



Mike Burrows

Kanban

Verstehen, einführen, anwenden

→ Aus dem Englischen übersetzt
von Florian Eisenberg und Wolfgang Wiedenroth



dpunkt.verlag



Mike Burrows ist Geschäftsführer und Principal Consultant von David J. Anderson and Associates (djaa.com). In seiner beruflichen Laufbahn, die sich von der Luftfahrt über das Bankwesen, das Energiewesen bis hin zum öffentlichen Dienst zieht, ist Mike bereits IT-Leiter, globaler Entwicklungsleiter und Softwareentwickler gewesen. Er ist akkreditierter Kanban Trainer (AKT) und Kanban Coaching Professional (KCP) und wird überall auf der Welt eingeladen, um auf Veranstaltungen Vorträge zu halten. Er bloggt auf *positiveincline.com* und twittert als @asplake und @KanbanInside.

Übersetzer:



Florian Eisenberg hilft Unternehmen als Trainer, Berater und Coach auf ihrem Weg zur Agilität. Dabei setzt er am liebsten auf Kanban und unterstützt es durch andere Vorgehensmodelle wie beispielsweise Scrum. Er schätzt an Kanban besonders den respektvollen Umgang mit den Beteiligten und die lösungsoffene Herangehensweise.

Nach seinem Studium der Informatik in Karlsruhe arbeitete er als Programmierer, Scrum Master und Product Owner, bevor er Berater wurde. Er ist einer der ersten akkreditierten Kanban Trainer (AKT) und Kanban Coaching Professionals (KCP). Sein Blog finden Sie auf *florianeisenberg.de*, er twittert als @fjeisenberg.



Wolfgang Wiedenroth ist ausgebildeter Fachinformatiker der Fachrichtung Anwendungsentwicklung und arbeitete insgesamt fünf Jahre als Scrum Master in verschiedenen Bereichen der IT. 2010 entdeckte er Kanban und verliebte sich in dessen erstes Prinzip »Beginne mit dem, was du gerade tust« und wurde ein wahrer Kanban-Enthusiast. Heute arbeitet er bei it-agile als Berater und hilft Unternehmen, mit Kanban ihre Prozesse und Umgebung zu verstehen und zu verbessern. Wolfgang ist akkreditierter Kanban Trainer (AKT) und Kanban Coaching Professional (KCP). Außerdem ist er Co-Founder der Limited WIP Society München. Er bloggt auf *agilemanic.com* und twittert als @wwiedenroth.

Mike Burrows

Kanban

Verstehen, einführen, anwenden

Übersetzt aus dem Englischen
von Florian Eisenberg und Wolfgang Wiedenroth



dpunkt.verlag

Übersetzung: Florian Eisenberg (post@florianeisenberg.de)
Wolfgang Wiedenroth (wolfgang.wiedenroth@it-agile.de)
Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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.

ISBN:
Buch 978-3-86490-253-6
PDF 978-3-86491-805-6
ePub 978-3-86491-806-3
mobi 978-3-86491-807-0

Deutsche Ausgabe der 1. amerikanischen Auflage 2015
Translation Copyright für die deutschsprachige Ausgabe © 2015 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Copyright der amerikanischen Originalausgabe © 2014 by Mike Burrows
Title of American original: Kanban from the Inside
Blue Hole Press · 72 Buckhorn Road · Sequim, WA 98382 · www.blueholepress.com

ISBN 978-0-9853051-9-2

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten.
Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung
des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung,
Übersetzung oder die Verwendung in elektronischen Systemen.
Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie
Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-,
marken- oder patentrechtlichem Schutz unterliegen.
Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor
noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der
Verwendung dieses Buches stehen.

❖ Für Sharon ❖

Geleitwort

Ich bin ein Amerikaner, der in der glücklichen Lage ist, einen ordentlichen Teil der Welt gesehen zu haben. Während meiner Reisen habe ich festgestellt, dass Amerikaner eine einzigartige Vorliebe für das Genre der »Selbsthilfe«-Bücher haben: Amerikaner lieben es, gesagt zu bekommen, dass sie etwas falsch machen und dass das atemberaubende Buch in ihren Händen den geheimen Code darüber enthält, wie sie dünner, schneller, stärker oder sogar schöner werden.

Mike ist Brite und die Welt kann sich glücklich schätzen, dass er sich entschieden hat, dieses Buch zu schreiben. Denn das Letzte, was wir brauchen, ist ein weiteres »Selbsthilfe-Software-Methode«-Buch, in dem der Autor verspricht, Entwicklungsorganisationen den geheimen Code dafür zu geben, wie ihre Leute besseren Programmcode schreiben, der besser zu den Kunden-, Stakeholder- und Marktbedürfnissen passt, mit weniger Fehlern und den aktuellsten Technologien. Das bedeutet bedauerlicherweise, dass einige Amerikaner – und möglicherweise auch einige andere, die von der Schule der »Selbsthilfe« genährt wurden – dieses Buch meiden könnten.

Das wäre äußerst schade.

In diesem ausführlichen Buch gibt uns Mike keinen Rat zur »Selbsthilfe«. Stattdessen gibt er uns etwas, was noch viel, viel mächtiger und beständiger ist: Werkzeuge und Prozesse zur Selbsterkenntnis.

Da Erkenntnis die Fähigkeit ist, das wirkliche Wesen einer Sache zu verstehen, ist Selbsterkenntnis die Fähigkeit, uns selbst zu verstehen. Aus der Perspektive dieses Buches meint das »Selbst« nicht uns als Individuen, sondern unser »Organisations-Selbst« – unser Unternehmen oder Entwicklungsteam oder unsere Entwicklungsorganisation.

Die mächtigen Werkzeuge, die Mike beschreibt – beispielsweise das Kanban-Board oder das Cumulative Flow Diagram –, werden auf der Grundlage einer Reihe von Werten eingeführt, die uns wirklich umfassende Mittel zur Verfügung stellen, um anhaltende Veränderungen zu erzielen.

Das Überraschendste ist vielleicht, dass Mike das tut, ohne zu fordern, zu beschwatschen, zu nötigen oder überhaupt darum zu bitten, dass Sie als Leser auch nur eines dieser Dinge tun. Stattdessen beginnt Mike mit einer tiefen Erkundung

der Werte und zeigt uns dann, wie wir so handeln können, dass es kongruent zu diesen Werten ist. Wenn Sie diese Dinge »wert«-schätzen, werden Sie diese Aktionen umsetzen wollen. Und wenn Sie sie umsetzen, werden Sie feststellen, dass Sie beginnen, die Verbesserungen zu erzielen, die Sie die ganze Zeit haben wollten.

Luke Hohmann

Gründer und CEO der Conteneo, Inc.

Vorwort zur deutschen Ausgabe

Die Diskussion über die Werte von Kanban wurde emotional geführt, als Mike sie in seinem Blog publizierte. Wieso sollte Kanban Werte bekommen? Die Community hatte sicher aus den Erfahrungen gelernt, die ein Teil der Gruppe mit den Werten von Scrum (Offenheit, Mut, Respekt, Commitment, Fokus) gemacht hatte: Wenige Menschen kennen sie und noch weniger Menschen wenden sie tatsächlich auch an. Warum sollten also Werte für Kanban wichtig sein? Sie bilden eine gemeinsame Sprache, aber das tun die Grundprinzipien und -praktiken auch. Sie geben diesen beiden allerdings eine »wertvolle« Grundlage: Sie verankern das Gefühl, das Kanban-Anwender kennen, wenn sie über die Methode sprechen, in der Sprache. Die meisten Kanban-Enthusiasten wissen, wie schwer es ist, Kanban allein anhand der Praktiken und Prinzipien zu vermitteln, denn es steht immer auch eine Philosophie, eine Geisteshaltung dahinter. Die ließ sich schwer in Worte fassen – bis die Werte die Brücke zwischen den Daten und den Emotionen bildeten. Die Community diskutierte intensiv über den Zweck, die Zusammensetzung, das Hinzufügen oder Weglassen einzelner Werte. Herausgekommen sind neun Werte, die in Beziehung zu den Praktiken und Prinzipien der Kanban-Methode stehen. Dieses Buch orientiert sich stark an diesen Werten und zeigt dabei, wie eng diese Beziehung ist. Wenn wir als Kanban-Trainer oder Coaches in Unternehmen gehen und uns eine bestehende Kanban-Implementierung ansehen, erkennen wir häufig anhand der Struktur und der Stimmung bei Meetings oder schon allein anhand des Kanban-Boards, dass etwas schiefläuft: fehlende WIP-Limits, Risiken werden nicht besprochen, Kundenwünsche nicht berücksichtigt, Überlast bei einzelnen Mitarbeitern, Aufgabenzuteilung auf einzelne Mitarbeiter etc. Denn obwohl einige Praktiken von Kanban augenscheinlich implementiert wurden, klingen viele Implementierungen trotzdem blechern, wenn man dagegen klopft – bildlich gesprochen. Bei jeder Kanban-Implementierung geben uns die Werte einen Ansatzpunkt, um über unsere spezifische Interpretation der recht allgemein gehaltenen Praktiken zu urteilen: Sind wir auf den Kunden fokussiert? Fließt unsere Arbeit tatsächlich? Verstehen wir das System und die bestehende Unzufriedenheit zur Genüge? Handeln wir mit dem Einverständnis aller und respektieren wir die Menschen im System? Nachdem Sie dieses Buch gelesen und sich hoffent-

lich die Werte zu eigen gemacht haben, werden Sie diese Fragen ganz natürlich für eine bestehende oder neue Implementierung stellen können. Fragen wie diese stellen erfahrene Kanban-Anwender ständig – sie sind für die richtige Implementierung der Prinzipien und Praktiken ungemein hilfreich und wichtig.

In der Reflexion über die Werte und ihre Anwendung liegt viel Lernpotenzial. Auch wir beide haben während der Bearbeitung des Buches wieder viele neue Aspekte und Interpretationen der Kanban-Methode gefunden. Ihnen wird das Buch natürlich nicht nur die Werte vermitteln, sondern auch den aktuellsten Stand der Kanban-Methode. Falls Sie die Methode bereits anwenden, aber die Werte noch nicht kennen, wird Ihnen das Buch einen neuen Blick auf Kanban bescheren. Und auch wenn bei der täglichen Arbeit selten die Zeit vorhanden ist, um über die Prozesse und ihre Anwendung zu reflektieren: Es lohnt sich, mit neuem Blick auf die eigene Arbeit zu schauen. Die Kanban-Methode sieht diese Feedbackzyklen zum Glück explizit vor!

Wir wünschen Ihnen viel Erfolg und freuen uns über einen Dialog mit Ihnen!

Florian Eisenberg
post@florianeisenberg.de

Wolfgang Wiedenroth
wolfgang.wiedenroth@it-agile.de

Hamburg, Juli 2015

Vorwort

Vielleicht haben Sie schon von Menschen mit T-förmiger Qualifikation gehört. Nun, dies ist ein T-förmig qualifiziertes Buch! Es besitzt in seinem Kern die Tiefe, die man erwartet, gleichzeitig aber auch eine gewisse Breite. Seine Tiefe bezieht sich natürlich auf die Kanban-Methode: Was sie ist; wie Anwender denken; Ratschläge, wie man sie anwendet; und so weiter. Die Breite kommt in Form einiger wichtiger Referenzpunkte außerhalb der Methode hinzu. Diese sind hoffentlich sowohl für die hilfreich, die Kanban noch nicht verwenden und von außen hineinsehen, als auch für jene, die Kanban anwenden und auf der Suche nach frischen Quellen der Inspiration sind.

Ich habe es mir zur Aufgabe gemacht, die »menschliche *>Beginne mit dem, was du gerade tust<*-Vorgehensweise für Veränderung« nicht als Produktivitätswerkzeug zu beschreiben, sondern als Managementmethode. Eine Methode, die um ein starkes Framework von Werten herum gebaut ist. Sie ist ein Weg, um Organisationen zu helfen, besser für ihre Mitarbeiter, Kunden und Stakeholder zu arbeiten.

Eine wichtige Eigenschaft dieser Managementmethode ist, dass sie auf vielen verschiedenen Ebenen angewendet werden kann: von der Ebene einzelner und kleiner Teams bis ganz nach oben zu strategischen Businessinitiativen. Seien Sie also nicht überrascht, wenn wir uns innerhalb weniger Abschnitte von einer Ebene auf die andere bewegen. Die Einfachheit, mit der das möglich ist, gibt schon eine deutliche Vorstellung davon, wie die Kanban-Methode funktioniert.

Ich schreibe als erfahrener Manager mit einem ausgeprägten technischen Hintergrund. In den vergangenen Jahren bin ich Entwicklungsleiter von Teams auf vier Kontinenten, Geschäftsführer, IT-Leiter und Interimsmanager gewesen. Es ist schon einige Zeit her, dass ich dafür bezahlt wurde, Anwendungen zu programmieren. Es ist aber immer noch eine meiner Leidenschaften und auch heute noch zieht es mich in Diskussionen über Systemarchitektur und Design.

Natürlich lässt sich das Buch ein wenig leichter lesen, wenn man etwas Wissen über die Softwareentwicklung besitzt, es ist aber keinesfalls eine Voraussetzung. Genauso werde ich keine Annahmen über Kenntnisse in Bereichen wie *Lean* und *Agile* treffen – dazu kommen wir, wenn wir sie brauchen. Sie sollten

vielmehr eine Art professioneller Neugier über das Management *kreativer Wissensarbeit* mit mir teilen und wie ich daran interessiert sein, wie man diese für alle Betroffenen verbessern kann.

Das Buch ist in drei Hauptteile gegliedert:

- Teil I stellt die Kanban-Methode (oder auch nur »Kanban«) auf eine neue Art vor – mit einem System von neun Werten. Ich begann Anfang 2013, über die Werte zu schreiben, und sie sind seitdem das Herzstück meiner Arbeit. Teil I schließt mit zwei noch jüngeren Konzepten: den drei *Agenden* und der *Kanban-Linse*. Die Agenden sind das Ergebnis der Zusammenarbeit mit meinem Freund und Kollegen David J. Anderson (dem Urheber der Kanban-Methode), die Kanban-Linse stammt von David direkt.
- In Teil II geht es um Breite. Statt zu behaupten, alle Antworten zu haben, beinhaltet die Kanban-Methode in einer ihrer Kernpraktiken den Term »unter Verwendung von Modellen«. Diese wenigen Worte ermutigen Anwender dazu, Anleihen aus anderen etablierten Wissensgebieten zu machen. Diese beinhalten Systemdenken, Lean, Agile und die Engpasstheorie, aber es gibt viele weitere, aus denen man wählen kann. Es ist nicht möglich, alle eingehend zu behandeln, aber ich hoffe, dass meine Kanban-zentrierte Perspektive dazu führt, dass Sie diese weiter erforschen und in Ihr Denken integrieren.
- Teil III beschreibt einen wiederholbaren Prozess, um die Kanban-Methode in einer Organisation zu implementieren. David nennt diesen Prozess *STATIK* oder »Systems Thinking approach to introducing Kanban«. STATIK stellt das Rückgrat unserer Basisschulung zu Kanban dar. Ich habe den Prozess aktualisiert, indem ich ihn mit den Agenden und Werten aus Teil I zusammengeführt habe und einige Modelle aus Teil II explizit referenziere.

Es ist Ihnen vielleicht schon aufgefallen, dass ich Kursivschrift verwende, wenn ich ein Wort oder eine Phrase bewusst als technischen Ausdruck verwende – *beginne mit dem, was du gerade tust* zum Beispiel ist Kanbans erstes Grundprinzip; *Serious Games* bezieht sich auf etwas Wohldefiniertes. Viele dieser Begriffe und Ausdrücke finden sich im Glossar wieder. Begriffe wie **Balance** und **Kundenfokus**, die fett gedruckt sind, beziehen sich auf die Werte von Kanban.

Inhaltsübersicht

Teil I	Kanban durch seine Werte verstehen	1
1	Transparenz	3
2	Balance	13
3	Kooperation	23
	Reflexion: Transparenz, Balance und Kooperation	29
4	Kundenfokus	31
5	Arbeitsfluss	37
6	Führung	47
	Reflexion: Kundenfokus, Arbeitsfluss und Führung	53
7	Verständnis	55
8	Vereinbarung	63
9	Respekt	71
10	Muster und Agenden	77
Teil II	Modelle	85
11	Systemdenken, Komplexität und die lernende Organisation	87
12	Engpasstheorie (TOC)	99
13	Agile	109
14	TPS und Lean	119

15	Ökonomische Ansätze zum Arbeitsfluss	129
16	Die Kanban-Methode	143
17	Kleinere Modelle	153
Teil III Implementierung		167
18	Quellen der Unzufriedenheit erkennen	169
19	Anforderungen und Leistungsfähigkeiten analysieren	175
20	Den Workflow modellieren	185
21	Serviceklassen finden	191
22	Kanban-Systeme gestalten	197
23	Ein Kanban-System einführen	207
Anhang		221
A	Demings 14 Punkte für gutes Management	223
B	Danksagung	225
C	Glossar	227
D	Literatur	237
	Index	245

Inhaltsverzeichnis

Teil I	Kanban durch seine Werte verstehen	1
1	Transparenz	3
1.1	Kernpraktik 1: Visualisiere	4
1.1.1	Visualisierung und Veränderung	5
1.2	Kernpraktik 4: Mache Prozessregeln explizit	7
1.2.1	Prozessregeln und Veränderung	8
1.3	Kernpraktik 5: Implementiere Feedbackzyklen	8
1.3.1	Das Standup-Meeting	9
1.3.2	Replenishment-Meetings	10
1.3.3	Weitere Meetings	10
1.3.4	Feedbackzyklen, die auf Metriken basieren	11
1.4	Transparenz als Wert	12
2	Balance	13
2.1	Kernpraktik 2: Parallelle Arbeit (Work in Progress, WIP) limitieren	13
2.2	Pull-Systeme in der Wissensarbeit	13
2.3	Balance zwischen Arbeitslast und Kapazität	15
2.4	Andere Möglichkeiten, WIP zu limitieren	16
2.5	Balance zwischen dringlichkeits- und termingetriebener Arbeit	17
2.6	Risikobasierte Kategorisierung und Serviceklassen	18
2.7	Balance zwischen Anforderungen und Leistungsfähigkeit	20
2.8	Stakeholder-Balance	20
2.9	Balance suchen	21

3	Kooperation	23
3.1	Kernpraktik 6: Erziele Verbesserung kooperativ und entwickle experimentell	23
3.2	Kooperation ist eine ernste Sache	23
3.3	Erziele Verbesserung kooperativ	24
3.4	Zu Kooperation ermutigen	24
3.5	Fokus auf Kooperation legen	25
3.6	Entwickle experimentell	26
3.7	Die und wir	27
Reflexion: Transparenz, Balance und Kooperation		29
4	Kundenfokus	31
4.1	Kernpraktik 3: Manage den Arbeitsfluss	31
4.2	Warum Kundenfokus?	31
4.3	Zufriedenheit garantiert	32
4.4	Quer über das Board	33
4.5	Upstream-Kanban	33
4.6	Bedürfnisse vorwegnehmen	36
5	Arbeitsfluss	37
5.1	Kernpraktik 3: Manage den Arbeitsfluss (nochmal)	37
5.2	Gleichmäßigkeit	37
5.2.1	Wie sieht Arbeitsfluss aus?	38
5.3	Quer über das Board (nochmal)	39
5.4	Es gibt immer einen größeren Kontext	40
5.5	Einige einfache Beispiele	41
5.6	Arbeitsfluss quer durch die Organisation	42
5.7	Arbeitsfluss managen, um Pünktlichkeit zu erzielen	45
6	Führung	47
6.1	Grundprinzip 4: Führung auf allen Ebenen	48
6.2	Möglichkeiten finden sich überall	48
6.3	Führung erzeugt Führung	49
6.4	Warum Führung wertschätzen?	50
6.5	Wie sieht Führung in Kanban aus?	51
Reflexion: Kundenfokus, Arbeitsfluss und Führung		53

7	Verständnis	55
7.1	Änderungen ohne Verständnis – drei Anti-Pattern	55
7.1.1	Selbstzufriedenheit	56
7.1.2	Draufgängertum	56
7.1.3	Einmischung	58
7.2	Einführung der J-Kurve	59
7.3	Ein Muster für zielgerichtete Veränderung	60
8	Vereinbarung	63
8.1	Verfolge evolutionäre Veränderung	63
8.2	Vereinbare das Verfolgen	65
8.3	Change Management	66
8.3.1	Der Change Agent	66
8.3.2	Der betreute Change Agent	67
8.3.3	Das Change-Team	69
9	Respekt	71
9.1	Fange nicht bei den Rollen an	71
9.2	»Sei wie Wasser«	72
9.3	Respekt für Menschen	73
9.3.1	Transparenz	73
9.3.2	Balance	74
9.3.3	Kooperation	74
9.3.4	Kundenfokus	74
9.3.5	Arbeitsfluss	75
9.3.6	Führung	75
9.3.7	Verständnis	75
9.3.8	Vereinbarung	76
9.4	Die menschliche »Beginne mit dem, was du gerade tust«-Methode	76
10	Muster und Agenden	77
10.1	»Noble Muster«	77
10.2	Agenden für Veränderung	78
10.3	Die Nachhaltigkeits-Agenda	79
10.4	Die Serviceorientierungs-Agenda	80
10.4.1	Die Kanban-Linse	80
10.5	Die Überlebensfähigkeits-Agenda	81

10.6	Die Agenda Ihrer Organisation	81
10.7	Wie erging es uns?	82
10.8	Fünf Jahre später	83

Teil II	Modelle	85
----------------	----------------	-----------

11	Systemdenken, Komplexität und die lernende Organisation	87
11.1	Systemdenken	87
11.1.1	Hebelpunkte und Systemdynamiken	88
11.2	Komplexität	89
11.2.1	Kausalität und das Cynefin-Framework	91
11.2.2	Komplexe adaptive Systeme	92
11.3	Wissen, lernen und die lernende Organisation	93
11.3.1	Demings System vom umfassenden Wissen	93
11.3.2	Argyris und das Doppelschleifen-Lernen	94
11.3.3	Die lernende Organisation	96
11.4	Systemdenken und Kanban	97
11.4.1	Das Design der Methode	97
11.4.2	Anwendung	98
12	Engpasstheorie (TOC)	99
12.1	Die fünf Fokussierungsschritte und der Process of Ongoing Improvement (POOGI)	100
12.2	Drum-Buffer-Rope (DBR)	101
12.3	Thinking Processes – Denkprozesse	102
12.4	Critical-Chain-Projektmanagement (CCPM)	103
12.5	Throughput Accounting – Durchsatzrechnung	105
12.6	Die Beziehung von TOC zu Kanban	105
12.6.1	POOGI und die fünf Fokussierungsschritte	106
12.6.2	DBR und CCPM	107
12.6.3	Die Denkprozesse	107
12.7	Dennoch	108
13	Agile	109
13.1	Drei agile Methoden	110
13.1.1	Feature-Driven Development (FDD)	110
13.1.2	Extreme Programming (XP)	111
13.1.3	Scrum	114

13.2	Kanban und Agile	116
13.2.1	Kompatibilität	116
13.2.2	Wann ist Kanban zu benutzen?	117
13.3	Das Modell Agile	118
14	TPS und Lean	119
14.1	Drei Lean-Werkzeuge	120
14.2	TPS und Lean im richtigen Licht	122
14.3	Verbesserungen bei Lean	123
14.4	Lean Product Development	124
14.5	Lean Startup	126
14.6	Lean/Agile-Hybriden	127
14.7	Kanban und Lean	127
15	Ökonomische Ansätze zum Arbeitsfluss	129
15.1	ROI und die Pareto-Diät	130
15.2	Verzögerungskosten	134
15.2.1	Die Kosten von Warteschlangen	137
15.3	Haltekosten	137
15.4	Optionen	139
15.4.1	Optionen haben einen Wert	140
15.4.2	Optionen verfallen	141
15.4.3	Binde dich nie frühzeitig, es sei denn, du weißt warum ..	141
15.5	Alles zusammengeführt	142
16	Die Kanban-Methode	143
16.1	Eine sehr kurze Historie	143
16.2	Grundprinzipien	145
16.3	Kernpraktiken	145
16.4	Kontextualisiertes Kanban	146
16.4.1	Personal Kanban	146
16.4.2	Portfolio-Kanban	147
16.4.3	Scrumban	148
16.5	Unterstützende Konzepte und Werkzeuge	149
16.6	Implementierungshilfe: STATIK	150

17 Kleinere Modelle	153
17.1 Zwei, die davongekommen sind	153
17.1.1 Littles Gesetz	153
17.1.2 Das Satir-Modell für Veränderung	156
17.2 Denkwerkzeuge und Coaching-Modelle	157
17.2.1 GROW	157
17.2.2 A3	158
17.2.3 Exkurs: Wie ein Profi überprüfen!	159
17.2.4 Der Lean Change Canvas	160
17.3 Gruppenmoderation und Spiele	160
17.3.1 Kaners Moderationsmodell	160
17.3.2 Serious Games	162
17.4 Modelle für kooperative Führung: Triaden und T-Formen	164
Teil III Implementierung	167
18 Quellen der Unzufriedenheit erkennen	169
18.1 Zwei Perspektiven	169
18.2 Zwei Fragen	170
18.3 Formate	171
18.4 Organisiere und erforsche	172
18.5 Folgemaßnahmen	173
19 Anforderungen und Leistungsfähigkeiten analysieren	175
19.1 Wissen, was Sie wem liefern und warum	176
19.1.1 Was	176
19.1.2 Wem	177
19.1.3 Warum	178
19.2 Quantitative Analyse	178
19.2.1 Flusseffizienz	181
19.3 Wie Arbeit ankommt	182
19.4 Passt alles zusammen?	183
20 Den Workflow modellieren	185
20.1 Grob aufzeichnen	185
20.2 Top-down-Dekomposition	186
20.3 Bottom-up-Organisation	188
20.4 Fazit	189

21	Serviceklassen finden	191
21.1	Entdecken, überprüfen	192
21.2	Beispiele	193
21.3	In Richtung einer gesunden Arbeitszusammenstellung	194
22	Kanban-Systeme gestalten	197
22.1	Umfang, Granularität und Status der Arbeitseinheiten	197
22.1.1	Aufeinanderfolgende Status	197
22.1.2	Parallele Status	198
22.1.3	Fehler	199
22.1.4	Abhängigkeiten	200
22.2	Andere Dimensionen	201
22.3	Hierarchische Boards und das Aufbrechen/Zusammenführen-Muster	202
22.4	Work in Progress limitieren	203
22.5	Review	205
23	Ein Kanban-System einführen	207
23.1	Den Einsatz planen	207
23.2	Die Agenda zusammenstellen: die drei P	209
23.2.1	Positionierung	209
23.2.2	Zweck (Purpose)	210
23.2.3	Prioritäten	210
23.3	Veränderung durch das System vorantreiben	211
23.3.1	Veränderungsinkremente identifizieren	212
23.3.2	Veränderung visualisieren	216
23.4	Abschließende Gedanken	220
Anhang		221
A	Demings 14 Punkte für gutes Management	223
B	Danksagung	225
C	Glossar	227
D	Literatur	237
	Weiterführende Literatur zu einzelnen Themen	240
	Index	245

Teil I

Kanban durch seine Werte verstehen

Die Herangehensweise, die ich zur Einführung von Kanban wähle, ist ein wenig unüblich, denn meist wird Kanban durch die Grundprinzipien und Kernpraktiken erklärt. Ich gehe einen halben Schritt zurück und beginne bei einem System von neun Werten. In der Reihenfolge der ersten neun Kapitel in Teil I sind dies die Werte **Transparenz**, **Balance**, **Kooperation**, **Kundenfokus**, **Arbeitsfluss** (Flow), **Führung** (Leadership), **Verständnis**, **Vereinbarung** und **Respekt**.

Durch diese Vorgehensweise entfallen keine Prinzipien und Praktiken von Kanban; jedes Kapitel referenziert eine oder mehrere direkt.

Der Einfachheit halber kennzeichne ich die Grundprinzipien als GP1–4:

- GP1: Beginne mit dem, was du gerade tust.
- GP2: Vereinbare, dass evolutionäre Veränderung verfolgt wird.
- GP3: Respektiere initial bestehende Prozesse, Rollen, Verantwortlichkeiten und Jobtitel.
- GP4: Ermutige dazu, Führung auf jeder Ebene der Organisation zu zeigen – vom einzelnen Mitarbeiter bis zum höheren Management.

Ebenso bezeichnen KP1–6 die sechs Kernpraktiken:

- KP1: Visualisiere.
- KP2: Limitiere die Menge paralleler Arbeit (Work in Progress, WIP).
- KP3: Manage den Arbeitsfluss.

KP4: Mache Prozessregeln explizit.

KP5: Implementiere Feedbackzyklen.

KP6: Erziele Verbesserung kooperativ, entwickle experimentell (unter Verwendung von Modellen und der wissenschaftlichen Methode).

Statt eine detaillierte technische Rechtfertigung für jede Technik darzustellen, hoffe ich, einen Einblick geben zu können, der erklärt, wie Kanban-Anwender organisatorische Probleme angehen. Mit Kanban-Anwendern meine ich Manager, Mitarbeiter und externe Experten, die die Methode kennen und anwenden. Die Werte helfen, die Intention der Prinzipien und Praktiken zu betonen, sodass Sie sie einfacher mit den Bedürfnissen Ihrer aktuellen Organisationssituation übereinbringen können.

Das zehnte und letzte Kapitel des ersten Teils fügt die 10 Werte in drei *Agenden* zusammen und führt ein weiteres nützliches Werkzeug ein, die *Kanban-Linse*.

1 Transparenz

Wir befinden uns ein paar Stockwerke weiter oben in unserem schönen, neuen Büro, von dem man die Nyugati-Haltestelle in Budapest überblicken kann. Es ist kurz nach 9:30 Uhr und unser Standup-Meeting läuft gerade. Unsere Stimmung ist ein wenig vergnügter als sonst, denn wir freuen uns auf Kaffee und Kuchen auf der anderen Straßenseite. Dort werden wir feiern, dass wir wieder etwas Neues gelernt haben. Die Rechnung geht dieses Mal auf mich – ich habe es am Tag zuvor geschafft, den Build über mehrere Stunden in einem schlimmen Zustand zu hinterlassen, und der Entwicklungsleiter sollte das eigentlich nicht tun.

Allerdings haben wir noch ein paar wichtige Dinge, um die wir uns kümmern müssen. Tibor und Maté drücken ihre Sorge über ein Ticket (eine gelbe Haftnotiz) aus, das seit Tagen in der »Released«-Spalte auf dem Whiteboard des Teams feststeckt. Gy bietet kurzerhand seine Hilfe an: Er ist sich recht sicher, dass er das Betriebsteam überzeugen kann, dass die neue Preiskurve, die durch das feststehende Ticket repräsentiert wird, zuverlässiger ist als die alte; wir sollten damit rechnen, dass sie in ein bis zwei Tagen voll in der Produktion implementiert ist. Gys Angebot wird dankbar akzeptiert und wir nehmen uns das nächste Ticket vor.

Diese wenigen Momente reichen aus, um einige Beispiele für Kanbans ersten Wert zu demonstrieren, **Transparenz**:

- Unsere Arbeit ist für alle sichtbar auf einem großen Whiteboard dargestellt. Auf dem Board sind Tickets, die Arbeitspakete repräsentieren, in Spalten organisiert; die Spalten korrespondieren zu Status, in denen sich Arbeitspakete befinden können, oder zu Schritten unseres Workflows.
- Wir verstehen die Wichtigkeit von regelmäßigen *Feedbacks* und wir halten ein *Standup-Meeting* ab.
- Wir haben einige der *Regeln*, die für unsere Arbeit gelten, herauskristallisiert; viele von ihnen sind in der Nähe des Boards aufgelistet. Eine dieser Regeln besagt, dass Entwickler die Verantwortung für ihre Arbeitspakete behalten, bis sie eine Kundenbestätigung bekommen haben, dass dieses Paket auch den gewollten Wert erzielt. Eine andere Regel schreibt vor, dass Lernen mithilfe von Kuchen verstärkt werden sollte – im ganz speziellen Fall die Ich-werde-diesen-Fehler-nie-wieder-machen-Art des Lernens.

Dieses Szenario spielte sich 2009-2010 ab, gerade als David Anderson Feedback zu den kürzlich formulierten *Grundprinzipien* und *Kernpraktiken* der Kanban-Methode einholte. Wir – Menschen wie ich, die die Ideen bereits anwendeten – diskutierten und verfeinerten sie über unsere Gruppenmailingliste. Innerhalb weniger Wochen gingen sie mit Davids »blauem Buch« [Anderson 2010] in den Druck. Wir hatten eine dokumentierte Methode!

Transparenz ist ein zentraler Aspekt von Kanban. Drei der sechs Kernpraktiken stehen damit in Verbindung:

KP1: Visualisiere.

KP4: Mache Prozessregeln explizit.

KP5: Implementiere Feedbackzyklen.

Sehen wir sie uns nacheinander an.

1.1 Kernpraktik 1: Visualisiere

Kanbans Ein-Wort-Praktik »Visualisiere« erscheint erst einmal sehr unspezifisch. Allerdings besitzen die meisten Kanban-Implementierungen eine bestimmte Art der Visualisierung, ein *Kanban-Board*. Diese Boards werden verwendet, um *Kanban-Systeme* zu implementieren. Das sind visuelle Arbeitsmanagementsysteme, die einige nützliche Eigenschaften haben. Wir werden diese Eigenschaften in den nachfolgenden Kapiteln genauer betrachten, für den Moment konzentrieren wir uns auf die visuellen Aspekte.

Hätten wir ein elektronisches Werkzeug verwendet, hätte unser Board vielleicht so wie in Abbildung 1–1 ausgesehen.

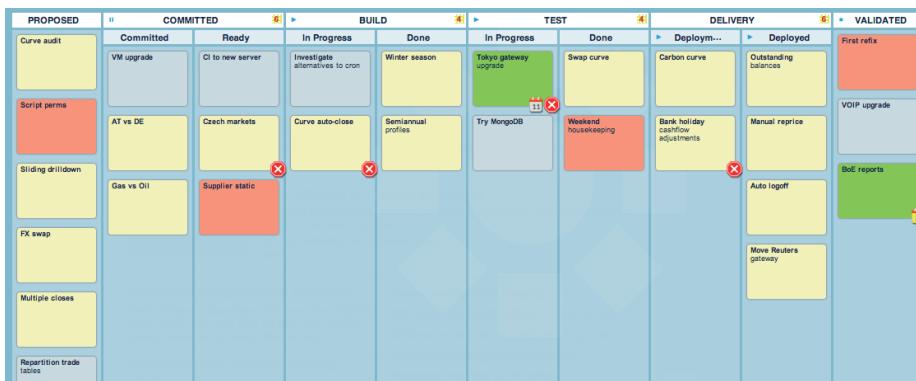


Abb. 1–1 Ein elektronisches Board

Statt Haftnotizen auf einem Whiteboard haben wir nun Icons, die wir über den Bildschirm ziehen können. Egal ob elektronisch oder physisch, diese werden als *kanban* bezeichnet, ein japanisches Wort, das als »visuelles Zeichen« oder auch

»Platzhalter« übersetzt werden kann. Wir bevorzugen die etwas umgänglicheren Begriffe *Karte* (oder *Ticket* – ich verwende die beiden synonym) und *Arbeitspaket*. Denken wir an das visuelle Design, tendieren wir zum ersten Begriff. Wir verwenden eher den zweiten, wenn der Fokus darauf liegt, was sie repräsentieren.

In diesem Buch repräsentieren Arbeitspakete festgelegte Teilstücke von Wissensarbeit: Dinge wie Produkteigenschaften, die entwickelt werden müssen, oder Serviceanfragen, die zu erfüllen sind. Diese Dinge müssen keine Verbindung zu Software haben – es gibt Beispiele aus Rechtsabteilungen, Personalabteilungen, dem Vertrieb und dem Büro des Geschäftsführers. Sie haben typischerweise gemeinsam, dass ein großer Teil der Arbeit in den Köpfen von Menschen oder auf ihren Computern erfolgt. Ohne das Board wäre diese Arbeit unsichtbar.

