
- ✓ Informationen aus den Daily Scrum-Meetings, Sprint-Burndown-Diagrammen und Taskboards verwenden, um den Fortschritt der einzelnen Sprints zu verfolgen.
- ✓ Mit Sprint-Retrospektiven erkennen, was gut funktioniert und was nicht, um Aktionspläne für Verbesserungen zu erstellen.
- ✓ Demonstration von Leistungen in Sprint-Reviews. Jeder innerhalb einer Organisation kann an einem Sprint-Review teilnehmen, auch Mitglieder anderer Scrum-Teams.

Eine verbesserte Projekttransparenz kann zu einer besseren Kontrolle und Vorhersagbarkeit des Projekts führen, wie in den folgenden Abschnitten beschrieben.

Verbesserte Projektsteuerung

Scrum-Teams haben zahlreiche Möglichkeiten, die Projektleistung zu kontrollieren und bei Bedarf Korrekturen vorzunehmen:

- ✓ Die Anpassung der Prioritäten während des gesamten Projekts in jedem Sprint-Intervall und anstatt bei wichtigen Meilensteinen ermöglicht es der Organisation, Projekte mit fester Dauer und zu festen Preisen durchzuführen und gleichzeitig den Veränderungen Rechnung zu tragen.
- ✓ Das Projektteam kann auf externe Faktoren wie beispielsweise die Marktnachfrage reagieren.
- ✓ Koordination im Daily Scrum ermöglicht es dem Scrum-Team, Probleme schnell anzugehen, sobald sie auftreten, und gemeinsam die Anforderungen zu erfüllen.
- ✓ Tägliche Aktualisierungen der Sprint-Backlogs bedeuten, dass die Sprint Burndown-Diagramme den Sprintfortschritt genau wiedergeben und dem Scrum-Team die Möglichkeit geben, Änderungen vorzunehmen, sobald es Probleme sieht.
- ✓ Persönliche Gespräche beseitigen Hindernisse bei der Kommunikation und der Problemlösung.
- ✓ Sprint-Reviews ermöglichen es den Projektbeteiligten, funktionierende Produkte zu sehen und den Produktbesitzern das Feedback zu geben, das sie benötigen, um sicherzustellen, dass das Projekt auf Kurs bleibt.
- ✓ Sprint-Retrospektiven ermöglichen es dem Scrum-Team, am Ende eines jeden Sprints fundierte Kursanpassungen vorzunehmen, um die Produktqualität zu verbessern, die Leistung des Entwicklungsteams zu steigern und die Projektprozesse zu verfeinern.

Die vielen Möglichkeiten, Scrum-Projekte zu überprüfen und anzupassen, ermöglichen es allen Mitgliedern des Projektteams – Entwicklungsteam, Product Owner, Scrum Master und

Stakeholdern –, die Kontrolle zu behalten und letztendlich bessere Produkte zu entwickeln.

Reduziertes Risiko

Scrum trägt dazu bei, das Risiko eines totalen Projektausfalls zu minimieren –große Mengen an Zeit und Geld ausgeben ohne einen Return on Investment zu erhalten –, indem früh greifbare Produktinkremente geliefert werden und das Scrum-Teams frühzeitig zum Scheitern gezwungen wird, falls es denn scheitern sollte. Dazu werden die folgenden Praktiken eingesetzt:

- ✓ Wenn die risikoreichsten Dinge zuerst erledigt werden, ist das die längste Startbahn, um Probleme zu lösen oder frühzeitig und kostengünstig zu scheitern.
- ✓ Entwicklung in Sprints, um zu gewährleisten, dass zwischen der ersten Projektinvestition und dem schnellen Scheitern oder der Validierung, dass ein Produkt oder ein Ansatz funktionieren wird, nur wenig Zeit vergeht.
- ✓ Ab dem ersten Sprint ein funktionierendes Produktinkrement zur Verfügung zu haben, sodass auch im Falle eines Projektabbruchs die wertvollsten und risikoreichsten Anforderungen entwickelt wurden und auf Wunsch an den Kunden geliefert werden können.
- ✓ In jedem Sprint die Anforderungen auf der Basis der Definition of Done entwickeln, damit Projektsponsoren fertiggestellte, brauchbare Funktionen erhalten, unabhängig davon, was mit dem Projekt in der Zukunft passieren kann.
- ✓ Ständiges Feedback zu Produkten und Prozessen durch
 - Daily Scrum-Meetings und ständige Kommunikation des Entwicklungsteams durch Kollokation,
 - regelmäßige Klärung der Anforderungen und Überprüfung und Akzeptanz der Features durch den Product Owner,

- Sprint-Reviews mit Stakeholder- und Kunden-Input über fertiggestellte Produktfeatures,
- Sprint-Retrospektiven, in denen das Entwicklungsteam die Prozessverbesserung diskutiert.
- Releases, in denen der Endanwender regelmäßig neue Features sehen und darauf reagieren kann.

Kapitel 22

Zehn wichtige Kennzahlen für Scrum

IN DIESEM KAPITEL

Ineffektive Kennzahlen vermeiden

Das Optimum aus den verfügbaren Daten herausholen

Den Nutzen und Wert von Scrum optimieren

Kennzahlen können in einem agilen Projekt mächtige Werkzeuge für die Planung, die Überprüfung, die Anpassung und das Verständnis des Fortschritts im Zeitverlauf. An den Erfolgs- und Ausfallquoten kann ein Scrum-Team erkennen, ob es positive Veränderungen vornehmen muss oder die gute Arbeit aufrechterhalten kann. Zahlen zu Dauer und Kosten können die Vorteile agiler Projekte hervorheben und die Finanzaktivitäten einer Organisation untermauern. Kennzahlen, die die Zufriedenheit der beteiligten Personen quantifizieren, können dem Scrum-Team helfen, Bereiche für die Verbesserung des Kundenkontakts und im Team selbst zu erkennen.



Doppelarbeit agil ist die Praxis des Managements, das neben Scrum-Artefakten, Ereignissen und geeigneten Metriken auch traditionelle Statusberichte und Meetings erwartet. Diese Praxis ist einer der wichtigsten Fallstricke bei Scrum-Projekten. Das Management ist auf der Suche nach einer Sache, während die Teams versuchen, eine andere zu tun. Dadurch werden Entscheidungen auf der Basis falscher Informationen getroffen, Teams werden durch die doppelte Arbeit überfordert, es besteht die Gefahr von Burnout-Problemen und der Nutzen von Scrum wird minimiert.

Dieses Kapitel beschreibt zehn wichtige Kennzahlen, die agile Teams als Richtschnur verwenden können.

Sprintziel-Erfolgsquoten

Eine Möglichkeit, um agile Projektperformance zu messen, ist die Quote, mit der die Sprintziele erreicht werden. Der Sprint benötigt eventuell nicht alle Anforderungen und Tasks im Sprint-Backlog, um das Ziel zu erreichen. Jedoch sollte ein erfolgreicher Sprint ein funktionierendes Produktinkrement ergeben, dass die Sprintziele erfüllt und das der Definition of Done des Scrum-Teams entspricht: entwickelt, getestet, integriert und dokumentiert.

Das Scrum-Team kann im Projektverlauf erfassen, wie oft es die Sprintziele erreicht und die Erfolgsquoten verwenden, um zu erkennen, ob sich das Team weiterentwickelt oder ob es Kurskorrekturen vornehmen muss. Sprinterfolgsquoten sind ein guter Ausgangspunkt für die Überprüfung und Anpassung. Die Teams sollten die Latte immer so hoch legen, dass die hundertprozentige Fertigstellung des Sprint-Backlogs nicht unbedingt das Ziel ist. Es ist wichtig, sicherzustellen, dass die begonnenen Anforderungen zu 100 Prozent fertig werden, jedoch sollten Teams die Ziele so hoch ansetzen, dass Erfolgsquoten von weniger als 100 Prozent bei den Sprint-Backlog-Einträgen als Chance zum Lernen und Verbessern betrachtet werden. Scrum Master sollten immer nach Möglichkeiten für ein störungsfreies

Arbeiten des Teams sorgen, damit dieses höhere Ziele setzen und erreichen kann, während es in jedem Sprint mehr und mehr erreicht. Sprint-Erfolgsraten sind ein nützlicher Ausgangspunkt für die Überprüfung und Anpassung.



Velocity ist kein Ziel, sondern ein Faktum, das sich nach Abschluss des Sprints ergibt. Eine zunehmende Velocity und eine zunehmende Fertigstellungsrate der Sprintziele sind Schlüsselindikatoren dafür, dass sich die Effizienz des Teams kontinuierlich verbessert.

Weitere Informationen über das Festlegen von Sprintzielen finden Sie in [Kapitel 8](#).

Defekte

Um wirklich agil zu sein, müssen Scrum-Teams agile Praktiken wie testgetriebene Entwicklung und kontinuierliche Integration implementieren (siehe [Kapitel 13](#)). Ohne solche Qualitätspraktiken sind Scrum-Teams nicht in der Lage, die Qualität so schnell zu liefern, wie es der Markt verlangt. Der Aufwand für manuelle Tests vor jedem Release ist einfach zu groß; außerdem lassen sich durch automatisierte Tests eine Vielzahl der Fehler, die sich eingeschlichen haben, erkennen und aufdecken.

Es ist unwahrscheinlich, dass ein Scrum-Team in der Lage sein wird, in diesen Bereichen Perfektion zu erreichen, sodass jedes Projekt wahrscheinlich einige Mängel aufweisen wird. Agile Techniken in Kombination mit dem Scrum-Framework helfen Entwicklungsteams, Fehler proaktiv zu minimieren.

Durch die Erfassung von Kennzahlen zu Defekten kann das Entwicklungsteam wissen, wie gut es Problemen vorbeugt und ob es seine Prozesse verbessern muss. Um Defekte zu erfassen und nachzuverfolgen, ist es hilfreich, sich die folgenden Zahlen anzusehen:

- ✓ **Defekte während der Entwicklung:** Wenn das Entwicklungsteam automatisiertes Testen und kontinuierliche Integration einsetzt, kann es in jedem Sprint die Anzahl der Defekte auf Build-Ebene erfassen.
Indem es die Anzahl der Build-Defekte erfasst, kann das Entwicklungsteam wissen, ob es Entwicklungsprozesse oder Umgebungsfaktoren anpassen muss, um die Defekte bereits früher in seinem Entwicklungsprozess abfangen zu können.
- ✓ **Defekte im Akzeptanztest (User Acceptance Test, UAT):** Das Entwicklungsteam kann die Anzahl der Defekte erfassen, die der Product Owner findet, wenn er in jedem Sprint die fertiggestellte Funktionalität überprüft. Durch die Erfassung der UAT-Defekte können das Entwicklungsteam und der Product Owner erkennen, ob die Prozesse für das Verständnis der Anforderungen verbessert werden müssen. Außerdem kann das Entwicklungsteam untersuchen, ob Anpassungen an den Werkzeugen für automatisierte Tests erforderlich sind.
- ✓ **Fehlerrate:** Neben der Erfassung der Gesamtanzahl der entdeckten Fehler ist es sinnvoll, sich die Häufigkeit pro User Story anzusehen (gelieferte Stories/gefundene Defekte). Zwei Fehler in 20 User Stories ergeben ein ganz anderes Bild als zwei Fehler bei zwei User Stories. Wenn ein Team seinen Arbeitsaufwand erhöht, sehen Sie, ob auch die Fehlerquote steigt.
- ✓ **Release-Defekte:** Das Entwicklungsteam kann die Anzahl der Defekte erfassen, die erst nach der Markteinführung sichtbar wurden.
Durch das Erfassen der Release-Defekte können das Entwicklungsteam und der Product Owner wissen, ob Änderungen am Akzeptanztest, am automatisierten Test oder am Entwicklungsprozess erforderlich sind. Eine große Anzahl von Defekten auf der Release-Ebene kann ein Indikator für größere Probleme im Scrum-Team sein.