Es hört sich vielleicht trivial an, aber es ist wichtig, dass diese Tickets beweglich sind. Man bewegt die Tickets von Spalte zu Spalte, während die Arbeit am dahinterliegenden Arbeitspaket voranschreitet. Das bedeutet, dass man mit einem Blick sehen kann, wie viel Arbeit sich in welchem Zustand der Fertigstellung befindet. Schaffen Sie das mal mit einer To-do-Liste!

Wenn das Board groß genug und das Ticketdesign deutlich genug ist, kann man von der anderen Seite des Raums erkennen:

- Welche Arbeit gerade blockiert ist (wartet auf etwas)
- Wer woran arbeitet
- Welche unterschiedlichen Arten von Arbeit existieren und in welcher Verteilung
- Wie viel Arbeit sich in welchem Grad der Fertigstellung befindet

Haben wir all diese Informationen ständig verfügbar, bekommen wir recht schnell ein Gefühl dafür, wie ein gesunder Zustand aussieht. Wenn man sich darauf einmal eingerichtet hat, lässt einen das Board sofort wissen, wenn Dinge von der Norm abgewichen sind und eine Untersuchung des Zustands oder korrigierende Maßnahmen notwendig sind.

1.1.1 Visualisierung und Veränderung

Die Visualisierung und andere Formen der Transparenz haben in Kanban einen doppelten Sinn: Zum einen soll die Notwendigkeit von Aktivität sichtbar gemacht werden und zum anderen soll den Menschen geholfen werden, gute Entscheidungen zu treffen. Beides funktioniert auf zwei Ebenen:

- Aktivität in Form von Arbeit, die getan werden muss; gute Entscheidungen bei der Auswahl der Arbeitspakete
- Aktivität in Form von Veränderung des Systems; gute Entscheidungen in der Rechtfertigung, im Umfang und in der Implementierung der Veränderung

Wie reagieren Sie, wenn das Board anzeigt, dass nicht alles so ist, wie es sein sollte? Hier sind einige übliche Antworten, die ein Manager oder Coach geben könnte:

1. *Egal, es wird sich schon alles selbst regeln, wie es das normalerweise auch immer tut.*
2. *Ich werde durch ein straffes Management eingreifen, bis die Situation vorüber ist.*
3. *Ich werde durch eine Veränderung des Systems eingreifen.*
4. *Verstehen sie, wie die Dinge so gekommen sind? Wollen sie das System entsprechend verändern? Wie sollte ich ihnen helfen?*
5. *Ich respektiere ihre Fähigkeiten, angemessene Änderungen in dieser Situation durchzuführen.*

Jede dieser Verhaltensweisen hat ihre Berechtigung, allerdings scheinen einige eine höhere Reife widerzuspiegeln als andere. Durch das Auslösen von Aktionen und das Unterstützen guter Entscheidungen führt Kanban die Menschen zu reifrem Verhalten wie in Punkt 4 oder 5. Organisationen, die diese Verhaltensweisen bewusst akzeptieren und die **Führungsstile** fördern, die solches Verhalten unterstützen, sind auf einem guten Weg zu höherer Reife.

Verhalten 5 (und zu einem geringeren Anteil Verhalten 4) enthält ein weiteres, sehr interessantes Element, die *Selbstorganisation*.

Selbstorganisation ist eine wunderbare Sache. Sie bedeutet nicht nur, dass Individuen autonom handeln können – obwohl das essenziell für ihr Wohlergehen ist. Selbstorganisation bedeutet auch, dass das System sich selbst neu organisieren und einstellen kann, um effektiver mit seinen Herausforderungen umgehen zu können. Selbstorganisation bringt in ihrer vollen Bedeutung Anpassungsfähigkeit und Resilienz. Da Selbstorganisation ohne Intervention von außen passieren kann, skaliert sie sehr gut. Sowohl für den Systembetrieb als auch für die Systemveränderung ist Selbstorganisation sowohl effektiv als auch menschenfreundlich.

Ein System zu verändern, das visuell gemanagt wird, sollte eine kostengünstige und schnelle Sache sein. Einfach eine Linie oder zwei auf dem Whiteboard ausradieren, eine weitere zeichnen und ein paar Klebezettel umherschieben (oder ein paar Klicks mit der Maus). Die Auswirkung dieser Änderungen kann sehr groß sein. Verglichen mit dem recht kleinen Implementierungsaufwand ist dieses Arbeiten am System eine Aktivität mit sehr großem Hebel.

Hier ist ein positiver Kreislauf im Spiel:

- Das Kanban-System organisiert die Arbeit.
- Die Menschen organisieren sich um die Arbeit herum.
- Aus ihrer neuen Perspektive sehen die Menschen, dass das Kanban-System die Arbeit besser organisieren könnte, als es das im Moment tut, und sie verändern es.

1.2 Kernpraktik 4: Mache Prozessregeln explizit

Kanban-Boards sind sehr effektiv darin, Arbeit zu organisieren, aber einige Aspekte des Systems werden manchmal nicht optimal durch die visuelle Sprache von Tickets, Farben, Spalten und Ähnlichem beschrieben. Manchmal machen ein paar wenige Worte in Form von Prozessregeln den Unterschied. Es handelt sich dabei nicht um Verordnungen von oben: Vielmehr sind sie ein Weg, um zwischen den Teilnehmern ein gemeinsames Verständnis darüber aufrechtzuerhalten, wie das System betrieben wird.

Ein weiterer Aspekt der **Transparenz**: Parallel zur Strategie, das Unsichtbare sichtbar zu machen, machen wir das Implizite explizit, und zwar dann (und nur dann), wenn wir glauben, dass es uns hilft, bessere und vorhersagbarere Entscheidungen zu treffen. Auch hier möchten wir einen Hebel ansetzen – wenige Worte, die vorsichtig ausgewählt werden, um die Absicht darzulegen, keine dicken Dokumente, die alle Eventualitäten abdecken. Ich habe schon Prozessregeln gesehen, die so kurz waren wie »*Demo!*« und über der betroffenen Spalte auf dem Board gehängt waren – das war alles, was benötigt wurde, um eine effektive Ver einbarung über die Arbeitsweisen zu stützen.

Viele Prozessregeln beschreiben, welche Kriterien Arbeitspakete erfüllen sollen, wenn sie eine Spalte erreichen oder sie verlassen, zum Beispiel:

- *Tickets in der »Bereit«-Spalte sollten nicht mehr als fünf Tage Arbeit in der Entwicklung benötigen.* In Budapest nannten wir sie die »Fünf-Tage-Regel«; später wurde sie zur »Zwei-Tage-Regel.«
- *Tickets dürfen nicht in die »Test«-Spalte rutschen, bis sie ein Peer-Review absolviert haben und die Ergebnisse dem Team demonstriert wurden.*

Auf unserem Board waren Stichworte wie »< 5 Tage Implementierung«, »Code-review« und »*Demo!*« als Erinnerung an die Erwartung des Teams absolut ausreichend.

Andere Prozessregeln können auch globalerer Natur sein:

- *Die Stabilität der Produktionsumgebung hat eine höhere Priorität als das Lösen von Fehlern in der Qualitätssicherung; beide haben höhere Priorität als Neuentwicklungen.*
- *Wird ein neues Stück Arbeit gezogen, wird der Auftraggeber informiert, falls es wahrscheinlich ist, dass es Auswirkung auf bestehende Arbeit hat.*

Diese Beispiele für Prozessregeln sind nicht universal anwendbar, aber sie können einfach in vergleichbare Kontexte übernommen werden. Das ist nicht untypisch. Manchmal existiert ein unterliegendes Prinzip, das sich leichter übertragen lässt als die Regel – »*Das Lernen feiern*« statt »*Kuchen essen*« zum Beispiel. Es ist aber auch keine Schande, wenn eine Regel komplett einmalig für die Situation ist. Der Kontext ist wichtig!

1.2.1 Prozessregeln und Veränderung

Wir fügen Prozessregeln hinzu, wenn wir glauben, dass die zusätzliche Klarheit uns dabei helfen wird, bessere Entscheidungen zu treffen oder diese effizienter zu treffen. Das ist häufig ein guter Zeitpunkt, um die darunterliegenden Denkweisen zu diskutieren, wie zum Beispiel:

- *Große Arbeitspakete haben verglichen mit kleineren eine überproportional große Wahrscheinlichkeit, problematisch zu sein.* (Das könnte auch mehr als nur ein Bauchgefühl sein – wir könnten Daten haben, die diese Hypothese belegen.)
- *Allgemein gesagt ist es besser, etwas fertigzustellen als etwas weiteres Neues zu beginnen.* (Das könnte als hausgemachte Lebensweisheit anerkannt sein – »Stop starting, start finishing!« – oder auf solider Theorie aufgebaut sein.)

Indem wir Prozessregeln explizit machen, erzwingen wir, dass das darunterliegende Denken überprüft wird. Passt die Realität nicht zu unserem Denken, sind wir ständig dabei, uns an den Regeln zu reiben. Das erzeugt Unbehagen, aus dem eine Reevaluation und weiteres Lernen entsteht.

Aus diesem Grund ist es gut, mit einfachen Prozessregeln zu beginnen, die die aktuelle Praxis wiedergeben (das, was wirklich in den meisten Fällen getan wird, unabhängig von den offiziellen Regeln), und von dort aus nach Bedarf zu verfeinern. *Beginne mit dem, was du gerade tust* ergibt hier genauso viel Sinn wie auch sonst. Schreib sie nieder, dann stelle dich der Herausforderung. Ist es immer sinnvoll, hier eine Demo zu machen? Könnte diese Zehn-Tage-Aufgabe vielleicht trotz allem doch okay sein?

Diese Strategie – vorher implizite Dinge explizit zu machen – bezieht sich auch auf die Kanban-Methode selbst. Das Vorhandensein von Feedbackzyklen und ihre Implementierung waren so offensichtlich, dass niemand daran dachte, sie im ursprünglichen Entwurf der Methode mit aufzunehmen.

1.3 Kernpraktik 5: Implementiere Feedbackzyklen

Trotzdem sie anfangs ausgelassen wurden, sind Feedbackzyklen essenziell. Ohne frühes Feedback werden Anzeichen, dass das System ungesund ist, übersehen. Oder sie werden ignoriert oder so halbherzig adressiert, dass niemand sicher ist, ob wir wirklich eine Verbesserung erzielt haben. Feedbackzyklen sind deshalb essenziell, damit Transparenz zu einem effektiven Treiber für die Veränderung gemacht werden kann.

Wie bei »Visualisiere« ist viel Platz für Interpretation und kreative Implementierung in der Formulierung dieser Praktik. Das ist Absicht. Um es etwas konkreter zu machen, schauen wir uns drei allgemein implementierte Beispiele für Meetings an, die regelmäßig die Möglichkeit für verschiedene Arten von Feedback bieten. Wir beenden diesen Abschnitt mit Feedbackzyklen, die auf Metriken basieren.

1.3.1 Das Standup-Meeting

Sollte ich eine agile Praktik vor allen anderen empfehlen, so wäre es diese. Ich habe erlebt, wie Teams Standup-Meetings abschaffen, weil sie ihren Wert nicht erkennen, um sie dann nur wenige Tage oder Wochen später wieder einzuführen, wenn die Dinge langsam auseinanderfallen. Vielleicht macht gerade die Vertrautheit mit dieser Praktik es so einfach, den Wert dahinter zu übersehen.

Standup-Meetings sind kurze Treffen – kurz genug, dass alle Teilnehmer für die Dauer komfortabel stehen können (oder sollten) –, die regelmäßig durchgeführt werden (häufig täglich). Die Geschwindigkeit kommt mit der Übung: den Ablauf kennen, Aktualisierungen auf einem angemessenen Detailniveau darstellen und die Disziplin besitzen, Konversationen auf danach zu vertagen.

Standup-Meetings können verschiedene Formen haben:

- Informell, ohne Agenda, unstrukturiert (kaum zu empfehlen)
- Befragung durch einen Manager (häufig ist das ein Projektmanager, der nach Aktualisierungen fragt)
- Nacheinander, Person für Person, vielleicht unter Verwendung des *Scrum*-Formats, in dem jeder Teilnehmer beschreibt, *was er gestern getan hat, was er heute vorhat und welche Hindernisse er hat*¹.
- Durchgehen der Arbeitspakete auf dem Kanban-Board, von rechts nach links, beginnend mit den Dingen, die kurz vor der Fertigstellung sind. Das Treffen wird beendet, wenn es unproduktiv wäre, weiter stromaufwärts zu schauen.
- Ähnlich wie oben das Board von rechts nach links durchgehen, aber nur die Pakete besprechen, die blockiert oder risikobelastet sind.

Sowohl das Scrum-Format als auch die beiden boardzentrierten Formate unterstützen unsere dahinterliegende Agenda, Transparenz um unsere Entscheidungen herum hinzuzufügen: Woran man arbeiten sollte und wie, wann man taktisch intervenieren sollte, um einem Arbeitspaket zu helfen, vorwärts zu kommen, und wann man zurücktreten und beobachten sollte, wie das System wirklich funktioniert.

Dabei gibt es natürlich auch ein soziales Element, das bei der Teambildung hilft. Teammitglieder profitieren immens von dem Treffen. Sie wissen dadurch nicht nur, was überhaupt passiert, sondern haben auch eine bessere Vorstellung von der Arbeit der anderen Teammitglieder. Wenn dadurch Vertrauen aufgebaut wird und die Bedürfnisse des Teams besser verstanden werden, wird die Kommunikation ehrlicher und direkter. Kurz gesagt, es sind fünfzehn gut investierte Minuten.

Es überrascht Sie wahrscheinlich kaum, dass ich eine starke Vorliebe für die boardgetriebenen Formate habe. Sie stärken die Idee, dass wir die Arbeit zusammen über die Ziellinie bringen wollen; sie sind mehr auf das Ziel der Arbeit fokussiert als auf den, dem die Arbeit zugewiesen ist. Ich folge dem allerdings

1. Die agile Methode Scrum wird in Kapitel 13 erklärt.

nicht sklavisch und ich habe gesehen, dass Teams, die mit den Techniken und miteinander angenehm umgehen, unbewusst, aber trotzdem noch angemessen zwischen den verschiedenen Stilen, sowohl formelle als auch informelle, hin- und herwandern können.

Halten Sie gleichwertige Treffen ab? Fragen Sie sich im Sinne von Explizit-Machen des Impliziten: Welche Absicht steckt hinter dem Format meines Treffens? Wird diese Absicht gänzlich umgesetzt? Fördern oder behindern die Teamtreffen selbstorganisiertes Problemlösen?

1.3.2 Replenishment-Meetings

Das Replenishment-Meeting ist eine weitere weitverbreitete Praktik². Dies ist das Forum, in dem die *Input Queue* (oder, falls Ihnen das lieber ist, das *Backlog*) eines Prozesses mit neuer Arbeit befüllt wird. Es ist außerdem eine großartige Möglichkeit, um die Kundenzufriedenheit zu überprüfen, Kundenbedürfnisse zu erforschen und diese Bedürfnisse mit den Fähigkeiten des Teams zu vergleichen. Es ist ein wichtiger Feedbackzyklus, denn es liefert eine externe Perspektive.

Ich werde aus zwei Gründen nicht auf die Durchführung von Replenishment-Meetings eingehen:

1. Sie sind sehr kontextabhängig. Wie man mit einem einzelnen, internen Kunden zusammenarbeitet, wird sich sehr von dem Umgang mit mehreren, externen Kunden unterscheiden. Gestalten Sie den Prozess passend zu Ihrer Situation und seien Sie offen für die Möglichkeit, dass es eine angemessene Lösung sein kann, Meetings nur nach Bedarf abzuhalten – oder überhaupt keine Meetings.
2. Ich habe schon zu häufig gesehen, dass Teams in die Falle tappen, sich auf Kosten der Kunden zu sehr auf die Bedürfnisse des Prozesses (oder noch schlimmer, des Lieblingsprozesses des Prozessbegleiters) zu fokussieren. Wem kommen die gestellten Hürden wirklich zugute?

1.3.3 Weitere Meetings

Es existieren weitere Meetings außerhalb des Lieferprozesses, die dafür vorgesehen sind, den Verbesserungsprozess mit Daten zu versorgen oder zu managen. Eine populäre agile Praktik, die von einigen Kanban-Implementierungen übernommen wurde, ist die *Retrospektive* auf Teamebene. In anderen Implementierungen wird sie durch eine eher spontane Herangehensweise ersetzt. Wir raten größerer Kanban-Implementierungen dringend dazu, abteilungsweise wöchentlich oder vierzehntägig »Service Delivery Reviews« abzuhalten (manchmal auch *System Capability Review* genannt) und bereichsweite *Operations Reviews* auf

2. Sollten Sie Scrum kennen, sein Replenishment-Meeting ist das *Sprint Planning Meeting*.

einer monatlichen Basis. Bei diesen Meetings teilen mehrere Teams ihre Leistungsdaten, Zwischenfallberichte und Aktualisierungen bezüglich angegangener Verbesserungen miteinander und (idealerweise) mit Kundenrepräsentanten und der Restorganisation.

Vielleicht führen Sie bereits diese oder ähnliche Meetings durch. Das Ziel ist natürlich nicht, noch weitere Treffen zu etablieren, sondern Bedingungen zu schaffen, in denen schnelles Feedback generiert wird und anhand dessen effektiv gehandelt werden kann. Sie sind es wert, mit Sorgfalt gestaltet zu werden.

1.3.4 Feedbackzyklen, die auf Metriken basieren

Das Diagramm in Abbildung 1–2 ist mein *Cumulative Flow Diagram* (häufig als CFD bezeichnet). Ich habe es folgendermaßen erstellt:

1. Alle paar Tage habe ich gezählt, wie viele Tickets sich in jeder Spalte des Boards befinden.
2. Fertiggestellte Arbeit, deren Tickets nicht mehr auf dem Board auftauchen, habe ich hinzugerechnet (dadurch wird dieses Diagramm kumulativ).
3. Die Ergebnisse habe ich in einem Excel-Diagramm visualisiert, sodass die fertiggestellte Arbeit unten dargestellt wird.

Heutzutage bekommt man diese Diagramme auch einfach von seinem favorisierten Online-Kanban-Werkzeug bereitgestellt.

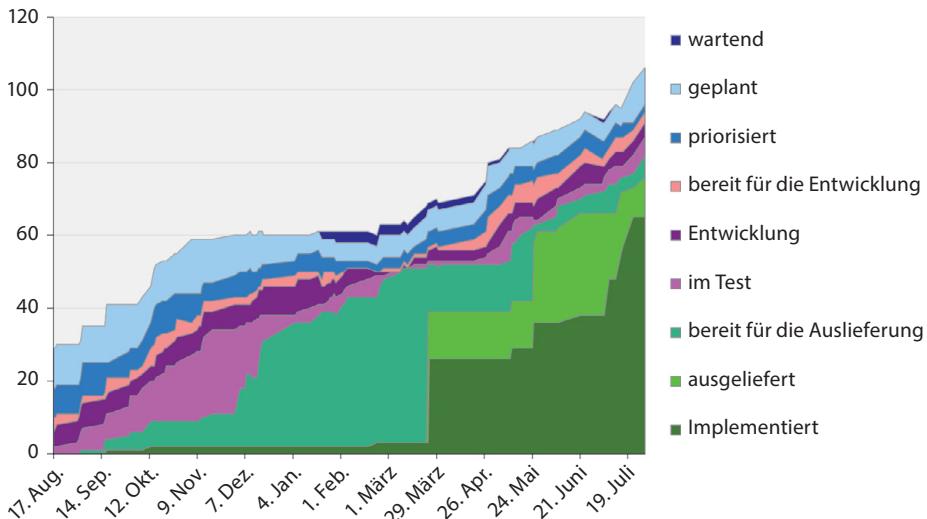


Abb. 1–2 Cumulative Flow Diagram (CFD) meines Projekts

Selbst wenn Sie noch nie eines dieser Diagramme gesehen haben, können Sie wahrscheinlich das »Aufblähen« von Work in Progress erkennen, das so typisch

für den Beginn eines Projekts ist, dann die treppenartigen Übergänge großer *Batches* von Status zu Status und gegen Ende die eher kontinuierliche Art der Auslieferung.

Ja, es gab einige deutliche Unebenheiten im Verlauf, aber es war gut zu sehen, dass wir erfolgreich sind. Ich konnte auch an anderen Dingen erkennen, dass eine Auslieferung erfolgt ist, aber dieses Diagramm gab mir ein klares Feedback darüber, dass sich die Durchlaufzeit durch den Prozess, der Durchsatz des Prozesses und sogar die Art der Auslieferung zu einem kontinuierlichen Stil hin veränderten. All das in nur einem Diagramm!

Andere Visualisierungen und Metriken helfen Anwendern beim Verstehen, wie Zeiten verteilt sind, und geben damit Einsicht sowohl in die interne Funktionsweise des Prozesses als auch darauf, wie die Dienstleistung vom Kunden erfahren wird.

1.4 Transparenz als Wert

Transparenz in Kanban ist zielgerichtet. Das heißt, sie ist absichtlich enthalten, um Menschen dabei zu helfen, gute Entscheidungen sowohl bei ihrer tagtäglichen Arbeit als auch bezüglich des Systems zu treffen. Gut ausgeführt unterstützt sie die Selbstorganisation – fügt einen Sinn, Anpassungsfähigkeit und Resilienz hinzu – und sie gibt den Menschen die Rückmeldung, dass Fortschritt erzielt wird. Auf mehreren Ebenen fordert sie heraus, dass die darunterliegende Logik artikuliert, hinterfragt und verfeinert wird.

Soweit es weitere Erwägungen wie **Respekt** erlauben, bedeutet mehr Transparenz, dass mehr Dinge an die Oberfläche gebracht werden und es damit mehr Anlässe zur Veränderung gibt. Und je reifer Organisationen werden, desto mehr steigt hoffentlich ihr Appetit auf Transparenz. Mehr ist mehr, um es etwas allgemeiner auszudrücken.

Transparenz alleine reicht aber nicht aus und sie ist auch kein Ersatz für effektive **Kooperation**; Veränderung ohne **Vereinbarung** ist nicht dauerhaft. Manchmal kombinieren wir Transparenz absichtlich mit anderen Werten, beispielsweise mit **Kundenfokus** im Replenishment-Meeting. Kurz gesagt: Der Schlüssel zum Erweitern der Grenzen der Transparenz in Ihrer Organisation könnte in den anderen acht Werten liegen.

2 Balance

2.1 Kernpraktik 2: Parallele Arbeit (Work in Progress, WIP) limitieren

Balance ist ein Wert, der sehr eng mit Kanbans zweiter Kernpraktik verknüpft ist:

KP2: Limitiere die Menge paralleler Arbeit (Work in Progress, WIP).

Der erste Teil dieses Kapitels zeigt, wie WIP-Limits auf einem Kanban-Board als Koordinationsmechanismus funktionieren, um ein *Pull-System* zu implementieren. Später erkunden wir, wo Balance in Kanban noch vorkommt.

2.2 Pull-Systeme in der Wissensarbeit

Sehen wir uns die »TEST«-Spalte auf dem Board in Abbildung 2-1 an.

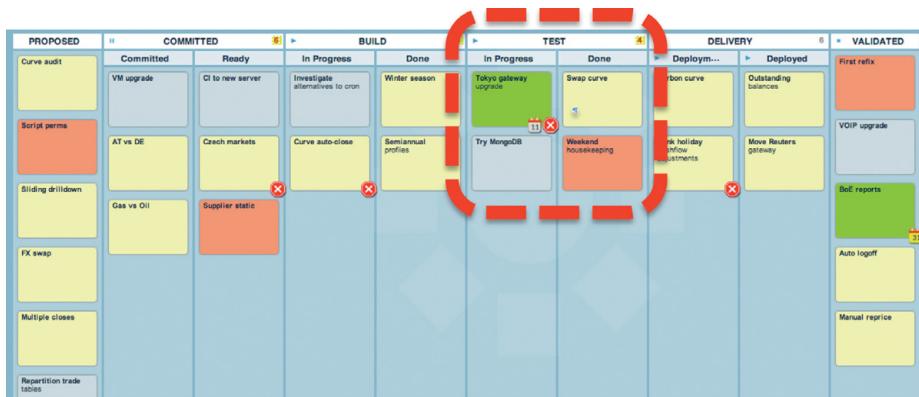


Abb. 2-1 Die TEST-Spalte besitzt ein WIP-Limit von 4.

In der rechten oberen Ecke der »TEST«-Spalte befindet sich eine »4«. Dies ist das sogenannte WIP-Limit: eine Begrenzung der Anzahl der Arbeitspakete, die sich zur gleichen Zeit in diesem Bereich befinden dürfen. In diesem Beispiel sollte die Gesamtanzahl der Arbeitspakete in der »TEST«-Spalte (inklusive der beiden Unterspalten mit den Überschriften »In Progress« und »Done«) vier nicht über-

schreiten. Unser elektronisches Werkzeug hat das Limit gelb hervorgehoben, weil dieser Teil des Boards aktuell bis zur vollen Kapazität gefüllt ist.

Sehen Sie sich jetzt in Abbildung 2–2 an, was passiert, wenn Tickets nach rechts, sowohl innerhalb des Bereichs als auch (und das ist noch wichtiger) aus dem Bereich heraus, bewegt werden.

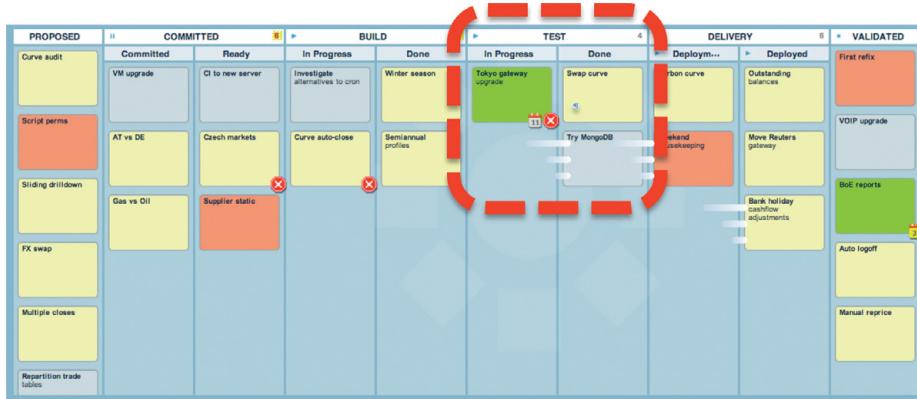


Abb. 2–2 Zwei Tickets bewegen sich nach rechts über das Board.

Der Testbereich des Boards ist nicht mehr voll. Der freie Platz, der in der Spalte »In Progress« entstanden ist, hat eine wichtige Bedeutung – er stellt ein Verfügbarkeitssignal dar, mit dem angezeigt wird, dass wir ein Ticket aus der vorhergehenden Spalte ziehen können.

So verhalten sich Pull-Systeme: Mit der Bewegung der Arbeit in Richtung der Fertigstellung (nach rechts auf dem Board) bewegen sich die Verfügbarkeitssignale stromaufwärts (nach links auf dem Board). Sie zeigen an, dass sich weitere Arbeitspakete vorwärts auf die frei gewordene Position begeben können, sobald sie bereit dazu sind. Dort, wo wir eine Reihe von Spalten mit (eigenen oder mit anderen Spalten geteilten) Limits haben, können diese Signale und Bewegungen sehr schnell stromaufwärts ziehen – das Gesamtsystem ist anscheinend in sich verbunden, was es ja auch sein sollte. Mit unserem Kanban-Board haben wir etwas Besonderes geschaffen: ein geeignetes Pull-System für unser unsichtbares Gut, die Wissensarbeit.

Ohne die WIP-Limits gehen die Pull-Signale verloren. Der nach links gerichtete Signalfluss stoppt, sobald er eine Spalte ohne ein Limit erreicht. In diesem Beispiel ist das die »PROPOSED«-Spalte am linken Ende des Boards.

Diese unbegrenzten (oder unendlichen) Warteschlangen haben eine besondere Bedeutung beim Messen der Zeit, die Arbeit benötigt, um sich durch das System zu bewegen. Wir nennen die Zeit, die ein Arbeitspaket durch die WIP-limitierten Spalten zwischen zwei unbegrenzten Warteschlangen benötigt, *Durchlaufzeit des Kanban-Systems*. Diese Durchlaufzeit kann sehr unterschiedlich zur

Kunden-Durchlaufzeit sein, der Durchlaufzeit, die aus Perspektive des Kunden gemessen wird.¹

2.3 Balance zwischen Arbeitslast und Kapazität

Sehen Sie sich das Board in Abbildung 2–2 aus der Sicht der Mitarbeiter an, die am ehesten verantwortlich für die TEST-Spalte sind. Sie könnten Testspezialisten sein oder dieselben Entwickler oder Analysten, die schon früher im Prozess für diese Arbeit verantwortlich waren; aufgrund der Selbstorganisation drückt das Board das nicht aus. Aber eine Sache stellt das Board sicher, dass die Menge an Arbeit in diesem Bereich die Kapazität nicht überschreiten wird (oder zumindest nicht sollte).

Das bedeutet, dass wir wichtige Schritte gemacht haben, um eine Überlastung von Mitarbeitern zu verhindern, indem wir eine Situation vermeiden, die im besten Fall unproduktiv und im schlechtesten inhuman ist. Sobald das Fertigstellen von Aufgaben höher priorisiert ist als das Beginnen neuer Arbeit, ist das Ergebnis eine Verbesserung der Qualität durch einen verbesserten Fokus und es wird Platz geschaffen. Und durch weniger Arbeitspakete in jedem Teil der Pipeline wird Arbeit im Allgemeinen sehr viel schneller fertiggestellt und man erhält sehr viel früher Feedback.

Von Zeit zu Zeit wird der Eindruck entstehen, die WIP-Limits seien falsch gesetzt. Zu hoch gesetzt scheinen sie wenig Auswirkung auf den Arbeitsfluss zu haben. Sieht man aber genau hin, wird deutlich, dass einige Arbeitspakete steckengeblieben sind, weil Personal fehlt, sie zu bearbeiten. Sind die Limits zu niedrig gesetzt, könnte ein zu großer Anteil von Arbeit zu einem beliebigen Zeitpunkt *blockiert* sein. Vielleicht sogar bis zu dem Punkt, an dem das System zu »verhungern« droht, der technische Ausdruck dafür, dass die Mitarbeiter nicht ausreichend Arbeit haben.

Diese Situationen sollten Auslöser für Diskussionen und Nachforschungen sein und gegebenenfalls korrigierende Aktionen der einen oder anderen Art nach sich ziehen. Der natürliche Reflex wäre vielleicht, das Limit anzupassen. Das sollte aber nicht zu voreilig geschehen. Eher sollte man sicherstellen, dass alle Beteiligten verstehen, welche Folge von Ereignissen zur aktuellen Situation geführt hat, und von dort aus weiterarbeiten.

WIP-Limits sollten am besten nicht nur als Hebel zum Durchsetzen von Regeln verstanden werden, sondern vielmehr auch als Feedbackmechanismen und Treiber für systemweite Verbesserungen. Wenn Sie den Work in Progress

1. Begriffe wie Durchlaufzeit und Cycle Time sollten vorsichtig verwendet werden. Es ist normalerweise empfehlenswert, sie in irgendeiner Weise zu qualifizieren, weil ihre Definition von der jeweiligen Sichtweise abhängt. Es ist auch hilfreich, wenn sie an einem wohldefinierten Ausgangspunkt starten und eindeutig in einem festgelegten Zustand von Fertigstellung, Auslieferung oder Validierung enden. Man sollte berücksichtigen, dass die technische Definition für verschiedene dieser Begriffe in der Literatur variiert und dies auch in Zukunft sicher zu einer Menge Verwirrung führen wird.

reduzieren, treten andere Probleme viel deutlicher zutage (und ja, das kann auch ein wenig schmerhaft sein). Behebt man diese, kann man den WIP weiter reduzieren, er könnte sogar von alleine sinken². Das ist ein weiterer sich selbst verstärkender und machtvoller Zyklus, der auch bei Toyota über mehrere Jahrzehnte erfolgreich genutzt wurde³.

Stellen wir diesen positiven Zyklus dem Teufelskreis gegenüber, der aus dem Denken heraus entsteht, dass zuallererst Mitarbeiter zu beschäftigen sind, dann ist die Standardantwort auf blockierte Arbeit nicht, das Problem zu lösen. Stattdessen wird neue Arbeit angefangen und damit der Work in Progress im System erhöht. Mehr WIP im System bedeutet, öfter darauf zu warten, dass andere fertig werden. Das Problem wird verstärkt. Verzögerungen und Multitasking haben beide negative Auswirkungen auf die Qualität, was wiederum zu – Sie können es schon erraten – mehr Nacharbeit, mehr Blockaden und mehr Work in Progress führt. Zu viel Arbeit gepaart mit geringer Qualität – ist das eine Kombination, die Sie vielleicht wiedererkennen?

2.4 Andere Möglichkeiten, WIP zu limitieren

Ich möchte nicht den Eindruck erwecken, dass Spalten-WIP-Limits die einzige Möglichkeit sind, Work in Progress zu limitieren. Sie sind zwar mächtig, aber manchmal funktionieren sie am besten in Kombination mit anderen Mechanismen. Diese fallen typischerweise in zwei Kategorien:

1. Begrenzung der Batch-Größe – die Größe (Budget und Dauer) von Projekten reduzieren, die Zeit zwischen zwei Releases, die Dauer von Sprints, die Größe von Features usw.
2. Steuerung der parallel ablaufenden Menge an Arbeitspaketen – Reduktion von Businessinitiativen (an denen sich die Projekte orientieren), der Anzahl von Projekten, an denen ein Team oder eine Abteilung parallel arbeitet, der Anzahl von Arbeitspaketen pro Person usw.

Wir haben jede Menge Hebel zur Verfügung! Drückt man den richtigen, wird es einfacher (sowohl psychologisch als auch technisch), andere zu bewegen. Begrenzt man die Menge paralleler Arbeit auf verschiedene Weisen, lässt sie sich auch auf ein Niveau senken, das vorher als unerreichbar erschien.

Die Situation in Budapest, mit der Kapitel 1 beginnt, entstand nicht über Nacht. Ich hatte mich einer Organisation angeschlossen, die süchtig nach Work in Progress war. Wie das in vielen jungen Unternehmen so ist, war es sehr schwer, »Nein« zu sagen. Unglücklicherweise passierte das nicht nur im Kundenkontakt, sondern hatte sich auch als interne Gewohnheit etabliert. Meeting für Meeting wuchsen die vielen existierenden To-do-Listen.

-
2. Kapitel 17 gibt mit dem *Gesetz von Little* eine Erklärung für diesen Effekt.
 3. Das Toyota-Produktionssystem (TPS) ist eines der Modelle, die in Teil II, Kapitel 14 erläutert werden.

Nach ein paar Wochen in meiner neuen Rolle und einem recht konventionellen Anwenden von Kanban in meinem Teil der Organisation gab es einen Schlüsselmoment in einem Managementmeeting. Mark Dickenson, der Geschäftsführer, gab bekannt, dass To-do-Listen der Geschichte angehörten und damit unverzüglich aufgegeben werden müssten. Es wäre eine massive Untertreibung zu sagen, dass ich darüber erfreut war – ich erkannte zu diesem Zeitpunkt, dass dies für uns alle ein bedeutsamer Durchbruch war.

Es gab noch weitere solche Durchbrüche. Einige Monate später unternahmen wir Schritte, um die Menge der Businessinitiativen auf die paar wenigen zu reduzieren, die die Geschäftsstrategie am effektivsten unterstützten. Dies hatte Auswirkungen auf die Projekte, von denen einige angehalten wurden, andere wurden komplett abgebrochen. Irgendwann hatten wir endlich die Bandbreite, die es uns ermöglichte, die Projekte ordentlich zu betrachten. Wir stellten fest, dass es Projekte gab, die auf Bearbeitung warteten und die viel wichtiger waren, als irgendwer gedacht hätte. Eines war sogar von kritischer Wichtigkeit.

Es war ermutigend, dass viele De-Priorisierungen nicht darauf warteten, dass ich (in der IT) mich für sie einsetzte – in mehreren Fällen wurden sie von ihren Sponsoren (meinen Kollegen im Managementteam) freiwillig vorgeschlagen. Ich weiß nicht, wie viele Projekte Sie schon gesehen haben, die nach dem Start auf diese Weise gestoppt wurden. Meiner Erfahrung nach passiert es aber selten und man weiß, dass etwas Besonderes geschieht, wenn es auch noch wiederholt vorkommt.

2.5 Balance zwischen dringlichkeits- und termingetriebener Arbeit

Nicht jede Arbeit ist gleich, das gilt speziell für Wissensarbeit. Viele Manager versuchen Vielfalt zu leugnen oder sie wollen sie wegorganisieren, ohne zu verstehen, dass es besser sein kann, sie zu akzeptieren.

Lassen Sie uns einen genaueren Blick auf einen kleinen Teil unseres Boards, wie in Abbildung 2-3 gezeigt, werfen.



Abb. 2-3 Die Arbeitspakete in »TEST«

Kümmern Sie sich dabei weniger um die Namen der drei Arbeitspakete, sondern konzentrieren Sie sich auf die visuellen Unterschiede. Fürs Erste schauen wir uns die beiden obersten an: »Tokyo gateway upgrade« und »Swap curve«.

»Tokyo gateway upgrade« ist ein grünes Ticket mit einem Kalendersymbol – es ist ein termingetriebenes Arbeitspaket. In diesem Beispiel ändert sich an einem bestimmten Tag die Schnittstelle zur Tokyoter Börse und unser System muss für diese Änderung angepasst werden. Falls diese Änderung zu spät geliefert werden würde, könnten wir in diesem speziellen Markt nicht mehr handeln. Eine Situation, die für die Organisation teuer werden könnte. Eine frühere Auslieferung bietet aber keinen zusätzlichen Wert, im Gegenteil: Es könnte sogar schaden.

»Swap curve« ist anders. Es beschreibt ein völlig unabhängiges Stück Funktion. Je früher sie geliefert wird, desto früher kann Nutzen daraus gezogen werden. Im Gegensatz zu unserem vorherigen Beispiel ist dieses Arbeitspaket nicht termingetrieben, sondern dringlichkeitsgetrieben.

Wenn wir die dringende Arbeit sicher vor der termingetriebenen Arbeit ausliefern können, ist das ein Gewinn. Die termingetriebene Arbeit bekommt jedoch Priorität, sollte der Termin in Gefahr sein. So oder so, wir erhalten gute Ergebnisse.⁴

Behandeln wir beide gleich, passieren ungute Dinge. Geben wir der von Dringlichkeit getriebenen Arbeit einen willkürlichen Termin, gefährden wir womöglich die wirklich termingetriebene Arbeit. Behandeln wir beide Arbeitspakete als dringend, vernachlässigen wir das zeitliche Risiko. Keine der beiden Herangehensweisen fördert eine gute Entscheidungsfindung.

Zu viele Teams leiden unnötig darunter, dass Arbeit an Termine gebunden wird. Entweder weil zu viel Arbeit in eine Timebox gepresst wird oder – schlimmer – jedem einzelnen Arbeitspaket ein Termin gegeben wird. Aus Schätzungen werden Zusagen. Produktivitätsziele dehnen Zusagen bis zur Belastungsgrenze aus und das noch, bevor es zu Störungen kommt.

2.6 Risikobasierte Kategorisierung und Serviceklassen

»Try MongoDB« (das übrig gebliebene Arbeitspaket) klingt weder termingebunden noch dringend. Ist es deswegen unwichtig? Es ist schwierig, eine sichere Aussage zu treffen. Aber wo werden wir in ein oder zwei Jahren stehen, wenn wir nie das Risiko eingehen, zu experimentieren, und nie die längerfristige Arbeit im Hintergrund machen, um unsere Fähigkeiten in Bezug auf Plattform, Produkt, Prozess oder Menschen zu entwickeln?

Ziehen wir unterschiedliche Arten von Arbeit oder unterschiedliche Arbeitsabläufe nicht in Betracht und berücksichtigen nur Termineigenschaften, decken

4. Ich vereinfache hier ein wenig – termingetriebene Arbeitspakete sind nicht immer so wertvoll, dass das Bedürfnis, sie pünktlich auszuliefern, alle anderen Überlegungen übertrumpft. In Kapitel 15 schauen wir etwas genauer auf die Ökonomie solcher Entscheidungen.

unsere Beispiele drei qualitative Kategorien ab. Kanban-Implementierungen besitzen meist vier:

■ **Beschleunigt**

Sehr dringende Arbeitspakete, für die wir aktuelle Arbeit unterbrechen, um uns sofort darum zu kümmern.

■ **Termingeschrieben (oder Fester Termin)**

Arbeit, deren Verzögerung nach einem bestimmten Datum zu einer erheblichen Strafe führt, die unverhältnismäßig zu jedem Gewinn ist, der aus verfrühter Lieferung entsteht. Ihre Risiken in der zeitlichen Planung werden aktiv gesteuert.

■ **Standard (oder Regulär⁵)**

Dringlichkeitsgetriebene Arbeit, die in einer mit dem Kunden abgesprochenen Reihenfolge geliefert oder durch eine Systemregel sequenziert wird. Abhängig vom Kontext können passende Regeln so einfach wie *first in, first out* (FIFO) sein, auf *Verzögerungskosten* (Kap. 15) basieren oder der persönlichen Wahl überlassen werden.

■ **Unbestimmbare Kosten**

Ausbau von Fähigkeiten, Experimente zu Technologien oder zum Markt, Investitionen in Mitarbeiter – immaterielle Arbeit, die mittel- und langfristig essenziell ist, deren direkter und kurzfristiger Geschäftswert sich jedoch schwierig messen lässt.

Kategorisierungen nach anderen Risikodimensionen sind auch möglich. Das in Ihrer Organisation bestehende Planungssystem hat womöglich schon eigene Kategorien – Abwandlungen wie Innovationen, Wartung, Support und Regulierung sind beispielsweise in Banken üblich.

Wenn Kategorisierungen auf unterschiedlichen Leistungsversprechen basieren, nennen wir diese *Serviceklassen*. Extern helfen wir den Kunden, die aus ihrer Sicht richtige Serviceklasse zu wählen, die am besten die Balance zwischen Risiko und Ergebnis reflektiert. Intern leiten Planungsregeln die Entscheidung zwischen Arbeitspaketen, um gute Ergebnisse zu erzielen. Mit einem gesunden Mix aus Arbeit, insbesondere mit genügend Puffer aus immaterieller und Standardarbeit, kann eine überraschend genaue Vorhersagbarkeit erreicht werden.

5. Marcus Hammarberg und Joakim Sundén empfehlen das weniger belastende Wort *Regulär* als das üblichere *Standard* und ich würde mich freuen, wenn es sich durchsetzt.

2.7 Balance zwischen Anforderungen und Leistungsfähigkeit

Innerhalb unseres Systems balancieren wir Arbeitslast und die Kapazität unseres Systems aus. Davon profitieren zum einen die Menschen, die im System arbeiten, zum anderen erreichen wir eine verbesserte Leistung und Vorhersagbarkeit. Damit muss es aber nicht aufhören.

Interessante Dinge passieren, wenn Sie die Leistungsfähigkeit Ihres Systems kennen, Tickets der einzelne Kategorien zu liefern: Sie können Ihren Kunden helfen, informiertere Entscheidungen zu treffen. Das hat Auswirkungen auf vorgelagerte Prozesse, und zwar darauf, wie Arbeit vom Kunden in das System fließt. Wird dies aktiv gemanagt, können die Anforderungen an das System über die Zeit geformt werden. Dieses Formen führt zu einem weiteren Ausbreiten der Balance über größere Bereiche und längere Zeithorizonte und damit zu einer weiteren Verbesserung der Ergebnisse.

Die Kategorisierung von Arbeitspaketen und Serviceklassen hilft, mehrere Zeithorizonte gleichzeitig zu steuern. Es ergibt keinen Sinn, immer mit herausragenden Lieferraten zu glänzen, denn man wirtschaftet sich schnell in den Ruin, wenn der Nachhaltigkeit keine Aufmerksamkeit geschenkt wird. Ebenso können wir das Geld unserer Investoren nicht nur für langfristige Arbeiten verwenden, die so nebulös oder bombastisch sind, dass nie ein Kundenwert geliefert werden wird.

Zusammengefasst hilft Kanban dabei, die Balance zwischen Anforderungen und Leistungsfähigkeit über mehrere Zeitspannen zu erreichen. Dies ist eine mächtige Managementstrategie, die innerhalb und außerhalb des Systems anwendbar ist.

2.8 Stakeholder-Balance

Eine Balance zwischen den Interessen verschiedener Stakeholder zu erreichen – Teammitglieder, Kunden, Topmanagement, Gesellschafter oder selbst die Gesellschaft im weiteren Sinne –, kann ziemlich anspruchsvoll sein.

Niemand hat die Kapazität, die Anliegen aller Stakeholder zu jeder Zeit gegeneinander abwägen zu können. Eine Daumenregel kann jedoch den richtigen Grundton setzen: Wenn eine sogenannte Verbesserung auf Kosten einer dieser Gruppen geht, denken Sie noch einmal nach und strengen Sie sich mehr an. Wenn es sich anfühlt, als würde ein Nullsummenspiel gespielt, passen Sie auf. »Gutes für Kunden, für die Organisation und für die Menschen, die die Arbeit verrichten« hat einen besseren Klang.

Verbesserungen, die diese Regel nicht beachten, scheitern häufig. Kunden haben nur eine begrenzte Toleranz für sich verschlechternde Produkte und Dienstleistungen. Von Organisationen kann man meist nicht erwarten, dass sie Veränderungen ohne Nachfrage zu weiterführenden Auswirkungen akzeptieren.

Mitarbeiter werden schlechter werdende Arbeitsumgebungen irgendwann verlassen; Produktivität, Qualität und Wohlbefinden nehmen derweil ab.

2.9 Balance suchen

Balance ist etwas Seltsames. Wir fühlen uns wirklich wohl, wenn sie da ist. Sie zu erreichen, benötigt jedoch Voraussicht, Wachsamkeit und Aufwand und manchmal sogar den gelegentlichen Durchbruch. Neben den bedeutenden technischen Vorteilen von WIP-limitierten Pull-Systemen ist es genau das, was Balance für mich zu einem so wichtigen Wert macht.

Um Balance in die Anwendung von Kernpraktik 2 zu bringen, versuchen Sie, die Phrase »Finde Wege« vor die Praktik zu stellen:

- Finden Sie Wege, die Menge paralleler Arbeit zu limitieren. Verwende dafür jeden verfügbaren Hebel.
- Finde Wege, die Menge paralleler Arbeit auf allen Ebenen der Organisation zu limitieren, und suche nach Wegen, die tiefer liegenden Ungleichgewichte aufzudecken, sobald das gegenseitige Vertrauen wächst.

Vielleicht finden Sie eine Formulierung, die für Ihre Organisation die Notwendigkeit für Balance in der Essenz erfasst. Es ist ein Trick, der auch bei anderen Praktiken funktioniert.

3 Kooperation

Wir spulen jetzt vor zu Kanbans sechster Kernpraktik, um **Kooperation** einzuführen. Kooperation ist ein enger Verbündeter der **Transparenz** und der **Balance**; bei allen dreien geht es um das Vorantreiben, Herbeiführen und Auslösen von Veränderung, und deren wachsende Präsenz ist ein guter Indikator, dass Dinge nachhaltiger werden.

3.1 Kernpraktik 6: **Erziele Verbesserung kooperativ und entwickle experimentell**

Diese Praktik enthält Kleingedrucktes. Die ganze Formulierung lautet:

KP6: Erziele Verbesserung kooperativ und entwickle experimentell (verwende dazu Modelle und die wissenschaftliche Methode).

Wir beschäftigen uns zuerst mit »erziele Verbesserung kooperativ«, dann sehen wir uns die Teile »entwickle experimentell« und »die wissenschaftliche Methode« gemeinsam an. Modelle sind Bestandteil von Teil II.

3.2 Kooperation ist eine ernste Sache

Ich finde es hilfreich, bei **Kooperation** an etwas recht Konkretes zu denken und mir dabei einige Beispiele berühmter kreativer Kooperationen ins Gedächtnis zu rufen. Lennon und McCartney, Watson und Crick, Marie und Pierre Curie – das sind Kooperationen, die enorme Auswirkungen hatten, und das nicht nur in ihrem speziellen Bereich, sondern auch in der allgemeinen Wahrnehmung. Es handelt sich dabei nicht nur um Menschen, die nett waren und generell miteinander gearbeitet haben; es sind Beziehungen, in denen das Ganze irgendwie größer ist als die Summe der Teile, wo kreative Energie sowohl zwischen den Involvierten existiert als auch in jedem Einzelnen.

Wir können nicht erwarten, dass jede Kooperation am Arbeitsplatz so spektakulär produktiv ist wie diese Beispiele, aber wenn unsere wissensbasierten

Organisationen nicht mehr Kreativität generieren kann als die jedes Einzelnen alleine, warum sollten Firmen dann noch existieren?¹

3.3 Erziele Verbesserung kooperativ

Genau so, wie man von einer effektiven Organisation erwarten würde, dass sie Kooperation in der Auslieferung ihrer Produkte und Dienste verwendet, heißt »erziele Verbesserung kooperativ«, dass ein Teil dieser kreativen Energie dafür verwendet wird, das darunterliegende Liefersystem zu verbessern.

Das wirft ein neues Licht auf einige der Konzepte zu **Transparenz**, die in Kapitel 1 besprochen wurden. *Feedbackzyklen* müssen mehrere Personen involvieren. Wenn Teams *selbstorganisiert* sind, entwickeln sich Verbesserungen und andere opportunistische Innovationen aus Kooperationen, die meistens recht natürlich entstehen, wenn die passenden Gruppen von Menschen beteiligt werden.

Es ist allerdings kein Widerspruch, wenn wir sagen, dass wir daran auch arbeiten müssen. Kontinuierliche, selbstorganisierte Verbesserung implementiert sich meistens nicht von selbst über Nacht. Manager hoffen manchmal, dass eine einfache Ankündigung neuer Regeln es schon richten wird; fast immer werden sie enttäuscht. Vielleicht haben Sie das auch schon selbst erlebt: Ein Meeting oder zwei, vielleicht ein paar Verbesserungen, dann verblasst der Elan, sobald klar wird, dass es gar nicht als »echte Arbeit« zählt. Wenn beim nächsten Mal der Aufruf zur Verbesserung erfolgt, ist die mittlerweile zynische Belegschaft nicht weiter interessiert.

In Kapitel 6 sehen wir uns an, welche Rolle **Führung** bei der Entwicklung dieser organisatorischen Fähigkeiten spielt. Zwischenzeitlich schauen wir uns ein Beispiel für eine Prozessveränderung aus der Praxis an, das im Kern Kooperation aufweist.

3.4 Zu Kooperation ermutigen

Ich habe mittlerweile bei einer Reihe von Gelegenheiten sehen können, wie verschiedene Formen des *Peer-Reviews* zu signifikanter Frustration führen können. Bei UBS waren wir stolz auf die Qualität unseres Codes und ein *Codereview* (eine Art Peer-Review) wurde abgehalten, um sowohl Qualität »durchzusetzen« als auch »Good Practices« zu verbreiten. In diesen beiden Punkten war die Praktik sehr erfolgreich.

1. Warum existieren Firmen überhaupt noch? Ökonomen streiten sich immer noch darüber. Siehe http://en.wikipedia.org/wiki/Theory_of_the_firm und http://en.wikipedia.org/wiki/Knowledge-based_theory_of_the_firm.

Es gab allerdings ein Problem: Code steckte tagelang (manchmal länger) fest und wartete darauf, dass der zugeteilte Senior-Entwickler Zeit für das Review finden würde. Diese Reviewer packten mehrere Reviews in ein großes Paket, weil es so »viel effizienter« sei. Das Ergebnis dieser Reviews war häufig signifikante Nacharbeit, die echte Kosten und zusätzliche Verzögerung mit sich brachte. Mehrere Verzögerungen und Nacharbeit – ein Rezept für Mühe und Qual.

Was war die Lösung? Kooperation! Der formale Prozess, den unser elektronisches Werkzeug unterstützte, musste in diesem Fall kaum verändert werden. Es reichte aus, dass wir uns darauf einigten, Verzögerungen und Nacharbeit als Beweis dafür anzusehen, dass Entwickler und Reviewer nicht ausreichend miteinander kooperiert hatten, während der Code geschrieben wurde.

Das klingt vielleicht trivial, aber der psychologische Effekt war groß. Vorher oblag es dem Entwickler, seinen Code beim Reviewer durchzubekommen. Codereview war mit anderen Worten eine Hürde. Jetzt füllte das informale Prinzip »keine Überraschungen« die Rolle einer *explizit gemachten Prozessregel* aus und beide Parteien fühlten die gemeinsame Verpflichtung, für einen ausgeglichenen Fluss der Arbeit zu sorgen.

Dieser kooperative Ansatz für das Problem der Reviews kann noch viel breiter umgesetzt werden und wir kommen in Kapitel 5 darauf zurück.

3.5 Fokus auf Kooperation legen

Man kann aus dem letzten Beispiel durchaus die Lehre ziehen, dass Reviews problematisch sein können – ich bin diesem Punkt schon oft genug begegnet, dass ich eine generelle Tendenz ableiten kann. Man kann aber noch etwas Tieferliegendes lernen: Es zahlt sich aus, die *Qualität* der Interaktionen in einem Prozess anzusehen und nicht nur wie viele davon stattfinden oder in welcher Reihenfolge sie vorkommen.

Das ist mehr als nur eine Technik, die man im Kontext einer bestimmten Verbesserungsmöglichkeit anwenden kann – es ist eine eigenständige Strategie. Schauen Sie sich die Qualität der Interaktionen um sich herum an. Erkennen Sie Schlüsselprozesse, die durch Interaktionen von geringer Qualität, Verzögerungen, Hin-und-her-Konversationen, Nacharbeit und Frustration gebremst werden? Falls das noch nicht schlimm genug ist: Sehen Sie Individuen oder Teams, die anscheinend selten die Möglichkeit haben, sich in einer qualitativ hochwertigen Kooperation einzubringen? Diese zu identifizieren könnte ein Auslöser für etwas ganz Besonderes sein.

Investieren Sie dann in die Fähigkeiten und Methoden der Kooperation. Die Agile-Community verdient wirklich Anerkennung dafür, dass sie diesem Thema Priorität eingeräumt hat; wir gehen in Teil II auf einige wichtige Kooperationswerkzeuge näher ein. In Teil III kommen wir darauf zurück und zeigen, wie einige

dieser Werkzeuge verwendet werden können, um den Kontext für eine Kanban-Implementierung oder Verbesserungen zu setzen oder zu erneuern.

Einstweilen gilt: Wenn Sie das nächste Mal von etwas überrascht oder frustriert werden, was wie ein Fehler einer einzelnen Person oder ein Fehler in einem Prozess aussieht, fragen Sie sich, ob es an der fehlenden Kooperation liegen kann. Drehen Sie »Manager mögen keine Überraschungen« zu etwas Positivem, indem Sie sich und die Menschen um Sie herum trainieren, diese Frage automatisch zu stellen.

3.6 Entwickle experimentell

Wenn »erziele Verbesserung kooperativ« dafür steht, wie Veränderung vorangetrieben wird, steht »entwickle experimentell« dafür, wie sie durchgeführt wird.

Kanban hat mit anderen Frameworks das Konzept eines Prozesses zur Problemlösung gemeinsam, der iterativ in einem Verbesserungszyklus angewendet wird. Die Engpasstheorie hat beispielsweise den POOGI (Process of ongoing Improvement); Six Sigma hat den DMAIC-Kreis. Es gibt noch viele mehr – den eigenen Verbesserungszyklus zu beschreiben, scheint ein beliebter Weg zu sein, sich von anderen Frameworks zu differenzieren.

Die Kanban-Community ist mit dem Großväterchen aller Verbesserungszyklen glücklich, dem kanonischen Verbesserungszyklus, bekannt auch als *Deming-Zyklus*, *Shewhart-Zyklus* (so nannte W. Edwards Deming ihn nach Walter A. Shewhart, dessen Ideen er weiterentwickelte), *PDCA-Zyklus* oder einfach nur *PDCA*. Das Akronym steht für »*Plan, Do, Check, Act*«; manchmal sieht man auch *PDSA* für »*Plan, Do, Study, Act*«.

Ein Hinweis zur Vorsicht: Die Wörter *Plan*, *Do*, *Check* und *Act* stehen nicht für das, was Sie vielleicht denken, wofür sie stehen – sie lenken sogar ein wenig in die falsche Richtung – bis man sie zusammen mit dem Wort »*Experiment*« verwendet:

- Plane (**Plan**) ein Experiment (basierend auf einer Hypothese).
- Führe das Experiment durch (**Do**).
- Prüfe (**Check** oder **Study**) die Resultate oder Ergebnisse des Experiments.
- Agiere (**Act**) nach den Resultaten, indem du die Hypothese oder das System entsprechend veränderst und deine Lernresultate mit anderen angemessen teilst.

Es erfordert einiges an Disziplin, um Veränderungen nach wissenschaftlicher Vorgehensweise durchzuführen, aber es gibt dem Wort »ergebnisorientiert« sicherlich eine neue, weitere Bedeutung!

3.7 Die und wir

Man könnte sagen, dass ein Team, das ein Problem gelöst hat, ein stärkeres Team ist. Das ist normalerweise ein Grund zum Feiern. Es gibt aber einen Punkt, über den hinaus die positiven Aspekte der Kooperation durch negative Aspekte geschmälert werden: Cliquenbildung, konkurrierendes Verhalten, Verstecken von Informationen, unfaire Annahmen über die Motive und Intentionen Außenstehender sowie »Blasen«, deren Wände zu Barrieren werden.

Dies scheint ein schwieriges Problem zu sein. Eines, das tief in der menschlichen Natur verwurzelt ist und sicherlich außerhalb der Domäne der Kanban-Methode liegt. Oder etwa nicht?

Kanban-Anwender stellen sich diesen negativen Teamverhaltensweisen eher nicht entgegen. Wir versuchen auch nicht, Teamzusammensetzungen oder die Wahrnehmung der Teammitglieder über die eigene Teamidentität so früh wie möglich zu klären. Stattdessen versuchen wir, diese Konflikte zu umgehen, indem wir die Arbeit so betrachten, dass organisatorische Strukturen und individuelle Rollen nicht besonders betont werden. Dies können Sie schon als nützliche Designrichtlinie für die Implementierung der **Transparenz, Balance und Kooperation** verwenden. Diese überraschend effektive Strategie wird jedoch immer deutlicher, wenn wir die restlichen sechs Werte untersuchen.

Reflexion:

Transparenz, Balance und Kooperation

Wir haben drei der neun Werte betrachtet und es ist Zeit, eine Bilanz zu ziehen. Gehen Sie einen Schritt zurück und reflektieren Sie für einen Moment:

- Mit welchen Praktiken werden bei Ihnen aus der Organisation heraus Veränderungen vorangetrieben? Welche Intentionen, Annahmen und Werte liegen hinter diesen Praktiken?
- Wie bereitwillig akzeptiert Ihre Organisation **Transparenz**? Wird Transparenz als Gefahr gesehen, gleichgültig behandelt oder als eine positive, treibende Kraft wertgeschätzt?
- Erreicht Ihre Organisation **Balance**? Versucht sie es überhaupt? Wie stark werden Teams und Einzelpersonen durch Ungleichgewichte beeinflusst? Wie viele können Sie identifizieren?
- Wird **Kooperation** in Ihrer Organisation so sehr geschätzt, dass sie gefördert und gezielt verbessert wird?

Welche der folgenden Maßnahmen wären in Ihrer aktuellen Umgebung am nützlichsten?

- **Transparenz:**
 - Arbeit sichtbar machen
 - Prozessregeln explizit machen
 - Feedbackzyklen implementieren (oder vorhandene Feedbackzyklen besser nutzen)
- **Balance:**
 - Einführung eines Pull-Systems
 - Einführung von risikobasierten Serviceklassen
 - Eine tragbare Zusammenstellung der Arbeit finden (nach Typ oder Zeithorizont)

■ Kooperation:

- Fördern gemeinschaftlicher Problemlösung
- Schritte unternehmen, die die Qualität der Interaktionen verbessern
- Verbesserungen als Experimente strukturieren

Wo würden Sie beginnen? Teil III (Implementierung) wird Ihnen ein paar Ideen dazu geben, falls Sie sich nicht sicher sind.

4 Kundenfokus

4.1 Kernpraktik 3: Manage den Arbeitsfluss

Ist das ein Fehler? Wie kommen wir von *Manage den Arbeitsfluss* zu **Kundenfokus**? Seien Sie für einen kleinen Moment nachsichtig mit mir — lassen Sie mich ein wenig schummeln und die Formulierung dieser Kernpraktik erweitern, um besser erklären zu können, was diese Kernpraktik wirklich bedeutet:

KP3 (erweitert): Manage den Arbeitsfluss, um Gleichmäßigkeit, Pünktlichkeit, und gute ökonomische Ergebnisse zu erreichen, und nehme Kundenbedürfnisse vorweg.

Der Schwerpunkt dieses Kapitels liegt auf Kundenbedürfnissen und wie man sie besser vorhersieht. Gleichmäßigkeit und Pünktlichkeit werden im nächsten Kapitel zu **Arbeitsfluss** behandelt. Behalten Sie »gute ökonomische Ergebnisse« im Hinterkopf, während Sie beide Kapitel lesen; ökonomische Entscheidungsfindung wird in Kapitel 15 behandelt.

4.2 Warum Kundenfokus?

Aufgabenfokus, Rollenfokus, Teamfokus, Projektfokus, Produktfokus, Technologiefokus ... und die Liste geht weiter. So viele Wege, die Sicht auf das zu verlieren, wofür wir eigentlich arbeiten!

In meinen Kursen gebe ich folgenden Rat:

Wissen Sie, was Sie liefern, an wen und warum.

Sie denken vielleicht, dass das nicht erwähnt werden müsste, aber es scheint wirklich ins Schwarze zu treffen. Kursteilnehmer sagen mir – unaufgefordert –, dass sie sich nie wirklich Gedanken darüber gemacht haben. Sie erwähnen es auf den Feedbackbögen als einen Schlüssepunkt, den sie mitnehmen. Es ist nicht so, dass die anderen Fokusse schlecht sind, der Kundenfokus hilft jedoch dabei, alle anderen in die richtige Perspektive zu bringen.

In diesem Kapitel untersuchen wir einige praktische Wege, bei denen der Kundenfokus den Arbeitsfluss verbessern kann, manchmal tut er das sogar erheblich. Wenn man darüber nachdenkt, ist es eigentlich gar nicht so überraschend: Wenn Sie aus der Perspektive eines anderen auf Ihre Arbeit schauen, ist es wahrscheinlich, dass Sie etwas Neues darüber lernen, wie sie funktioniert.

4.3 Zufriedenheit garantiert

Erinnern Sie sich an die Prozessregel aus dem Anfangsszenario von Kapitel 1:

- *Entwickler bleiben für Arbeitspakete verantwortlich, bis sie die Rückmeldung vom Kunden erhalten haben, dass sich die Pakete als wertvoll erwiesen haben.*

Diese Prozessregel wurde relativ spät ergänzt. Wir hatten einen Prozess entwickelt, der effektiv genug schien. Wir sammelten Anforderungen, bauten neue Funktionen, testeten diese und gaben sie frei. Nach einer Weile wurden wir anspruchsvoller: Wir fügten unserem Board eine weitere Spalte hinzu, um Funktionalität verfolgen zu können, die zwar veröffentlicht war, jedoch weitere Schritte benötigte, um als vollständig erachtet zu werden.

Zu häufig fanden wir heraus, dass Funktionen ausgeliefert worden waren, die nie verwendet wurden. Funktionen, die gewünscht worden waren! Wie konnte das passieren?

Unsere neue Prozessregel sollte etwas adressieren, von dem wir dachten, dass es schlechtes Kundenverhalten ist. Warum um Dinge bitten, die nicht gebraucht werden? Wie wäre es, uns Bescheid zu geben, wenn ihr eure Meinung ändert? Allerdings wurde schnell klar, dass die neue Prozessregel das Verhalten auf beiden Seiten änderte. Das Schließen einer Feedbackschleife war der Katalysator für die Zusammenarbeit mit dem Kunden (ein Wert direkt aus dem Agilen Manifest¹) auf einem Niveau, das wir vorher noch nicht erlebt hatten.

Alle wussten, dass der Prozess in einer womöglich schwierigen Konversation enden würde. Trotzdem gaben sich unsere Entwickler und internen Kunden gleichermaßen die größte Mühe, die finalen Einführungsschritte zu schaffen: Zeitpläne klären, Leute angemessen informiert und geschult haben, Stammdaten aufräumen usw. Wenn nötig, wurden diese Schritte schon vorab getestet, oftmals in Zusammenarbeit. Das wiederum beeinflusste die Art und Weise, wie entwickelt und spezifiziert wurde. Jetzt, da offensichtlich wurde, dass Erfolg von einem gemeinsamen Engagement abhing, änderte sich sogar, wie Arbeit am Anfang des Prozesses priorisiert wurde.

Ich übertreibe nicht, wenn ich sage, dass die Auswirkung dieser Regeländerung weit über das hinaus ging, was ich mir vorgestellt hatte. Ein wenig Demut ist angebracht: Wir hatten keine schlechten Kunden, nur schlechte Beziehungen, die nicht effektiv genug waren.

1. Siehe Kapitel 13 zum *Agile Manifesto*.

4.4 Quer über das Board

Unser Katalysator war eine Prozessregel, die (bildlich gesprochen) an der rechten Spalte unseres Kanban-Boards angebracht war und irgendwie den gesamten Prozess mit Kundenfokus infizierte. Um verstehen zu können, wie eine Umsetzung wie unsere möglicherweise wiederholbar ist, ist es hilfreich, das Board mit ein paar gezielten Fragen noch einmal zu betrachten, und zwar von rechts nach links, Spalte für Spalte:

- Wessen Bedürfnisse werden in diesem Teil des Prozesses untersucht? Wessen nicht und welche Risiken stellt dies dar?²
- Was lernen wir in dieser Phase, was wir nicht schon früher wissen oder lernen konnten? Auf welche Art und Weise helfen uns die Aktivitäten in dieser Phase zu erreichen, was gebraucht wird?
- Was muss noch gelernt werden? Werden Unklarheiten besser durch Vorangehen oder Zurückgehen behandelt?

Arbeiten wir uns mit dieser Logik zurück zum Beginn des Prozesses, entwickeln wir nicht mehr, um vorgegebene Anforderungen zu erfüllen, sondern damit Bedürfnisse erfüllt werden, die noch entdeckt und erforscht werden müssen. Wir blicken nicht andauernd zurück, um uns zu versichern, dass wir auch »korrekt nach Spezifikation« entwickeln, sondern schauen nach vorn, um Bedürfnisse zu erfüllen, die sich noch offenbaren. Wir vertrauen auch nicht übermäßig einem angeblich wasserdichten Prozess, um unsere Arbeit zu erledigen; stattdessen suchen wir nach Wegen, um effektiver lernen zu können.

Kreative Wissensarbeit handelt nicht nur davon, was wir schon wissen. Es ist ein *Prozess der Wissenserarbeitung*. Benutzen Sie Ihr Kanban-Board, um stets daran erinnert zu werden: »Was wissen wir nicht?«

4.5 Upstream-Kanban

Probieren Sie mal eines der ambitionierteren Board-Designs aus, das ich für meine persönliche Organisation verwendet habe. Die Schlüsselfunktion dieses Designs besteht aus den drei Spalten unter »IDEEN«. Beachten Sie, wie die absteigenden WIP-Limits einen Trichter suggerieren.

IDEEN			ZUGESAGT		
Priorität 3 15	Priorität 2 10	Priorität 1 5	Bereit 3	In Arbeit 3	Fertig

Abb. 4-1 Ein Beispiel für ein Personal-Kanban-Board-Design

2. Vergessen Sie hier nicht die internen Stakeholder – Audit, Sicherheit, Finanzen, Support usw. Haben sie ein Veto, sollten sie wie Kunden behandelt werden (siehe [Middleton & Sutton 2005]).

Diese Art von Design ist hilfreich, wenn Sie sich über die Organisation Ihrer Ideen und zukünftigen Aufgaben genauso viele Gedanken machen wie über Ihr aktuelles Arbeitsaufkommen. Es gibt zwei wichtige Tricks, um dieses Design effektiv zu nutzen:

1. Sie lernen die Spalten links außen auszublenden, bis es an der Zeit ist, die Spalten weiter rechts zu priorisieren oder aufzufüllen. Schon bald wird dies zu einem unbewussten Verhalten und es besteht kein Zweifel, dass Sie ein Pull-System betreiben. Wollen Sie die »Bereit«-Spalte neu auffüllen, tun Sie dies nun aus der relativ überschaubaren Spalte »Priorität 1« mit ihren fünf Aufgaben. Das ist eine viel einfachere Aufgabe, als aus allen Ideen wählen zu müssen.
2. Sie gleichen aus zwischen Ihrer Ungeduld, neue Ideen zum Board hinzuzufügen, und Ihrer Bereitschaft, Dinge zu entfernen, die es wahrscheinlich sowieso nicht in die »Bereit«-Warteschlange schaffen würden. Entfernte Tickets können für eine spätere Überprüfung archiviert oder ganz vernichtet werden. (Ich bevorzuge es, einzelne Tickets schnell in einen abseits liegenden Wartebereich zu verschieben und mich um diese in regelmäßigen Abständen »in einem Rutsch« zu kümmern).

WIP-Limits erinnern uns natürlich daran, uns an diese beiden Tricks zu halten. Hat sich das Board erst einmal gefüllt, werden Sie sehen, dass Sie öfter Dinge herunter priorisieren. Das ist gut!

Die Idee für dieses Design kam mir durch eine *Personal-Kanban*-Technik (siehe [Benson & DeMaria Barry 2011]), das Prioritätsieb. Es erscheint vielleicht ungewöhnlich, dass ich jetzt zu Personal Kanban abschweife (wir kommen in Teil II noch einmal kurz darauf zurück), ich nutze dieses Design jedoch als ein Modell für Upstream-Kanban. Der Begriff wurde geschaffen, um die Verwendung eines Kanban-Systems stromaufwärts des Lieferungsprozesses zu beschreiben. Upstream-Kanban hilft dabei, Bedürfnisse zu verwalten und Ideen zu entwickeln, sodass zu jeder Zeit gute Auswahlmöglichkeiten bestehen, wenn Lieferkapazität verfügbar wird.

Das Design verstärkt zwei Konzepte, die Portfoliomanager einiger unserer größten Unternehmen gerne vergessen zu scheinen:

- Wir generieren mehr Ideen, als wir überhaupt nutzen können – tatsächlich sollten wir uns sogar sorgen, wenn es anders wäre. Über die Zeit werden wir mehr Ideen anhäufen, als wir sinnvoll bewältigen können, ganz zu schweigen, dass wir sie alle implementieren können.
- Ideen kommen nicht nur anhand ihres eigenen Wertes voran – sie befinden sich in Konkurrenz zu anderen Ideen. Zudem können zu jeder Zeit neue Ideen in den Wettkampf eintreten.