Die Anzahl der Defekte und ob die Defekte zunehmen, abnehmen oder gleich bleiben, sind gute Kennzahlen, um in Sprint-

Retrospektiven Diskussionen über Projektprozesse und Entwicklungstechniken zu entfachen.

Produkteinführungszeit – Time to Market

Produkteinführungszeit ist die Zeit, die ein agiles Projekt braucht, um durch den Release von einsatzfähiger Funktionalität an den Kunden Wert zu erzeugen. Organisationen können Wert auf unterschiedliche Art interpretieren:

- ✓ Wenn ein Produkt direkt Einnahmen generiert, dann ist sein Wert das Geld, das damit verdient wird.
- ✓ Wenn ein Produkt intern in der Organisation verwendet wird, besteht sein Wert darin, dass die Mitarbeiter in der Lage sind, es zu verwenden, und er enthält subjektive Faktoren, die darauf basieren, was das Produkt kann.

Bedenken Sie beim Messen der Produkteinführungszeit folgende Punkte:

- ✓ Messen Sie die Zeit vom Projektstart, bis Sie erstmalig Wert sehen.
- ✓ Manche Scrum-Teams veröffentlichen am Ende jedes Sprints neue Produktfeatures für den Einsatz. Für Scrum-Teams, die mit jedem Sprint releasen, ist die Produkteinführungszeit einfach die Sprintdauer in Tagen gemessen.
- ✓ Andere Scrum-Teams planen Releases nach mehreren Sprints und stellen Produktfeatures in Gruppen bereit. Bei Scrum-Teams, die längere Releasezeiten einsetzen, ist die Produkteinführungszeit die Anzahl der Tage zwischen den Releases.

Die Produkteinführungszeit hilft Organisationen, den dauerhaften Nutzen agiler Projekte zu erkennen und zu quantifizieren. Die

Produkteinführungszeit ist vor allem für Unternehmen mit Umsatz generierenden Produkten wichtig, da sie bei der jährlichen Finanzplanung hilft. Sie ist ebenfalls wichtig, falls Sie ein sich selbst finanzierendes Projekt haben – ein Projekt, dessen Entwicklung aus den Einnahmen des Produkts bezahlt wird.

Return on Investment

Return on Investment (ROI) sind die durch ein Produkt generierten Einnahmen abzüglich der Projektkosten: Geldeinnahmen versus Geldausgaben. ROI ist in agilen Projekten fundamental anders als in traditionellen Projekten. Agile Projekte haben das Potenzial, bereits mit dem ersten Release Einnahmen zu generieren; außerdem kann der Umsatz mit jedem neuen Release zunehmen.

Vergleichen Sie die beiden folgenden Szenarien, um den Unterschied zwischen ROI bei traditionellen und agilen Projekten voll und ganz zu verstehen. Die Projekte in beiden Beispielen weisen identische Projektkosten auf; die für die Fertigstellung benötigte Zeit ist ebenfalls gleich. Wir ignorieren die zusätzliche Dokumentation, Meeting- und anderen Kosten eines Wasserfallprojekts und geben zur Vereinfachung des Vergleichs davon aus, dass sich diese auf dem Scrum-Level befinden. Beide Szenarien beginnen am 1. Januar und sind in der Lage, je Monat 100.000 € Umsatz zu generieren, wenn alle Anforderungen fertig sind.

Bei Szenario A handelt es sich um ein Wasserfallprojekt. Das Release erfolgt am 30. Juni des gleichen Jahres, nachdem alle Anforderungen fertiggestellt sind. Dieses Projekt kann ab Juli monatlich 100.000 € Umsatz generieren. Bis zum Ende des Jahres (6 Monate) beträgt der Umsatz also 600.000 €.

Bei Szenario B werden die Features mit dem höchsten Wert und dem größtem Risiko inkrementell nach 4 einwöchigen Sprints ab dem 31. Januar veröffentlicht, also fünf Monate früher als bei Projekt A. Der monatliche Umsatz ist am Anfang geringer, nimmt

aber nach dem ersten Release mit jedem Monat zu, er beträgt im Februar 50.000 €, im März 60.000 €, im April 70.000 €, im Mai 80.000 €, im Juni 90.000 € und ab Juli, wenn das gesamte Projekt fertiggestellt ist, 100.000 €.

Aufgrund der zusätzlichen Umsätze in den fünf Monaten von Februar bis Juni beträgt der Gesamtjahresumsatz von Projekt B 950.000 €, also 350.000 € mehr als in Szenario B.



So wie die Produkteinführungszeit sind die ROI-Kennzahlen eine hervorragende Möglichkeit, damit eine Organisation den kontinuierlichen Wert eines agilen Projekts verstehen kann. ROI-Kennzahlen helfen, Projekte von Anfang an zu rechtfertigen, da Firmen Projekte anhand des ROI-Potenzials finanzieren können. Organisationen können den ROI für einzelne Projekte, aber auch für die gesamte Organisation verfolgen.

Gesamtprojektdauer und Kosten

Um den ROI zu berechnen, berechnen Sie zunächst die Dauer und die Kosten, die selbst effektive Inputs und Metriken für Scrum-Projekte sein können. Scrum-Projekte sollten schneller und kostengünstiger durchgeführt werden als herkömmliche Projekte.

Ein höherer ROI durch kürzere Projektlaufzeiten und geringere Kosten sollte kann guter Indikator dafür sein, dass Scrum-Teams immer mehr das Swarming einsetzen, das Thrashing reduzieren und die Gesamteffizienz verbessern.

Neue Anforderungen innerhalb von ROI-Budgets

Die Fähigkeit eines agilen Projekts, schnell hohe Rendite zu generieren, bietet Organisationen eine einzigartige Möglichkeit, zusätzliche Produktentwicklungen zu finanzieren. Neue Produktfunktionalität kann zu höheren Produkteinnahmen führen.

Angenommen, dass das Projektteam für das obige Beispielprojekt ein neues Feature ausgemacht hat, dessen Fertigstellung vier weitere einwöchige Sprints pro Monat benötigt und das dazu führt, dass die Einnahmen aus dem Produkt von monatlich 100.000 € auf monatlich 120.000 € steigen. Wenn in Szenario B das neue Feature früher erkannt worden wäre, wäre der Return on Investment sogar noch höher.

Falls ein Projekt bereits Einnahmen generiert, kann es für eine Organisation sinnvoll sein, diese Einnahmen in neue Entwicklungen zu stecken, die zu mehr Umsatz führen.

Kapitalumschichtung

Wenn bei einem agilen Projekt die Kosten für weitere Entwicklungen höher sind als der Wert dieser zukünftigen Entwicklungen, ist es an der Zeit, das Projekt zu beenden.

Der Product Owner priorisiert die Anforderungen zum Teil danach, ob sie in der Lage sind, Umsatz oder Wert zu generieren. Falls im Backlog nur noch Anforderungen übrig bleiben, die wenig Umsatz generieren oder geringen Wert bringen, kann ein Projekt enden, bevor das Projektteam das gesamte Budget verbraucht hat. Die Organisation kann dann die übrig gebliebenen Finanzmittel des alten Projekts verwenden, um ein neues, wertvolleres Projekt zu starten. Die Praxis, ein Budget von einem Projekt auf ein anderes zu übertragen, wird *Kapitalumschichtung* genannt.

Um zu berechnen, ob es sinnvoll ist, das Projekt zu beenden, benötigen Sie diese Kennzahlen:

- ✓ den Geschäftswert (W) der verbleibenden Anforderungen im Produkt-Backlog
- ✓ die tatsächlichen Kosten (TK) der Arbeit, die für das Fertigstellen der Anforderungen im Produkt-Backlog anfallen
- ✓ die Alternativkosten (AK) oder den Wert, falls das Scrum-Team an einem neuen Projekt arbeitet

Wenn $W < TK + AK$ (oder $AK + TK > W$, wie in [Kapitel 5](#) beschrieben), dann kann das Projekt beendet werden. Die Kosten, die Sie in das Projekt stecken, sind höher als der Wert, den Sie aus dem Produkt erzielen.

Kapitalumschichtung erlaubt es einer Organisation, effizient Geld für wertvolle Produktentwicklung auszugeben und die Gesamtinvestitionsrentabilität der Organisation zu maximieren.

Umfragen zur Zufriedenheit

Die höchste Priorität eines Scrum-Teams in einem agilen Projekt besteht darin, den Kunden früh und oft zufrieden zu stellen. Gleichzeitig strebt das Scrum-Team danach, jedes einzelne Teammitglied zu motivieren und nachhaltige Entwicklungspraktiken zu fördern.

Ein Scrum-Team kann davon profitieren, tiefer in die Erfahrungen des Kunden und der Teammitglieder einzutauchen. Eine Möglichkeit, um messbare Informationen darüber zu erhalten, wie gut ein Scrum-Team die agilen Prinzipien verkörpert, besteht im Durchführen von Umfragen über die Zufriedenheit.

Umfragen zur Kundenzufriedenheit messen die Erfahrungen des Kunden mit dem Projekt, dem Prozess und dem Scrum-Team. Das Scrum-Team kann mehrmals während eines Projekts Kundenumfragen durchführen, auch zu Projektbeginn, um einen Benchmark für zukünftige Vergleiche zu erhalten. Das Scrum-Team kann die Umfrageergebnisse verwenden, um die Prozesse zu untersuchen, positive Praktiken fortsetzen und das Verhalten anpassen, wo es nötig ist.