Es passiert etwas Besonderes, wenn sich Tickets in die »Bereit«-Spalte bewegen: Dieses System hat einen sehr eindeutigen *Zusagepunkt*. Links davon freuen wir uns, wenn Tickets sich zurückbewegen oder weggeworfen werden – das ist eine wichtige vom System zu erfüllende Aufgabe. Rechts davon würden wir annehmen, dass etwas falsch läuft, wenn ein Ticket nicht schnell genug vorankommt, und wir würden uns über den unnötigen Aufwand ärgern, wenn Tickets vor Beendigung eingestellt werden würden.

Genauso wie innerhalb des Lieferprozesses hängt die Effektivität stromaufwärts von den bereits besprochenen Werten ab:

■ Transparenz

Das System muss die schwierigen Entscheidungen, die getroffen werden müssen, sichtbar machen. Die Grundprinzipien, nach denen entschieden wird, sollten explizit sein. Entscheidungen sind der Fokus von Feedbackschleifen (z.B. Priorisierungsmeetings).

■ Balance

Die Menge paralleler Arbeit im System wird reguliert, um einerseits einen ausreichenden Nachschub an qualitativ hochwertigen Ideen zu haben, und andererseits, um Entscheidungen rechtzeitig zu erzwingen. Falls das nötig ist, kann durch die WIP-Zuordnung nach Kunden, Etatposten, Risikokategorien, strategischen Initiativen usw. noch mehr Kontrolle erlangt werden.

■ Kooperation

Die Arbeit, um Tickets für die weitere Entwicklung zu qualifizieren, wird gemeinsam von den Erstellern und denen erledigt, die die Tickets liefern. Statt Risiko frühzeitig in das System zu transportieren, halten sich alle Beteiligten (von denen es einige geben kann) ihre Optionen offen, bis eine Zusage notwendig ist.

Kundenfokus fügt dem Ganzen noch Folgendes hinzu:

- Wessen Bedürfnisse glauben wir, mit diesen Ideen zu erfüllen?
- Erfüllen wir Bedürfnisse schnell genug?
- Was sagen uns die Daten? Was sagen uns die Menschen?
- Was könnte hinter diesen Bedürfnissen stecken?
- Wessen Bedürfnisse können vielleicht nicht erfüllt werden?
- Wie können wir das überprüfen?

Zusammengefasst: Können wir ein besseres Gespür dafür entwickeln, was gebraucht werden wird?

4.6 Bedürfnisse vorwegnehmen

Wenn Sie nur eine Sache aus diesem Kapitel mitnehmen sollten, würde ich mir wünschen, dass es ein mentaler Wechsel ist. Er soll Sie davon wegbringen, das zu tun, nachdem gefragt wurde, dem Entgegennehmen von Befehlen, Abarbeiten von Anfragen, Erfüllen von Anforderungen usw., und Sie hinlenken zu einer Neuausrichtung des Prozesses in Richtung des Entdeckens und Erfüllens von Bedürfnissen. Es ist ein Wechsel von einer internen Perspektive (was wir zu wissen meinen) zu einer externen (was gibt es dort draußen noch zu entdecken). Es ist zusätzlich auch ein Wechsel von der Vergangenheit (was uns gesagt wurde) zur Zukunft (wenn die Bedürfnisse des Kunden erfüllt sind).

Der Schwerpunkt auf die Zukunft wird in den abschließenden Worten des Kundenversprechens von Toyota sehr schön deutlich. Folgendes habe ich auf einer Tafel hinter dem Schalter des Kundenservice meines örtlichen Toyotahändlers entdeckt:

... die Bedürfnisse der Menschen und der Gesellschaft in Bezug auf Mobilität vorzeitig erahnen

Denken Sie an einen Dienst, auf den Sie persönlich angewiesen sind. Wären Sie nicht begeistert, würde der Anbieter Ihre Bedürfnisse schon vorzeitig erahnen? Welche Innovation müsste er einführen, damit dies passiert? Können Sie diese Art zu Denken auf Ihre Arbeit übertragen?

5 Arbeitsfluss

Als unser Team anfangs zusammenkam, hatte nur ein kleiner Teil der Mitglieder Erfahrungen mit definierten Prozessen, agil oder anderweitig. Für die Hälfte von ihnen war die verwendete Technologie neu – selbst die Programmiersprache war unbekannt –, dazu kam, dass sie noch nie Erfahrungen mit Techniken wie Continuous Integration oder Versionskontrollsystmen gemacht hatten. Trotzdem produzierten sie Anwendungen von beträchtlicher Größe und Komplexität und schafften zusätzlich einen Prozess, der für seine Qualität bekannt war und mit dem sie in der Lage waren, Anforderungen innerhalb von Stunden oder Tagen an ihre Kunden auszuliefern.

Wie war dieser Wandel möglich? Wir haben das Team nicht in einem neuen Prozess geschult oder die Mitarbeiter gebeten, neue Rollen zu übernehmen. Statt dessen haben wir Kanban verwendet, die »*Beginne mit dem, was du gerade tust*«-Methode. Geradezu einfach: Wir organisierten unsere Arbeit visuell, identifizierten unsere größten Probleme in Bezug auf den **Arbeitsfluss** (Flow) und kümmerten uns darum, diese Probleme zu beseitigen.

5.1 Kernpraktik 3: Manage den Arbeitsfluss (nochmal)

Hier noch einmal meine erweiterte Version der dritten Kanban-Kernpraktik:

KP3 (erweitert): Manage den Arbeitsfluss, um Gleichmäßigkeit, Pünktlichkeit, und gute ökonomische Ergebnisse zu erreichen, und nehme Kundenbedürfnisse vorweg.

Am Ende des Kapitels behandeln wir Arbeitsfluss in Bezug auf *Pünktlichkeit*. Der Schwerpunkt dieses Kapitels ist der *Gleichmäßigkeit* gewidmet.

5.2 Gleichmäßigkeit

Kanbans Wertschätzung für Gleichmäßigkeit ist nicht ungewöhnlich. In manchen Kreisen ist diese noch viel größer: Es ist fast eine Besessenheit! Das Managertraining bei Toyota beinhaltet stundenlanges Beobachten von Produktionslinien, bei

dem nach der kleinsten Abweichung vom Ideal der Gleichmäßigkeit Ausschau gehalten wird. Hierbei wird impliziert, dass jeder noch so gleichmäßige Produktionsprozess es wert ist, verbessert zu werden, und dies auch möglich ist.

Die Idee, auf Produktionslinien zu starren, lässt sich nicht auf kreative Wissensarbeit übertragen, selbst wenn wir Kanban verwenden. Nichtsdestotrotz fühlen wir uns schnell unwohl, wenn wir sehen, dass Arbeit unterbrochen, blockiert und unerledigt bleibt oder für Nacharbeiten zurückgeschickt wird, selbst dann, wenn die vorherrschenden geschäftlichen Bedingungen zu der Zeit diese Umstände rechtfertigen. Wir sind einen großen Schritt weiter, wenn dieses Unwohlsein die Frage aufwirft »Was an unserem Prozess lässt dies so leicht geschehen?«.

In seinem Buch *The Culture Game* [Mezick 2012] erinnert uns Daniel Mezick daran, dass wir nur auf wenige Dinge gleichzeitig explizit achten können. Bewusst oder unbewusst treffen wir Entscheidungen, mit welchen Dingen wir uns bewusst befassen wollen, wenn wir unser Arbeitsmanagementsystem designen und ausstatten. Mit der Einführung von **Transparenz** in unseren Workflow haben wir solch eine Entscheidung getroffen. Um es mit den Worten von Mezick zu sagen: »*Kanban achtet gezielt auf Arbeitsfluss.*«

5.2.1 Wie sieht Arbeitsfluss aus?

Hier ein paar Anzeichen dafür, dass Arbeit beginnt, im Fluss zu sein:

- Eine gute Anzahl von Arbeitspaketen bewegt sich zwischen den Standup-Meetings. Wie viele genau und wie oft ist eine Funktion aus Größe der Arbeitspakete, Teamgröße und dem Intervall zwischen den Meetings. Meistens fühlt es sich beruhigend an, wenn man täglich zumindest ein bisschen Bewegung sieht. Lernen Sie, große Arbeitspakete in kleinere (aber trotzdem werthaltige) Einheiten herunterzubrechen, falls Fortschritt nicht gut sichtbar ist.
- Relativ zur Anzahl der verfügbaren Personen ist eine gute Anzahl der Pakete erkennbar frei von Blockaden, sodass Sie sicher sein können, dass aktuell und in Zukunft Fortschritt erzielt wird.
- Man spürt (und misst vielleicht), dass Arbeit schneller und vorhersagbarer fertiggestellt wird. Für Arbeitspakete gleicher Größe oder Art werden die Durchlaufzeiten typischerweise kürzer und sie bewegen sich in einem engeren Bereich.

Wenn Sie diese Anzeichen nicht sehen, haben Sie ein Ziel. Falls Sie denken, dass Sie die Anzeichen erkennen, fragen Sie sich, ob dieser Arbeitsfluss weiter verbessert werden kann bezüglich einer oder mehrerer dieser breiten Messkriterien – in der Erwartung, dass dem so ist.

Obwohl es gut ist, den Arbeitsfluss zu sehen oder sogar zu quantifizieren, sollte man nicht die Wichtigkeit des *Fühlens* unterschätzen. Es fühlt sich gut an, wenn Arbeitspakete fortschreiten, wenn eine Arbeitsmenge nicht von Problemen

dominiert wird, wenn eine Vorhersage mit einer gewissen Sicherheit abgegeben werden kann. Und es fühlt sich für alle besser an – Kunden, das Unternehmen und die Menschen, die die Arbeit tun.

5.3 Quer über das Board (nochmal)

Das vorherige Kapitel ermutigt Sie dazu, den eigenen Workflow bezüglich **Kundenfokus** zu betrachten. Unsere Untersuchung begann am Punkt der Kundenzufriedenheit und arbeitete sich vorsichtig den Strom aufwärts bis zum ursprünglichen Kundenbedürfnis. Wir haben jeden Punkt durch die Linse der Wissensentdeckung erforscht. Wir können den gleichen Ansatz mit **Arbeitsfluss** verfolgen, aber mit einer anderen Linse.

Warum sollten wir auch hier rückwärts arbeiten? Es zwingt uns zum Fokus darauf, was wirklich benötigt wird, damit Arbeit fließen kann, statt dass wir an jedem Punkt alle Wege und alles Verfügbare, das wir verwenden können, aufzählen. Metaphorisch (und vielleicht sogar physisch) stehen wir an der rechten Seite des Kanban-Boards und denken darüber nach, was die Arbeit typischerweise davon abhält, glatt bis zur Fertigstellung über das Board gezogen zu werden, statt auf der linken Seite zu stehen und uns Wege zu überlegen, wie wir Dinge schneller in einen wachsenden Stapel in der Mitte des Systems drücken können. Kanbans Protokoll für Standup-Meetings ist ein getarntes Denkwerkzeug!

Wie sieht unsere neue Linse aus? Wir konzentrieren uns auf den Fluss der Arbeit und stellen uns für jede Spalte Fragen wie diese:

- Wie verlassen Arbeitspakete diesen Prozessschritt? Welche Kriterien sagen uns, dass sie dafür bereit sind? Wie werden diese Kriterien ausgedrückt? Wie wird diese Bereitschaft signalisiert, sodass stromabwärts entsprechend gehandelt werden kann?
- Wie viel Zeit verbringen Arbeitspakete typischerweise in diesem Zustand? Wie viel dieser Zeit (wenn überhaupt) wird in aktiver Arbeit im Gegensatz zu reiner Wartezeit verbracht?
- Welche sind die größten Quellen für Unvorhersagbarkeit – liegen sie in der Arbeit oder in der Wartezeit? Warten die Einheiten typischerweise auf interne Verfügbarkeit oder darauf, dass externe Abhängigkeiten aufgelöst werden?
- Welcher Anteil der Kapazität dieses Zustands wird durch Nacharbeit verbraucht? Oder von sogenanntem *Failure Demand* (siehe [Seddon 2003]), der nur deshalb auftaucht, weil vorher fertiggestellte Arbeit es nicht geschafft hat, das Kundenbedürfnis adäquat zu befriedigen?
- Wie kommen Arbeitspakete in diesem Zustand an? Wie wissen wir, dass sie für eine Bearbeitung bereit sind?

Vorsicht: Diese Fragen haben folgende Annahmen:

1. Der Prozess hat sinnvolle Ziele, um die richtigen Bedürfnisse zu adressieren.
2. Der Workflow hat einen sinnvollen Umfang. Er beginnt mit der richtigen Art von Fragen und endet mit der richtigen Art von Resultaten.
3. Der Workflow ist sinnvoll organisiert. Er ist so gestaltet, dass er so schnell wie möglich wertvolle Lernresultate liefert.

Ich habe genügend Gegenbeispiele gesehen, um zu wissen, dass das gefährliche Annahmen sein können. Um nur zwei Beispiele zu nennen:

- Ich habe einmal geholfen, einen Prozess zu richten, dessen Ziel es war, Änderungsvorschläge zum Systemdesign entweder zu bestätigen oder abzulehnen. Es zeigte sich, dass dieser sogenannte Change-Management-Prozess vollkommen von allen Implementierungsprozessen entkoppelt war; seine Ziele und sein Umfang waren entweder schlecht gewählt oder unpassend. Unnötig zu sagen, dass sämtliche Kundenzufriedenheit, die aus diesem Prozess herauskam, eher kurzlebig war.
- Projektbasierte Softwareentwicklungsprozesse gehen häufig davon aus, dass das Ende des Projekts die richtige Zeit ist, um über Akzeptanztests nachzudenken. Das verzögert die Möglichkeit eines hohen Lerneffekts bis sehr spät im Prozess.

5.4 Es gibt immer einen größeren Kontext

Wir haben bereits gesehen, wie **Kundenfokus** Teams dazu einlädt, einen breiteren Kontext in Betracht zu ziehen als den ihrer eigenen Arbeit. **Arbeitsfluss** tut das Gleiche, und das in mehreren Dimensionen.

Beschäftigt man sich mit dem Fluss der Arbeit, werden wir häufig dazu gezwungen, darauf zu achten, wie Dinge, von denen wir abhängen – Material, Informationen, Ideen, sogar Menschen –, in unserem Teil des Systems ankommen. Dinge neigen nicht unbedingt dazu, aus Zufall zur rechten Zeit anzukommen. Wir können auch nicht erwarten, dass Arbeit unsere Hände zur rechten Zeit verlässt, wenn wir uns keine Gedanken über unsere vier Wände hinaus gemacht haben, um sicherzustellen, dass die Ausgänge auch frei sind.

Sie haben mittlerweile vielleicht realisiert, dass Kanban sich nicht um die individuelle oder Teamproduktivität dreht. Ja, wir arbeiten häufig daran, das System im Rahmen unserer Einflussmöglichkeiten zu verbessern. Wir beschränken uns aber nicht nur auf die Probleme, die wir alleine lösen können. Wir strecken häufig die Hand aus und arbeiten mit anderen Menschen an größeren Problemen, die sich durch das Gesamtsystem ziehen. **Kooperation** hilft uns hier nicht nur, Dinge fertigzustellen. Sie hilft uns auch sicherzustellen, dass unsere kollektiven Bemühungen sich auf das Wesentliche konzentrieren.

5.5 Einige einfache Beispiele

Es folgt eine unvollständige Liste von Verbesserungen, die wir in der Zeit von 2009 bis 2010 durchführten. Einige von ihnen sehen vielleicht sehr softwareentwicklungsspezifisch aus, aber die meisten enthalten Aspekte, die auch in andere Domänen übertragen werden können. Sie sind »von rechts nach links« geordnet, also rückwärts durch den Entwicklungsprozess.

- Wie bereits im vorhergehenden Kapitel beschrieben, fügten wir einen Schritt zur Validierung durch den Kunden an das Ende des Prozesses. Unsere ursprüngliche Motivation war es, den Übergang von »fast ausgeliefert« zu »tatsächlich implementiert« zu beschleunigen. Rückblickend war die wirkliche Auswirkung dieser Veränderung, dass sie eine spezielle Art der Zusammenarbeit mit dem Kunden katalysierte, die auf gute Ergebnisse fokussiert war. Sie führte zu signifikanten Verringerungen in der Gesamtdurchlaufzeit, Nacharbeit, von gemeldeten Fehlern und abgebrochener Arbeit.
- Automatisierte Auslieferung. Was uns ursprünglich mehrere Stunden intensiver manueller Arbeit mit einer hohen Fehlerwahrscheinlichkeit kostete, konnte nun in wenigen Minuten mit hoher Ausfallsicherheit durchgeführt werden. Es wurde zudem noch die Option geschaffen, Dinge schnell rückgängig machen zu können, falls doch etwas nicht richtig aussah.
- Automatisiertes Testen. Wir begannen mit grundlegenden *Unit Tests*, aber automatisierten mehr und mehr auch die Akzeptanztests. Wir lernten, erst den Testcode zu prüfen, bevor der Anwendungscode durchgesehen wurde, und konnten damit eine Vielzahl substantieller Verbesserungen in der Codebasis erreichen. Dies verbesserte nicht nur die Qualität der Tests, die wir gerade geschrieben hatten, sondern machte zukünftige Änderungen an den Tests ebenfalls einfacher.
- Die »Demo!«-Regel. Jede Neuentwicklung musste dem gesamten Team (nicht nur dem Reviewer) vorgestellt werden, bevor das Ticket von »Entwickeln« zu »Test« bewegt werden durfte. Diese Regel hinderte einige Arbeitspakete daran, zu früh voranzuschreiten, und schuf gute Möglichkeiten zur Wissensverbreitung und Ideengenerierung.
- Wir blieben beim Prinzip »keine Überraschungen« bei den Codereviews, allerdings in einer moderneren Form. Wir erachteten Codereviews als unnötig, wenn der Code im *Pair Programming* entwickelt wurde (intensives, kooperatives Programmieren, bei dem zwei Entwickler gemeinsam an nur einem Bildschirm arbeiten).
- Die »Fünf-Tage-Regel«. Diese gemeinschaftliche Regel besagte, dass Arbeitspakete nicht mehr als geschätzte fünf Tage Entwicklungsarbeit umfassen sollten. Später wurde diese zur »Zwei-Tage-Regel«. Wir verfeinerten diese Regel mit einer Ausnahme für Arbeit, die nicht einfach zerlegt werden konnte. Die Ausnahme wurde selten in Anspruch genommen.

- Tägliche statt wöchentliche Wechsel durch den Support. Mit unserer alten, wöchentlichen Supportaufgabe litt die Entwicklungsarbeit regelmäßig unter langen Unterbrechungen. Das hatte schwerwiegende Auswirkungen auf unsere Vorhersagbarkeit. Als das neue System eingeführt wurde, bei dem täglich gewechselt wurde, verbesserten wir die Regeln für die morgendliche Übergabe der Supporttickets. Dadurch stellten wir sicher, dass ausreichend Hilfe verfügbar war, wenn die Menge der Supporttickets hoch war (ein intelligentes Ausbalancieren des Flusses der Supportarbeit gegen den Fluss der Entwicklungsarbeit).
- Wir untersuchten genauer, woher Supportanfragen kamen, und handelten lokale Regeln mit den Teilen der geschäftlichen Anforderungsseite aus, die die meiste Arbeit generierten. Die Last auf dem Support verringerte sich, als allgemeine Probleme beseitigt wurden und die Geschäftsteams ihr Wissen mit ihren Kollegen teilten.

5.6 Arbeitsfluss quer durch die Organisation

Je größer Organisationen werden, desto mehr Dienste sind an ihren Workflows beteiligt. Typischerweise sind einige Dienste dediziert für einen Workflow vorgesehen, während andere über mehrere verteilt sind oder sogar outgesourct sind. Manche werden als Teil eines »Haupt«-Workflows verstanden, andere steuern nur nach Aufforderung etwas bei.

Abbildung 5–1 stellt einige Probleme der Skalierung dar.

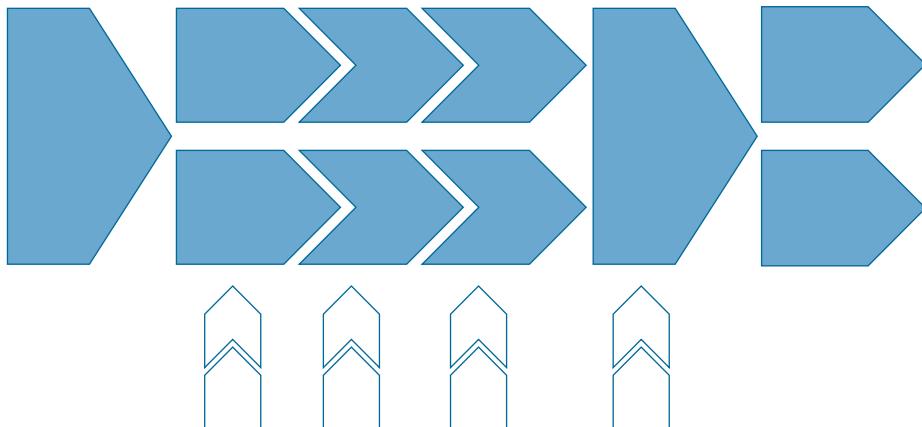


Abb. 5–1 Gemeinsam genutzte und abhängige Dienste umfassen zwei Hauptworkflows.

In diesen Größenordnungen gibt es viele Gelegenheiten für Verzögerungen und Frustration.

- Die Gruppe, die Arbeitsaufträge entgegennimmt, ist eine andere als diejenige, die die Arbeit dann wirklich durchführt. »Fire and Forget«-Kommunikation (insbesondere E-Mail) führt dazu, dass Arbeit bei der Übergabe verlorengeht.
- Es existieren unterschiedliche Verständnisse zwischen Personen oder Teams bezüglich eines »Fertig«-Status und eines »Bereit für die Weiterbearbeitung«-Status. Diese Qualitätsprobleme können sowohl Fehler in den Prozessregeln als auch in der Kommunikation sein; häufig ist es beides.
- Aktivitäten werden auf der falschen Seite einer organisatorischen Grenze durchgeführt, und zwar an einem anderen Ort als dem, wo die erforderlichen Fähigkeiten und Informationen verfügbar sind.
- Der Prozess hat unnötige Schritte, d.h. Schritte, die einfach von anderen Schritten übernommen werden könnten, oder Schritte, die herausgelöst und früher durchgeführt werden sollten, um früheres Feedback zu ermöglichen.
- Den gemeinsam genutzten Diensten, von denen der Hauptworkflow abhängig ist, wird unzureichend Aufmerksamkeit geschenkt. Dieses Problem strahlt in beide Richtungen ab: Die Abhängigkeit eines Dienstes ist das Problem eines anderen Dienstes, das von ihm aus stromaufwärts auftritt. Bei letzterem besteht das Problem sowohl in der Sichtbarkeit (Aufträge kommen zu spät von Personen, die nicht unbedingt verstehen, wonach sie fragen) als auch im Kapazitätsmanagement (Aufträge werden ohne die Kenntnis der gesamten Arbeitslast abgeschickt).
- Arbeit wird in zu großen *Batches* verwaltet; zum Beispiel Multi-Monats-Projekte oder sogar Projekte, die sich über Jahre erstrecken und die in der Zwischenzeit keinen Kundennutzen liefern. Das Problem wird noch dadurch katastrophal verschlimmert, dass diese großen Projekte gestartet werden, ohne dass viele Gedanken daran verschwendet werden, welche Auswirkung sie auf bereits vorhandene Arbeit (oder andere große Arbeitspakete) haben.

Üblicherweise treten diese Probleme in Kombination miteinander auf. Das Resultat ist nicht nur eine extrem schlechte Leistung auf Organisationsebene, sondern echte Qualen auf der menschlichen Ebene. Wer möchte schon konstant überlastet sein und dabei das Gefühl haben, dass dabei nie etwas Sinnvolles erreicht wird?

Manche werden Ihnen raten, alles wegzuerwerfen und neu aufzusetzen, wenn die Dinge so schlimm stehen. Es steckt schon etwas Ironie darin: Dieselben Menschen, die sich für inkrementelle und evolutionäre Ansätze in der Produktentwicklung einsetzen, scheinen nur zu vernarrt darauf zu sein, destabilisierende und revolutionäre Veränderungen in menschenbasierten Systemen zu empfehlen – in denen die Ergebnisse noch viel unsicherer sind.

Die gute Nachricht ist, dass es schwer ist, einen Prozess zu finden, der so komplex ist, dass Probleme wie die gerade genannten nicht identifiziert und nacheinander adressiert werden könnten. Die folgenden Praktiken sind häufig überraschend einfach und stehen alle unübersehbar in Beziehung zu den Kanban-Praktiken:

- Arbeit in der richtigen Granularität von Anfang bis Ende verfolgen, häufig in verschiedenen Detaillierungsstufen.
- Die Batchgröße der Arbeit mit WIP-Limits und Regeln für die Größe von Arbeitspaketen kontrollieren – und den Hunger auf Lieferversprechen weit in die Zukunft schieben.
- Work in Progress funktionsübergreifend steuern.
- Funktionsübergreifend Prozessregeln vereinbaren und kooperativ instandhalten.
- Abhängigkeiten zu geteilten Diensten visuell verwalten; umgekehrt die stromaufwärts liegende Arbeit aus diesen Diensten heraus steuern.
- Die Granularität der Aktivitäten oder Status, in denen Arbeit gesteuert wird, verändern (wieder auf den unterschiedlichen Detailstufen, falls notwendig).
- Inkrementell den Prozess reorganisieren und dabei der funktionalen Organisation erlauben, im Windschatten zu folgen.
- Leistung so messen und berichten, wie es für den Kunden relevant ist (viele dieser Messungen werden zum Arbeitsfluss¹ sein)

Organisationen geraten häufig noch tiefer in Probleme, weil Menschen meinen, dass die Antwort wäre, sich noch mehr anzustrengen. Sie glauben, dass besseres Projektmanagement ein Problem des Kapazitätsmanagements löst (die Projektleiter bekommen derweil den Schwarzen Peter zugeschoben), dass bessere Steuerung in den funktionalen Silos zu besserer Gesamtleistung führt (obwohl es das sehr häufig schlimmer machen kann) oder dass die Menschen sich einfach mehr anstrengen sollten, um besser zu sein (während das System an sich fundamental unzuverlässig ist).

Kanbans ungewöhnliche Lösung für dieses Problem besteht nicht darin, direkt die Rolle des Projektmanagements oder des Bereichsleiters anzugehen; ganz sicher sollen diese nicht mit anderen Dingen ersetzt werden. Stattdessen gibt es Managern (und anderen natürlich auch) Werkzeuge an die Hand, um die Arbeit und ihren Fluss auf eine neue Art und Weise zu sehen, sowie Kontrollmechanismen für WIP, die extreme Auswirkungen auf die Probleme der Verzögerung und der Unvorhersagbarkeit haben. Mit dem richtigen Umfang wachsen die

1. In einigen Teilen der Community wurde *Messe und Manage den Arbeitsfluss* statt *Manage den Arbeitsfluss* verwendet. Ich fand das nicht besonders gut. Zu messen ist wichtig, die Wortwahl hat den Messungen jedoch zu viel Bedeutung geschenkt. Die Aufnahme von *Implementiere Feedbackzyklen* (KP5) hat dann den Teil überflüssig gemacht.

Verbesserungen und das nötige neue Denken Hand in Hand. Falls dann Rollen verändert werden müssen: kein Problem!

Der richtige Ausbreitungsbereich sollte nicht als gegeben angenommen werden. Wir werden typischerweise (und wenig überraschend) als Berater eingekauft, um einen funktionalen Bereich zu beraten oder zu schulen, der unter Last steht: Es ist häufig der am stärksten überlastete Bereich des Unternehmens. Wir müssen unsere neuen Sponsoren daran erinnern, dass eine rein intern fokussierte Verbesserungsinitiative begrenzt ist, selbst wenn sie auf Kanban basiert.

Um ehrlich zu sein: Eine rein intern fokussierte Kanban-Initiative ist kastriert. Ihr fehlen im Speziellen die Werte der Methode, die einen Blick nach außen werfen, **Kundenfokus** und **Arbeitsfluss** (gemeinsam mit **Führung**, mit der wir uns im nächsten Kapitel befassen). Diese Werte geben den eher nach innen orientierten Werten und Praktiken der früheren Kapitel eine Richtung. Egal, ob auf Kanban basierend oder nicht, falls einer Veränderungsinitiative in Ihrer Nähe die Puste ausgeht: Könnten die Dinge wieder zum Laufen gebracht werden, wenn auf der richtigen Ebene der Organisation einem oder mehreren dieser Werte Aufmerksamkeit geschenkt wird?

5.7 Arbeitsfluss managen, um Pünktlichkeit zu erzielen

Es wäre ein Fehler zu denken, dass das Management des Arbeitsflusses nur aus dem Entfernen von Hindernissen besteht, wenn diese auftauchen. Typischerweise sehen wir in der Wissensarbeit, dass Arbeitspakete sowohl inhaltlich als auch im Wert stark variieren und dass die Arbeitsbelastung insgesamt über die Zeit stark schwankt. Das bedeutet, dass immer ein Ort existiert, an dem wir Arbeit proaktiv steuern können:

- Ungewöhnliche Risiken und Abhängigkeiten müssen früh und effektiv adressiert werden.
- Mittelfristig muss der erwarteten Arbeitslast mit einer adäquaten Kapazität entgegengetreten werden.
- Langfristig müssen eventuell ganz neue Fähigkeiten entwickelt werden.

Man muss nicht immer davor zurückschrecken, die wichtigsten Dinge etwas enger zu begleiten, selbst wenn man sich der Selbstorganisation verschrieben hat. Nicht alle Arbeitspakete sind gleich und manche verdienen es mehr als andere, besondere Aufmerksamkeit des Managements zu bekommen. Termine – zumindest wichtige Termine – sollten eingehalten werden. Und wenn es durch einen geschäftlichen Nutzen gerechtfertigt ist, ist es auch völlig in Ordnung, ein wenig Vorhersagbarkeit zu opfern, damit einige Dinge mit hohem Wert an den Warteschlangen vorbeihuschen können.

Eine Eigenschaft eines gut entwickelten und wohlkalibrierten Pull-Systems besteht darin, dass die meisten tagtäglichen Entscheidungen über die Reihenfolge einfach zu treffen sein sollten. Sie können diese Dinge einfach passieren lassen und zur selben Zeit das richtige Gefühl entwickeln, was wann beschleunigt werden muss. Sobald Sie diese Vorgehensweise formalisieren können, können Sie dem System helfen, sich weiter zu entwickeln. Je reifer das System wird, desto weniger häufig sind diese Interventionen notwendig.

6 Führung

Führung (Leadership) – einer der markantesten Werte von Kanban – wird am besten durch die anderen acht Werte beschrieben. Mit den ersten fünf, nämlich **Transparenz, Balance, Kooperation, Kundenfokus und Arbeitsfluss**, haben wir die sechs *Kernpraktiken* der Kanban-Methode eingeführt. In ihrer typischen Reihenfolge sind sie:

- KP1:** Visualisiere.
- KP2:** Limitiere die Menge paralleler Arbeit (Work in Progress, WIP)
- KP3:** Manage den Arbeitsfluss.
- KP4:** Mache Prozessregeln explizit.
- KP5:** Implementiere Feedbackzyklen.
- KP6:** Erziele Verbesserung kooperativ, entwickle experimentell (unter Verwendung von Modellen und der wissenschaftlichen Methode).

Jede dieser Praktiken bietet viele Möglichkeiten, Führung auszuüben.

Es fehlen noch **Verständnis, Vereinbarung und Respekt**. Diese Führungsdisziplinen stehen für die ursprüngliche Menge der Grundprinzipien von Kanban:

- GP1:** Beginne mit dem, was du gerade tust.
- GP2:** Vereinbare, dass evolutionäre Veränderung verfolgt wird.
- GP3:** Respektiere initial bestehende Prozesse, Rollen, Verantwortlichkeiten und Jobtitel.

Diese Prinzipien sind grundlegend auf zwei Arten: Sie halten die Kanban-Methode als evolutionäre Herangehensweise fest verankert, sowohl technisch als auch philosophisch, und sie beschreiben einige wichtige Verpflichtungen und Verhalten von **Führung**, die essenziell für seine initiale Einführung und den kontinuierlichen Betrieb sind.

Führung ist also die Brücke zwischen den beiden Wertemengen und verbindet die Praktiken und die Prinzipien.

6.1 Grundprinzip 4: Führung auf allen Ebenen

Das vierte Grundprinzip heißt:

GP4: Ermutige dazu, Führung auf jeder Ebene der Organisation zu zeigen – vom einzelnen Mitarbeiter bis zum höheren Management.

Achten Sie darauf, wie integrativ diese Sprache ist. Sie respektiert die aktuelle Struktur Ihres Unternehmens, schließt niemanden aus und sie zwingt auch keine falsche Gegensätzlichkeit zwischen Führung und Management auf. Man darf aber auch nicht unterschätzen, wie bedeutsam das Prinzip ist – es ist eine starke Erklärung. Als Grundprinzip gibt es den Hinweis, dass eine Kanban-Initiative folgenschwer unterlaufen werden kann, falls man es nicht schafft, diese Art von Führung *auf allen Ebenen* zu pflegen. Doch dies bringt eine Herausforderung mit sich: Was ist, wenn diese Art von Führung in Ihrem Unternehmen nicht selbstverständlich vorhanden ist?

Glücklicherweise lässt Kanban einen nicht alleine mit der Lösung des Problems. Wenn Veränderung stimuliert wird, werden kleine und größere Gelegenheiten zur Führung eröffnet. Je ausgedehnter, wiederholbarer und sichtbarer dieser Prozess ist, desto positiver wird die Auswirkung auf die Unternehmenskultur sein.

6.2 Möglichkeiten finden sich überall

Lassen Sie uns das überprüfen. Wo kann man diese Möglichkeiten zur Führung finden?

■ Transparenz

In der Wissensarbeit werden Dinge nicht von alleine sichtbar oder explizit; Führungskräfte entscheiden sich, dies zu tun. Das gilt sowohl für die kleinen Details – der Wortlaut einer Prozessregel beispielsweise – wie auch für die größeren Dinge wie zum Beispiel institutionalisierte Feedbackzyklen.

■ Balance

Wo sind wir überlastet und warum? Sind unsere Schmerzpunkte offensichtlich oder versteckt die Menge an Arbeit sie? Ist der Mix der Arbeit richtig? Möglichkeiten zur Führung stecken im Stellen dieser Fragen und auch in den Entscheidungen, die daraus folgen.

■ Kooperation

Jemanden einem anderen vorstellen, eine Verbindung suchen, ein Problem teilen, bemerken, wie Menschen interagieren – all das können Führungshandlungen sein.

■ Kundenfokus

Es erfordert Führung, anzuerkennen, dass der Prozess vielleicht ineffektiv darin ist, die wirklichen Kundenbedürfnisse zu entdecken und zu befriedigen.

■ Arbeitsfluss

Streben Sie danach? Was ist heute blockiert? Wo treten Blockaden wiederholt auf? Warum ist das so? Dies sind tagtägliche Fragen der Führung.

■ Führung

Zu Führung zu ermutigen, kann echte Führungsqualitäten auf der Seite des Ermutigenden erfordern. Kanbans Art von Führung breitet sich nicht nur aus, sie verstärkt sich sogar. Führung folgt aus Führung.

6.3 Führung erzeugt Führung

Meine Geschichte aus Budapest illustriert diesen letzten Punkt.

Früh in unserer evolutionären Reise fingen wir mit der agilen Praktik der *Retrospektiven* an, indem wir uns regelmäßig zusammensetzten und über die kürzlich geschehenen Ereignisse reflektierten. Anfänglich fanden wir diese Treffen nicht sonderlich effektiv. Ein offensichtliches Hindernis war meine Anwesenheit als Führungskraft. Wir lernten auf diese Weise, dass von Managern modierte Retrospektiven so etwas wie ein Anti-Pattern sind; ich musste mich irgendwie zurückziehen.

Eine Kaskade von »kleinen Führungshandlungen« folgte:

1. Ich ziehe mich zurück und schlage vor, dass das nächste Treffen stattfindet, wenn ich nicht im Büro bin (das war einfach, da ich wochenweise wechselnd zu Hause in England und in Ungarn arbeitete).
2. Krisztian meldet sich freiwillig, um die nächste Retrospektive zu leiten, und beginnt sofort mit Vorbereitungen für das Treffen. Die Retrospektive findet ohne meine Beteiligung während meiner Abwesenheit statt.
3. Ich komme zurück nach Budapest und sehe das Board zum ersten Mal mit geschriebenen Prozessregeln annotiert. Bemerkenswerterweise sind einige dieser Regeln schärfster als irgendetwas, was ich ohne deutlichen Widerstand des Teams selbst hätte einbringen können.

Ich sah das erweiterte Board und ich spürte, dass es nicht länger nur mein Board war – es war das Board des Teams, *unseres*. Es hörte auch nicht beim Board auf. Würde ich beispielsweise versuchen, Arbeit in die »zu testen«-Spalte zu bewegen, würde ich sofort gefragt werden: »Mike, ist dein Code bereits überprüft worden?« und ich müsste zugeben, dass dem nicht so war. Das Team hatte insofern Besitz vom Prozess ergriffen, als es das Selbstvertrauen hatte, sogar den Teamleiter herauszufordern. Darüberhinaus bedeutete dieser neugefundene Besitzanspruch über den Prozess eine neue Stufe im Bemühen um Qualität.

6.4 Warum Führung wertschätzen?

Geschichten wie diese zeigen, dass es keinen Widerspruch gibt zwischen dem Wertschätzen von Führung und gleichzeitigem Ermutigen von Selbstorganisation, wie sie in Kapitel 1 beschrieben wurde. Die beiden müssen sich nicht als Gegensätze gegenüberstehen.

Es ist mir nicht entgangen, dass einige Menschen und Communitys sich mit Führung schwer tun – vielleicht haben sie zu viel schlechtes Management erfahren oder es fällt ihnen schwer, die Begriffe Führung und Gleichheit in Einklang zu bringen. Gehen wir mal darauf näher ein.

Zum ersten Punkt: Schlechtes Management ist ein wirkliches Problem, das explizit aufgezeigt werden sollte. Alle Manager auf einen Streich zu verurteilen, wäre allerdings mit mindestens zwei von Kanbans Werten inkompatibel – **Verständnis und Respekt**. Anwender der menschlichen »Beginne mit dem, was du gerade tust«-Methode müssen disziplinierter sein.

Zum zweiten Punkt: Führung kann allen offenstehen, wenn die Umgebung es nicht aktiv unterbindet. Menschen können wählen zu führen oder nicht (und das ist okay); die wirkliche Last liegt auf den Schultern der Menschen, die bereits Führungspositionen besetzen und es zulassen, dass Führung aufblühen kann. Es sollte angemerkt werden, dass schwächere Ausdrücke wie Delegation und Bevollmächtigung (Empowerment) keinen ausreichenden Ersatz darstellen – zu häufig sind sie so eng beschnitten, dass sie keine wirkliche Möglichkeit zur Führung darstellen.

Um das klarzustellen: Wir messen Führung nicht den hohen Wert bei, weil wir hierarchische Organisationsstrukturen und Managementstile schätzen. Die brauchen wohl kaum unseren Segen! Wenn überhaupt, wird Kanbans Art von Führung *auf allen Ebenen* nur sehr schwer in manche dieser Kulturen hineinpassen. Der Grund, warum wir Führung einen so hohen Wert beimesse, ist, dass die Beziehung zwischen Führung und Veränderung einfach zu stark ist für eine Methode, die sich sehr um Veränderung kümmert, um sie zu ignorieren. Veränderung kann spontan sein, sogar zufällig, aber meist ist sie die Frucht von Führung. Und Führung, die keine Veränderung herbeiführt, ist was genau?

6.5 Wie sieht Führung in Kanban aus?

Weiter oben in diesem Kapitel haben wir einige wenige Wege aufgelistet, wie die ersten sechs Kanban-Werte auf Möglichkeiten zur Führung und Veränderung hinweisen. Es lohnt sich, diese noch einmal zu betrachten, um wenigstens teilweise ein Bild davon zu bekommen, was Kanban vorschlägt, was Organisationen von ihren Führungskräften benötigen. Dies gilt besonders, aber nicht ausschließlich für die Menschen in formalen Führungspositionen. Man benötigt kein besonderes Abzeichen, um in Kanban zu führen.

Aus dem Verbund **Transparenz**, **Balance** und **Kooperation** entsteht ein Bedürfnis nach Führungskräften, die sich dem Erhalt einer Umgebung verschreiben, in der Möglichkeiten zur Veränderung leicht erkannt und systematisch nachverfolgt werden können. Diese Führungskräfte streben bessere Ergebnisse für alle Beteiligten an und helfen bei der Schaffung von Bedingungen, in denen Menschen an der ständig präsenten Herausforderung der Verbesserung zusammenarbeiten.

Mit den Werten **Kundenfokus**, **Arbeitsfluss** und **Führung** stellen die Führungskräfte sicher, dass Veränderung im richtigen Kontext mit Ausrichtung und Ziel durchgeführt wird. Führungskräfte realisieren, dass einige der wichtigsten Erfolgskriterien extern statt intern liegen. Sie schätzen die zentrale Rolle des Arbeitsflusses zur effektiveren Bedienung von Kunden, zur Verbesserung der Sicherheit und des Wohlfühlens der Menschen innerhalb des Systems und für Verbesserung des Nettoprofits des Unternehmens. Die besten Führungskräfte in der Wirtschaft sind diejenigen, die Führung in anderen wachsen lassen, und sie begrenzen ihren Einsatz nicht nur auf die Bereiche, die sie kontrollieren.

Das ist nur der Anfang. Welche Bedingungen werden benötigt, damit diese Art von Führung gedeiht? Die nächsten drei Kapitel behandeln die drei verbleibenden Werte, nämlich die Führungsdisziplinen **Verständnis**, **Vereinbarung** und **Respekt**. Diese dringen zum Kern des Ansatzes der Kanban-Methode für Veränderung vor.

Reflexion: Kundenfokus, Arbeitsfluss und Führung

Stellen Sie sich vor, Sie gehen durch Ihre Organisation und stellen die Frage: »Was liefern wir wem und warum?« Welche Art von Antworten würden Sie erwarten?

- Wäre die Aussage »Zufriedenheit garantiert« überall richtig?

Falls ein Mangel an gleichmäßigem Arbeitsfluss Probleme verursacht, wo würde es in Ihrer Organisation am meisten wehtun? Wer würde es bemerken und wie würde es bemerkt werden?

Wie gut beschreiben die folgenden Wörter die Änderungen, denen Sie kürzlich begegnet sind (oder für die Sie verantwortlich waren)?

- Verordnung
- Kaskade
- Kampagne
- Big Bang
- ablenkend
- irrelevant
- auferlegt
- zufällig

Finden Sie dort, wo manche der Wörter gepasst haben, andere Formulierungen, die Sie sich eher wünschen würden?

Stellen Sie sich vor, ein Vorstandsmitglied besucht unangekündigt Ihren Arbeitsplatz. Sie bekommen von ihm drei Fragen gestellt:¹

- Was ist der Prozess?
- Wie können wir feststellen, dass er funktioniert?
- Wie verbessert er sich?

Wie ist Ihre Reaktion? Was könnte hier passieren?

1. Diese Fragen folgen dem Stil von Toyotas Führungsroutine oder Kata – vgl. Kapitel 14.

7 Verständnis

Der Wert **Verständnis** gehört zu Kanbans erstem Grundprinzip.

GP1: Beginne mit dem, was du gerade tust.

Angesichts des Titels dieses Kapitels wäre es verziehlich, sollten Sie sich gewundert haben, dass die Formulierung dieses Prinzips nicht hätte lauten sollen:

Starte mit Verständnis

So schön es auch klingen mag, es wäre falsch. Das könnte nämlich folgendermaßen verstanden werden: »Ändern Sie nichts, bis Sie ein großes Ist-Analyse-Projekt abgeschlossen haben« oder: »Starte damit, dein Denken an meines anzugeleichen« Wir beabsichtigen keines von beidem.

Mit Kanban reicht schon ein bisschen von der richtigen Art von Verständnis, um die gewünschte Reaktion zu beeinflussen. Danach erwarten wir eine Weiterentwicklung des Verständnisses, während die »Kanbanisierung« greift und Kanban-Systeme und die mit ihnen verbundenen Praktiken implementiert werden sowie eine Ausbreitung und Weiterentwicklung gefördert werden.

Der Fokus auf *was du gerade tust* ist durchaus beabsichtigt – es geht darum, Änderungen in der gegenwärtigen Realität zu verankern, jetzt und während sie sich weiterentwickelt. **Verständnis** repräsentiert die initiale Zusage und die andauernde Disziplin, diese Verankerung beizubehalten. Es ist schnell umsetzbar (Teil III erklärt, wie es geht) und doch beinhaltet es auch etwas Tieferes – es ist eine offene Einladung, Systemdenken und dessen verwandte Modelle zu erforschen (siehe Kap. 11).

7.1 Änderungen ohne Verständnis – drei Anti-Pattern

Drei einflussreiche Personen haben die essenzielle Beziehung zwischen Änderung und Verständnis betont. Sie identifizierten drei Management-Anti-Pattern – weitverbreitete Verhaltensmuster des Managements, die über ein Verständnisdefizit hinwegtäuschen und von denen jedes zu folgenreichen Konsequenzen führen kann:

- Russell Ackoff beklagte *Selbstzufriedenheit* [Ackoff 1991], die Sünde der Tatenlosigkeit beim Management.
- Jim Collins warnt in seinem Buch *Good to Great* [Collins 2001] vor *Draufgängertum*, bei dem das Management die Veränderungsfähigkeit der Organisation leichtsinnig als selbstverständlich sieht.
- William Edwards Deming – den wir in Kapitel 3 das erste Mal getroffen haben und in Kapitel 11 wiedersehen werden – sprach oft von *Einmischung*, der Tendenz des Managements, Unvorhersehbarkeit so zu adressieren, dass sie tatsächlich schlimmer gemacht wird.

Hüten Sie sich vor *Selbstzufriedenheit*, *Draufgängertum* und *Einmischung*: Veränderungen, die zu langsam, zu schnell oder zu zufällig sind.

7.1.1 Selbstzufriedenheit

Gerechtfertigt oder nicht, es ist einfach, dem Management Selbstzufriedenheit vorzuwerfen, wenn die Organisation von Ereignissen überrollt wird. Spektakuläres Scheitern ist jedoch vergleichsweise selten. Was ist aber mit einem langsamen Abfallen, Unachtsamkeit gegenüber Minderleistung oder mit dem Versäumnis, schädigendes Verhalten (egal auf welcher Ebene) zu erkennen oder zu behandeln, und so weiter?

Selbstzufriedenheit ist heimtückisch. Es ist schwer, etwas dagegen zu tun, weil es schwierig ist, es sich selbst einzustehen; und die Liste der Bereiche, in denen man der Selbstzufriedenheit beschuldigt werden könnte, ist tatsächlich sehr lang.

7.1.2 Draufgängertum

Geschichten von Draufgängertum sind nicht schwer zu finden. Die folgende Nachricht schaffte es auf die Titelseite des »Unternehmen & Märkte«-Teils der *Financial Times*:

G4S schiebt Skandale auf übermäßig aggressive Übernahmestrategie

Der neue Chef der G4S hat eine kurzfristige und übermäßig aggressive Übernahmestrategie für eine Reihe von Skandalen bei der Sicherheitsgruppe verantwortlich gemacht.

Einen Tag, nachdem das Betrugsdezernat eine Untersuchung zu Verträgen zwischen G4S und dem Vereinigten Königreich eingeleitet hat, meldete sich der Geschäftsführer Ashley Almanza zu Wort: »Ich glaube, dass wir als Managementteam nicht immer auf unsere Grundwerte fokussiert waren und ein kurzfristiger Fokus diese bisweilen untergraben hat.«

(Financial Times, 6. November 2013)

Ich habe nichts mit der Geschichte zu tun und ich weiß nicht, wie sie ausgehen wird, aber es sieht so aus, als ob das Überleben der gesamten Firma möglicherweise in Gefahr ist. Leider ist dies keineswegs eine einzigartige oder seltene Situation. Ich habe in meiner eigenen Karriere Erfahrungen aus erster Hand dazu machen können, was es heißt, wenn eine große Firma durch das Handeln oder die Tatenlosigkeiten ihrer leitenden Angestellten bis ins Mark erschüttert wird.

2008 schrieb der globale Finanzdienstleistungsriese UBS – zu diesem Zeitpunkt war er seit zehn Jahren mein Arbeitgeber – Verluste von knapp 38 Milliarden Dollar durch amerikanische Hypotheken ab. Das ist eine Zahl, die man nur schwer greifen kann. Um eine Vorstellung dazu zu geben: 38 Milliarden Dollar sind 5 Dollar für jeden Mann, jede Frau und jedes Kind auf diesem Planeten. Es sind wesentlich mehr, als die von den Vereinten Nationen geschätzten jährlichen Kosten von 30 Milliarden Dollar für ein Programm, das im selben Jahr vorgestellt wurde, um den Welthunger zu stoppen.¹ Sogar für eine Schweizer Bank, die daran gewöhnt ist, Einlagen in Größen zu erhalten, von denen ihre Konkurrenten weltweit nur träumen können, sind 38 Milliarden Dollar Verlust eine Menge Geld.

Die UBS war sich sehr bewusst, dass sie schwer angeschlagen war, und hatte schon eine Kapitalspritze von der Government of Singapore Investment Corporation (GIC) erhalten und bekam später weitere Unterstützung von der Schweizer Notenbank. Machen wir einen Sprung in die Gegenwart: Der Mitarbeiterstamm der UBS ist um Tausende gesunken und ihr Aktienpreis stagniert noch immer bei weniger als einem Viertel des Preises vor der Krise.

UBS war keinesfalls nur ein unschuldiges Opfer der Weltwirtschaft. Was besonders schwer zu verkraften war in dieser schwierigen Situation – und glauben Sie mir, es war für alle schwer, die diese Situation durchleben mussten –, war, welchen Teil des Grabs man selbst geschaufelt hatte, in dem die Bank beinahe gelandet wäre.

Auf Basis einer Vergleichsstudie externer Berater setzte UBS eine Zeit lang auf starkes Wachstum in relativ konkurrenzschwachen Bereichen, um ihre Marktposition zu stärken. UBS unternahm große Anstrengungen im Hypothekendarlehenmarkt, von dem der Subprime-Markt einen Teil ausmacht. Das vom Geld angefeuerte Wachstum des Geschäfts und der Mitarbeiterzahlen war deutlich größer als das der Fähigkeit zum Risikomanagement, was für eine Schweizer Bank ungewöhnlich ist. Ihre Verbindlichkeiten gerieten derart außer Kontrolle, dass der anschließende Fall fast unausweichlich wurde.²

-
1. »UN says solving food crisis could cost \$30 billion«, New York Times, June 4, 2008, <http://www.nytimes.com/2008/06/04/news/04iht-04food.13446176.html>.
 2. Die Ereignisse sind von UBS selbst in ihrem Investorenbericht aus dem April 2008 gut dokumentiert, Shareholder Report on UBS's Write-Downs http://www.static-ubs.com/global/en/about_ubs/investor_relations/agm/2008/agm2008/inagenda_jcr_content/par/linklist_9512/link.277481787.file/bGluay9wYXR0PS9jb250ZW50L2RhbS91YnMvZ2xvYmFsL2Fib3V0X3Vicy9pbnZlc3RvcI9yZWxhdGlvbnMvMTQwMzMzXzA4MDQxOFNoYXJlaG9.

Ich bin kein Fan von Vergleichsstudien, aber hier gebe ich weder der Studie noch den hochgesteckten Zielen der Bank die Schuld. Herausforderungen können motivierend, vereinend – ja sogar edel – sein. Das Problem stand in Verbindung mit Veränderungen: Es wurde zugelassen, dass die Veränderungen dem Wissen und der Leistungsfähigkeit der Organisation davonrannten. Wenn Individuen und Organisationen bedroht sind, ist *Draufgängertum* ein Sicherheitsrisiko.

Die meisten von uns werden keine Erfahrung mit einem Fehler der Größenordnung von UBS machen, aber Beispiele für Draufgängertum sind nicht schwer zu finden: aus der Luft gegriffene Liefertermine; für Projekte, deren Wert fragwürdig ist, werden Karrieren, Familienleben und Gesundheit aufs Spiel gesetzt; fehlgeschlagene Veränderungsprojekte nach fehlgeschlagenem Veränderungsprojekt.

7.1.3 Einmischung

Einmischung wurde als »Überreaktion auf Variation«⁶ definiert, was ich der engeren und technischeren Definition vorziehe: »Anpassungen aufgrund von Common-Cause-Variation, die zu zusätzlicher Variation führen.«³ Für mich sieht es so aus, als sollte Einmischung durch die menschliche Natur (die Übereifrigkeit einzugreifen) definiert werden statt durch sein Ergebnis.

Vielleicht klingt das Folgende bekannt:

- »Das darf auf keinen Fall noch einmal passieren«
als Antwort auf etwas, was für den Manager ungewöhnlich ist, was sich aber sehr wahrscheinlich nicht wiederholen wird.
- »Ab sofort ...«
als Einleitung für eine Verordnung
- »Diese Umstrukturierung wird folgende Vorteile mit sich bringen ...«
als Verkündung, dass das Pendel nun in die andere Richtung schwingt und der nächste Umbruch ansteht.
- »Es muss etwas getan werden«
einfach um jemandes Gewissen zu beruhigen.

Es ist einfach, mit diesen Dingen zynisch umzugehen, aber in der Wissensarbeit (und im Geschäftsleben generell) ist die unnötige Unvorhersagbarkeit wohl der kleinste Teil der Kosten, die durch Einmischung entstehen. Ich sehe zwei größere Kostenfaktoren:

1. Wenn Mitarbeitern wiederholt einzelne Entscheidungen abgenommen werden, verlieren sie Autonomie. Im Extremfall fühlen sich Menschen im Detail gesteuert oder sogar unsicher. In anderen Worten: Einmischung verursacht direkte menschliche Kosten.
3. Quelle: <http://curiouscat.com/management/deming/tampering>.

2. Wenn weitentfernte Manager Verordnungen als Strom von einzuführenden, kontextlosen Änderungen bekanntgeben, führt das zu einem »Regelzuwachs«, wie wir es nennen. Prozesse werden mehr und mehr begrenzt, was im Gegenzug Innovationen reduziert, zu Verzögerungen führt und die ökonomischen Kosten signifikant steigen lässt. Das sind nicht die Markenzeichen wachsender Prozessreife, sondern Zeichen, dass eine Sklerose beginnt. Die ökonomischen Kosten sind dauerhaft.

7.2 Einführung der J-Kurve

Wenn Ihre Organisation Kanban einführt, hat sie den Kampf gegen die Selbstzufriedenheit schon aufgenommen (das soll nicht heißen, dass eine Methode oder ein Werkzeug die komplette Heilung herbeiführen kann). Sie beginnt, die Notwendigkeit von Veränderung sichtbar zu machen, und sollte bald Daten haben, die aufzeigen, dass sie sich in die richtige Richtung bewegt. Erlaubt man Kanban in dem Rahmen zu operieren, den es braucht, ist es von Natur aus anti-selbstzufrieden.

Um die zwei anderen Anti-Pattern, Draufgängertum und Einmischung, zu adressieren, braucht es ein wenig mehr. Die J-Kurve ist hilfreich dabei, dies zu visualisieren. Auf die Kurve kommen wir im nächsten Kapitel noch einmal zurück und schauen uns in Kapitel 17 eine beachtenswerte Analyse einer J-Kurve an. Vorerst beschränken wir uns aber auf ihre allgemeine Form (siehe Abb. 7-1).



Abb. 7-1 Die J-Kurve

Die vertikale Achse bezieht sich typischerweise auf Größen wie *Leistung*, *Leistungsfähigkeit*, *Fitness* oder *Wohlbefinden* (alle vier funktionieren in diesem Kontext). Die J-Kurve zeichnet die Auswirkung von Veränderungen über die Zeit auf. Ihre Form legt drei Fragen nahe:

1. Werden wir am Ende besser abschneiden, als wir es zu Beginn taten?
Wie sicher können wir sein, dass wir uns nach der Änderung über bessere Leistung, Leistungsfähigkeit, Fitness oder Wohlbefinden freuen können?
2. Werden wir den Übergang aushalten können?
Ist der zwischenzeitliche Verlust von Leistung, Leistungsfähigkeit, Fitness oder Wohlbefinden nachhaltig?
3. Sind wir geduldig genug?
Wie viel Zeit darf vergehen, bevor der Drang, die Reise abzubrechen, zu stark wird, eine andere Initiative fehlschlägt und der Change Agent gefeuert wird?⁴

Je hastiger oder großspuriger die Veränderung konzipiert ist, desto wahrscheinlicher ist es, dass eine dieser Fragen negativ beantwortet wird.

- Mein Freund Andy Carmichael sagt dazu:
 - *Evolutionäre Veränderung führt uns von einem Punkt, an dem wir überlebensfähig sind, zum nächsten Punkt, an dem wir überlebensfähig sind.*
- Worauf ich antworte:
 - *Und die Reise muss auch überlebbar sein!*

Sie haben bis jetzt überlebt; ein wenig Anerkennung für das, *was man gerade tut*, muss es das wert sein. Genauso wie Wissen über das umgebende Terrain (ein paar Ideen über mögliche Auswirkungen, wenn man von dem, was man gerade tut, sich entfernt), sodass eine sichere Richtung für die Reise gewählt werden kann.

7.3 Ein Muster für zielgerichtete Veränderung

Falls dies zu abstrakt war, lassen Sie uns das Prinzip Verständnis etwas konkreter machen:

GP1 (erweitert): Beginne mit dem, was du gerade tust, und verstehe.

- den Zweck des Systems,
- wie es dem Kunden dient,
- wie es für diejenigen innerhalb des Systems funktioniert,
- wie es Kunden unzufrieden macht und Mitarbeiter frustriert,
- wie es sicher verändert werden kann.

4. ... oder befördert! Ich verdanke David den Satz »Change Agent wird gefeuert«.

Dies umfasst sowohl Praktik als auch Prinzip – es ist sogar eine sehr konzentrierte Zusammenfassung von STATIK⁵, der in Teil III beschriebenen Einführungsmethodik. Diese funktioniert sowohl für die Einführung von Kanban als auch für nachfolgende Veränderungen.

Versteht man den Zweck des Systems, führt es uns von *was wir gerade tun* zu warum wir dem Kunden so dienen, wie wir es gerade tun. Mein damaliges Team in Budapest brachte es von »*Wir bauen und unterstützen Risikomanagementsysteme für Energie*« zu »*Wir bauen und unterstützen die Systeme, die es unserer Firma ermöglichen, unseren Kunden dabei zu helfen, ihre Energierisiken zu verwalten*«. Für ein Team, das sich schwer damit tat, über ein »*Wir sind Softwareentwickler*« hinwegzukommen (und wir waren definitiv nicht von dieser Art), war das Verstehen des Zwecks ein großer Schritt.

Man nimmt Unzufriedenheiten und Frustrationen als Indikatoren dafür, dass es entweder Hindernisse gibt, die das System davon abhalten, seinen Zweck zu erfüllen, oder dass wir diesen Zweck noch nicht vollends verstehen. Diese Hindernisse, die möglichen Gründe und ihre Auswirkungen anzuerkennen, ist ein erster wichtiger Schritt hin zu wirkungsvollen Veränderungen.

In Budapest waren die Frustrationen von Kunden und des Teams gegenseitige Spiegelbilder. Man konnte also annehmen, dass sie denselben Grund hatten. Verständlicherweise wollten unsere (internen) Kunden ihre Bedürfnisse schneller erfüllt sehen. Seitens der Entwickler waren wir frustriert durch ständige Wechsel in den Prioritäten und Verspätungen, die anscheinend von der Kundenseite verursacht wurden. Wir mussten irgendwie enger zusammenarbeiten.

Wie in den vorherigen Kapitel aufgezeigt, implementierten wir über einen Zeitraum von mehreren Monaten eine schnelle Abfolge von Veränderungen, die die Frustrationen nacheinander adressierten. All diese Veränderungen waren sicher genug und umkehrbar und keine erzeugte signifikante Ängste (jedenfalls nicht in der Größenordnung der Frustrationen, die sie adressierten). Dies war *evolutionäre Veränderung*, die bewusste Strategie, die zu unserem nächsten Wert **Vereinbarung** gehört.

5. Genau genommen fasst es STATIK-0 zusammen: »*Verstehe den Zweck des Systems*« ist Schritt 0 des »fortgeschrittenen Modus« von STATIK.

8 Vereinbarung

Kanbans zweites Grundprinzip ist sehr direkt:

GP2: Vereinbare, evolutionäre Veränderung zu verfolgen.

Etwas ausführlicher heißt es: Vereinbare, dass Veränderung notwendig ist; vereinbare, sie mit einer evolutionären Strategie zu verfolgen.

8.1 Verfolge evolutionäre Veränderung

Das Wort *verfolgen* ist mit Bedacht gewählt. Es ist so viel stärker als *anwenden* und es beschwört die Energie und Beharrlichkeit herauf, jeglicher *Selbstzufriedenheit* zu begegnen. Es erinnert uns gleichzeitig daran, dass es ein andauernder Prozess ist. Verfolgen kombiniert Herausforderung und Verpflichtung.

Die Strategie von beabsichtigter, evolutionärer Veränderung kann nicht heißen »lassen wir die Natur entscheiden«. Sondern:

- Sie startet (und startet immer wieder) dort, wo man gerade ist – was man gerade tut, in der aktuellen Marktumgebung usw.
- Sie ist ergebnisoffen – es geht nicht darum, einem Plan bis zu einem konzipierten Ende zu folgen, dessen Tauglichkeit nur erraten werden kann.
- Sie akzeptiert es, dass jede Änderung (sowohl in der Umgebung als auch intern erzeugt) Auswirkungen und Reaktionen bewirkt, die nicht vorhergeschenken werden können; Veränderung ist unvermeidbar *komplex* und dynamisch.
- Sie ist das Verfolgen von Fitness, statt »Veränderung um der Veränderung willen«. Fitness kann in absoluten Begriffen definiert werden, ist aber letztendlich die Eignung für einen bestimmten Zweck relativ zur Konkurrenz, die das Überleben bestimmt.
- Indirekt geht es auch um *Anpassungsfähigkeit*. Das Verfolgen von Fitness beinhaltet regelmäßige Veränderung. Jede Organisation, die evolutionäre Veränderung verfolgt, entwickelt die Fähigkeit zur Anpassung, also die Fähigkeit zur Reaktion auf Veränderung. Das tut sie unabhängig davon, ob sie explizit daran arbeitet (wobei ich der Meinung bin, sie sollte es gerade explizit tun).

Starten mit dem, was Sie gerade tun, immer und immer wieder, unbefristet, unerbittlich. Ihre Konkurrenz wird nicht nachlassen, also sollten Sie das selbst auch nicht; der ideale Endpunkt (falls es jemals so etwas geben sollte) wird sich mit der Änderung der Umgebung ändern.

Wir können die J-Kurve aus dem vorherigen Kapitel verwenden und mehrere J-Kurven miteinander verbinden (siehe Abb. 8–1).

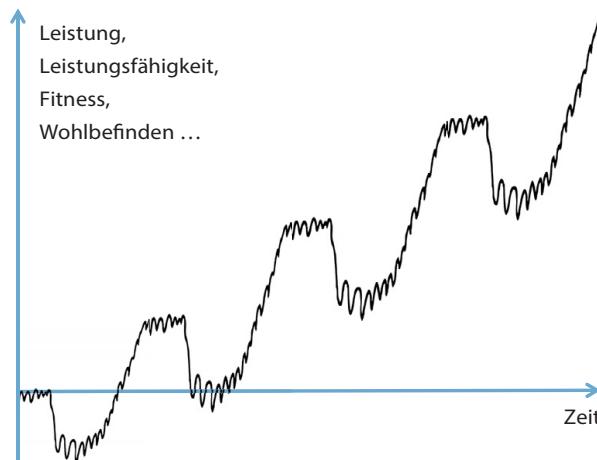


Abb. 8–1 Eine Abfolge kleiner J-Kurven

In Abbildung 8–1 sind zwei wichtige Dinge vereinfacht worden, verglichen mit den Erfahrungen, die man üblicherweise macht:

1. Es ist unwahrscheinlich, dass jede Veränderung erfolgreich ist. Es ist dementsprechend nur in den extremsten Situationen angemessen, Veränderungen durchzuführen, die beim Fehlschlagen sehr wahrscheinlich bleibende Schäden verursachen. Im Normalfall sollten Veränderungen *safe-to-fail* sein. Die Grafik sollte unter Umständen eher wie in Abbildung 8–2 aussehen.
2. In der Praxis ist das »Experimente-in-Progress-Limit« solcher *Safe-to-fail-Experimente* nicht maximal eins. Lässt man verschiedene Experimente nebeneinander laufen, erhöht das die Wahrscheinlichkeit, etwas unerwartet Wertvolles zu entdecken.

Eine Nebenbemerkung: Auch in der Produktentwicklung gilt die Strategie der starken Parallelisierung: Viele verschiedene Experimente durchführen und diverse Gruppen von Menschen Ideen erzeugen lassen. Genau wie die Selbstorganisation erhöht Diversität die Resilienz und inhomogene Gruppen lösen Probleme schneller. Die Parallelen sollten nicht überraschen – der Service ist natürlich Teil des Produkts.

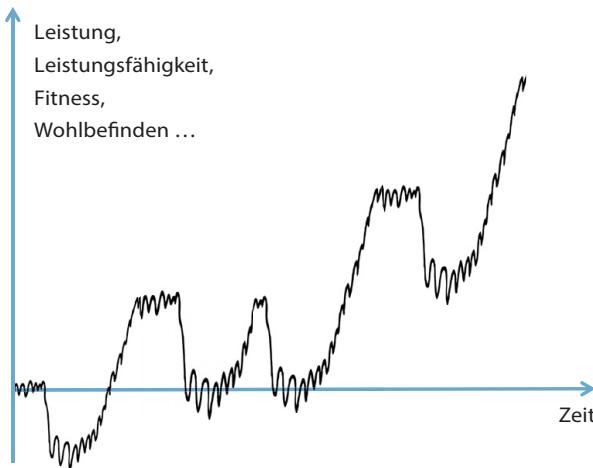


Abb. 8-2 Nicht jedes Experiment wird erfolgreich sein.

8.2 Vereinbare das Verfolgen

In Kapitel 5 habe ich darum gebeten, dass Sie den Spielraum Ihrer Kanban-Implementierung nicht als gegeben ansehen. Jetzt möchte ich daran erinnern, dass das **Vereinbarungs-Prinzip** (GP2, in anderen Worten) mit dem Wort *vereinbaren* beginnt. Man sollte sich der involvierten Menschen und derer, die wahrscheinlich betroffen sind, nicht sicher sein.

Als Analogie fällt mir dazu ein Autoaufkleber ein. Der dürfte wahrscheinlich jedem bekannt sein, der in den späten Siebzigern und danach im Vereinigten Königreich gelebt hat:

Ein Hund ist fürs Leben, nicht nur für Weihnachten.¹

Vereinbarung gilt nicht nur für das Kick-off. Die meisten Veränderungen, die durch Kanban entstehen, werden nur durch Vereinbarung hängen bleiben. Man sollte also so starten, wie man auch weitergehen will. Teil III wird Ihnen dabei helfen. Er beschreibt ein hochgradig wiederholbares Vorgehen, das konzipiert wurde, um *Vereinbarung in der Praxis* zu generieren, statt der halbherzigen *Vereinbarung aus Prinzip*. Das zweite ist schnell vergessen, sobald es Arbeit zu erledigen gibt.

1. Clarissa Baldwin. 1978. *The Dogs Trust*. <http://www.dogstrust.org.uk/az/a/adogisforlife>.

8.3 Change Management

Vereinbarung ist etwas, was Leute zusammenschweißt; es ist eine überlebenswichtige, soziale Fähigkeit und ein Prozess. Im Kontext der Organisation ist Vereinbarung eine nützliche Abkürzung für eine wertvolle Fähigkeit, das *Change Management*.

Die grundlegende Idee des Change Management ruft eine Reihe von Emotionen hervor, manchmal auch alle zugleich. Gut gemacht bemerken wir es kaum – wir fühlen uns als Teil der Entscheidung und es kümmert uns nicht besonders, woher es kommt. Schlecht gemacht stört es und wir wehren uns dagegen.

Diese Gefühle scheinen darauf zu beruhen, wie die Veränderung eingeführt wurde und wie die positiven Auswirkungen der Veränderung aussehen. Wir könnten keinen Widerspruch darin sehen, sowohl die Notwendigkeit einer Veränderung vollständig zu akzeptieren und uns trotzdem gleichzeitig persönlich von der Aussicht darauf angegriffen zu fühlen.

Es gibt Momente, da ist dieser Unmut vollkommen gerechtfertigt. Schnell werden Versuche durchschaut, Veränderung mit unglaublicher Krisenstimmung durch das Heraufbeschwören von Bildern wie der *brennenden Plattform* zu beschleunigen, wenn klar ist, dass weder Leben noch Lebensgrundlagen in unmittelbarer Gefahr sind. Ultimaten (als unterschwellige Bedrohung) und Erlasse (kein Versuch, Vereinbarung zu erreichen, nur Einwegkommunikation) rufen selten Wohlwollen hervor. Oder man wird desorientiert zurückgelassen, wenn das Management Veränderungen unzureichend oder ungeschickt kommuniziert, sodass man gar nicht mehr weiß, was überhaupt los ist.

Drei Modelle für Change Management heben die Rolle von Vereinbarung im Veränderungsprozess hervor:

1. Veränderung wird von einem *Change Agent* geleitet.
2. Veränderung wird von einem *Change Agent* in Kombination mit einem *Mentor* geleitet.
3. Veränderung wird von einem *Change-Team* geleitet.

8.3.1 Der Change Agent

Vereinfacht gesagt sind Change Agents Menschen, die dafür sorgen, dass Veränderungen geschehen. Es ist ein weiter Begriff: Eventuell folgen sie einem Implementierungsplan, rufen Veränderungen durch die Kommunikation einer Vision hervor, gehen als Beispiel voran usw. Sie könnten die Urheber von Veränderung oder Anstifter dazu sein oder sie können damit beauftragt sein, diese Veränderung durchzuführen.

Change Agents positionieren sich oft an den Verknüpfungspunkten dieser vier Gruppen von Stakeholdern:

1. Die Menschen, deren Arbeitsablauf durch die Veränderung am stärksten betroffen sein wird. Der Fokus des Change Agent liegt häufig auf ihnen.
2. Die etwas außerhalb Stehenden, deren Mitarbeit die Implementierung der Veränderung wahrscheinlich einfacher machen wird.
3. Die Kunden, die direkt oder indirekt von der Veränderung profitieren.
4. Jene, denen der Change Agent gegenüber persönlich Rechenschaft schuldig ist – Linienmanager, ein Projektausschuss, eventuell Gesellschafter.

Viele von uns können sich an wirklich exzellente und lobenswerte Beispiele erinnern, wie dieses Modell verwendet wurde. Kapitel 4 des »blauen Buches« beschreibt, wie Dragos Dumitriu (heute ein Freund und Kollege von mir) versuchte, die Probleme von Microsofts XIT-Team aus den verschiedenen Blickwinkeln der Teammitglieder, deren Kunden und dem Rest der Organisation zu verstehen. Mit einer Idee, wie es vorzugehen kann, betrieb Dragos eine geschickte Shuttle-Diplomatie, die ausreichend Vertrauen für ein radikales Experiment aufbaute. Dragos vermied auf clevere Art und Weise das große Entscheidungstreffen, bis der Erfolg gesichert war. Das XIT-Team entwickelte sich, obwohl es zuvor noch dem Untergang geweiht war, (in den Worten des Kapiteltitels) »von den Schlechtesten zu den Besten in fünf Quartalen«.