Umfragen zur Teamzufriedenheit messen die Erfahrungen der Mitglieder des Scrum-Teams mit der Organisation, der Arbeitsumgebung, den Prozessen, mit anderen Projektteammitgliedern und mit ihrer Arbeit. Jeder im Scrum-Team kann Teamumfragen durchführen. So wie bei den Kundenumfragen kann sich das Scrum-Team dafür entscheiden, im Projektverlauf Teamumfragen durchzuführen. Mitglieder des

Scrum-Teams können die Umfrageergebnisse verwenden, um regelmäßig das persönliche Verhalten und das Verhalten des Teams anzupassen. Das Scrum-Team kann die Ergebnisse ebenfalls verwenden, um auf organisatorische Probleme zu reagieren. Im Lauf der Zeit können die Teamumfrageergebnisse einen quantitativen Blick darauf ermöglichen, wie das Team reift und sich entwickelt.

Sie können informelle schriftliche Umfragen erstellen oder eines der zahlreichen Online-Umfragetools verwenden. In manchen Firmen steht Umfragesoftware über die Personalabteilung zur Verfügung.

Fluktuation im Team

Scrum-Projekte sorgen tendenziell für eine bessere Arbeitsmoral. Eine Möglichkeit, um die Arbeitsmoral zu quantifizieren, ist, die Personalfluktuations zu messen. Obwohl Fluktuation nicht immer direkt mit der Arbeitszufriedenheit zusammenhängt, kann es hilfreich sein, sich die folgenden Metriken anzusehen:

- ✓ **Fluktuation im Scrum-Team:** Eine geringe Fluktuation im Scrum-Team kann ein Zeichen für ein gesundes Teamumfeld sein. Eine hohe Fluktuation im Scrum-Team kann auf Probleme im Projekt, in der Organisation, in der Arbeit, bei einzelnen Mitgliedern des Scrum-Teams, mit Burnout, mit einem ineffektiven Product Owner, der das Entwicklungsteam zwingt, bestimmte Verpflichtungen einzugehen, mit nicht kompatiblen Persönlichkeiten, mit einem Scrum Master, der es nicht schafft, Hindernisse zu beseitigen oder mit der allgemeinen Dynamik im Team hinweisen.

Teammitglieder können kündigen oder Manager »stehlen« Teammitglieder für andere Projekte. So oder so, das Ergebnis ist ein erhöhtes Thrashing, das organisatorische Probleme aufdeckt, die angegangen werden sollten.

- ✓ **Fluktuation in der Firma:** Eine hohe Fluktuation in der Firma, auch wenn sie nicht das Scrum-Team betrifft, kann die

Arbeitsmoral und Effektivität beeinflussen. Eine hohe Fluktuation in der Firma kann ein Zeichen für Probleme in der Organisation sein. Wenn eine Firma agile Praktiken einführt, kann es sein, dass die Fluktuation abnimmt.

Wenn das Scrum-Team die Kennzahlen zur Fluktuation kennt und die Gründe hinter diesen Metriken versteht, kann es in der Lage sein, Aktionen zu unternehmen, die die Arbeitsmoral fördern und das Arbeitsumfeld verbessern. Wenn die Fluktuation hoch ist, fragen Sie nach dem Warum.

Projektverschleiß

Organisationen mit Projektportfolios sollten sich mit der Quote abgebrochener Projekte befassen. Kapitalumschichtung ist nicht zu verwechseln mit dem Hin- und Herschieben von Teams zwischen Projekten je nach Lust und Laune von Führungskräften. Die Nachverfolgung von Projektlaufzeiten anhand von Kapitalumschichtungsanalysen kann Tendenzen aufzeigen, Projekte vorzeitig zu beenden oder länger als nötig laufen zu lassen.

Ausgehend von diesen Trends können Portfoliomanager nach Gründen suchen, warum Projekte abgebrochen werden. Hohe Fluktuation kann auf Probleme im Zusammenhang mit Thrashing, Planung, Priorisierung, Hindernissen und Cross-Funktionalität hinweisen.

Weitere Informationen darüber, wie Scrum das Portfoliomanagement verbessert, finden Sie in [Kapitel 13](#).

Vielseitigkeit der Qualifikationen

Erfahrene, entwickelte Scrum-Teams sind typischerweise crossfunktionaler als weniger gereifte und entwickelte Scrum-Teams. Durch die Beseitigung von Single Points of Failure

kommen Teams schneller voran und produzieren qualitativ hochwertigere Produkte. Das Erfassen der Vielseitigkeit der Qualifikationen erlaubt es Scrum-Teams und Vorgesetzten, das *Wachstum* der Crossfunktionalität zu beurteilen.

Beginnen Sie damit, die vorhandenen Fähigkeiten und das jeweilige Qualifikationsniveau zu erfassen, die in den folgenden Organisationsstrukturen vorhanden sind:

- ✓ Fähigkeiten und Qualifikationsniveau jeder Person
- ✓ Fähigkeiten und Qualifikationsniveau jedes Teams
- ✓ Fähigkeiten und Qualifikationsniveau in der Organisation

Wenn im Laufe der Zeit die Anzahl der Fähigkeiten und das Qualifikationsniveau der einzelnen Personen zunehmen, verschwinden die durch Kompetenzlücken verursachten Einschränkungen. In agilen Teams geht es um Fähigkeiten, nicht um Titel. Sie brauchen Teammitglieder, die jeden Tag ihren Beitrag zum Erreichen des Sprintziels leisten, ohne das Risiko von Single Points of Failure.

Verhältnis Manager- Produktersteller

Wahrscheinlich hat sich in großen Organisationen eine erhebliche Ebene des mittleren Managements entwickelt. Viele Organisationen haben nicht verstanden, wie sie gut funktionieren können, ohne dass es mehrere Manager gibt, die sich um Personal, Ausbildung und die technische Richtung bei Fragen der Softwareentwicklung kümmern. Sie müssen jedoch versuchen, ein ausgewogenes Verhältnis zwischen den Managern und den Personen herzustellen, die Produkte produzieren.



Jeder Euro, der für jemanden ausgegeben wird, der organisatorische Prozesse steuert, ist ein Euro, der nicht für einen Produktersteller ausgegeben wird.

Erfassen Sie das Verhältnis Manager zu Produkterstellern, um Aufblähung zu erkennen und um nach Wegen zu suchen, die in Mitarbeiter, die keine Produkte erstellen, investierten Finanzmittel zu minimieren.

Kapitel 23

Zehn wertvolle Ressourcen für Scrum

IN DIESEM KAPITEL

Unterstützung für erfolgreiche Scrum-Übergänge finden
Sich in Scrum-Communities einbringen
Auf Scrum-Ressourcen für das lebenslange Lernen zugreifen

Zahlreiche Organisationen, Websites, Blogs und Unternehmen stellen Informationen und Unterstützung zu Scrum bereit. Um Ihnen den Einstieg zu erleichtern, haben wir eine Liste der wichtigsten Ressourcen zusammengestellt, die Ihre Reise in die Scrum-Welt unterstützen können.

The Scrum Alliance

<http://scrumalliance.org>

The Scrum Alliance ist ein gemeinnütziger Berufsverband, der das Verständnis und den Einsatz von Scrum fördert. Der Verband erreicht diese Ziele durch die Durchführung von Scrum-Weiterbildungen, Zertifizierungskursen, die Organisation internationaler und regionaler Scrum-Kongresse und durch Unterstützung lokaler Scrum-User-Communities. Auf der Website der Scrum Alliance finden Sie vielfältige Blogeinträge, Informationsschriften, Fallstudien und weitere Tools, um Scrum zu erlernen und anzuwenden. The Scrum Alliance bietet unter anderem die folgenden Zertifizierungen an:

✓ Certified ScrumMaster (CSM)

- ✓ Advanced Certified ScrumMaster (A-CSM)
- ✓ Certified Scrum Product Owner (CSPO)
- ✓ Advanced Certified Scrum Product Owner (A-CSPO)
- ✓ Certified Scrum Developer (CSD)
- ✓ Advanced Certified Scrum Developer (A-CSD)
- ✓ Certified Scrum Professional (CSP)
 - CSP für ScrumMaster (CSP-SM)
 - CSP für Product Owner (CSP-PO)
 - CSP für Entwickler (CSP-D)
- ✓ Certified Team Coach (CTC)
- ✓ Certified Enterprise Coach (CEC)
- ✓ Certified Agile Leadership (CAL)

The Agile Alliance

<http://agilealliance.org>

The Agile Alliance ist eine globale agile Community, die das Ziel hat, die 12 Agilen Prinzipien sowie gebräuchliche agile Praktiken unabhängig vom konkreten Ansatz zu verbreiten. Die Website von The Agile Alliance enthält zahlreiche Ressourcen wie Artikel, Videos und Präsentationen sowie ein Verzeichnis unabhängiger agiler Community-Gruppen weltweit.

Scrumguides.org

<http://scrumguides.org>

Jeff Sutherland und Ken Schwaber, Mitbegründer von Scrum, stellen unter <http://scrumguides.org> den Leitfaden *Der Scrum Guide: Der gültige Leitfaden für Scrum: Die Spielregeln* in mehr als 30 Sprachen zur Verfügung. Der Scrum Guide ist sowohl online als auch im PDF-Format verfügbar und kann kostenlos verwendet werden. Auf weniger als 20 Seiten beschreibt der

Leitfaden die Scrum-Theorie und definiert jede Scrum-Rolle, jedes Artefakt und jedes Ereignis.

Scrum.org

<http://scrum.org>

[Scrum.org](http://scrum.org) bietet Tools und Ressourcen für Scrum-Praktiker an sowie Prüfungen und Zertifizierungen, wie

- ✓ Professional Scrum Master (PSM I, II, III)
- ✓ Professional Scrum Product Owner (PSPO I, II)
- ✓ Professional Scrum Developer (PSD)

Scruminc.com (Scrum at Scale)

<http://www.scruminc.com>

Jeff Sutherland und sein Team bieten Scrum-Training, Ressourcen und Zertifizierungen an, einschließlich Scrum at Scale.

ScrumPLoP

<http://scrumplop.org>

Pattern Languages of Programs (PLoP) sind Methoden zur Beschreibung von Designpraktiken innerhalb von Fachgebieten. Zu diesen Methoden werden häufig Konferenzen mit dem Ziel organisiert, gemeinsam zu lernen. ScrumPLoP veröffentlicht Muster von Scrum-Profis, von denen viele von Jeff Sutherland, dem Mitbegründer von Scrum, geschrieben wurden. Diese praktischen Muster wurden von Organisationen erfolgreich eingesetzt, um mit Scrum zu beginnen.

Scaled Agile Framework (SAFe)

<http://scaledagileframework.com>

Das Scaled Agile Framework (SAFe) ist eine Wissensbasis für die Implementierung agiler Praktiken und ein Framework für die Implementierung von Scrum at Scale. Verwenden Sie die interaktive Übersichtsgrafik auf der Startseite um klicken Sie sich durch die Highlights der Rollen, Teams, Aktivitäten und Artefakte.

SAFe ist ein eingetragenes Warenzeichen von Scaled Agile Inc.