Ich vermute, dass auf jede gute Geschichte zahlreiche andere kommen, die nur wenige feiern würden. Geschichten, in denen Folgendes vorkommt:

1. Der Change Agent glaubt, dass seine oder ihre größte Aufgabe darin besteht, den von Teammitgliedern ausgehenden *Widerstand gegen die Veränderung zu überwinden*.
2. Kollegen wird nur spärlich Beachtung geschenkt – sie werden eher »auf dem Laufenden gehalten« als eingebunden.
3. Der Kunde wird kaum erwähnt.
4. Das Management erhält bis zu dem Moment gute Nachrichten, an dem die Initiative abgebrochen wird.

Diese Beispiele zeigen vier Gruppen von Stakeholdern, von denen jede einzelne vernachlässigt wird. Sehen wir uns an, wie ein guter Mentor in solch einer Situation behilflich sein kann.

8.3.2 Der betreute Change Agent

Typischerweise unterstützt die Rolle des Mentors jemandem dabei, die Dinge so zu sehen, wie sie wirklich sind, und verhilft dadurch dem Betreuten (dem Change Agent), zu besseren Entscheidungen zu gelangen. Die Aufgabe des Mentors ist es jedoch nicht, die Situation durch die Augen des Betreuten tiefgehend zu analysie-

ren; es ist mehr ein Prozess, mit Authenzität und Souveränität auf relevante Erfahrungen zurückzugreifen und sicherzugehen, dass die Gedanken des Change Agent einer ausreichenden Selbstüberprüfung unterzogen werden (»führen durch Analogie«, sozusagen).

In der XIT-Geschichte war David Anderson der Mentor von Dragos. Beide arbeiteten zu der Zeit bei Microsoft, waren jedoch keine direkten Kollegen. Es waren eher Davids veröffentlichte Arbeiten, die Dragos ihn aufsuchen ließen. David war in Dragos Situation nicht direkt beteiligt (sie kannten sich kaum), er konnte Dragos jedoch dabei behilflich sein, auf eine neue Art auf das Problem zu schauen.

Im Lean-Modell (Kap. 14, siehe auch Kap. 17) ist der Mentor eher stärker involviert. Er ist also nicht unbedingt unabhängig von der Situation. Die Linien zwischen Mentor, Manager und Coach können in diesem Modell etwas verschwommen erscheinen. Dies mag dazu auffordern, sich selbst Gedanken zu machen, wie man die Rollen interpretiert, aber das Modell scheint zu funktionieren. Es funktioniert, weil der Fokus der Beziehung auf der Situation und deren Problemen liegt und nicht auf dem betreuten Change Agent.

Hier sind ein paar Fragen, mit denen der Mentor den Prozess sanft führen kann. Sie sind gruppiert nach den beteiligten Gruppen und ihren Herausforderungen im Change Management:

1. **Welche Lösungen hat das Team für das Problem in Betracht gezogen?**
Eine heikle Frage, die herauszufinden hilft, ob das Team in richtiger Art und Weise beteiligt wird. Sie kann dabei auch helfen, dem Problem einen besseren Rahmen zu geben, falls der Change Agent zu sehr auf eine Lösung fokussiert ist.
2. **Was genau haben deine Kollegen (und andere Betroffene) vereinbart zu tun?**
Es reicht nicht, nur zu informieren; hier führt der Mentor seinen Schützling in Richtung Vereinbarungen, die auch hängen bleiben. Es ist eine Chance, breiteren Nutzen zu finden, den man ansonsten nicht in Erwägung gezogen hätte.
3. **In welcher Art und Weise nützt es dem Kunden?**
Es ist sehr unwahrscheinlich, dass sich eine Veränderung als Verbesserung qualifiziert, wenn sie keinen Nutzen für den Kunden hat.
4. **Wie sieht das Ganze aus Sicht der Organisation aus?**
Ist es kostenwirksam? Orientiert es sich an Strategie, Regeln und Werten?

Und es gibt eine fünfte Überlegung:

5. **Lösen wir das richtige Problem?**
Auch hier geht es wieder um den Rahmen, aber auch um den Umfang. Wählt man die Reichweite zu klein, verpasst man eine Möglichkeit; wählt man sie zu groß, hat der Change Agent nur geringe Chancen, einen bedeutenden Erfolg zu erzielen.

8.3.3 Das Change-Team

So viele Stakeholder! Da sind die Impulsgeber für Veränderung, jene, die Details der Veränderung gestalten, die davon Betroffenen und die Profiteure. Hinzu kommen noch die Mentoren, Coaches und Manager der eben Genannten. Was wäre, wenn das zum größten Teil dieselben Leute wären – ein Team? Wäre das nicht einfacher?

Diese Dinge passieren natürlich nicht zufällig. Meine österreichischen Freunde und Kanban-Experten Klaus Leopold und Sigi Kaltenecker unterrichten die Strategien und Fähigkeiten, die benötigt werden, um ein solches *Change-Team* zu bilden, Teil dessen zu sein und es zu führen. Ich werde ihre Arbeit hier nicht wiederholen (ich möchte Sie jedoch dazu ermutigen, ihr Buch [Leopold & Kaltenecker 2013] zu lesen oder eine ihrer Schulungen zu besuchen). Wir sollten allerdings eine kurze Pause machen und darüber nachdenken, was passiert, wenn man all diese Leute vor ein Kanban-Board stellt.

Der Arbeitsfluss und seine Hindernisse sind gemeinsam sichtbar. Falls es ein größeres Problem zu lösen gibt, kann der Einfluss auf das Design des Prozesses – der direkt von allen Anwesenden zu sehen ist – durchgespielt werden. Wenn Kompromisse geprüft werden müssen, kann dies anhand der tatsächlichen Arbeitspakete geschehen.

Was das Change-Team hier macht, unterscheidet sich kaum von dem, was sonst vor dem Board passiert. Das ist natürlich genau der Punkt: Kanban ist die Integration einer Veränderungsfähigkeit in den wirklichen Betrieb der Organisation.

Jedes Team, das Kanban nur als Produktivitätswerkzeug verwendet, verpasst diesen entscheidenden Punkt. Eventuell entscheiden Sie sich trotz dieses Wissens dafür, es weiterhin »unter dem Radar« zu belassen, in der Hoffnung, dass es zu einer nicht zu stoppenden, viralen Verbreitung kommt. Wann ist jedoch der richtige Zeitpunkt, zu dem man die Organisation dazu einlädt, bewusste Entscheidungen bezüglich Geschwindigkeit und Zielen zu treffen? Ich fühle mich mit dieser Herangehensweise ausgesprochen unwohl.

Die Kanban-Methode basiert auf **Vereinbarung**. Der Kontext und der Ton für alles Folgende wird dadurch gesetzt, dass man sich über die Grundsätze verständigt. Sieht man sich die Implementierungen detaillierter an, repräsentieren Einverständnis und Vereinbarung Fähigkeiten, für deren Weiterentwicklung sich die Organisation entscheiden kann. Für einzelne Führungskräfte ist Vereinbarung eine Schlüsseldisziplin. Ad-hoc-Implementierungen riskieren, all diese Dinge zu verpassen.

9 Respekt

Der neunte und letzte Wert ist die dritte unserer drei *Führungsdisziplinen*. Was wäre, wenn jede Veränderung mit **Verständnis, Vereinbarung und Respekt** umgesetzt werden könnte?

Respekt untermauert das dritte Grundprinzip von Kanban:

GP3: Respektiere initial bestehende Prozesse, Rollen, Verantwortlichkeiten und Jobtitel.

Hin und wieder merkt jemand an, dass das Prinzip überflüssig sei. Warum sollte die menschliche »*Beginne mit dem, was du gerade tust*«-Methode es nötig haben, dieses Prinzip explizit zu benennen?

Mir fallen mindestens drei wichtige Gründe dafür ein, dass es aufgenommen wurde:

1. Es beinhaltet einen pragmatischen Change-Management-Rat: Beginne deine Initiative nicht mit dem Begutachten der Rollen.
2. Es deutet auf eine zugrunde liegende Philosophie hin: Finde Wege um Hindernisse herum, um Veränderung zu erzielen.
3. Kein anderes Prinzip oder andere Praktik drückt so direkt »*Respekt für die Menschen*« aus.

9.1 Fang nicht bei den Rollen an

Lassen Sie uns ein paar frühere Ratschläge zusammenfassen und dabei die Betonung etwas ändern.

- Beginne mit dem, was du gerade *tust*, und sieh dir an, wie das, *was du gerade tust*, die Bedürfnisse der Menschen innerhalb und außerhalb des Systems erfüllt (oder auch nicht erfüllt).
- Organisiere die Arbeit; lasse zu, dass sich die Menschen um die Arbeit herum selbst organisieren, und erlaube dem System, sich zu verändern.

Ein voreiliger Fokus auf Rollen würde diesen Prozess unwiderruflich schwächen. Falls die Rollen überhaupt formal definiert sein müssen, können sie an den Prozess angepasst werden, sobald – im echten Kontext – bewiesen wurde, dass dieser seinen Zweck ausreichend erfüllt. Bis dahin machen wir es den Menschen sehr viel einfacher, diese Anpassungen selbst zu finden. Warum sollten wir das Thema erzwingen?

9.2 »Sei wie Wasser«

Ein voreiliger Fokus auf die Rollen hat einen Effekt, der höchst unerwünscht ist: Er provoziert Widerstand. Wie logisch es vielleicht auch sein mag, es ist schwer, von anderen zu hören, dass Rollenveränderungen nötig sind. Ganz gleich wie positiv die Absicht ist oder wie gründlich es erklärt wird, die gehörte Nachricht beinhaltet nur Unzulänglichkeit und Belanglosigkeit. Keiner möchte hören, dass er die Bedürfnisse der Organisation nicht erfüllt; eine Abwehrhaltung ist fast unvermeidbar.

Die Kanban-Überlieferung nimmt sich den Satz »*sei wie Wasser*« aus einer eher ungewöhnlichen Quelle. Bruce Lee war berühmt für seine Kampfkünste, er war jedoch weniger bekannt für seine Philosophie. Er benutzte diese Redewendung mehr als nur einmal; wir beziehen uns auf die folgende Erweiterung:

Sei wie Wasser, das seinen Weg durch Spalten findet. Sei nicht starr, sondern passe dich dem Objekt an, und du wirst einen Weg herum oder hindurch finden.¹

Die Kanban-Methode kommt mit dem Warnhinweis, dass etablierte Rollen als Fels anzusehen sind. Das lässt uns manchmal den rollenzentrierten Ansatz anderer Frameworks hinterfragen. Warum sollte man so beginnen, dass man sagt, dass bestimmte Rollen (Produktmanager, Projektmanager, Businessanalysten, Tester usw.) gebraucht und wertgeschätzt werden und andere nicht? Warum sollte man den Felsen attackieren, wenn man ihn so leicht umgehen kann und es auch sonst so viel zu tun gibt?

Es wird natürlich andere Felsen geben. Ob man sich ihnen stellt oder um sie herum geht, ist eine Gefühlssache. Wenn Sie wählen können, empfehlen wir den Pfad der Anpassungsfähigkeit, jenen, der den Veränderungsprozess am Laufen hält. Eine andere gute Orientierungshilfe ist **Respekt**: Stellt man sich dem Felsen nur, weil er der nächste auf der Liste ist oder weil Bedürfnisse von Menschen nicht erfüllt werden können, bis das Thema geklärt ist?

1. Es ist mir nicht gelungen, die definitive Quelle dieses längeren Zitats zu finden. Ob es Bruce Lee selbst war oder einer seiner vielen Jünger oder Kommentatoren, der es geschrieben hat, kann ich nicht verlässlich sagen.

9.3 Respekt für Menschen

Ich gebe hier gerne zu, dass wir Toyota und Lean gegenüber etwas schuldig sind. Mehrere Werte von Kanban korrespondieren mit den Prinzipien von Lean, keiner jedoch so sehr wie **Respekt**. *Respekt für Menschen* wurde schon vor Langem als eine der sogenannten »Säulen« von Lean erkannt. Neben *kontinuierlicher Verbesserung* ist er heute (seit 2001) eine der beiden Säulen des »Toyota Way«, der offiziellen Managementphilosophie von Toyota.

Allerdings soll dieses Kapitel nicht erklären, was *Respekt für Menschen* innerhalb Toyota oder für Anwender der Lean-Methoden bedeutet; das wird in Kapitel 14 behandelt. Stattdessen soll das Kapitel zeigen:

1. Wie die Kanban-Methode der Überprüfung, die **Respekt** darstellt, standhält.
2. Wie **Respekt** bei der Kanban-Implementierung leiten kann.

Da wir das Ende der wertebasierten Einführung der Kanban-Methode erreichen, ist es eine gute Gelegenheit, Wert für Wert zu rekapitulieren.

9.3.1 Transparenz

Mit **Transparenz** fordert uns Kanban auf zu visualisieren, Regeln explizit zu machen und Feedbackschleifen einzuführen. Diese Praktiken sind dann respektvoll, wenn sie mit der Erwartung durchgeführt werden, dass Menschen bessere und sinnvolle Entscheidungen treffen, wenn ihnen die richtigen Informationen, Werkzeuge und Gelegenheiten gegeben werden.

Respekt muss in beide Richtungen gehen. Man sollte nicht vergessen, dass die Organisation mit den Ergebnissen dieser Entscheidungen leben muss; vielleicht verdienen die Führungskräfte der Organisation ein wenig Respekt dafür, dass sie das Umfeld geschaffen haben, in dem solche Entscheidungen getroffen werden können.

In einigen fortgeschrittenen Konzepten von Transparenz ist Respekt inhärent. Vielfalt willkommen zu heißen – also zum Beispiel die Idee abzulehnen, dass jedes Stück Arbeit mit einem Enddatum versehen werden muss –, ist nicht einfach nur »nett«. Es verbessert die Ergebnisse insgesamt und steigert die Resilienz des Systems. Die Resilienz wird noch weiter gesteigert, wenn die Mitarbeiter den Raum haben, mit relevanter Autonomie zu handeln, und Systeme sich selbst organisieren können. Wer will schon in einem fragilen System arbeiten? Ich sicher nicht.

Gibt es so etwas wie »zu viel Transparenz«? Die radikaleren unter meinen Freunden vertreten mit Nachdruck den Standpunkt, dass Transparenz unbegrenzt sein sollte. Selbst wenn sie recht haben (und es gibt einige interessante Beispiele von extrem transparenten Unternehmen), würde ich gern Folgendes fragen: Ist es möglich, Transparenz gedankenlos und rücksichtslos zu implementieren? Ich bin sicher, dass es geht.

9.3.2 Balance

Für die **Balance** ist das erste und wichtigste Ziel von Kanban, Anforderungen und Leistungsfähigkeit aneinander anzupassen. Das wichtigste Werkzeug dafür ist die Limitierung paralleler Arbeit (Work in Progress). Die Limitierung hilft den Menschen dabei, sowohl Überlastung als auch Aushungern zu vermeiden und ihre Hindernisse besser sichtbar zu machen. Dies erscheint erst einmal respektvoll; aber auch hier kann Respekt zur Überprüfung der Maßnahmen dienen. Mit aggressiven Limits voranzupreschen, ohne die individuelle Leistungsfähigkeit oder Präferenz mit einzubeziehen, kann negative Folgen nach sich ziehen.

Balance erweitert Respekt auf die anderen beteiligten Gruppen. Risiken und Vorteile können gerecht verteilt werden. Respekt sollte dazu führen, dass man »Nullsummen-Verbesserungen« ablehnt, die einen Teil der Beteiligten auf Kosten der anderen profitieren lässt.

9.3.3 Kooperation

Im Respekt für Menschen scheinen die Grundlagen dieser Aspekte von **Kooperation** zu liegen:

- Kooperativ zusammenzuarbeiten, um etwas von Wert zu liefern
- Kooperativ tiefliegende Probleme zu lösen und so das System zu verbessern
- Kooperation sowohl als Ziel als auch als Weg zu behandeln und damit größere Chancen auf qualitativ hochwertigere Zusammenarbeit zu schaffen

Natürlich möchte nicht jeder zu jeder Zeit kooperativ arbeiten. Menschen brauchen auch Raum für sich selbst. Noch einmal, anhand des Respekts kann man dies prüfen.

9.3.4 Kundenfokus

Auf der Suche nach dem Sinn unserer Arbeit ist es nicht schlecht, bei »Respekt für den Kunden« zu starten. Befriedigung findet sich in der Erfüllung und noch mehr im Vorausahnen von Bedürfnissen; ich habe keinen Zweifel daran, dass **Kundenfokus** ein menschlicher Wert ist. Wird unsere Fähigkeit vorauszuschauen bei einem Mangel an respektvollen Konversationen und Selbstbeobachtung wesentlich wachsen können? Unwahrscheinlich.

9.3.5 Arbeitsfluss

Den **Arbeitsfluss** wertzuschätzen bedeutet, einen so großen Wert auf Gleichmäßigkeit und Pünktlichkeit zu legen, dass entsprechende Verbesserungen am System unternommen werden. Arbeitsfluss kann gemessen werden, sollte aber nicht nur auf Zahlen reduziert werden – er muss auch erfahren und genossen werden.

Hat man den Fluss der Arbeit einmal zu schätzen gelernt, schmerzt seine Abwesenheit. Nichtsdestotrotz ist manchmal ein abrupter Richtungswechsel gerechtfertigt; bei solchen Ereignissen könnte man sagen: »*Wert sticht Fluss*«², und man lebt für eine kurze Zeit mit den Unannehmlichkeiten. Verfallen wir aber zu häufig ins Akzeptieren, könnte man unsere Selbstverpflichtung zu Arbeitsfluss und seinen Vorteilen – ökonomisch wie menschlich – hinterfragen.

9.3.6 Führung

Bei *Führung auf allen Ebenen* geht es offensichtlich um Menschen allgemein und nicht nur um den Menschen, der zufällig die höchste Position innehat oder die Rolle des Change Agent besetzt. Kanbans **Führungsansatz** bedeutet, das Potenzial des Unternehmens als ein selbsterhaltendes System zu sehen, gewünschtes Verhalten so zu leben, dass es dauerhaft erhalten bleibt, Gelegenheiten zu ergreifen und für andere zu schaffen und angemessene Risiken einzugehen. So verändern sich Systeme von innen heraus.

Um Führungsexzesse zu begrenzen, gibt es die **Führungsdisziplinen Verständnis, Vereinbarung und Respekt**. Und falls Sie immer noch mit dem Wert Führung zu kämpfen haben, sollten Sie stattdessen über *Chancen* reflektieren.

9.3.7 Verständnis

Verständnis ist der Aufruf, das System nicht nur als Funktionen und Aktivitäten wahrzunehmen, sondern auch die Menschen innerhalb und außerhalb des Systems mit ihren Fähigkeiten, Bedürfnissen und auch ihren Frustrationen wertzuschätzen. Bei *beginne mit dem, was du gerade tust* geht es auch um Sicherheit und Überlebensfähigkeit. Denn obwohl wir nicht dafür argumentieren würden, dass Unternehmen unbedingt überleben müssen, würden wir denen, die von ihnen abhängen, keinen unnötigen Schaden wünschen.

Um verständnisvoll zu führen, müssen wir versuchen, *Selbstzufriedenheit, Draufgängertum* und *Einnischung* zu vermeiden. Vom Respekt für persönliche und unternehmerische Sicherheit kann man sich gut leiten lassen.

2. Das ist die erste Hälfte dessen, was David den »*Lean Decision Filter*« nennt: »Wert sticht Fluss und Fluss sticht das Beseitigen von Verschwendungen.« Er stellt heraus, dass das Beseitigen von Verschwendungen an dritter Stelle der Priorität steht, im Gegensatz zu einigen populären Beschreibungen von Lean. Wert und Fluss stehen sich meistens nicht entgegen.

9.3.8 Vereinbarung

In gemeinsamer **Vereinbarung** gibt man ein spezifisches und langfristiges Versprechen ab, evolutionäre Veränderung zu verfolgen und indirekt die Selbstveränderungsfähigkeit der Organisation zu entwickeln. Diese Investition in die Anpassungsfähigkeit ist der zweite Teil einer Überlebensstrategie und gleichzeitig eine Investition in Menschen.

Wer eine Veränderung anstößt (jegliche Veränderung, nicht nur die großen Initiativen), muss eine besondere Rolle spielen. Man kann dort disziplinierte Führung vorleben, indem man erwartet, dass die Veränderung respektvoll und mit Verständnis und über Vereinbarung durchgeführt wird. Das soll nicht heißen, dass Vereinbarung eine langweilige, konsensgetriebene Monokultur mit sich bringt; evolutionäre Veränderung braucht Kreativität, die es wiederum nötig macht, dass Diversität willkommen geheißen und respektiert – ja sogar gefeiert – wird.

9.4 Die menschliche »*Beginne mit dem, was du gerade tust*«-Methode

Wenn wir Kanban als »die menschliche ›*Beginne mit dem, was du gerade tust*‹-Methode« darstellen, hebt das die beiden Dinge besonders hervor, in denen sich Kanban von anderen Methoden unterscheidet. Es sind sogar stärkere Unterscheidungsmerkmale als die Klebezettel und WIP-Limits und das soll was heißen!

Kanban ist menschlich, nicht nur weil es edle Ideale hat. Auch nicht, weil es zu besseren Bedingungen für die Menschen führt. Sondern weil es so gestaltet wurde, dass es auf konsistent respektvolle Art und Weise eingeführt wird. Edle Ziele rechtfertigen keine unsicheren, organisatorischen Transformationen. Das Ziel, bessere Arbeitsbedingungen für die Mitarbeiter zu schaffen, entschuldigt nicht, zwischenzeitlich rücksichtslos mit Menschen umzugehen. Meines Wissens können nur wenige Methoden die bei der Implementierung zugrunde liegende Integrität von Kanban für sich beanspruchen.

Es liegt auch etwas einzigartig Respektvolles in *beginne mit dem, was du gerade tust*. Es respektiert die Reise all der Mitarbeiter – frühere und aktuelle –, die die Organisation dort hingebracht haben, wo sie jetzt ist. Für all das Gute und Schlechte respektiert es den Kontext, es feiert das Überleben und es macht es einfacher anzunehmen, dass alle Menschen ihr Bestes getan haben – egal wie naiv versöhnlich die Annahme auch sein mag. Ich mag das.

10 Muster und Agenden

Ich habe Sie Stück für Stück durch die Werte geführt, vom konkretesten und bekanntesten bis zum abstraktesten und philosophischsten, und Ihnen so die Kanban-Methode vorgestellt. Ich hoffe, dass ich Ihnen außerdem ein Gefühl davon vermitteln konnte, wie es sich anfühlt, Kanban wirklich zu verwenden.

Ich gebe gerne als einer der Ersten zu, dass es aufwendig ist, sich neun Werte zu merken, und ich habe schnell gemerkt, dass ein wenig Struktur dabei hilft. Sie haben schon ein paar Andeutungen gesehen: die Reflexionen nach Kapitel 3 und 6 zum Beispiel und die Bezeichnung der *Führungsdisziplinen* und ihre Anwendung auf die drei letzten Werte.

Ich merkte, dass ich auf Gruppen von Werten einen anderen Blick bekam, nachdem ich sie benannt hatte. Diese Namen haben sich jedoch als instabiler her-ausgestellt als die Werte und es hat etwas gedauert, sie endgültig festzulegen. Nach einer intensiven Zusammenarbeit, die sich über zwei Konferenzen mit insgesamt sechs Tagen erstreckte und an der David Anderson, Markus Andrezak, Andy Carmichael, Patrick Steyaert und ich selbst beteiligt waren, hatten wir uns irgendwann auf drei Namen für die *Agenden für Veränderung* von Kanban geeinigt.

Bevor wir uns diese anschauen, möchte ich Steve Tendon ehrenvoll erwähnen. Er war einer der ersten, der die neun Werte nahm und mit ihnen etwas anstellte, was ich wirklich nicht erwartet hatte.

10.1 »Noble Muster«

Steves Interesse gilt der Überschneidung von Kanban und zwei weiteren wichtigen Methoden: der *Engpasstheorie* (siehe Kap. 12) und *Scrum* (siehe Kap. 13).

In seinem Buch *Hyper-Productive Knowledge Work Performance, The Tame-Flow Approach and Its Application to Scrum and Kanban* [Tendon & Müller 2015] bringt Steve die Werte mit drei »Mustern« zusammen:

1. Gemeinschaft des Vertrauens
Verständnis, Vereinbarung, Respekt, Transparenz und Kooperation
2. Einigkeit über den Zweck
Arbeitsfluss, Kundenfokus und Balance
3. Führung (allein für sich)

Gemeinschaft des Vertrauens und *Einigkeit über den Zweck* sind Steves »noble Muster«: hoch wünschenswert, selten vollumfänglich umgesetzt sichtbar und mit den produktivsten Umgebungen assoziiert. Nicht ganz so nobel scheint **Führung** zu sein; vielleicht sieht er sie als zu normal an (wenn das nur stimmen würde!) oder sie wird allzu oft mangelhaft ausgeführt.

Steve fand Gruppierungen, die sich von meinen unterscheiden, und seine Namen dafür sind anschaulich. Seine Muster sind vor meiner Arbeit an den Werten entstanden, aber die Gruppierungen sind sinnvoll und der Gesamteffekt ansprechend und gut zu merken. Ich liebe es, wie sie unterschwellig zusätzliche Bedeutungen in den Werten hervorrufen. Sehr erfreulich!

10.2 Agenden für Veränderung

Schafft Kanban auf neutrale Art und Weise nur die Bedingungen für Veränderung oder ist eine gewisse Vorprägung dazu enthalten? Brauchen die Methode, ihre Anwender und die Organisation eine Richtung – vielleicht in der Form eines externen *True North* (Kap. 14) – oder steuern sie sich selbst? Wir haben uns diese Frage als Community schon mehrmals gestellt.

Inzwischen sollten Sie genug von Kanban gesehen haben, dass Sie nicht überrascht sein dürften, wenn die Antwort eher offen ist:

- Ja, Kanban schafft die Bedingungen für Veränderung – genaugenommen ist die Entwicklung der Veränderungsfähigkeit der Organisation eines seiner Hauptziele.
- Ja, egal ob als »True North« oder eine evolutionäre »Fitness-Funktion« ausgedrückt, es ist wichtig, den Erfolg messen zu können.
- Ja, Kanban beinhaltet nicht nur ein paar eingebaute Prägungen, es ist auch hilfreich, diese explizit zu machen.

In der Liste von Jas ist ein »Nein« angedeutet: Nein, Kanban ist nicht neutral. Heute scheint dies offensichtlich: Wie könnte etwas so mit Werten Durchsetzes – wo für den größten Teil »je mehr, desto besser« gilt – jemals als neutral bezeichnet werden?

Nun kommen die *drei Agenden für Veränderung* von Kanban in Spiel. Man kann sich diese als Herangehensweisen zur Einführung vorstellen, die man anhand der Bedürfnisse, Wünsche und Zielsetzung der Organisation wählen und kommunizieren kann:

■ Die Nachhaltigkeits-Agenda

Sie beschreibt, wie Kanban auf einer individuellen oder Teamebene häufig eingeführt wird. Oft wird sie durch die Ablösung von nicht auszuhaltenden Praktiken und Arbeitsaufkommen motiviert, die durch **Transparenz, Balance** und **Kooperation** erreicht werden kann.

■ Die Serviceorientierungs-Agenda

Mit dieser Agenda kommen **Kundenfokus, Arbeitsfluss** und **Führung** hinzu. Diese Werte führen zu einem Vorgehen, bei dem wir beim Einbeziehen der Beteiligten und der Einführung viel mehr nach außen schauen. Typischerweise liegt die Motivation hier in der signifikanten Verbesserung der Arbeit für den Kunden.

■ Die Überlebensfähigkeits-Agenda

Dies ist die offenkundigste kulturelle Agenda der drei Agenden. Mit den Führungsdisziplinen **Verständnis, Vereinbarung** und **Respekt** kommen wichtige, persönliche und organisatorische Verpflichtungen hinzu, Fitness durch Anpassungsfähigkeit und die Fähigkeit zu Veränderungen zu verfolgen.

10.3 Die Nachhaltigkeits-Agenda

Leiden Einzelne und Teams unter der schieren Last nicht beendeter Arbeit, kann die Einführung eines Kanban-Systems für schnelle Linderung sorgen. Verbesserungen in der Vorhersagbarkeit können dabei helfen, Vertrauen zu gewinnen. Das führt wiederum zu mehr Raum, um weitere Möglichkeiten für Veränderungen zu entdecken.

Wo »*was du jetzt tust*« gut definiert ist, ist eine Ausrichtung an den Werten **Transparenz, Balance** und **Kooperation** unkompliziert und selten von Kontroversen begleitet. Wo es nicht gut definiert ist, stellt Kanban eine risikoarme Möglichkeit dar, es besser zu verstehen.

Weil sie die »nachhaltige Geschwindigkeit« des Agilen Manifests wiedergibt, spricht die Nachhaltigkeits-Agenda häufig Teams an, die sich selbst als agil bezeichnen würden (siehe Kap. 13). Zudem können viele Praktiken und Artefakte der Teams im Sinne der Transparenz, Balance und Kooperation interpretiert werden. Letzteres ist sogar eine Schlüsselidee des Agilen Manifests (das selbst eine Aufstellung von Werten ist).

Ein wenig Vorsicht ist jedoch geboten: Obwohl Transparenz und Balance einfach skalieren (bis zur Ebene von Unternehmensinitiativen und -themen), gilt dies nicht für Kooperation. Für sich allein bringt diese Agenda eher indirekt Vorteile für externe Beteiligte und ihr Fokus kann sich schnell nach innen wenden. Ohne dass von außen Verbesserung gefordert wird, wird der Fortschritt sich tendenziell verlangsamen und eventuell sogar zum Stillstand kommen.

10.4 Die Serviceorientierungs-Agenda

Während die Nachhaltigkeits-Agenda am besten als ein praktikbasiertes Vorgehen beschrieben wird, beruht die Serviceorientierungs-Agenda auf dem Engagement anderer. Als Einzelkämpfer kommt man bezüglich **Kundenfokus, Arbeitsfluss und Führung** nicht weit!

Fortschritt kommt dabei durch folgende Faktoren:

- Einen kritischen Blick auf das, was man gerade tut. Man verwendet die Werkzeuge, die in Kapitel 4 und 5 beschrieben werden.
- Die Verbesserung der Koordination von Anfang bis Ende und der damit verbundenen Vermeidung von sich anhäufendem (oder schlimmer verlorenem) Work in Progress an den Übergabepunkten zwischen Diensten
- Das Formen der Arbeitslast stromaufwärts
- Das Erlangen eines besseren Verständnisses der Lücke zwischen den Erwartungen des Kunden und der Leistungsfähigkeit des Systems und dem Einbau von Feedbackschleifen, um die Lücke zu kontrollieren
- Das Verfolgen von Veränderungen, um sicherzugehen, dass kleinen und großen Verbesserungen die angemessene Anerkennung gegeben wird

Die Serviceorientierungs-Agenda ist überall dort die natürliche Wahl, wo es klare externe Treiber für Veränderung gibt. Mit diesen kommt Förderung ins Spiel, was die Implementierung der größeren Feedbackschleifen vereinfacht.

10.4.1 Die Kanban-Linse

2013 hat David Anderson mit einer anderen Neuerung, der *Kanban-Linse*, zusammengefasst, wie die Serviceorientierungs-Agenda eingeführt werden kann. Diese beschreibt, wie man »*was man gerade tut*« als eine Reihe von Diensten sieht, die sich verbessern lassen.

Erkenne an:

1. **Kreative Wissensarbeit ist serviceorientiert.**
Viel von ihrem Wert liegt in den Bedürfnissen anderer Menschen, die sie direkt oder indirekt erfüllt.
2. **Die Erbringung einer Dienstleistung beinhaltet einen Arbeitsablauf.**
Aktivitäten sind von anderen Aktivitäten abhängig; Zustände folgen Zuständen; Arbeit beginnt und endet außerhalb des Systems.
3. **Der Arbeitsablauf beinhaltet eine Folge von Aktivitäten zur Entdeckung von Wissen.** In der Tat ist das eine entscheidende Charakteristik von kreativer Wissensarbeit.

Anschließend:

1. **Bilde den Arbeitsablauf von Aktivitäten zur Entdeckung von Wissen ab.**
Erschließe ein grobes Verständnis der Sequenz von Schlüsselaktivitäten und Status der Arbeitspakete, die zur Fertigstellung nötig sind.
2. **Achte darauf, wie und warum Arbeit eintrifft.**
Verstehe die Motivation, Mechanismen und Muster des Eingangs.
3. **Verfolge die Arbeit**, während sie durch und zwischen Diensten fließt. Visualisiere ihren Fortschritt bis zur Fertigstellung und Validierung.

Verglichen mit der Nachhaltigkeits-Agenda, die hauptsächlich vertikal skaliert, skaliert die Serviceorientierungs-Agenda Kanban-Systeme explizit horizontal über Dienste hinweg, vom Anfang bis zum Ende und zwischen voneinander abhängigen Diensten.

10.5 Die Überlebensfähigkeits-Agenda

Wenn es um die Überlebensfähigkeits-Agenda geht, scheint das Wort »Wahlmöglichkeit« zu schwach – es ist eine strategische, unumgängliche Entscheidung. Es geht darum, evolutionäre Veränderung zu institutionalisieren, um so eine anpassungs- und widerstandsfähigere Organisation entstehen zu lassen, während diese das Ziel verfolgt, fit für ihren Zweck zu werden oder zu bleiben.

Im Gegensatz zu den ersten beiden Agenden, die von unten herauf bzw. aus der Mitte heraus starten, beginnt die Überlebensfähigkeits-Agenda mit der Verpflichtung der Organisationsführung zu **Verständnis, Vereinbarung und Respekt**. Der Rest folgt schon, wenn Förderung, Rahmen und Zielvorgaben sichergestellt sind.

10.6 Die Agenda Ihrer Organisation

Egal wie auch immer wir sie strukturieren, letztendlich geht es bei den Werten Kanbans um Menschen. Menschen mit Transparenz über und Balance in ihrer Arbeit. Menschen, deren Beziehungsbedürfnisse – Möglichkeiten zur Kooperation, Führung usw. – verstanden werden. Menschen, die an einem Workflow mitwirken können, der die Bedürfnisse von Kunden sinnvoll erfüllt. Menschen, die mit dem Wissen im Hinterkopf arbeiten können, dass obwohl Veränderung ständig präsent ist, diese sicher und zweckgerichtet ist.

Falls Ihre Organisation sich selbst auf lange Sicht schwer tun würde, sich mit diesen Zielen zu identifizieren, müssen Sie womöglich akzeptieren, dass Kanban eventuell nicht gut passt. Für die meisten Organisationen ist das jedoch eine Frage von Prioritäten. Welche Werte erfassen am besten, wie sich Ihre Organisation vom jetzigen Standpunkt aus vorwärts bewegt? Wie würden Sie diese insgesamt beschreiben? Dieses Thema wird gegen Ende von Teil III adressiert.