LeSS

<http://less.works>

Large-Scale Scrum (LeSS) ist eine Scrum-Scaling-Methode, die zwei Frameworks zur Verfügung stellt, bekannt als LeSS und LeSS Huge. Wie bei SAFe bietet auch die Website von LeSS eine grafische Oberfläche, die die Frameworks hervorhebt.

InfoQ

www.infoq.com/scrum

InfoQ ist eine unabhängige Online-Community mit einem herausragenden Bereich zu agilem Projektmanagement. Sie finden dort Nachrichten, Artikel, Video-Interviews, Video-Präsentationen sowie Minibooks, die alle von Experten in agilen Techniken verfasst sind. Die Ressourcen auf InfoQ sind in der Regel von hoher Qualität, die Inhalte sind sowohl einmalig als auch im Hinblick auf die Probleme, mit denen agile Projektteams konfrontiert werden, wichtig.

Platinum Edge

www.platinumedge.com

Besuchen Sie unseren Blog, um sich über den aktuellen Stand bei Praktiken, Werkzeugen und innovativen Lösungen zu informieren, die im Rahmen unserer Arbeit mit Unternehmen der

Global-1000-Rangliste und der dynamischen agilen Community entstanden sind.

Außerdem bieten wir die folgenden Dienstleistungen an:

- ✓ **Agiles Auditing:** Untersuchung Ihrer aktuellen organisatorischen Strukturen und Prozesse, um eine agile Implementierungsstrategie zu entwickeln, die zu besseren Ergebnissen führt.
- ✓ **Personalbeschaffung:** Dank unseres Zugangs zu den besten agilen und Scrum-Talenten – sie wurden bei uns ausgebildet – helfen wir Ihnen dabei, die richtigen Leute zu finden (Scrum Master, Scrum Product Owner und Scrum-Entwickler), um Ihre Scrum-Projekte ins Laufen zu bringen.
- ✓ **Ausbildung:** Maßgeschneiderte agile und Scrum-Schulungen und Zertifizierung, unabhängig von Ihrem Wissensstand. Neben den maßgeschneiderten und nicht auf eine Zertifizierung ausgerichteten Schulungsangeboten bieten wir die folgenden Zertifizierungen an:
 - Ausbildung zum Certified Scrum Master (CSM)
 - Ausbildung zum Certified Scrum Product Owner (CSPO)
 - Ausbildung zum Certified Scrum Developer (CSD)
 - SAFe Scaled Agile, Ausbildung und Implementierungen
 - Prüfungsvorbereitung für PMI Agile Certified Practitioner (PMI-ACP)
- ✓ **Transition:** Kein Faktor ist wichtiger für zukünftigen Erfolg als sachgerechtes Coaching. Neben den verschiedenen Ausbildungen bieten wir eingebettetes agiles Coaching und Mentoring, um zu gewährleisten, dass im richtigen Leben die richtigen Praktiken eingesetzt werden.



Verwenden Sie die Schummelseite dieses Buches, wenn Sie damit beginnen, das in den vorigen Kapiteln beschriebene Scrum-Framework zu implementieren. Dort finden Sie hilfreiche Ressourcen, um bezüglich Scrum auf dem Laufenden zu bleiben.

Stichwortverzeichnis

3-3-5-Framework [1](#)

3M [1](#)

5-6-7-Framework [1](#)

7Shifts [1](#)

360-Grad-Feedback [1](#)

"Sprints

Definition“ [1](#)

A

Abbruchrate, als Metrik [1](#)

Abnahme

als Schritt beim Erstellen von Anforderungen [1](#)

Absolute Schätzung [1](#)

A-CSD, Advanced Certified Scrum Developer [1](#)

A-CSM, Advanced Certified Scrum Master [1](#)

A-CSPO, Advanced Certified Scrum Product Owner [1](#)

Affinitätsschätzung [1](#)

Affordable Care Act [1](#)

Agile Alliance [1](#)

Agile Prinzipien [1](#)

Agile Release Train, ART [1](#)

"Agiles Auditing„ [1](#)

agile Säule [1–2](#)

Agiles Manifest [1](#), [2](#)

Agiles Marketing-Manifest [1](#)

Agiles Transition-Team [1](#)

Agilität

Agile Prinzipien [1](#)

im Marketing [1](#)

Säulen der Verbesserung [1](#)

Werte [1](#)

Wurzeln [1](#)

Akademische Ausbildung [1](#)

Aktuelle Prozesse, in Implementierungsstrategie [1](#)

Akzeptanzkriterien [1](#)

als Ergebnis der Verfeinerung des Produkt-Backlogs [1](#)

Alltagspflichten [1](#)

als Schritt beim Erstellen von Anforderungen [1](#)

Alternativkosten (AK) [1](#), [2](#), [3](#)

Amazon [1](#), [2–3](#)

Anforderung

Code-Refactoring berücksichtigen [1](#)

Eintragstyp Produkt-Backlog [1](#), [2](#)

nicht abgenommene [1](#)

nicht fertiggestellte [1](#)

Schätzpoker [1](#)

unterteilen [1](#)

Unterteilungsstufen [1](#)

Anforderungsbereiche, LeSS Huge [1](#)

Anpassung

agile Säule [1](#)

für Familien [1](#)

Kurskorrektur [1](#)

beim Release [1](#)

kundenspezifische Projekte [1](#)
Medienlandschaft [1](#)
Release-Größe [1](#)
Anpassungsfähigkeit [1](#)
Anreize schaffen [1](#)
Anthem Healthcare [1](#)
Antimuster [1](#)
Antipattern [1](#)
Apple [1](#)
Arbeitsmoral [1](#)
Arbeitssicherheit [1](#)
Arbeitsumfeld
 Effizienz [1](#)
Arbeitszufriedenheit [1](#)
Architektur
 Besitzer [1](#)
 modulare [1](#)
 sich entwickelnde [1](#)
Area Product Owner [1](#)
Aristoteles [1](#)
Artefakte [1–2](#)
 Definition [1](#)
 verwässerte [1](#)
ART-Modell, Agile Release Train [1](#)
AT&T Wireless [1](#)
Atos Origin [1](#)
Audit durchführen [1](#)
Aufgaben [1](#)

Auftragstaktik [1](#)

Ausarbeitung, als Schritt beim Erstellen von Anforderungen [1](#)

Ausbildung [1](#)

mangelnde [1](#)

Auslieferfähig [1](#)

Ausschreibungsverfahren [1](#)

Automatisierte Tests [1](#)

Automobile [1](#), [2](#), [3](#)

Toyota [1](#)

Autonomie

Saab-Mehrzweckkampfflugzeug [1](#)

Autoren [1](#)

B

Bauen-Messen-Lernen-Kreislauf [1](#)

Baugewerbe

Angebote [1](#)

Arbeitssicherheit [1](#)

Kundenbeteiligung [1](#)

Scrum-Rollen [1](#)

Subunternehmer [1](#)

Transparenz [1–2](#)

Beitragende Persönlichkeit [1](#)

Benutzerakzeptanztest [1](#)

Beobachtungseffekt [1](#)

Beziehungen mit Scrum finden [1](#)

Big Data [1](#)

Bildgebung [1](#)

Bildungswesen

Beispielansatz [1](#)

Curriculum [1](#)

erausforderungen [1](#)

Klassengröße [1](#)

Klassenzimmer [1](#)

leistungsschwache Schüler [1](#)

Blockaden identifizieren [1](#)

Blueprint High School [1](#)

Bots [1](#)

Bronowski, Jacob [1](#)

Budget

Notfall [1](#)

Reiseplanung [1](#)

Burndown-Diagramme [1](#)

Burnout [1](#)

Burnup-Diagramme [1](#)

Businessaspekte und Technik aufeinander ausrichten [1](#)

Business Development

Marketing [1](#)

Vertrieb [1](#)

BYOA, bring your own application [1](#)

C

CafePress [1](#)

Canon [1](#)

Card – Conversation – Conformation [1](#)

Card – Conversation – Conformation [1](#)

CEC, Certified Enterprise Coach [1](#)

Certified Scrum Professional (CSP-SM, CSP-PO) [1](#)
Chicago Tribune, Tageszeitung [1](#)
CI, continuous integration [1](#)
Cirillo, Francesco [1](#)
Cisco [1](#)
CMG Partners [1](#)
Cockburn, Alistair [1](#)
Codequalität [1](#)
Code-Refactoring [1](#)
College [1](#)
Cone of Uncertainty [1](#)
Continuous Integration, CI [1](#)
CPO, Chief Product Owner [1](#)
Credit Hours [1](#)
CRM, Customer Relationship Management [1](#)
Cross-Funktionalität [1](#)
Crossing the Chasm [1](#)
CSD, Certified Scrum Developer [1](#), [2](#), [3](#)
CSM, Certified Scrum Master [1](#), [2](#), [3](#), [4](#)
CSP, Certified Scrum Professional [1](#)
CSP für Entwickler (CSP-D) [1](#)
CSP für Product Owner (CSP-PO) [1](#)
CSP für Scrum Master (CSP-SM) [1](#)
CSPO, Certified Scrum Product Owner [1](#), [2](#), [3](#)
CubeSat, John Hopkins [1](#)
Cumulative Flow Diagrams, CFDs [1](#)
Curriculum, Umfang des [1](#)
Cyberangriffe [1](#)

Cycle time [1](#)

D

Daily Scrum

Begriff [1](#)

Durchführung [1](#)

effektiver machen [1](#)

Familienalltag [1](#)

Gesundheitswesen [1](#)

in Large-Scale Scrum (LeSS) [1](#)

Innovationskultur [1](#)

Kanban [1](#)

Kundendienst [1](#)

planen [1](#)

Subunternehmer beteiligen [1](#)

Zweck [1](#)

Daodejing [1](#)

Das öde Land [1](#)

Data Warehouse [1](#)

Daten

Big Data [1](#)

Migration großer Datenmengen [1](#)

Dating

Definition of Done [1](#)

Fokussierung versus Multitasking [1](#)

Freundschaften nutzen [1](#)

in Schichten [1](#)

mit Scrum [1](#)

Roadmap to Value [1](#)

Teamarbeit [1](#)

Vision festlegen [1](#)

Decomposition [1](#)

Dedizierte Entwicklungsteams [1](#)

Defekte [1](#)

im Akzeptanztest [1](#)

während der Entwicklung [1](#)

Definition of Done [1](#)

Bildungswesen [1](#)

Dating [1](#)

Familienprojekt [1](#)

Kundendienst [1](#)

Nachrichtenmedien [1](#)

Pilotprojekt [1](#)

regulatorische Anforderungen [1](#)

Derby, Esther [1](#)

Derby-und-Larsen-Prozess [1](#)

Deseret Digital Media [1](#)

Designverbesserungen

Eintragstyp Produkt-Backlog [1](#)

DevOps, Development and Operations [1](#)