10.7 Wie erging es uns?

In meinen achtzehn Monaten in Budapest führten zwei IT-Teams Kanban ein, das zweite wenige Monate nach dem ersten. Hier ist eine Retrospektive der Ereignisse von 2009 und 2010, verfasst in der Sprache der Werte und Agenden:

- Die Nachhaltigkeits-Agenda fühlte sich intern ganz natürlich an und es steht nachweislich fest, dass andere Teile der Firma zu verstehen lernten, was wir taten. Das erste Team fand sich sehr schnell mit seinem Kanban-System zurecht und verfeinerte es weiter. Die zweite Einführung ging sogar noch schneller, da alle von den Erfahrungen der ersten Implementierung profitierten. Dann wurde klar, dass wir das Projektportfolio unter Kontrolle bekommen mussten; der Grad, bis zu dem die Projektsponsoren eine Senkung des Portfolio-WIP anboten, erscheint mir auch heute noch bemerkenswert.
- Die Serviceorientierungs-Agenda stellt die Richtung und den Antrieb unserer Implementierung gut dar. Kapitel 4 erklärt, wie wir unseren **Kundenfokus** erneuerten. Wie in Kapitel 5 beschrieben, waren wir immer auf den **Arbeitsfluss** bedacht und mit der Zeit veränderte sich unsere Leistungsfähigkeit. Gelegenheiten für **Führung** (Kap. 6) gab es zuhauf. Manche entstanden durch die geschaffenen Bedingungen, andere durch meine ungewöhnliche, sich abwechselnde An- und Abwesenheit.
- Unsere Beziehung mit der Überlebensfähigkeits-Agenda war mehr implizit als explizit und eher intern als extern:
 - Man kann durchaus sagen, dass **Transparenz** dem **Verständnis** vorangestellt. Für beide Teams war der wichtigste Grund, Kaban einzuführen, die Notwendigkeit, besser begreifen zu können, wo wir standen. Ich hatte genügend Erfahrung als Manager, um eine vage Idee davon zu haben, auf was wir uns organisatorisch einließen. Ich kann aber nicht behaupten, dass wir gezielt damit begannen, ein gemeinsames Verständnis zu erlangen oder mehrere externe Perspektiven strukturiert aufzunehmen.
 - Wir praktizierten **Vereinbarung** innerhalb des Teams, doch erst später realisierte ich, dass wir uns zu sehr auf die externe Unterstützung meines Geschäftsführers verlassen hatten und ich nicht hart genug für eine Vereinbarung mit meinen Kollegen im Management gearbeitet hatte. Das verlangsamte einige Entscheidungen und verspätete manche wichtige Lieferungen.
 - An **Respekt** fehlte es nicht – der war vorhanden.

Innerhalb der Grenzen meiner formalen Zuständigkeit folgten wir am Ende erfolgreich allen drei Agenden. Hätte ich die gleiche Chance noch einmal und würde explizit mit einer Serviceorientierungs-Agenda an die Sache herangehen, würde ich ähnliche Ergebnisse (gelieferte Projekte, kürzere Durchlaufzeiten, ein fokussierteres Projektportfolio und nachhaltiger Fortschritt) erwarten, dieses Mal jedoch früher.

Es ist mehr als nur ein wenig ernüchternd zu sehen, wie die Überlebensfähigkeits-Agenda so gut beschreibt, was wir hätten verbessern können. Wir benötigten eine **Vereinbarung** und wir hätten sie womöglich früher erreicht, hätten wir mehr dafür getan, ein gemeinsames **Verständnis** darüber zu erlangen, wo wir standen.

Ein mit der Überlebensfähigkeit verbundenes Nachwort gibt es noch zu meiner Geschichte. Mein Hauptziel bestand darin, die Dauerhaftigkeit der Organisation zu sichern, wenngleich nicht unbedingt in ihrer existierenden Form. Meine Aufgabe war, die IT-Funktionen, ihre wichtigsten Applikationen und ihre Infrastruktur in eine akzeptable Form zu bringen, sodass sie bei einer potenziellen Fusion kein Risiko darstellen würden. Nachdem das geschafft war, verließ ich die Firma im Guten und half bei der Fusion mit einem größeren Konkurrenten. Bald sah es wie eine umgekehrte Übernahme aus: Mark Dickinson, mein damaliger Geschäftsführer, wurde bald danach Geschäftsführer der fusionierten Firma. Arbeit erledigt!

10.8 Fünf Jahre später

Wir spulen zur Gegenwart vor. Wir wissen mittlerweile, dass unsere Erfahrung in hohem Maß wiederholbar ist. Teams, die Kanban einführen, spüren die Vorteile schnell. Dort, wo von Anfang bis Ende auf den Fluss der Arbeit geachtet und die Kundenbedürfnisse antizipiert werden, sind dramatische Verbesserung von Dienstleistungen möglich. Anscheinend sind es nur wenige Firmen, die sich ganz explizit dazu entschließen, Anpassungsfähigkeit und Fitness für ihren Zweck zu verfolgen, es gibt aber auch außerhalb von Davids Buch einige interessante Beispiele.

In der Zwischenzeit hat sich die Kanban-Gemeinde ihre eigene Identität geschaffen und ihr Wissen ist gewachsen und gereift. Es ist allerdings nicht so, dass sie meint, für alles eine Antwort zu haben (oder auch nur anstrebt, das zu sagen). Stattdessen schauen wir nach *Modellen*, manche von ihnen sind anerkannt und genießen einen gewissen Ruhm, andere sind weniger bekannt. Sie sind Gegenstand von Teil II.

Teil II

Modelle

Teil I hat anhand der Werte beschrieben, wie Anwender von Kanban organisatorische Herausforderungen angehen und wie die Kanban-Methode funktioniert. Hier und da habe ich beiläufig auf Einflüsse von außen hingewiesen, aber ich wollte nicht, dass diese Hinweise vom Hauptthema ablenken.

In Teil II nehmen die Werte eine unterstützende Rolle ein. Die Bühne gehört den *Modellen*, die eine besondere Beziehung zur Kanban-Methode und ihrer Community haben und die für sich genommen einiges erreicht haben. Die Werte dienen hier als Grundlage für Vergleich und Integration.

Erinnern wir uns an die sechste Kernpraktik der Kanban-Methode:

KP6: Erzielle Verbesserung kooperativ, entwickle experimentell (unter Verwendung von Modellen und der wissenschaftlichen Methode).

Die Formulierung »*unter Verwendung von Modellen*« bringt etwas erfreulich Flexibles mit sich. Es kann bedeuten,

- die Beispiele anderer zu replizieren,
- einem bestimmten Muster, einer Vorlage oder Routine zu folgen, die dem eigenen Tun und Denken Struktur gibt,
- die Welt anhand einer Menge von definierten Annahmen zu verstehen oder
- gewisse Ergebnisse als Konsequenz dieser Annahmen zu erwarten.

Wenn das herangezogene Modell einen Umfang von der Größe des Systemdenkens und verwandter Disziplinen – Engpasstheorie, Lean oder Agile – besitzt, kann »*unter Verwendung von Modellen*« all diese Dinge gleichzeitig bedeuten. Diese Modelle werden kurz in den Kapiteln 11 bis 14 vorgestellt.

Jedes dieser Modelle repräsentiert einen wichtigen Paradigmenwechsel:

- Systemdenken oder Systemisches Denken ist eine Bewegung weg vom Reduktionismus hin zu einer ganzheitlichen Sicht auf Systeme.
- Die Engpasstheorie ersetzt den traditionellen Fokus auf die Kosten von Verbesserungen mit einem Fokus, der sich mehr an den Systemzielen orientiert.
- Lean will Effizienz durch das Verfolgen des Arbeitsflusses statt durch die maximale Auslastung von Ressourcen erreichen.
- Agile ersetzt geplante, projektgetriebene Auslieferung mit einem evolutionären und kooperativen Vorgehen.

Dieses Buch wird einen Anfänger nicht zum Experten für irgendeines dieser Modelle machen. Dafür gibt es andere Bücher. Um jedoch die Intention hinter der Kanban-Methode vollkommen begreifen zu können, müssen diese Paradigmenwechsel verstanden werden.

Kapitel 15 behandelt einige ökonomische Konzepte, die für das effektive Managen des **Arbeitsflusses** essenziell sind.

Kapitel 16 fasst die Prinzipien, Praktiken und andere Kernkonzepte zusammen, die zur Kanban-Methode gehören. Es behandelt außerdem spezielle Ausführungen der Kanban-Methode wie Personal Kanban, Portfolio-Kanban und Scrumban.

Schließlich behandelt Kapitel 17 ein paar kleinere Modelle. Einige von ihnen helfen dabei, ein tieferes Verständnis der vorherigen Teile des Buches zu erlangen. Andere wiederum liefern nützliche Werkzeuge für Teil III.

11 Systemdenken, Komplexität und die lernende Organisation

*Systemdenken*¹ und die verwandte Wissenschaft der *Komplexität* decken gemeinsam ein großes Spektrum verschiedener Fächer ab. Dieser sehr kurze Ausflug durch diese Gebiete beginnt mit *Hebelpunkten* und der Systemdenken-Methodologie der *Systemdynamik*. Über *Nichtlinearität* und *Unvorhersehbarkeit* machen wir einen kurzen Abstecher zu *Komplexität*, *Kausalität*, dem *Cynefin-Framework* und *komplexen adaptiven Systemen*. Am Ende dieser Tour steht dann die Gesamtintegration in das Konzept der *lernenden Organisation*.

Am Ende dieses Kapitels zeigen wir, wie all diese Konzepte das Design der Kanban-Methode beeinflusst haben und uns dabei helfen können, sie effektiver zu verwenden.

11.1 Systemdenken

Systemdenken beschäftigt sich weitestgehend damit zu verstehen, wie Systeme als Ganzes funktionieren. Im Gegensatz zum *Reduktionismus* oder dem *analytischen* Vorgehen, bei dem versucht wird, die Systemteile voneinander isoliert zu verstehen, ist Systemdenken ganzheitlich. Systemdenken richtet den Blick auf Beziehungen, Interaktionen und Einflüsse zwischen den Komponenten und auf das Verhalten und die Folgen, die aus diesen entstehen.

Systemdenken wurde auf soziale, ökonomische und politische Systeme angewendet, die teilweise wirklich groß sind. Ein bekanntes Beispiel ist die wegweisende Studie *The Limits to Growth (Die Grenzen des Wachstums)* von Donella H. Meadows et al. [Meadows et al. 1972], die die Beziehungen zwischen ökonomischem Wachstum und begrenzten Ressourcen wie Mineralien und nicht erneuerbarer Energie untersucht.

Etwas geläufiger dürfte die Arbeit von John Seddon sein, der Systemdenken auf das Design von Dienstleistungstätigkeiten im öffentlichen Sektor angewendet hat. Gerald M. Weinberg, ein bekannter Autor zu Psychologie und Anthropolo-

1. Ich behandle Systemdenken, Agile und Lean als Nomen, wenn ich sie im Zusammenhang mit Wissensgebieten, Herangehensweisen und Communitys verwende. Ich versuche, sie nicht als Adjektive zu verwenden.

gie in der Softwareentwicklung, verwendet Systemdenken, um zahlreiche Phänomene zu erklären, die wir regelmäßig in unseren Organisationen beobachten.

Andere einflussreiche Systemiker sind:

- W. Edwards Deming, Pionier im Bereich des *Qualitätsmanagements*, der eine Schlüsselrolle beim Wiederaufbau der japanischen Industrie nach dem zweiten Weltkrieg spielte.
- Peter Drucker, der – schon 1959 – den Begriff *Wissensarbeiter* prägte und dessen Anliegen dieses Buch dienen möchte.
- Mehrere Generationen der Familie Toyoda und Taiichi Ohno, der Vater des *Toyota-Produktionssystems (TPS)*, aus dem *Lean* entstanden ist (siehe Kap. 13).

Es scheint, als würden sich Systemiker durch zwei Dinge vom Rest der Menschen unterscheiden:

1. Die Disziplin, das größere System gemeinsam mit dem umgebenden Kontext zu betrachten
2. Die Fähigkeit, die versteckten Einflüsse zwischen Systemkomponenten und zwischen dem System und seinem Umfeld zu identifizieren

Beides sind wertvolle Fähigkeiten, die sich anzueignen lohnen. Je besser wir die Beziehungen zwischen unseren Systemen und ihrem Umfeld verstehen, desto wahrscheinlicher ist es, dass wir effektive Eingriffe identifizieren und implementieren können.

11.1.1 Heelpunkte und Systemdynamiken

Systemdenken hilft uns dabei, *Heelpunkte* zu identifizieren. Das sind Stellen im System, wo ein Eingriff überproportional effektiv ist. Mit etwas Glück liegen diese im Bereich unserer Kontrolle oder (falls nicht) in unserem Einflussbereich. Wenn nicht, ist aber auch noch nicht alles verloren. Systemdenken hilft uns eventuell dabei, *gemeinsame Orientierung* in Form von Eingriffen zu finden, von denen das große Ganze profitiert.

Donella H. Meadows ist verantwortlich für einen Teil des Systemdenkens, den ich besonders mag: die *12 Heelpunkte zur Intervention in ein System*. Interessanterweise sind sie von 12 bis 1 abwärts sortiert, beginnend mit dem am wenigsten effektiven Heelpunkt:

12. Kennzahlen

Konstanten und Parameter wie Subventionen, Steuern und Vorgaben

11. Puffer

Das Verhältnis der Größe von Beständen zu ihrem Durchsatz

10. Puffer-Fluss-Strukturen

Physische Systeme und ihre Knotenpunkte

9. Verspätungen

Das Verhältnis von Zeit zur Häufigkeit von Systemänderungen

8. Ausgleichende Feedbackschleifen

Die Stärke des Feedbacks, relativ zu den Auswirkungen, die es korrigieren soll.

7. Verstärkende Feedbackschleifen

Die Stärke des Zugewinns durch antreibende Schleifen

6. Struktur der Informationsflüsse

Wer hat Zugang zu Informationen und wer nicht.

5. Regeln

Leistungsanreize, Strafen und Beschränkungen

4. Selbstorganisation

Die Macht, die Systemstruktur zu erweitern, zu ändern oder weiterzuentwickeln

3. Ziele

Der Zweck des Systems

2. Paradigmen

Die Geisteshaltung, aus dem das System – seine Ziele, Struktur, Regeln, Verspätungen und Parameter – erwächst

1. Überschreiten von Paradigmen

Die Terminologien der Interventionen 11, 10, 8, 7 und 6 bedürfen eventuell weiterer Erklärung. Meadows war Anwenderin der *Systemdynamik*, einer Methodik, die aus einem Bereich der industriellen Konzeption entstanden ist. *Puffer* repräsentierten in erster Linie also eine Anzahl von physikalischen Materialien und *Fluss* bezog sich auf den Transfer dieser Materialien von einer Stelle oder Aktivität zur nächsten. Je breiter die Systemdynamik eingesetzt wurde, desto mehr wurde der Begriff Puffer für andere greifbare Dinge wie Tierpopulationen oder Geld und nicht greifbare Dinge wie Qualität oder Vertrauen verwendet. Je abstrakter der Puffer, desto subtiler der Fluss.

11.2 Komplexität

Das Verhalten von Systemen, deren Signale und Feedbackschleifen verzögert zu erkennen sind, ist oft schwer vorherzusagen. Das gilt selbst dann, wenn die Systeme vollkommen *deterministisch* und immun gegen Zufälle sind. Diese Systeme können nicht durch die Summe ihrer Teile beschrieben werden – sie sind *nicht linear*. Einige überraschend einfache Systeme sind so sensibel bezüglich Aus-

gangsbedingungen, Messfehlern oder den kleinsten Perturbationen (vgl. den *Schmetterlingseffekt*), dass ihr Verhalten chaotisch erscheint.

Auch Geschäftssysteme können darunter leiden. Ein klassisches Beispiel ist der *Peitscheneffekt*. Er ist manchmal auch bekannt als der *Forrester-Effekt*, benannt nach Jay Forrester, einem der Gründer der Systemdynamiken. Der Effekt beschreibt größere Schwankungen in Lieferketten. Lieferanten haben keine genauen Informationen und nehmen eventuell an, dass eine kurzfristige Änderung in der Nachfrage langfristige Bedeutung hat. Das bringt sie dazu, zeitgleich sowohl die unmittelbare Nachfrage zu bedienen als auch ihren Pufferbestand zu erhöhen. Das führt zu einem verstärkten Signal die Lieferkette aufwärts. Der nächste Lieferant in der Kette verstärkt das Signal wiederum. Bei einer ausreichend langen Lieferkette können Lagerbestände massiv schwanken.²

Ein anderer Weg, den Effekt der Verstärkung noch weiter zu vergrößern, führt über Feedbackschleifen. Nicht erkannte oder schlecht konzipierte Feedbackschleifen können verheerend für ein System sein. Weinberg beschreibt in [Weinberg 1992] den in Abbildung 11–1 gezeigten, weit verbreiteten Teufelskreis.



Abb. 11–1 Teuflisch oder engelsgleich? Eine Feedbackschleife in der Qualität.

In einer pessimistischen Deutung dieses Bilds führt der Verständnisverlust für den Wert von Qualität zu einer Reduzierung der Motivation, Qualität zu erreichen. Das wiederum reduziert das Verständnis davon, wie man Qualität erreicht, und damit die Motivation, den Wert zu bemessen. So dreht sich die Spirale immer weiter nach unten.

Glücklicherweise gibt es auch eine positive Deutung des Bilds. Mit den richtigen Anfangskonditionen oder einer starken, der sich nach unten drehenden Spirale entgegengesetzten Intervention kann sich diese *selbstverstärkende Schleife* positiv verhalten. Das Verständnis fördert die Motivation, das Erreichen verstärkt das Verständnis usw.

2. Das *Bierspiel* ist eine Simulation, die diesen Effekt demonstriert.

Diese Systeme sind leicht zu verstehen, aber die Konsequenzen für einen Manager sind tiefgreifend. Das Verhalten des Systems im Ganzen kann nicht an einem einzelnen seiner vielen Teile festgemacht werden. Es ist schwer, Ursache von Wirkung zu trennen. Das System zu verstehen, wird dem Manager mit Sicherheit helfen, sinnvolle Eingriffe vorzunehmen. Aber er wäre wirklich mutig, wenn er sich darauf festlegt, wie dieser Eingriff ausgehen wird. Manager bestimmen das Verhalten eines Systems nicht; sie interagieren lediglich damit.

11.2.1 Kausalität und das Cynefin-Framework

Auf diesen Ideen aufbauend beschreibt Dave Snowden im *Cynefin framework* [Snowden & Boone 2007] fünf Domänen, die Kausalität innerhalb von Systemen charakterisieren:

- **Offensichtlich** (früher Einfach)

Beobachtetes lässt sich einfach kategorisieren, wir können dementsprechend darauf reagieren, eventuell mit einer *Best Practice*, deren Ergebnis schon vorab bekannt ist.

- **Kompliziert**

Obwohl für einen ungeübten Betrachter nicht offensichtlich, ist es für einen Experten möglich, die Situation gut genug zu verstehen, um eine passende *Good Practice* zu wählen, deren Ergebnis vorhersagbar ist.

- **Komplex**

In der Domäne des *entstehenden Verhaltens* können wir nur rückblickend die Auswirkungen unserer Intervention komplett verstehen.

- **Chaos**

Kausale Zusammenhänge verschwinden völlig, selbst rückblickend.

- **Unordnung**

Der Zustand der Unwissenheit darüber, welche der anderen vier Domänen zutrifft.

Diese Domänen sind voneinander abgegrenzt. Insbesondere die *komplexe Domäne* ist nicht einfach eine »kompliziertere« Version der *komplizierten Domäne*. Die Herausforderung der komplexen Domäne – die Domäne, in der sich die interessantesten Unternehmen befinden – liegt in ihrer Nichtlinearität. Wir können keine noch besseren Experten finden, die noch fähiger darin sind, präzise Ergebnisse vorherzusagen. Strategien, die in der simplen und komplizierten Version zu funktionieren schienen, funktionieren entweder gar nicht oder funktionieren eine Zeit lang und scheitern dann.

Cynefin sorgt in seiner Beschreibung der komplexen Domäne für ein wenig Beruhigung: Wir können darauf hoffen, aus Dingen rückblickend einen Sinn zu erkennen. In anderen Worten können wir schauen, wie Dinge jetzt sind (oder

waren) und sie als das Produkt von Entscheidungen oder Umständen, intern oder extern, verstehen.

Wenn wir das andersherum betrachten, werden wir bald damit anfangen, die Ergebnisse unserer derzeitigen Interventionen zu sehen. Wir können eine Validierung unserer Interventionen planen und so bestimmen, ob sie vorteilhaft oder schädlich waren oder nur begrenzte Wirkung hatten. Mit ein wenig Vorausdenken können wir eventuell einige dieser möglichen Auswirkungen vorhersagen und uns darauf vorbereiten, die vorteilhaften aktiv zu *verstärken* und die schädlichen abzuschwächen. Diese *Safe-to-fail-Experimente* verbessern den *PDCA-Zyklus* (Kap. 3) in den Fällen, in denen Interventionen leicht Schaden anrichten können oder in denen erhoffte, emergente Vorteile früh erkannt und gefördert werden müssen.

Cynefin erlaubt uns außerdem, manche Interventionen so zu sehen, dass wir mit ihnen Aspekte des Systems von einer Domäne in eine andere bewegen können – sie also aus dem Chaos herausziehen (oder vorsichtig eintauchen), etwas Kompliziertes vereinfachen usw. Die einzelnen Praktiken anderer Frameworks (agile Methoden zum Beispiel) können in diesem Licht brauchbar erscheinen.

11.2.2 Komplexe adaptive Systeme

Für komplexe Systeme zu entwerfen, bedeutet nicht einfach, bessere Experimente zu entwerfen. Wenn Dinge stets unvorhersehbar sind, müssen wir *Resilienz* und *Anpassungsfähigkeit* aufbauen. Die Komplexitätswissenschaft bietet dafür ein Modell: *komplexe adaptive Systeme (CAS)* – mehrere Ebenen lose gekoppelter, selbstorganisierter Systeme.³

Die Anpassungsfähigkeit von CAS ist lebenswichtig für Systeme, die in konkurrierenden Umgebungen leben, in denen nur der *Fitteste* Erfolg haben wird. Adaptive Systeme haben weit größere Chance, Strukturen zu finden, mit denen sie ihren Vorsprung behalten. Auch ohne Bedrohung können adaptive Systeme mit guten Feedbackschleifen bessere *Fitness* erlangen. Das sind Strategien für *evolutionäre Veränderung*.

3. Beachten Sie, dass Meadows Selbstorganisation als eine der effektivsten Interventionspunkte identifiziert hat – es ist auf Platz vier auf ihrer Liste.

11.3 Wissen, lernen und die lernende Organisation

Was wäre, wenn nicht die momentane Struktur der Organisation das eigentliche Problem ist, sondern die Denkweise, aus der heraus sie entstanden ist? Natürlich lässt sich beides nicht einfach voneinander trennen. Weinberg beobachtete:

*Wenn sich die Denkweise ändert, ändert sich die Organisation,
und umgekehrt.*

[Weinberg 1992]

Daraus folgt, dass Anpassungsfähigkeit (die Leichtigkeit, mit der sich ein Prozess oder eine Praktik ändert) mit der Lust am Lernen (die das Denken ändert) Hand in Hand geht. Das sind gute Nachrichten, denn egal, auf welcher Seite der Gleichung gearbeitet wird, es wird immer einen Einfluss auf die andere haben. Noch besser ist, Sie arbeiten gleichzeitig an beiden.

Wir beenden diesen Abschnitt mit einer ganz bestimmten Vision der lernenden Organisation, nämlich der von Peter Senge. Zuvor benötigen wir jedoch weitere Teile des Puzzles. Wir betrachten das von Managern benötigte Wissen, so wie es W. Edwards Deming beschrieben hat, und danach das von Chris Argyris beschriebene einfache, aber mächtige Modell organisatorischen Lernens.

11.3.1 Demings System vom umfassenden Wissen

Am Anfang dieses Kapitels habe ich W. Edwards Deming (1900-1993) als einen maßgebenden Systemdenker vorgestellt. Ehrlich gesagt gibt es kaum einen Systemdenker, der einen größeren Einfluss auf die Kanban-Community hatte, obwohl Eli Goldratt (siehe Kap. 12) schon nahe herankommt.

Deming begann mit dem Systemdenken als Fundament und fügte ihm drei weitere Elemente hinzu, um daraus sein *System des Umfassenden Wissens (SdUW)* für Führungskräfte zu formen:

1. **Verständnis eines Systems**
Geschäftssysteme gemeinsam im Kontext ihrer Lieferanten und Kunden verstehen
2. **Wissen über Abweichung**
Ursachen und Auswirkungen von Abweichungen in der Qualität und die korrekte Nutzung von statistischen Methoden verstehen
3. **Theorie des Wissens**
Das Management zum Lernen herausfordern, durch das systematische Entwickeln, Testen und Anwenden von Theorien
4. **Wissen über Psychologie**
Menschen und ihre Motivation verstehen; außerdem die Psychologie von Veränderung verstehen

Jeder dieser Punkte ist für sich genommen hilfreich, aber das Ganze ist sehr viel umfassender und hebt signifikante Schwächen des üblichen Managementdenkens hervor. Zum Beispiel:

- Es ist Verschwendug (oder schlimmer), wenn sich ein Manager in ein System *eimmischt* und jede beliebige Abweichung so angeht, als würde sie eine *zurechenbare Ursache* besitzen. Genauso sollte man Abweichungen, die möglicherweise eine spezifische Ursache im System haben, nicht einfach ausblenden, als wären sie ein Resultat natürlicher Streuung⁴.
- Viele Unternehmen versuchen, ihre Mitarbeiter durch monetäre und andere Belohnungen zu motivieren, die genauso stark durch Schwankungen, Systemverhalten und dem derzeitigen Geschäftsklima wie durch den persönlichen Beitrag bestimmt werden. Es ist seit Langem bekannt, dass diese Strategien hochgradig schädlich sein können. Diese Praktiken bleiben jedoch bestehen, weil das System die befördert, die diese Praktiken unterstützen.

Deming erweiterte sein System vom Umfassenden Wissen um die *14 Punkte des Managements*, zu finden in Anhang A. Es sieht mehr als zwanzig Jahre später so aus, als gäbe es noch viel zu lernen!

11.3.2 Argyris und das Doppelschleifen-Lernen

Wie lernen Organisationen? Eine Reihe von signifikanten Beiträgen zu diesem Thema werden Chris Argyris, einem Verhaltenspsychologen und Wirtschaftswissenschaftler, zugeschrieben. Drei davon sind in der Kanban-Community sehr bekannt⁵:

- Die *Leiter der Schlussfolgerungen*. Sie hilft zu verstehen, warum Kollegen zu sehr unterschiedlichen Schlussfolgerungen kommen und wie man damit umgeht.
- *Vertretene Theorie* gegenüber *handlungsleitender Theorie*. Dies stellt einen Weg dar, Möglichkeiten zum Lernen zu erkennen. Man achtet auf die Unterschiede zwischen dem, was Menschen und Organisationen behaupten, und dem, was sie wirklich tun.
- *Doppelschleifen-Lernen*. Dieses einfache Modell beschreibt sehr elegant einen Prozess des tiefen Lernens.

Die ersten beiden sind nicht Teil dieses Buches. Doppelschleifen-Lernen ist allerdings sehr relevant. In Abbildung 11–2 ist es dargestellt.

4. Russell Ackoff, ein weiterer großartiger Systemdenker, hätte den letzteren der beiden Fehler vermutlich als ein »Auslassungsfehler« kategorisiert. Er bezeichnete diese als die heimtückischsten Fehler aller Fehler des Managements.

5. Vielen Dank an Benjamin Mitchell, dass er so für Argyris Arbeit eingetreten ist.

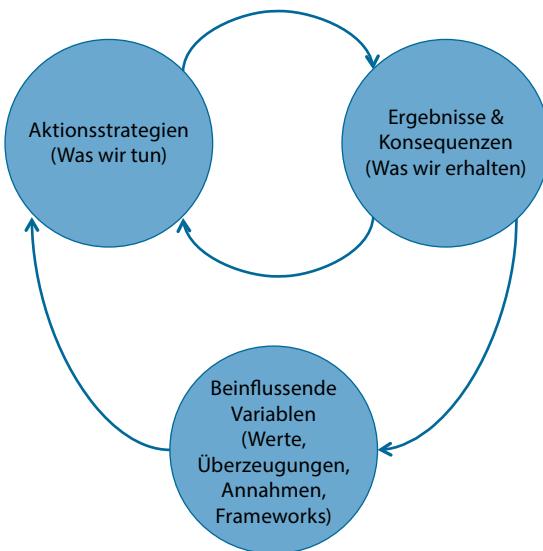


Abb. 11-2 Doppelschleifen-Lernen

Die innere Schleife in Abbildung 11-2 beschreibt das Einschleifen-Lernen. Die Mehrheit des Lernens besteht aus einer Schleife: Wir passen unsere *Aktionsstrategien* – unsere nächsten Ziele, Pläne und Schritte – an das an, was wir wahrnehmen. Einschleifen-Lernen ist etwas, was wir einfach machen, sogar unbewusst. Einschleifen-Lernen ist sehr effizient, wenn es das Ziel ist, ein System in einer vorhersehbaren Umgebung unter Kontrolle zu halten.

Wir wissen, dass Unternehmen sich nicht in vorhersagbaren Umgebungen befinden. Selbst ihre internen Systeme können auf trügerische Art und Weise nicht vorhersagbar sein. Deshalb können wir nicht annehmen, dass Organisationen ewig konkurrenzfähig bleiben, wenn sich die Mitarbeiter nicht von Zeit zu Zeit mit einem gründlicheren Lernen auseinandersetzen, dem sogenannten Doppelschleifen-Lernen.

Beim Doppelschleifen-Lernen werden nicht einfach Anpassungen vorgenommen, wenn die Ergebnisse nicht den Erwartungen entsprechen, sondern das Lernen adressiert tiefere Probleme. Annahmen werden auf den Prüfstand gestellt. Gedankenmodelle werden verworfen und durch neue ersetzt. Ob implizit oder explizit, die Veränderung im Denken spiegelt sich in neuen Arbeitsweisen wider.

Dieses Verlernen und Wiedererlernen ist harte Arbeit! Realistisch gesehen ist niemand in der Lage, auf Dauer in Doppelschleifen zu lernen. Genauso wenig können Organisationen aber erwarten, lebensfähig zu bleiben, wenn es lange ausbleibt. Irgendwie müssen die Bedingungen für häufiges Doppelschleifen-Lernen geschaffen werden.

11.3.3 Die lernende Organisation

Wie Demings *System vom Umfassenden Wissen* startet Senges Modell mit Systemdenken. Verglichen mit Deming ist Senge jedoch weniger auf den Manager fokussiert und dafür mehr auf Einzelne und Teams. Seine fünf Charakteristiken einer lernenden Organisation lauten:

1. Systemdenken

Die Organisation wird als Ganzes verstanden und gemanagt.

2. Individuelle Selbstverbesserung

Einzelne verpflichten sich dem Prozess des Lernens.

3. Mentale Modelle

Annahmen und Theorien werden explizit gemacht, für Überprüfungen geöffnet und in einer vertrauensvollen Umgebung infrage gestellt (die Bedingungen für Doppelschleifen-Lernen).

4. Gemeinsame Vision

Die Herausforderung, die die Organisation vereint und ihr Energie gibt – ein höherer Zweck.

5. Lernen im Team

Der soziale Mechanismus, der individuelles Lernen beschleunigt; die Strukturen, die organisationales Lernen durch das Gelernte Einzelner schafft.

Der letzte Punkt ist eine Anspielung auf den Titel von Senges Bestseller: *The Fifth Discipline (Die fünfte Disziplin)* [Senge 2006]. Senge beschreibt die lernende Organisation als:

... eine Gruppe von Menschen, die gemeinsam daran arbeiten, ihre Kapazitäten zu verbessern, um Ergebnisse zu erzielen, die sie als wirklich wichtig erachten.

[Fulmer & Keys 1998]

In der Beschreibung ist für die meisten Organisationen nicht genügend Geschäftszweck enthalten, um eine komplette, vereinende Vision zu formen. Für jemanden wie mich, der im Bereich der Organisationsentwicklung arbeitet, ist sie überaus überzeugend. Wenn ich sehe, wie sich die Organisationen, mit denen ich arbeite, auf dieses Ziel hinbewegen, sehe ich Ergebnisse, die *ich* als wirklich wichtig erachte.

11.4 Systemdenken und Kanban

11.4.1 Das Design der Methode

Mehrere Elemente der Kanban-Methode haben ihre Wurzeln in den kurz angerissenen Modellen dieses Kapitels. Im Speziellen:

- Die **Transparenz**-Praktiken (Kap. 1) schaffen neue Hebelepunkte und machen das System damit offener für Herausforderungen und Verbesserungen. Außerdem können sie Selbstorganisation fördern, eine Strategie für Resilienz bei Ungewissheit.
- Kernpraktik 5, *Erziele Verbesserung kooperativ und entwickle experimentell*, verbindet **Kooperation** (Kap. 3), Wissen, Experimentieren, geteiltes Lernen und evolutionäre Veränderung.
- Kanban teilt sich mit Lean (Kap. 14) einen systemischen Ansatz zum Thema **Führung** (Kap. 6). Auf längere Sicht erhalten Organisationen die Führung, die sie verdienen: die Art Führung, die ihr System rekrutiert, ermutigt und fördert. Daraus folgt, dass die beständigsten Organisationen solche sind, die ein Augenmerk darauf gehabt haben.
- Auch wenn es das implizit tut, zeigt das erste Grundprinzip, *beginne mit dem, was du gerade tust*, in Richtung Systemdenken und evolutionärer Veränderung. Ich habe ihm ein wenig mehr Gewicht verliehen, indem ich den Wert **Verständnis** (Kap. 7) aus diesem Prinzip abstrahiert habe. Das Ziel dahinter ist, dass Organisationen Verständnis wertschätzen und die Disziplin haben, es als Wegbereiter der Veränderung zu etablieren.
- Evolutionäre Veränderung ist explizit im zweiten Grundprinzip enthalten: *Vereinbare, dass evolutionäre Veränderung verfolgt wird* (siehe Kap. 8, **Vereinbarung**).

Weiter oben in diesem Kapitel habe ich deutlich gemacht, dass die Kanban-Methode Raum für Interpretationen lässt. Das ist eine Stärke. Sie ist ausreichend klar artikuliert, sodass sich eine Community darum bilden kann. Dennoch wird sie mit genügend Diversität angewendet, wodurch diese Community weiter lernt, spezifische Praktiken entwickelt und Erfahrungen austauscht. Es ist befriedigend zu sehen, wie die Kanban-Community selbst in gewisser Weise alle fünf Charakteristiken der lernenden Organisation von Senge aufweist.

11.4.2 Anwendung

Weder Kanban noch Systemdenken sollten einmalige Übungen oder Praktiken für Experten »im Elfenbeinturm« sein, die vom Ort, wo die »echte Arbeit« erledigt wird, ferngehalten werden. John Gall empfiehlt (auch als Galls Gesetz bekannt):

Ein komplexes System, das funktioniert, ist ausnahmslos aus einem simplen System entstanden, das funktioniert hat. Die umgekehrte Behauptung scheint auch wahr: Ein aus dem Nichts konzipiertes komplexes System funktioniert nie und kann auch nicht dazu gebracht werden, zu funktionieren. Man muss mit einem funktionierenden simplen System von vorn beginnen.

[Gall 2003]

Machen Sie sich keine Sorgen, dass jeder Schritt gleich beim ersten Mal richtig sein muss. Hält man das System mit Safe-to-fail-Experimenten in Bewegung, gibt es eine Grenze, wie viel Schaden jeder Schritt anrichten kann, und lokale Schwierigkeiten werden bald ausgebügelt. Unter diesen Umständen ist *Suboptimierung* – die lokale Verbesserung, die Dinge global schlechter macht – ein geringeres Problem als der Dynamikverlust durch die Angst vor ihr. Man sagt ja auch: »Das Perfekte ist der Feind des Guten.«

Sehen Sie mit diesen evolutionären Gedanken im Kopf das Folgende als eine Übung an:

- Denken Sie über diese *Bestände* nach und identifizieren Sie andere mit ähnlicher Signifikanz aus Ihrem derzeitigen Kontext:
 - Unbefriedigte Bedürfnisse
 - Risikobereitschaft
 - Kulanz
 - Qualität
 - Vertrauen

Was beeinflusst diese Punkte? Wie passen sie mit den neun Werten aus Teil I zusammen?