Dienstleistung [1](#)

Dienstleistungssektor

Bildungswesen [1](#)

Gesundheitswesen [1](#)

Militär und Strafverfolgung [1](#)

Dinge

die bekannt sind [1](#)

die man nicht wissen kann [1](#)

die man wissen könnte [1](#)

Disruptive Innovation [1](#)

Disziplin

fehlende [1](#)

Dokumentation

als Schritt beim Erstellen von Anforderungen [1](#)

permanently sichtbare [1](#)

Dokumentiert, Faktor der Definition of Done [1](#)

Doppelarbeit agil [1](#)

Dropbox [1](#)

Durchführen

Audit [1](#)

Durchlaufzeit (lead time) [1](#)

Durchschnittliche Antwortgeschwindigkeit, als Metrik [1](#)

Durchschnittliche Bearbeitungszeit, als Metrik [1](#)

Durchsetzungsfähigkeit, Scrum Master [1](#)

E

eBay [1](#)

Effektivität

Daily Scrums [1](#)

Effizienz des Umfelds [1](#)

Eigentumswohnungen

als Verbindlichkeit [1](#)

Eigenverantwortung

mehr durch Scrum [1](#)

Einfachheit

Gesundheitswesen [1](#)

Einstein, Albert [1](#)

Einstellung [1](#)

Elektrofotografie [1](#)

Eliot, T.S. [1](#)

Elwer, Pat [1](#)

Emergent Architecture [1](#)

Emotionale Intelligenz [1](#)

Energieeffizienz [1](#)

Engagement, Scrum-Wert [1](#)

Enterprise Architect, SAFe [1](#)

Entscheidungsfreudigkeit [1](#)

Entwickelt, Faktor der Definition of Done [1](#)

Entwickler

einstellen [1](#)

Entwicklung

Defekte während der [1](#)

Hardware- [1](#)

als Schritt beim Erstellen von Anforderungen [1](#)

Entwicklungsdauer (cycle time) [1](#)

Entwicklungsgeschwindigkeit [1](#)

Entwicklungsteam

als Scrum-Rolle [1](#)

Cross-Funktionalität [1](#)

dedizierte [1](#)
Herausforderungen [1](#)
Kollokation [1](#)
Schätzung des Produkt-Backlogs [1](#)
Scrum of Scrums [1](#)
Selbstorganisation [1](#)
Selbstverwaltung [1](#)
Sprint-Review-Prozess [1](#)
und Scrum [1](#)
Verantwortlichkeit [1](#)
warum sie Scrum lieben [1](#)
Entwicklung und Betrieb [1](#)
Entwurf, als Schritt beim Erstellen von Anforderungen [1](#)
Epic Owner, SAFe [1](#)
Epics [1](#)
Equifax [1](#)
Ereignisse
 verwässerte [1](#)
"Erfolgsfaktoren
 und Implementierungsstrategie [1](#)
Erfolgskontrolle [1](#)
Ergebnissicherheit [1](#)
ERP, Enterprise Resource Planning [1](#)
Estimable, Merkmale für User Stories [1](#)
Executive Action Team (EAT) [1](#), [2](#)
Executive Meta Scrum (EMS) [1](#)–[2](#)
Externe Kräfte [1](#)
Externe Unterbrechungen [1](#)

eXtreme Programming [1](#), [2](#)

F

Facebook [1](#), [2](#), [3](#)

Familien

Alltagspflichten [1](#)

Anpassung [1](#)

mit Scrum kommunizieren [1](#)

Projektplanung [1](#)

Strategie festlegen [1](#)

Taskboard [1](#)

Überprüfung [1](#)

Vision festlegen [1](#)

Familienleben

Definition of Done [1](#)

Fokussierung versus Multitasking [1](#)

Herausforderungen [1](#)

Partnersuche mit Scrum [1](#)

Release-Planung [1](#)

Schlafengehen der Kinder [1](#)

Sprint-Planung [1](#)

Teamvereinbarung [1](#)

Thrashing [1](#)

Vision festlegen [1](#)

FBI [1](#)

Features [1](#)

Feedback

Benutzer- [1](#)

Entwicklung Videospiele [1](#)

Gesundheitswesen [1](#)

Medienlandschaft [1](#)

Prozess [1](#)

Stakeholder [1](#)

Feedbackschleife [1](#)

Kundendienst [1](#)

testen in [1](#)

Feedbackzyklus

Medienlandschaft [1](#)

Fehlerrate [1](#)

Feiler, Bruce [1](#)

Fertigung [1](#)

Internet of Things [1](#)

schnellste Markteinführung [1](#)

Shareholder-Value [1](#)

strategisches Kapazitätsmanagement [1](#)

Festlegen

Familienstrategie [1](#)

Projektvision im Familienleben [1](#)

Fibonacci-Folge [1](#)

Finanzen

Budgetplanung [1](#)

Einführung [1](#)

Kreativkultur [1](#)

minimum viable product [1](#)

Rechnungslegung [1](#)

Verschwendung [1](#)

Finanzielle Freiheit [1](#)

Finanzierung inkrementelle [1](#)

Fist of Five [1](#)

Fitnessziele [1](#)

Flexibilität

Entwicklung Videospiele [1](#)

Softwareentwicklung [1](#)

Fluktuation [1](#)

im Scrum-Team [1](#)

im Unternehmen [1](#)

Fly-over-Projekt [1](#)

Fokusgruppen [1–2](#)

Fokus, Scrum-Wert [1](#)

Fokussierung

versus Multitasking [1](#)

Follow-up, Verkaufstrichter [1](#)

Ford, Henry [1](#)

Ford Motor Co. [1](#)

Formalitäten

widerstehen [1](#)

Fortsetzungsgeschichten [1](#)

Fotokopierer [1](#)

Frameworks

3-3-5 [1](#)

5-6-7 [1](#)

Scrum [1](#)

Frühes Lernen [1](#)

Fuji-Xerox [1](#)

Fulgham, Chad [1](#)

Funktionaler Test [1](#)

G

General Electric [1](#)

Genoms [1](#)

Geschäftsergebnis, besseres als Vorteil der Nutzung des agilen Frameworks [1](#)

Geschäftswert (W) [1](#)

Geschwindigkeit [1](#)

Gespräche

persönliche [1](#)

Gesundheitswesen

Bildgebung [1](#)

Fehler minimieren [1](#)

Herstellung medizinischer Geräte [1](#)

Kostensenkung [1](#)

Qualitätssicherung [1](#)

schnelle Markteinführung [1](#)

Sicherheit medizinischer Geräte [1](#)

verbesserte Qualität [1](#)

Vorschriften einhalten [1](#)

Getestet, Faktor der Definition of Done [1](#)

Gewaltenteilung [1](#)

Gewichtsziele [1](#)

Gewinn- und Verlustpotenzial [1](#)

Gilden [1](#)

Global Positioning System [1](#)

Gmail [1](#)

Google [1](#), [2](#)

GORE-TEX [1](#)

GPS [1](#)

Green Berets [1](#)

Größe

Releases [1](#)

Größenbestimmung, relative [1](#)

H

Haftnotizen [1](#)

Halbmarathon [1](#)

Ha-Phase, Shu-Ha-Ri-Konzept [1](#)

Hardware-Entwicklung [1](#)

Beispiele [1](#)

Harvard Business Review [1](#)

Häufigkeit

Releases [1](#)

Häuser

als Verbindlichkeit [1](#)

Hawthorne-Effekt [1](#)

Hawthorne Works [1](#)

Healthcare.gov [1](#)

Herausforderungen

Bildungswesen [1](#)

erkennen [1](#)

Familienalltag [1](#)

für Entwicklungsteams [1](#)

IT-Sicherheit [1](#)

Portfoliomanagement [1](#)

und Implementierungsstrategie [1](#)

Hier herunterladen, Schaltfläche [1](#)

Hochzeitsplanung [1](#)

Holokratie [1](#)

Honda [1](#)

Howey, Hugh [1](#)

Human Resources [1](#)

Anreize schaffen [1](#)

benötigte Fähigkeiten [1](#)

leistungsschwache Teammitglieder [1](#)

Organisationsstrukturen [1](#)

richtige Kultur schaffen [1](#)

Vergütung [1](#)



IaaS, Infrastructure as a Service [1](#)

Identifizieren

Herausforderungen [1](#)

Pilotprojekt [1](#)

Talente [1](#)

Implementierung, schnellere [1](#)

Implementierungsstrategie [1](#)

Incentives [1](#)

Independent, Merkmale für User Stories [1](#)

InfoQ [1](#)

Informationsüberflutung [1](#)

Inkrementelle Finanzierung [1](#)

Inkrementell releasen [1](#)

Inkrement, Produkt- [1](#)

Innovation

Kultur [1](#)

versus Stabilität [1](#)

Innovationskultur [1](#)

Integration

als Schritt beim Erstellen von Anforderungen [1](#)

Integrationsteam [1–2](#)

Integriert, Faktor der Definition of Done [1](#)

Intel [1](#)

Intelligenz, emotionale [1](#)

International Conference on Object-Oriented Programming,
Systems, Languages, and Applications [1](#)

Internet of Things (IoT) [1](#)

Internet-Ressourcen

Agile Alliance [1](#)
Agiles Manifest [1](#)
Agile Programmierung – für Ihre Familie [1](#)
Agile Projektentwicklung bei Intel [1](#)
Agiles Manifest [1](#)
Agiles Marketing-Manifest [1](#)
Dan Pinks TED Talk [1](#)
Healthcare.gov [1](#)
InfoQ [1](#)
Large-Scale Scrum (LeSS) [1](#)
Platinum Edge [1](#)
Rückwärts rennen, Webinar [1](#)
Scaled Agile Framework (SAFe) [1](#)
Schätzpokerkarten [1](#)
Scrum Alliance [1](#), [2](#)
Scrum Guide [1](#)
Scrum Guides [1](#)
Scrum Inc. [1](#)
[Scrum.org](#) [1](#)
ScrumPLoP [1](#)
Studienschulden USA [1](#)
Traning für Halbmarathon [1](#)
Intrapreneurship [1](#)
INVEST-Modell [1](#)
IoT, Internet of Things [1](#)
Irreversible Kosten [1](#)
iSense Prowareness [1](#)
Iterativer Prozess [1](#)
Iterative Unterrichtsmodelle [1](#)

IT-Management und Operations

Babyboomer-Ruhestands-Lücke [1](#)

Big Data [1](#)

Data Warehouses [1](#)

Dienstleistung versus Kontrolle [1](#)

Gewinn- und Verlustpotenzial [1](#)

Innovation versus Stabilität [1](#)

Unternehmensressourcenplanung [1](#)

IT-Sicherheit [1](#)

J

JAS 39E Saab Gripen [1](#)

Johns Hopkins CubeSat [1](#)

Johnson, Spencer [1](#)

J.P. Morgan [1](#)

J Sainsbury PLC (UK) [1](#)

K

Kanban

und IT-Management [1](#)

Kanban-Board [1](#)

Kanbantafeln

Alltagspflichten in Familie [1](#)

Kapazität

Sprint-Backlog [1](#)

Kapitalrendite, erhöht durch Scrum [1](#)

Kapitalumschichtung [1](#)

Kegel der Unsicherheit [1](#)

Kennzahlen

Defekte [1](#)
Fluktuation im Scrum-Team [1](#)
Fluktuation im Unternehmen [1](#)
Kapitalumschichtung [1](#)
Kompetenzen [1](#)
Produkteinführungszeit [1](#)
Qualifiktionen [1](#)
Return on Investment [1](#)
Sprintziel-Erfolgsquote [1](#)
Time to Market [1](#)
Überblick [1](#)
Verhältnis Manager-Produktersteller [1](#)
Zufriedenheit [1](#)

Klarheit

als Ergebnis der Verfeinerung des Produkt-Backlogs [1](#)

Klassengröße [1](#)

Klassenzimmer [1](#)–[2](#)

Klein, Merkmale für User Stories [1](#)

Kleinstmöglich durchführbare Produkt [1](#)

Kognitive Konsistenztheorie [1](#)

Kollokation [1](#), [2](#)

Kommunikation

Familie [1](#)
Herausforderung beim Dating [1](#)
mit Scrum [1](#)
Qualität [1](#)
und Beziehungen [1](#), [2](#)
und Familien [1](#)
Wikispeed, modulares Auto [1](#)

Kontinuierliche Integration [1](#), [2](#)

Kontinuierliche Verbesserung [1](#)

Konversion, Verkaufstrichter [1](#)

Konvertierung von Legacy-Systemen [1](#)

Konzentration [1](#)

Körpersprache [1](#)

Kosten

irreversible [1](#)

reduzieren durch inkrementelle Finanzierung [1](#)

Kosteneinsparungen, Videospiele [1](#)

Kostensenkung

Gesundheitswesen [1](#)

Kräfte

externe [1](#)

Kreativität

motivieren zu [1](#)

Kreativkultur [1](#)

Kultur

Innovations- [1](#)

Kreativ- [1](#)

Unternehmens- [1](#)

Kumulative Flussdiagramme [1](#)

Kunden

beteiligen [1](#)

Informationsüberflutung [1](#)

wichtigste Stakeholder [1](#)

Kundenbeteiligung [1](#)

Kundendienst

Definition of Done [1](#)

Feedbackschleife [1](#)

Informationsüberflutung [1](#)

Metriken [1](#)

Produkt-Backlog [1](#)

Shadowing [1](#)

Wahrnehmung durch Kunden [1](#)

Kunden-Feedback

Wikispeed, modulares Auto [1](#)

Kundenservice [1](#)

Kundenspezifische Projekte [1](#)

Kundenzufriedenheit

als Metrik [1](#)

größer durch Scrum [1](#)

Kurskorrektur [1](#)

L

Länge

Sprints [1](#)

Lao Tzu [1](#)

Large-Scale Scrum (LeSS) [1](#)

kleineres Framework [1](#)

LeSS Huge Framework [1](#)

Website [1](#)

Larsen, Diana [1](#)

Lasttest [1](#)

Last-Test [1](#)

Leads, Verkaufstrichter [1](#)

lead time [1](#)

Lean [1](#)

Lean-agile Führungskräfte [1](#)

Lean Startup [1](#)

Lean-Startup-Modell [1](#)

Lebensziele

Ausbildung [1](#)

Fitness [1](#)

Gewichtsabnahme [1](#)

Reisen [1](#)

Ruhestand [1](#)

Schule [1](#)

Wohlbefinden [1](#)

Lebenszyklus, Sprints [1](#)

Legacy-Anwendungen [1](#)

konvertieren [1](#)

Lehrplan [1](#)

Leistungsbeurteilungen [1](#)

Leistungstest [1](#), [2](#)

Lernen

Antimuster [1](#)

Bedürfnis nach Sicherheit [1](#)

externe Kräfte [1](#)

Feedbackschleife [1](#)

frühes [1](#)

Innovationskultur [1](#)

Kurskorrektur während des Fluges [1](#)

Testen in der Feedbackschleife [1](#)

Transparenz [1](#)

Unterstützung für [1](#)

Veränderungen positiv sehen [1](#)

Leser ändern sich [1](#)

LeSS Huge-Framework [1](#)

LeSS, Large-Scale Scrum [1](#), [2](#)

kleines Framework [1](#)

Lieferverzögerung [1](#)

Like-Button [1](#)

Lockheed Martin [1](#)

Los Angeles Department of Water and Power [1](#)

M

Macht des Product Owners [1](#)

Malware [1](#)

Mangelnde Ausbildung [1](#)

Mangelnde Schulung [1](#)

Marketing

Entwicklungen [1](#)

Scrum im [1](#)

Scrum in Aktion [1](#)

und soziale Medien [1](#)

Markteinführung

schnelle [1](#)

Markteinführung, schnelle

Gesundheitswesen [1](#)

Markteinführungszeit [1](#)

verkürzt durch Scrum [1](#)

Marshmallow Challenge, TED Talk [1](#)

Marshmallow-Test [1](#)

Materielle Güter [1](#)

Baugewerbe [1](#)

Fertigung [1](#)

Hardware-Entwicklung [1](#)

Wasserfall [1](#)

Wohnungsbau [1](#)

McDonald's [1](#)

Medien [1](#)

Medizinische Geräte, Herstellung und Sicherheit [1](#)

Medizintechnik [1](#)

Mehrabian, Albert [1](#), [2](#)

Mehr Schüler pro Lehrer [1](#)

Meinungsführer [1](#)

Mentoren

Scrum- [1](#)

Mentoring-Programm [1](#)

Meta-Scrum [1](#)

Chief Product Owner (CPO) [1](#)

Metriken [1](#)

für Scrum [1](#)

Kundendienst [1](#)

neu definieren [1](#)

Microchip [1](#)

Microsoft Office [1](#)

Migration

große Datenmengen [1](#)

Militär und Scrum [1](#)

Minuten Manager [1](#)

Mission Command [1](#)

Mitarbeiterbindung, bessere [1](#)

Mitarbeiterzufriedenheit, größere als Vorteil der Nutzung des agilen Frameworks [1](#)

Mitarbeiterzuordnung [1](#)

Mitwirkende Persönlichkeit [1](#)

MMBD, Multi-Mission Bus Demonstrator [1](#)

Modell für Veränderungen nach Satir [1](#)

Mögliche Herausforderungen in der Implementierungsstrategie [1](#)

Monetarisierung, schnellere [1](#)

Moore, Geoffrey [1](#)

Mortensen, Brady [1](#)

Möwen-Management [1](#)

Multitasking [1](#)

versus Fokussierung [1](#)

Musikindustrie [1](#)

Mut, Scrum-Wert [1](#)

MVP-Konzept, Dropbox [1](#)

MVP, Minimum Viable Product [1](#), [2](#), [3](#)

N

Nachrichtenmedien [1](#)

Definition of Done [1](#)

Scrum-Teams [1](#)

Sprint-Flexibilität [1](#)

NASA [1](#)

National Public Radio (NPR) [1](#)

Native Advertisement [1](#)

Navy Seals [1](#)

Negotiable, Merkmale für User Stories [1](#)

Nenner [1](#)

New New Product Development Game [1](#)

Nicht abgenommene Anforderungen [1](#)

Nicht fertiggestellte Anforderungen [1](#)

Nonverbale Signale [1](#)

NPR, National Public Radio [1](#)

O

Offenheit, Scrum-Wert [1](#)

OOPSLA 95 (International Conference on Object-Oriented Programming, Systems, Languages, and Applications) [1](#)

Open-Source-Hardware [1](#)

Operations und IT-Management

Babyboomer-Ruhestands-Lücke [1](#)

Big Data [1](#)

Data Warehouses [1](#)

Dienstleistung versus Kontrolle [1](#)

Gewinn- und Verlustpotenzial [1](#)

Innovation versus Stabilität [1](#)

Migration großer Datenmengen [1](#)

Unternehmensressourcenplanung [1](#)

Organisationen

Flukuation im Unternehmen [1](#)

Organisationsstrukturen [1](#)

Overhead

Eintragstyp Produkt-Backlog [1](#), [2](#)

P

Paarprogrammierung [1–2](#)

im Studium [1](#)

PaaS, Platform as a Service [1](#)

Paralyse durch Analyse [1](#)

Partnersuche [1](#)

Pattern Languages of Programs (PLoP) [1](#)

PayPal/TIO [1](#)

Peer-to-Peer-Kommunikation

Wikispeed, modulares Auto [1](#)

Personalbeschaffung [1](#)

Personal einstellen [1](#)

Personalwesen [1](#)

Personas [1](#), [2](#)

Persönliche Gespräche, Bedeutung [1](#)

Persönliche Unterbrechungen [1](#)

Persönliche Verantwortung

im Familienalltag [1](#)

Petraeus, David [1](#)

Pharmaunternehmen [1](#)

Phase II, Sprint-Planung [1](#)

Phase I, Sprint-Planung [1](#)

Phasenweiser Ansatz [1](#)

Phishing [1](#)

Pilotprojekt

identifizieren [1](#)

Klarheit [1](#)

loslegen [1](#)

Pink, Dan [1](#)

Planen

Daily Scrum [1](#)

Hochzeit [1](#)

Reisen [1](#)

und Prioritätensetzung [1](#)

Platinum Edge

Website [1](#)

Platinum Edge Dienstleistungen [1](#)

Platinum Edge Estimation Poker [1](#)

Platinum-Prinzipien [1](#)

PlayStation [1](#)

PMI Agile Certified Practitioner (PMI-ACP) [1](#)

Pomodoro-Technik [1](#)

Portfoliomanagement [1](#)

Herausforderungen [1](#)

Large-Scale Scrum (LeSS) [1](#)

Lean Startup [1](#)

Scrum-at-Scale-Ansatz [1](#)

Scrum für große Portfolios skalieren [1](#)

Scrum-of-Scrums-Modell [1](#)

vertikaler Schnitt [1](#)

Vorteile des SAFe-Modells [1](#)

Work-in-Progress-Grenzwerte [1](#)

Postproduktion, Entwicklungsstufe Videospiele [1](#)

Präproduktion, Entwicklungsstufe Videospiele [1](#)

Pre-Silizium-Infrastruktur und -Readiness [1](#)

Prinzipien

agile [1](#)

für Projekte [1](#)

Platinum- [1](#)

Priorisieren

Anforderungen auf der Roadmap [1](#), [2](#)

Priorisierung

Familienprioritäten [1](#)

Product Owner

- als Scrum-Rolle [1](#)
- Baugewerbe [1](#)
- einstellen [1](#)
- Hauptverantwortlichkeiten [1](#)
- ineffektive [1](#)
- Large-Scale Scrum [1](#)
- LeSS Huge [1](#)
- Macht des [1](#)
- Nachrichtenmedien [1](#)
- Priorisierung der Anforderungen [1](#)
- Scrum at Scale [1](#)
- Scrum of Scrums [1](#)
- Sprint-Review-Prozess [1](#)
- Verantwortlichkeit [1](#)
- warum sie Scrum lieben [1](#)

Product Owner Agent [1](#)

Produkt-Backlog

- Arten von Einträgen [1](#)
- Ausarbeitung [1](#)
- Definition [1](#)
- Einführung [1](#)
- in Large-Scale Scrum (LeSS) [1](#)
- Kundendienst [1](#)
- Nachrichtenmedien [1](#)
- Schätzung [1](#)
- Umfangssteuerung [1](#)
- User Stories [1](#)

Produkte

- Medien [1](#)

Produkteinführungszeit [1](#)

Produktfehler, weniger [1](#)

Produktinkrement [1](#)

Produktion

Entwicklungsstufe Videospiele [1](#)

Produkt-Release

Anpassung während [1](#)

Größe anpassen [1](#)

häufig [1](#)

Produkt-Roadmap [1](#)

Aufwandsschätzung [1](#)

Ausrichtung am gleichen Ziel [1](#)

einfache Werkzeuge [1](#)

langfristige Perspektive [1](#)

Schätzung [1](#)

Schritte zur Erstellung [1](#)

Umfangssteuerung [1](#)

Zeitraumen festlegen [1](#)

Professional Scrum Developer (PSD) [1](#)

Professional Scrum Master (PSM I, II, III) [1](#)

Professional Scrum Product Owner (PSPO I, II) [1](#)

Projektausfall, durch Scrum minimiert [1](#)

Projekte

kundenspezifische [1](#)

Projektmanager, Baugewerbe [1](#)

Projektplanung

Anforderungen unterteilen [1](#)

Begrifflichkeiten [1](#)

Familie [1](#)

häufig verwendete Techniken [1](#)

Produkt-Backlog [1](#)

Produkt-Roadmap [1](#)

Überblick [1](#)

Projektsteuerung

durch Scrum verbessert [1](#)

Prozesse

verwässerte [1](#)

Prüfbar, Merkmale für User Stories [1](#)

Pull-Mechanismus [1](#)

Push-Modelle [1](#)

Q

Qualifikationen, vielseitige [1](#)

Qualität

bessere durch Scrum [1](#)

Gesundheitswesen [1](#)

Kommunikation [1](#)

Qualitätssicherung

Gesundheitswesen [1](#)

R

Reduzieren

Single Points of Failure [1](#)

Refactoring [1](#)

Medienbranche [1](#)

Regulatorischen Anforderungen [1](#)

Regulatorische Vorschriften [1](#)

Reiseplanung [1](#)

Relative Größenbestimmung [1](#)

Relative Schätzung [1–2](#)

Release-Defekte [1](#)

Release-Plan

Schritte für [1](#)

Ziele [1](#)

Release-Planung

Aufwandsschätzung [1](#)

Familien [1](#)

Releases

Anpassung während der [1](#)

Größe anpassen [1](#)

Häufigkeit [1](#)

marktgerechte [1](#)

Nachrichtenmedien [1](#)

Release-Sprints [1](#)

Release-Ziele [1](#)

emotionales Wohlbefinden [1](#)

Studium [1](#)

Republican National Committee [1](#)

Requirement Area, LeSS Huge [1](#)

Respekt, Scrum-Wert [1](#)

Return on Investment (ROI) [1](#)

Kennzahlen [1](#)

Ri-Phase, Shu-Ha-Ri-Konzept [1](#)

Risiken

Früherkennung [1](#)

identifiziert als Ergebnis der Verfeinerung des Produkt-Backlogs [1](#)

Risiko

durch Scrum reduziert [1](#)

Roadmap to Value

Dating [1](#)

emotionales Wohlbefinden [1](#)

Fitness [1](#)

Gewichtsabnahme [1](#)

Schule und Studium [1](#)

Überblick [1](#)

und Kundendienst [1](#)

Verkaufsprozess [1](#)

Rollen

außerhalb von Scrum [1](#)

im Baugewerbe [1](#)

Product Owner [1](#)

Product Owner Agent [1](#)

Scrum [1](#)

Scrum Master [1](#)

Scrum-Mentoren [1](#)

Stakeholder [1](#)

Rückkopplungsschleife [1](#)

Rückwärts rennen [1](#)

Rückwärts rennen, Webinar [1](#)

Ruhestand

finanzielle Freiheit sichern [1](#)

Finanzplanung [1](#)

Schuldentilgung [1](#)

sparen für Notfall [1](#)

S

Saab-Mehrzweckkampfflugzeug [1](#)

SaaS, Software as a Service [1](#), [2](#), [3](#)

SAFe, Scaled Agile Framework [1](#), [2](#)

Portfolioebene [1](#)

Programmebene [1](#)

Teamebene [1](#)

Vorteile [1](#)

Website [1](#)

Satir, Virginia [1](#)

Säulen der Verbesserung [1](#)

Scaffolding [1](#)

Schall und Rauch [1](#)

Schätzbar, Merkmale für User Stories [1](#)

Schätzpoker [1](#)

Schätzung

Affinitäts- [1](#)

Gebräuchliche Schätztechniken [1](#)

Produkt-Backlog [1](#)

relative [1](#)

Schätzpoker [1](#)

Schreiben vs. visualisieren [1](#)

Schritt-für-Schritt-Plan, in Implementierungsstrategie [1](#)

Schuldentilgung [1](#)

Schule [1](#)

Schulung

mangelnde [1](#)

Schwaber, Ken [1](#), [2](#)

Scope Creep [1](#), [2](#)

Scrum

allgemeine Vorteile [1](#)

als Framework [1](#)

Anwendung in Medienbranche [1](#)

Artefakte [1](#)

Beispielansatz im Klassenzimmer [1](#)

Budgetplanung [1](#)

Dating [1](#)

Entwicklungsteams [1](#)

Ereignisse [1](#)

für emotionales Wohlbefinden [1](#)

für Fitnessziele [1](#)

für Gewichtsabnahme [1](#)

für große Portfolios skalieren [1](#)

für Marketing [1](#)

für Schule und Studium [1](#)

für Vertrieb [1](#)

Grundlagen [1](#)

Hochzeitsplanung mit [1](#)

Kommunikation mit [1](#)

Marketing [1](#)

Mentoren [1](#)

Personal einstellen [1](#)

Reiseplanung [1](#)

Ressourcen [1](#)

Rollen [1–2](#)

Rollen im Baugewerbe [1](#)

Teamarbeit [1](#)

Teams [1](#)

Überblick [1](#)

Umstellung auf [1](#)

und Dating [1](#)
und Organisationsstrukturen [1](#)
und Softwareentwicklung [1](#)
und soziale Medien [1](#)
unechtes [1](#)
Ursprünge [1](#)
User-Communities [1](#)
Verkaufsprozess [1](#)
Werte [1](#), [2](#)
Zertifizierungen [1](#)

Scrum Alliance [1](#), [2](#)

Scrum at Scale [1](#)

Scrum-at-Scale-Modell [1](#)

Product Owner skalieren [1](#)
Scrum Master skalieren [1](#)
Synchronisation [1](#)
Website [1](#)

Scrum-Coach [1](#)

Scrum-Frameworks

3-3-5 [1](#)
5-6-7 [1](#)
LeSS Huge [1](#)
LeSS, Large-Scale Scrum [1](#)
über [1](#)

Scrum Guide [1](#)

Scrum Guide, Website [1](#)

Scrum-Lösungen [1](#)

Scrum Master [1](#)

als Scrum-Rolle [1](#)

als Servant Leader [1](#)

Baugewerbe [1](#)

Durchsetzungsfähigkeit [1](#)

einstellen [1](#)

Nachrichtenmedien [1](#)

Scrum at Scale [1](#)

Scrum of Scrums [1](#)

Servant Leader [1](#)

Sprint-Retrospektive [1](#)

Sprint-Review-Prozess [1](#)

Verantwortlichkeit [1](#)

warum sie Scrum lieben [1](#)

Scrum of Scrums

Entwicklungsteam [1](#)

Scrum-of-Scrums-Modell [1](#)

Entwicklungsteam [1](#)

Product Owner [1](#)

Scrum Master [1](#)

Scrum of Scrums of Scrums [1](#), [2](#)

ScrumPLoP [1](#)

Scrum-Rollen

verwässerte [1](#)

Scrum-Teams

Shu-Ha-Ri-Konzept des Erlernens [1](#)

vielseitige Qualifikationen [1](#)

Selbstorganisation [1](#)

Selbstverlag [1](#)

Selbstverwaltung [1](#)

Semesterwochenstunden [1](#)

Servant Leader

Scrum Master als [1](#)

Services [1](#)

Shadowing [1](#)

Kundendienst [1](#)

Shareholder-Value [1](#)

SHIFT Media Manager [1](#)

Shu-Ha-Ri [1](#)

Shu-Phase, Shu-Ha-Ri-Konzept [1](#)

Sicherheit

Bedürfnis nach [1](#)

finanzielle Freiheit [1](#)

Sicherheitsherausforderungen [1](#)

Sicherheitstest [1](#)

Sichtbarkeit [1](#)

Sprint-Planung für Familien [1](#)

verbessert durch Scrum [1](#)

Silo (Roman) [1](#)

Silo-Teams [1](#)

Single Points of Failure [1](#), [2](#)

Skalieren [1](#)

Skalierungstests [1](#)

Skillman, Peter [1](#)

Small, Merkmale für User Stories [1](#)

Social Media [1](#)

Softwareanwendungen [1](#)

Softwareentwicklung

Flexibilität [1](#)

Refactoring [1](#), [2](#)

und Scrum [1](#)

Sony [1](#)

Sozialkompetenz [1](#)

Sparen

für Notfälle [1](#)

Spaßfaktoren

Entwicklung Videospiele [1](#)

Spätere Auslieferung [1](#)

Spotify [1](#)

Sprint-Backlog

Arbeiten mit [1](#)

Burndown-Diagramme [1](#)

Daily Scrum [1](#)

Informationen in [1](#)

Kapazität festlegen [1](#)

Los Angeles Department of Water and Power [1](#)

Push versus Pull [1](#)

Sprints priorisieren [1](#)

Sprintdauer

Nachrichtenmedien [1](#)

Sprintlängen

bestimmen [1](#)

Sprint-Planung [1](#)

Aufwandsschätzung [1](#)

Familien [1](#)

Kanban [1](#)

Kundendienst [1](#)

Large-Scale Scrum (LeSS) [1](#)

Lean Startup [1](#)

LeSS Huge [1](#)

Nachrichtenmedien [1](#)

Phase I [1](#)

Phase II [1](#)

Sprint-Retrospektive [1](#)

Derby-und-Larsen-Prozess [1](#)

Ergebnisse [1](#)

Gesundheitswesen [1](#)

in Kanban [1](#)

Kundendienst [1](#)

Prozess [1](#)

Scrum Master [1](#)

Überprüfung und Anpassung [1](#)

Ziele [1](#)

Sprint-Retrospektiven

Large-Scale Scrum (LeSS) [1](#)

Lean Startup [1](#)

LeSS Huge [1](#)

Subunternehmer beteiligen [1](#)

Wichtigkeit von [1](#)

Sprint-Review [1](#)

in Kanban [1](#)

Kundendienst [1](#)

Large-Scale Scrum (LeSS) [1](#)

Lean Startup [1](#)
LeSS Huge [1](#)
Nachrichtenmedien [1](#)
Product Owner [1](#)
Produktinkremente [1](#)
Prozess [1](#)
Schule/Bildungswesen [1](#)
Scrum Master [1](#)
Stakeholder [1](#)
Stakeholder-Feedback [1](#)
Subunternehmer beteiligen [1](#)
Wichtigkeit von [1](#)
Wikispeed, modulares Auto [1](#)

Sprints

Anzahl schätzen [1](#)
Daily Scrum [1](#)
definieren [1](#)
Flexibilität und Nachrichtenmedien [1](#)
Komponenten [1](#)
Länge planen [1](#)
Lebenszyklus [1](#)
priorisieren [1](#)
Release- [1](#)
Schule/Bildungswesen [1](#)
Sprint-Retrospektive [1](#)
Sprint-Review [1](#)
Team-Taskboard [1](#)
Wikispeed, modulares Auto [1](#)
Ziele [1](#)

Sprintziel

Erfolgsquoten [1](#)

Stabilität

versus Innovation [1](#)

Stakeholder [1](#)

Feedback von [1](#)

Kunden als wichtigste [1](#)

Rolle [1](#)

Sprint-Review-Prozess [1](#)

Startup-Firma [1](#)

Story Points [1](#)

und Velocity [1](#), [2](#)

Strategisches Kapazitätsmanagement [1](#)

Stress reduzieren [1](#)

Studienschulden [1](#)

Subunternehmer [1](#)

Sunk Costs [1](#)

Sutherland, Jeff [1](#), [2](#), [3–4](#)

Swarming [1](#), [2](#)

Vorteile [1](#)

Synchronisation, Scrum at Scale [1](#)

Systemtest [1](#)

T

Talente identifizieren [1](#)

Taskboard [1](#)

Alltagspflichten in Familie [1](#)

Familie [1](#)

Hochzeitsplanung [1](#)

Tasks [1](#)

Tatsächliche Kosten (TK) [1](#), [2](#), [3](#)

Teamarbeit [1](#)

Teambildung

Reisen [1](#)

Studium [1](#)

Teammitglieder

schlechte Auswahl [1](#)

Team moral

besser durch Scrum [1](#)

Teams

Arbeitsmoral [1](#)

denken und handeln als [1](#)

Executive Action [1](#)

Incentives auf Teamebene [1](#)

John Hopkins CubeSat [1](#)

Leistungsbeurteilungen [1](#)

leistungsschwache Teammitglieder [1](#)

Nachrichtenmedien [1](#)

neue bilden [1](#)

Scrum-of-Scrums-Modell [1](#)

Selbstorganisation [1](#)

selbstverwaltende [1](#)

Titel [1](#)

Work-in-Progress-Grenzwerte [1](#)

Zufriedenheit [1](#)

Team-Taskboard [1](#)

Swarming [1](#)

Umgang mit nicht abgenommenen Anforderungen [1](#)

Umgang mit nicht fertiggestellten Anforderungen [1](#)

Teamunterbrechungen [1](#)

Teamvereinbarung

Familie [1](#)

Telefonica Digital [1](#)

Tesla [1](#)

Testable, Merkmale für User Stories [1](#)

test-driven development, TDD [1](#)

Testen

in der Feedbackschleife [1](#)

Testgetriebene Entwicklung [1](#), [2](#)

Tests

als Schritt beim Erstellen von Anforderungen [1](#)

The Marshmallow Challenge, TED Talk [1](#)

Themen [1](#)

Themes [1](#)

The New New New Product Development Game [1](#)

The Waste Land [1](#)

Thrashing [1](#), [2](#), [3–4](#), [5](#)

Familien [1](#)

Titel [1](#)

Toyota [1](#), [2](#)

Traceability Matrix [1](#)

Training

Halbmarathon [1](#)

Transparenz [1](#)

Baugewerbe [1–2](#)

Gesundheitswesen [1](#)

mangelnde über Ziele [1](#)

mehr durch Scrum [1](#)

Sprint-Planung für Familien [1](#)

Wikispeed, modulares Auto [1](#)

Transparenz, agile Säule [1](#)

TTD, test-driven development [1](#)

Twitter, Tweets mit Fotos [1](#)

U

Über [1](#)

Übergangsteam

mobilisieren [1](#)

Überprüfung

agile Säule [1](#)

für Familien [1](#)

Umfangsbestimmung

als Ergebnis der Verfeinerung des Produkt-Backlogs [1](#)

Umfeld

Effizienz maximieren [1](#)

motivierendes [1](#)

unangemessenes [1](#)

Umfragen [1](#)

Zufriedenheit [1](#)

Umgang

mit nicht abgenommenen Anforderungen [1](#)

mit nicht fertiggestellten Anforderungen [1](#)

Umgebung

unangemessene [1](#)

Unabhängig, Merkmale für User Stories [1](#)

Unechtes Scrum [1](#)

Unit-Test [1](#)

Unterbrechungen

durch Teammitglieder [1](#)

externe [1](#)

persönliche [1](#)

Unternehmen

Kultur [1](#)

Strategie [1](#)

Ziele [1](#)

Unternehmensressourcenplanung [1](#)

Unterrichtsmodelle, iterative [1](#)

Unterstützung für das Lernen [1](#)

Unterstützung in Transitionsphase [1](#)

Upfront-Engineering [1](#)

User Stories [1](#)

Epics [1](#)

INVEST-Modell [1](#)

Merkmale [1](#)

V

Valueable, Merkmale für User Stories [1](#)

Velocity [1](#), [2](#)

Veränderungen

Bauprojekte [1](#)

Modell nach Satir [1](#)

positiv sehen [1](#)

Verantwortlichkeit

falsch zugeordnet [1](#)

im Familienalltag [1](#)

Verbesserung

Eintragstyp Produkt-Backlog [1](#)

Verbesserungen

Bedürfnis nach Sicherheit [1](#)

Eintragstyp Produkt-Backlog [1](#)

externe Kräfte [1](#)

Feedbackschleife [1](#)

Hardware-Entwicklung [1](#)

Kurskorrektur während des Fluges [1](#)

Saab-Mehrzweckkampfflugzeug [1](#)

Vergütung [1](#)

Verhaltenstests [1](#)

Verhandelbar, Merkmale für User Stories [1](#)

Verkaufsabschluss, Verkaufstrichter [1](#)

Verkaufsanruf, Verkaufstrichter [1](#)

Verkaufstrichter

Follow-up [1](#)

Konversion [1](#)

Leads [1](#)

Verkaufsabschluss [1](#)

Verkaufsanruf [1](#)

Verkaufszyklen [1](#)

vertical slicing [1](#)

Vertikale Aufteilung [1](#)

Vertikaler Schnitt [1](#)

Verwässerte Prozesse [1](#)

Vision [1](#)

Kundendienst [1](#)

Visualisieren statt schreiben [1](#)

Vorhersagbarkeit [1](#)

Vorschriften [1](#)

Vorteile

und Implementierungsstrategie [1](#)

von Scrum [1](#)

W

Wake, Bill [1](#)

Walmart [1](#)

Warteschlange [1](#)

Warteschlangentheorie [1](#)

Wartung

Code-Refactoring [1](#)

Eintragstyp Produkt-Backlog [1](#), [2](#)

Washington Post, Tageszeitung [1](#)

Wasserfallansatz

erst alles planen, dann entwickeln [1](#)

irreversible Kosten [1](#)

Sturz des [1](#)

Wasserfallartefakte [1](#)

Wasserfallmethode [1](#)

Websites

Agile Alliance [1](#)

Agile Programmierung – für Ihre Familie [1](#)

Agile Projektentwicklung bei Intel [1](#)

Agiles Manifest [1](#)

Healthcare.gov [1](#)

InfoQ [1](#)

Large-Scale Scrum (LeSS) [1](#)

Platinum Edge [1](#)

Rückwärts rennen, Webinar [1](#)

Scaled Agile Framework (SAFe) [1](#)

Schätzpokerkarten [1](#)

Scrum Alliance [1](#), [2](#)

Scrum-at-Scale-Modell [1](#)

Scrum Guide [1](#)

Scrum Guides [1](#)

Scrum Inc. [1](#)

[Scrum.org](#) [1](#)

ScrumPLoP [1](#)

Studienschulden USA [1](#)

Traning für Halbmarathon [1](#)

Werkzeuge, Produkt-Roadmap [1](#)

Wert des Projekts (W) [1](#)

Werte

Agiles Manifest [1](#)

Scrum [1](#)

Wertschöpfungsplan

Überblick [1](#)

Wertvoll, Merkmale für User Stories [1](#)

Wert (W) [1](#)

Wichtigkeit, Pilotprojekt [1](#)

Widerstehen von Formalitäten [1](#)

Wiggins, Alexis [1](#)

Wikispeed [1](#)

WIP (Work in Progress) [1](#), [2](#)

Wohnungsbau [1](#)

Wool Omnibus (Roman) [1](#)

Work-in-Progress-Grenzwerte [1](#)

Work-in-progress-Limits

Dating [1](#)

Work in Progress (WIP)-Limits [1](#)

Wujec, Tom [1](#)

Würmer [1](#)

X

Xerographie [1](#)

Xerox [1](#)

Y

YouTube [1](#)

Z

Zähler [1](#)

Zappos [1](#)

Zerlegen von Anforderungen [1](#)

Zertifizierungen [1](#)

Zettabyte [1](#)

Ziele

- mangelnde Transparenz [1](#)

- Release-Plan [1](#)

- Sprint-Retrospektive [1](#)

- Sprints [1](#)

- Unternehmen [1](#)

Zielgerichtete Entwicklung [1](#)

Zukünftige Prozesse, in Implementierungsstrategie [1](#)

Zusammenarbeit

- bessere durch Scrum [1](#)

WILEY END USER LICENSE AGREEMENT

Besuchen Sie www.wiley.com/go/eula, um Wiley's E-Book-EULA einzusehen.