- Sehen Sie sich noch einmal aufmerksam die Heelpunkte von Meadows an. Welche Parallelen zur Kanban-Methode können Sie finden? Welche ihrer Interventionen könnten bei Ihnen Anwendung finden? Wie könnte Kanban helfen?
- Versuchen Sie so viele Aspekte Ihrer derzeitigen Situation wie möglich auf die Domänen des Cynefin-Frameworks abzubilden. Wie könnten Sie die Praktiken von Kanban verwenden, um Aspekte von einer in die andere Domäne zu bewegen?

12 Engpasstheorie (TOC)

Die Engpasstheorie (Theory of Constraints, TOC) ist das konzeptuelle Framework hinter Eliyahu (»Eli«) Goldratts wichtigstem Bestseller *The Goal (Das Ziel)* [Goldratt & Cox 2004]. Es besteht aus den folgenden Teilen:

- Process of Ongoing Improvement (POOGI)
- Drum-Buffer-Rope (DBR)
- Logical Thinking Process
- Critical-Chain-Projektmanagement (CCPM)
- Throughput Accounting

Diese Bestandteile können unabhängig voneinander verwendet werden und jeder Teil hat seine eigene Gruppe von Experten. Doch wie wir sehen werden, gibt es auch einen zufriedenstellenden Zusammenhang im Ganzen.

In Anbetracht der Zugänglichkeit und Popularität von Goldratts Literatur stellt sich die Frage, warum seine Methoden nicht weiter verbreitet sind? Dieser Frage gehen wir im letzten Teil dieses Kapitels nach, in dem wir die vergangenen und bestehenden Beziehungen zwischen TOC und Kanban betrachten.

TOC stellt eine große und wichtige Ansammlung von Wissen dar, das nicht einfach zusammengefasst werden kann. Haben Sie dies vor Augen, wenn Sie diesen Überblick lesen. TOC-Stoff kann ziemlich bildhaft und lustig sein, oft mit Figuren wie »Herbie« (ein Charakter aus *Das Ziel*, der eine personifizierte Beschränkung der Kapazität darstellt), »Murphy« (abgeleitet von Murphys Gesetz, wird als Spitzname für *Abweichung* verwendet) und dergleichen. Am anderen Ende des Spektrums gibt es eine ziemlich einschüchternde Anzahl TOC-eigener Akronyme und technischer Begriffe. Manchmal werden beide Stile in derselben Quelle verwendet. Keines der beiden Extreme ist mein Stil und ich versuche die Schlüsselkonzepte der Engpasstheorie und ihre wichtigsten Begriffe in einer Sprache zu erklären, die, wie ich hoffe, klar verständlich ist.

12.1 Die fünf Fokussierungsschritte und der Process of Ongoing Improvement (POOGI)

Der *Verbesserungskreislauf* der TOC, bekannt als *Process of Ongoing Improvement (POOGI)*, ist durch die *fünf Fokussierungsschritte* festgelegt. Sie umfassen vier Schritte in einem Kreislauf (der fünfte Schritt führt uns wieder an den Anfang):

1. **Identifiziere** den Engpass des Systems (den Faktor, der die größte Schuld daran trägt, dass das System keine höhere Leistung erzielen kann).
2. **Entscheide**, wie man den Engpass voll auslastet (stelle sicher, dass der Engpass immer voll ausgelastet arbeitet und ihm nie etwas fehlt, was er benötigt).
3. **Ordne** alles andere den vorherigen Entscheidungen **unter** (richte den Rest des Systems an den Bedürfnissen des Engpasses aus).
4. **Behebe** den Engpass des Systems (finde Wege, die Kapazität am Engpass zu erhöhen).
5. **Wiederhole.** Falls der Engpass in den vorherigen Schritten aufgelöst wurde, gehe zurück zu Schritt 1. **Warnung:** Erlaube nicht, dass Trägheit einen Systemengpass verursacht!

Es liegen ein paar versteckte Annahmen in der Formulierung dieser Punkte, die explizit gemacht werden sollten:

- Es gibt nur einen *Engpass*, der von Interesse ist; die Arbeit an anderen Engpässen wäre Verschwendungen. (Manche Formulierungen erlauben eine kleine Menge bedeutender Engpässe, doch die Sprache und die damit verbundene Nachricht verliert dadurch an Wirkung.)
- Der Engpass nimmt im Prozess üblicherweise die Form eines *Flaschenhals* an und stellt den Schritt dar, der die Geschwindigkeit der anderen Schritte im Prozess limitiert. Dieser wird auch als *kapazitätsbegrenzte Ressource (capacity-constrained resource (CCR))* bezeichnet. Manchmal begegnen Sie auch einer *nicht sofort verfügbaren Ressource (non-instant availability (NIA))*. Bei ihr ist insgesamt genügend Kapazität vorhanden, jedoch nicht unbedingt zu dem Zeitpunkt, an dem man sie braucht (Linienverkehr ist ein gutes Beispiel dafür).
- Dem (ökonomischen) Ziel des Systems ist meistens durch die Verbesserung des *Durchsatzes* am besten gedient; *Inventar*, *Durchlaufzeiten* oder *Abweichungen* werden meist zweitrangig betrachtet (sie bleiben aber wichtig).
- Irgendwann, nachdem alle internen Engpässe beseitigt worden sind, wird der Engpass außerhalb liegen. Wenn das passiert, wird das System typischerweise von einem Flaschenhals auf der Seite der Zulieferung oder durch ein Defizit in der Nachfrage relativ zur Kapazität eingeschränkt.

Diese Annahmen können durch einen Schritt 0 infrage gestellt werden:

0. Definiere das Ziel oder den Zweck des Systems.

Für den Zweck dieses Buches verwenden wir *POOGI-0* als POOGI, wobei Schritt 0 explizit inbegriffen ist.

Mir ist aufgefallen, dass Softwareprojekte meist mit der Annahme gemanagt werden, dass der Engpass das Entwicklungsteam sei (und es immer sein wird). Es wird als wichtig erachtet, vor allem die Entwickler immer voll auszulasten. Und zwar selbst dann, wenn es keine qualitativ hochwertigen Anforderungen gibt oder die Lieferfähigkeit des Teams nicht zur Fähigkeit des Kunden passt, die geplanten Vorteile der neu erhaltenen Funktionen zu nutzen. Nur wenn wir diese Annahmen hinterfragen, können wir sicherstellen, dass die Aufmerksamkeit des Managements auf den richtigen Dingen liegt.

12.2 Drum-Buffer-Rope (DBR)

Drum-Buffer-Rope ist ein Produktionsplanungssystem aus der Engpasstheorie. Als solches spielt DBR die gleiche Rolle in der Engpasstheorie wie Kanban-Systeme in der Kanban-Methode.

Nehmen wir an, dass wir mit Schritt 1 der fünf Fokussierungsschritte eine Aktivität *identifiziert* haben, die an einem späteren Punkt des Gesamtprozesses liegt und eine Beschränkung, wie vielleicht einen Integrations- oder Inspektionspunkt, darstellt. Für den Fall, dass kein solcher Engpass gefunden wird, behandeln TOC-Anwender die letzte Versandaktivität als solch einen Engpass und wenden DBR dort an.

Drum (denken Sie an »Trommelschlag«) ist ein im Voraus aufgestellter Arbeitsplan für den Engpass. Schritt 2 der fünf Fokussierungsschritte besagt, dass der Engpass ausgeschöpft werden soll. Wir planen also so, dass der Engpass immer mit hoher, aber nachhaltiger *Auslastung* beschäftigt ist.

Buffer ist die Zeit, die es jeder Art von Arbeitspaket, Komponente oder Unterbaugruppe erlaubt, durch den vorgesetzten Prozess zum Engpass zu gelangen. Während dieser Zeit wenden wir *Buffer-Management* an. Das heißt:

- Diese Zwischenprodukte werden auf ihrem Weg durch den vorgesetzten Prozess zum Engpass beobachtet.
- Jedem Teil wird ein roter, gelber oder grüner Status zugewiesen, gemessen an der verbrauchten Zeit seines Zeitpuffers.
- Die zeitkritischsten Teile, die mit einem roten Status, werden in Richtung des Engpasses geführt. Praktiken wie *tägliche Standup-Meetings* und *Swarming*, die wir aus Kanban oder Scrum kennen, gehören ebenfalls zu TOC.

Buffer-Management hilft dabei, den vorgelagerten Prozess effektiv den Bedürfnissen des Engpasses *unterzuordnen* (Schritt 3 der fünf Fokussierungsschritte). Erfolgreiches Buffer-Management zeichnet sich dadurch aus, dass am Engpass zu jeder Zeit ein Vorrat an Arbeit zur Verarbeitung bereitsteht¹.

Rope verbindet die Eingabe von Buffer mit seiner Ausgabe. Arbeit wird nach einem Zeitplan in das System gegeben, entsprechend dem Arbeitsplan der begrenzenden Aktivität (Drum). Rope macht aus DBR eine interessante Art von *Pull-System*; Arbeit wird mit so viel Vorlauf in das System gezogen, dass es vom Engpass zur geplanten Zeit bearbeitet werden kann.

Als Resultat hat DBR zwei Zeitpläne: einen stromabwärts am Engpass und einen an einem Punkt irgendwo stromaufwärts. Statt detailliert zu planen, was genau zwischen diesen beiden Punkten passiert, wird die Arbeit Tag für Tag durch den Prozess gemanagt und die einzelnen Teile entsprechend ihrer Zeitkritikalität priorisiert. Mit einer nachhaltigen Geschwindigkeit, die durch den Engpass vorgegeben wird, und einem vernünftigen Zeitpuffer sollte ein stabiler und vorhersagbarer Fluss entstehen.

12.3 Thinking Processes – Denkprozesse

Wie Kanban erkennt TOC, dass Verbesserung Veränderung bedeutet und dass die Menschen Veränderungen als schwierig empfinden. TOC beinhaltet eine Reihe von Denkwerkzeugen, die dazu geschaffen wurden, um den *Veränderungswiderstand* zu adressieren.

TOC beschreibt eine Reihe sogenannter *Widerstandsschichten* (*layers of resistance*). Die Anzahl und die genaue Bezeichnung haben sich erst über die Zeit entwickelt; Kelvyn Youngman hat diese Veränderungen auf seiner exzellenten Website, *A Guide to Implementing the Theory of Constraints (TOC)* (www.dbrmfg.co.nz), verfolgt. Das aktuelle Neun-Schichten-Modell, das in Tabelle 12–1 zu sehen ist, wird Efrat Goldratt zugeschrieben.

1. Beachten Sie, dass sich die Bedeutung von »Buffer« [zu Deutsch »Puffer«; Anm. der Übersetzer] in Kanban und TOC ein klein wenig unterscheidet. Eventuell beleidigen Sie einen TOC-Puristen, wenn Sie behaupten, DBR anzuwenden, indem Sie einfach einen Puffer mit Arbeit vor dem Engpass verwalten. Bei Kanban ist der Puffer eine Warteschlange, die für einen stetigen Vorrat an Arbeit zu sorgen hat. Bei TOC ist der Puffer ein definierter Zeitraum.

Widerstand	Werkzeug
Es gibt kein Problem	
Keine Einigung über das Ausmaß des Problems	Gegenwartsbaum
Keine Einigung über die Art des Problems	Konflikt-Wolke
Keine Einigung über die Richtung der Lösung	Dilemma-Wolke
Keine Einigung über die Vollständigkeit der Lösung	Zukunftsbaum
Weitere negative Ergebnisse sind sichtbar	Zukunftsbaum mit Vorbehalten
Hindernisse sind sichtbar	Voraussetzungsbaum
Es ist nicht klar, wie genau die Lösung implementiert werden kann	Umsetzungsbau
Nicht definiert/Angst	

Tab. 12-1 Efrat Goldratts Modell der neun Schichten des Widerstands gegen Veränderung

Werden TOC-Anwender mit einem bestimmten Widerstand aus der linken Spalte von Tabelle 12-1 konfrontiert, wenden sie das korrespondierende Werkzeug in der rechten Spalte an. Mit diesen Werkzeugen beschreiben und verändern sie das Problemgebiet grafisch und suchen nach Wegen, das Problem zu lösen. Diese und einige weitere Werkzeuge komplettieren die *Denkprozesse* von TOC.

Das Ziel der Denkprozesse besteht darin, einen Zielzustand zu definieren, der durch eine Reihe von Transformationen erreicht werden kann. TOC deckt mit seiner eigenen Projektmanagementmethode *Critical-Chain-Projektmanagement* (CCPM) auch die Durchführung ab.

12.4 Critical-Chain-Projektmanagement (CCPM)

Critical-Chain-Projektmanagement (CCPM) wendet den Gedanken von Drum-Buffer-Rope auf das Planungs- und Kontrollproblem von Projekten folgendermaßen an:

1. Starte mit dem Netzplan von Projektaufgaben, ihren Schätzungen und Abhängigkeiten
2. Plane Aufgaben so, dass:
 - a) keine Aufgabe gestartet wird, bevor nicht all ihre Abhängigkeiten erfüllt sind (keine »Nicht-Standard«-Abhängigkeiten zwischen Aufgaben),
 - b) Multitasking eliminiert wurde (kein »Ausgleich« von »Ressourcen«² zwischen Aufgaben),
 - c) die Projektlaufzeit minimiert ist (auf eine erste Näherung).
2. Nein, ich mag »Ressourcen« als PM-Sprache für »Menschen« auch nicht. Wir benutzen das Wort hier im generellen Sinne.

3. Identifiziere die *kritische Kette* – die Sequenz der Aufgaben, die die Gesamtdurchlaufzeit des Projekts bestimmt.
4. Teile Schätzungen von Aufgaben in zwei Komponenten: in die erwartete Dauer und den verbleibenden Sicherheitspuffer. Typischerweise passiert dies nach einer Faustregel wie zum Beispiel, dass der Sicherheitspuffer in jeder Schätzung ein Drittel des gesamten Umfangs ausmacht.
5. Lege alle Sicherheitspuffer zu einem *Buffer* zusammen. Diese Buffer befinden sich entweder am Ende eines Projekts (Projektpuffer) oder schützen jede Abhängigkeit (Versorgungspuffer).

Abbildung 12–1 zeigt einen einfachen Plan mit seinem Projektpuffer und einem Versorgungspuffer.

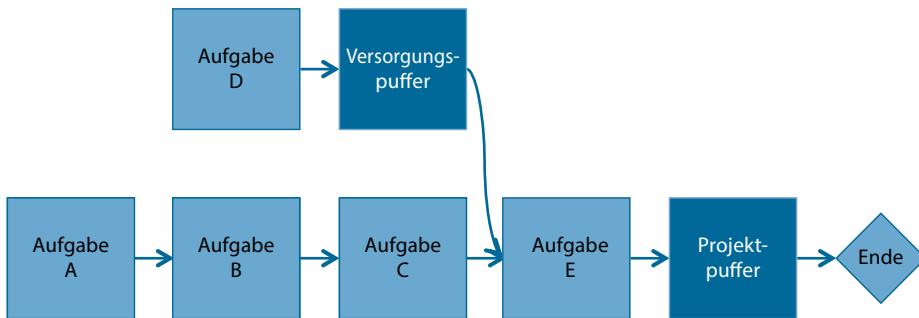


Abb. 12–1 Ein simpler Plan mit Projektpuffer und Versorgungspuffer

Schritt 2 der CCPM entspricht der *Drum* in DBR; aus diesem Schritt sollte ein grober Plan entstehen, der offensiv ist, jedoch keine unsicheren Annahmen macht. DBR vermeidet explizit die Tricks zur »Planverdichtung«, die Projektmanager in Schwierigkeiten bringen.

Entsprechend dem *Rope* von DBR wird Arbeit so geplant, dass sie begonnen wird, sobald ihr korrespondierender Buffer anfängt, verbraucht zu werden.

Sowie die Arbeit begonnen worden ist, übernimmt wie bei DBR das *Buffer-Management*. *Buffer-Penetration-Diagramme* können dem Projektmanager dabei helfen; diese zeigen den Pufferverbrauch gegenüber der Zeit oder dem Projektfortschritt an. Eine interessante Variante davon sind *Fieberdiagramme* (Abb. 12–2); farbige Hintergründe erlauben es, den Rot-Gelb-Grün-Status jedes einzelnen Puffers direkt abzulesen.

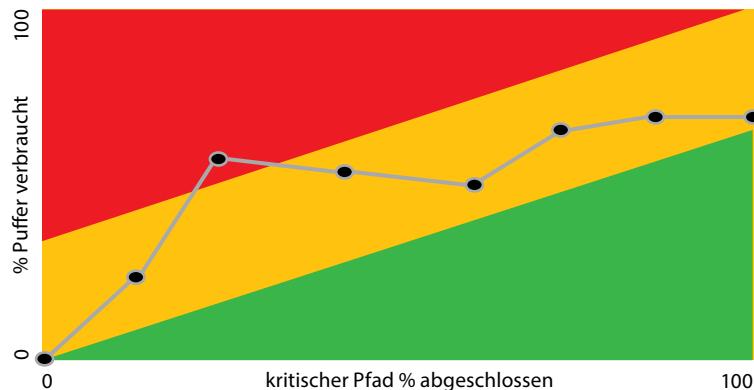


Abb. 12-2 Ein Fieberdiagramm

12.5 Throughput Accounting – Durchsatzrechnung

TOC hat sein eigenes Rechnungsmodell, *Throughput Accounting (Durchsatzrechnung)*, dessen ausführliche Behandlung den Rahmen dieses Buches sprengen würde. Es zielt darauf ab, die vermeintlichen Prioritäten des klassischen Managements umzukehren, das durch das übliche *Kostenrechnungsmodell* getrieben wird:

1. Reduziere Kosten.
2. Reduziere benötigte Investitionen oder Inventar.
3. Erhöhe den Durchsatz, der hier als Verkaufserlöse minus *echte variable Kosten* (*totally variable costs (TVC)*) pro Zeiteinheit definiert wird.

In der Logik der Durchsatzrechnung sieht es im Kostenrechnungsmodell zwar gut auf dem Papier aus, wenn Kosten, investiertes Kapital oder Inventar reduziert werden. Das könnte aber vielleicht für den Durchsatz schädlich sein und damit den Interessen der Organisation entgegenstehen.

12.6 Die Beziehung von TOC zu Kanban

Kapitel 8 bezieht sich am Rande auf eine Geschichte aus dem »blauen Buch« – David Andersons Kanban-Buch –, in der sich eine interessante Transition entfaltet. Sein Kapitel »Extreme Verbesserungen in fünf Quartalen« beschreibt, was später die erste Fallstudie zu Kanban werden sollte. Das Kapitel erzählt die Geschichte der Softwarewartungsabteilung der XIT-Einheit von Microsoft, die es durch die Einführung eines Pull-Systems und einer Reihe inkrementeller Veränderungen geschafft hat, sich bis zur Unkenntlichkeit zu verändern.

Dieses Pull-System war zuerst als simples Drum-Buffer-Rope-System konzipiert (Davids erstes Buch basierte sehr stark auf der Engpasstheorie); erst später

wurde es als ein Kanban-System neuinterpretiert. Nebenbei wurde die Kanban-Methode geboren. Es scheint, als würden wir Lean (Kap. 13) weit mehr Verdienst zuschreiben als TOC; ist das fair?

Schauen wir uns dazu die Hauptelemente von TOC an, um diese Frage zu beantworten.

12.6.1 POOGI und die fünf Fokussierungsschritte

Wie viele andere Kanban-Trainer auch erwähne ich in meinen Trainings gerne POOGI und die fünf Fokussierungsschritte. Ich erinnere mich noch genau an eine Schulung, an der das gesamte Mittelmanagement einer kleinen Firma teilnahm. Uns wurde klar: Wenn die Firma einen Engpass hatte (und das musste sie), wurde dieser durch einen der anwesenden Manager repräsentiert.

Alle Augen richteten sich auf die Finanzmanagerin. Jedem war klar, wie wichtig es war, dass sie das, was sie brauchte, dann bekam, wenn sie es brauchte. Ihre Kollegen beschlossen, sie und ihr Team von allem zu befreien, was unnötig Mehrarbeit verursachte, wie dem Hinterherrennen nach Papieren, Klärungen und Korrekturen. Besondere Priorität wurde allem eingeräumt, was Geld einbrachte. Jeder im Raum war davon überzeugt, dass sich die Dinge verbessern würden. Nicht nur für dieses eine Team, sondern auch für die Firma als Gesamtes.

Das ist eine großartige Geschichte, aber Kanban ist nicht »POOGI mit Kanban-Boards«. Der Engpass – typischerweise als Flaschenhals dargestellt – ist selten das erste Angriffsziel eines Kanban-Anwenders. Falls Sie das heikle Problem überwinden können, dass keine Person und kein Team gern als Flaschenhals bezeichnet wird, haben Sie immer noch das Problem, den Flaschenhals zu identifizieren. Wie sicher können Sie sich über die Lokalisierung Ihres Flaschenhalses sein, wenn die Menge paralleler Arbeit groß ist, es zwischen Arbeitstypen Durchlaufzeiten in zig verschiedenen Größenordnungen gibt, Menschen sich häufig zwischen verschiedenen Aktivitäten hin und her bewegen und (mehr oder weniger) jeder den Anschein macht, gleichermaßen beschäftigt zu sein?

Vielleicht sollte der *Flaschenhals* zu der Liste von Werkzeugen, Konzepten und Metaphern hinzugefügt werden, die sich nicht so einfach aus der Produktion in die Wissensarbeit übertragen lassen, wie manche einen glauben lassen wollen. Kapitel 14 stellt noch mehr davon vor.

Der Flaschenhals genießt vielleicht keinen Platz in der ersten Reihe der Kanban-Methode, POOGI wird jedoch noch lange gelehrt werden. Es fördert Verständnis und es liegt immer noch eine gewisse Stärke in der Idee, dass Sie immer weiter Ihre Systembeschränkungen identifizieren und adressieren müssen. Das ist gerade dann der Fall, wenn Sie offen sind für die Möglichkeit, dass viel breiter gefasste Beschränkungen existieren – das Fehlen von Wissen, Feedback, Lernen, Vertrauen usw. oder der Hang zu nicht hilfreichen Denkweisen.

12.6.2 DBR und CCPM

Natürlich haben in der Kanban-Methode die Kanban-Systeme Drum-Buffer-Rope als das Planungswerkzeug der Wahl abgelöst. Beides sind Pull-Systeme und beide können effektiv sein. Was sie voneinander unterscheidet, ist nicht so sehr die Visualisierung; vielmehr ist es die Methode, sie einzuführen. Implementieren Sie DBR, ändern Sie grundlegend die Art und Weise, wie Sie Arbeit einplanen und den Fluss der Arbeit kontrollieren. Obwohl die Behauptung falsch wäre, dass Kanban-Systeme nicht im geringsten disruptiv wären (schlussendlich wollen wir mit ihnen immer noch Veränderungen herbeiführen), werden diese Systeme typischerweise auf existierende Prozesse aufgesetzt.

Manche in der Kanban-Community sehen DBR und CCPM als Ergänzung zu Kanban (siehe zum Beispiel das Buch meines Freundes Steve Tendon [Tendon & Müller 2013]). Und sicherlich gibt es einige interessante Parallelen, die es wert sind, weiter erforscht zu werden. Das *Rope*-Konzept empfiehlt *explizite Regeln* für den Umgang mit termingetriebener Arbeit und wie sie gestartet wird. Das Buffer-Management fügt eine Variante von Risikomanagement zu *Manage den Arbeitsfluss* hinzu, eventuell erweitert durch Fieberdiagramme und verwandte Metriken. Sind Sie mit größeren, traditionell gemanagten Projekten konfrontiert, schauen Sie sich an, wie CCPM mit Abweichung umgeht, speziell im Hinblick auf Abhängigkeiten.

Das soll nicht heißen, dass ich daran glaube, dass Wissensarbeit am besten in Einheiten der Größe »Projekt« erledigt werden soll oder dass Arbeitspakete von vornherein als termingetrieben behandelt werden sollen; lassen Sie uns diese speziellen Werkzeuge für die seltenen Fälle bewahren, die sie wirklich verdient haben.

12.6.3 Die Denkprozesse

»Schichten des Widerstands« und »Widerstände gegen Veränderung überwinden«? Diese Sätze selbst erzeugen vielleicht schon Widerstand, doch das lässt sich leicht beheben (Youngman empfiehlt »Stufen der Akzeptanz«) und die Kategorisierung scheint hilfreich.

Die Denkprozesse als Denkwerkzeuge? Ich kann nicht behaupten, ein erfahrener Anwender darin zu sein, aber ich habe schon erstaunliche Ergebnisse anderer gesehen. Vielleicht werde ich eines Tages lernen, wie man sie benutzt.

Auch hier wieder eine Warnung zur Vorsicht. Zur Debatte steht nicht die potenzielle Macht, die in den Werkzeugen steckt, sondern ihre Absicht. Bei Kanban geht es darum, die Lernfähigkeit innerhalb Ihrer Organisation zu fördern. Das erreichen Sie nicht dadurch, dass Sie dem Transitionsplan eines externen Experten oder eines internen Spezialisten folgen. Ein Zielzustand wird Sie nur für eine Weile wettbewerbsfähig halten; auf lange Sicht gewinnt das Lernen. Meiner Meinung nach sind projektbasierte Veränderungsansätze deshalb keine nachhaltige Lösung.

12.7 Dennoch ...

Lassen Sie sich nicht von meiner Vorsicht und meinen Warnungen abschrecken – die Engpasstheorie mag vielleicht keine Grundlage der heutigen Kanban-Methode sein, aber eine Rolle als Quelle für Erkenntnis und Inspiration ist ihr sicher. Lesen Sie *Das Ziel* – es wird nicht viel Zeit in Anspruch nehmen und Sie werden belohnt werden. Steigen Sie tiefer in die Engpasstheorie ein und Sie können davon ausgehen, dass Sie sich einige Fragen stellen werden.

13 Agile

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

- *Individuen und Interaktionen mehr als Prozesse und Werkzeuge*
- *Funktionierende Software mehr als umfassende Dokumentation*
- *Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung*
- *Reagieren auf Veränderung mehr als das 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.

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

© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.

So steht es im Manifest für Agile Softwareentwicklung [Agile Manifesto], dem Ergebnis eines bahnbrechenden Treffens im Februar 2001 im Skiorf Snowbird in Utah. Eine Bewegung war geboren.

Ich will ehrlich sein: Als ich **Zusammenarbeit (Kooperation)** als einen von Kanbans Werten ausgemacht hatte, war das ein bewusstes Nicken in Richtung des Agilen Manifests. Es ist einfach, auch andere Werte in das Manifest hineinzulesen: **Respekt, Arbeitsfluss und Kundenfokus** kommen einem fast sofort in den Sinn. Es sind ein paar unterschwellige Resonanzen enthalten wie »Reagieren auf Veränderung mehr als Befolgen eines Plans«, was einen Hinweis auf Anpassungsfähigkeit durch *evolutionäre Lieferung* gibt. Evolution als Methode wird nicht als Wert, aber indirekt in der Präambel, »*erschließen bessere Wege, Software zu entwickeln*«, erwähnt.

Weitere Beweise für diese und andere Werte findet man in den **Zwölf Prinzipien Agiler Softwareentwicklung**, einem Zusatz zum Manifest:

Wir folgen diesen Prinzipien:

1. Kundenzufriedenheit durch frühe Auslieferung wertvoller Software
 2. Anforderungsänderungen selbst spät in der Entwicklung willkommen heißen
 3. Funktionierende Software wird regelmäßig ausgeliefert (eher in Wochen als Monaten)
 4. Funktionierende Software ist das wichtigste Fortschrittsmaß
 5. Nachhaltige Entwicklung und die Fähigkeit, ein gleichmäßiges Tempo halten zu können
 6. Enge und tägliche Zusammenarbeit zwischen Fachexperten und Entwicklern
 7. Gespräche von Angesicht zu Angesicht sind die beste Art der Kommunikation
 8. Projekte werden rund um motivierte Individuen errichtet, denen Vertrauen geschenkt wird
 9. Ständiges Augenmerk auf technische Exzellenz und gutes Design
 10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell
 11. Selbstorganisierte Teams
 12. Regelmäßige Anpassung an sich verändernde Umstände

Schauen wir uns einige dieser »besseren Wege« an, statt diese Prinzipien in abstrakter Weise zu analysieren. Genauer gesagt sehen wir uns drei Methoden an, die dem Treffen in Snowbird unterschiedlich stark einen Teil des Kontexts geliefert haben und die selbst interessante Modelle sind.

13.1 Drei agile Methoden

13.1.1 Feature-Driven Development (FDD)

Jeff de Luca entwickelte FDD 1997 für ein spezielles Projekt bei einer Bank in Singapur. Wie im »blauen Buch« nachzulesen ist, erscheint FDD in der Geschichte Kanbans 2004, als David Anderson es bei Motorola einführt.

Eine Übersicht des FDD-Prozesses ist in Abbildung 13-1 zu sehen.

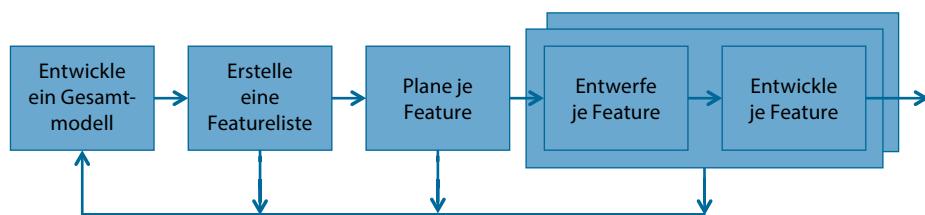


Abb. 13–1 Feature-Driven Development

Wie alle agilen Methoden sollte FDD nicht als traditioneller, phasenbasierter Projektprozess verstanden werden. Als Ergebnis der Entwicklungaktivität entstehen Inkremeante von der Größe eines Features über den Verlauf des Projekts. Wenn man das Ganze von hinten nach vorne betrachtet, geht jeder Entwicklung ein Entwurf voraus, der Feature für Feature gemacht wird. Eine Planung (*Plane je Feature*, dem *Erstelle eine Featureliste* vorausgeht) wird nur durchgeführt, wenn sie gebraucht wird, und in Einheiten, deren Größe die Arbeit weniger Wochen nicht überschreitet.

Eine der auffälligsten Eigenschaften von FDD ist die erste Aktivität, *Entwickle ein Gesamtmodell*, einschließlich eines technischen Modells (ein Objektmodell, meist in grafischer Notation der *Unified Modeling Language, UML*) und begleitender Anmerkungen. Zu Beginn ist dieses Modell noch sehr oberflächlich; es entwickelt sich und wird ständig durch das Feedback der anderen vier Aktivitäten so weit präzisiert, »*wie es jetzt nötig ist*«. Dies passiert speziell durch die »*Entwerfe je Feature*«-Aktivität.

Aus dem Diagramm wird nicht klar, dass die *Projektklienten* (FDD-Begriff für Kunden und andere Stakeholder) über die gesamte Lebensdauer des Projekts an der Modellierungsaktivität beteiligt sind. Die agilen Schlüsselemente sind vorhanden: Die Methode ist kooperativ, anpassungsfähig und inkrementell. Sie erkennt an, dass Wissen gesammelt und über die Zeit neu bewertet und durch Interaktion mit der realen Welt getestet werden muss.

13.1.2 Extreme Programming (XP)

Zum Ende des Jahres 1999 oder zu Beginn des Jahres 2000 legte mir unsere interne Post überraschend ein neuveröffentlichtes Buch mit einem ungewöhnlichen Titel auf den Tisch. Eine kleine Aufmerksamkeit meines damaligen Managers Thomas (Thommi) Suessli. Es handelte sich um Kent Becks *Extreme Programming Explained: Embrace Change* [Beck 1999]. Ich muss zugeben, dass mich das Buch anfangs etwas verwirrt hat, aber ich blieb dran und las später noch weitere Bücher der Serie.

Genauso wie FDD geht XP dem Agilen Manifest voraus. Es nimmt fünf Werte für sich in Anspruch: Kommunikation, Einfachheit, Feedback, Mut und (den später hinzugefügten) Respekt. Zu Einfachheit kann ich keine direkte Analogie in den neun Kanban-Werten finden (nicht, dass es da einen Konflikt gäbe), doch Kommunikation und Feedback korrespondieren gut mit Kooperation und Transparenz; es gibt auch sicherlich eine Verwandtschaft zwischen Mut und Führung; und Respekt findet sich natürlich direkt wieder.

Die Unterschiede zwischen XP und FDD sind beachtlich. Vergleichen Sie das Diagramm von FDD in Abbildung 13–1 mit dem von XP in Abbildung 13–2, um sich die Unterschiede zu verdeutlichen.

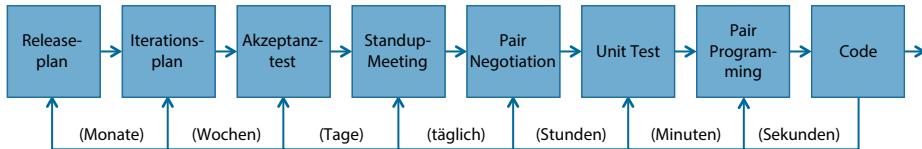


Abb. 13–2 Der XP-Prozess

Mehrere Unterschiede fallen auf:

- Es gibt keine Modellierungsphase. Vielmehr liegen die kürzesten Feedbackzyklen im rechten Teil des Bildes (um *Code* herum) und nicht im linken Teil.
- Es gibt genauso wenig eine explizite Aktivität für den Entwurf (aber es gibt eine Reihe anderer Aktivitäten zwischen der Planung und dem Programmieren). In XP entsteht der Entwurf nach und nach.
- Zwei Aktivitäten gehören zum Test: *Akzeptanztests* (kurioserweise erscheint dies viel früher im Prozess, als man erwarten würde) und *Unit Tests* (hier werden zum Code gehörende Tests kurz vor dem eigentlichen Code geschrieben, den sie prüfen sollen).
- Zwei Aktivitäten beinhalten Paare: *Pair Negotiation* (Austausch zwischen Entwicklern) und *Pair Programming*.

Der Schlüssel liegt im Namen: Der Kern von XP ist wirklich das Programmieren. Warum sollte man ein Modell erstellen, wenn der Code selbst das Modell sein kann? Das Programmieren wird durch das Entfernen von Irrelevantem, die intensive Zusammenarbeit in Paaren und die engen Feedbackzyklen »extrem« gemacht.

Mein liebstes XP-inspiriertes Zitat lautet:

*Wenn es weh tut, mach es öfter und fördere den Schmerz.*¹

Das klingt verrückt, aber es ist inspirierend! Sie sind sich nicht sicher, wie Sie Ihren Code testen sollen? Schreiben Sie den Test zuerst. Sie finden Akzeptanztests schmerhaft? Integrieren Sie sie in den Produktdesignprozess. Ausliefern tut weh? Planen Sie, es viel regelmäßiger zu tun (sogar kontinuierlich). XP ist um die Einsicht herum gebaut, dass diese Schmerzpunkte eigentlich Chancen für hochwertiges Feedback sind; dann gibt es einem Mut, die Drehregler auf Maximum zu drehen.

Dieses Streben nach schnellem Feedback hat die schnelle Entwicklung in Praktiken und Werkzeugen auf den unteren Ebenen befeuert. Neue Gruppen sind um diese herum entstanden, haben sie weiter verfeinert und arbeiten hart daran, sie möglichst weit zu verbreiten. Durch die Open-Source-Mentalität der Entwicklung ist die Zahl von Einführungen und Innovationen stets sehr hoch gewesen.

1. »If it hurts, do it more frequently, and bring the pain forward.« – Es gibt verschiedene Versionen dieses Zitats. Meine Quelle für diese bestimmte Version ist das exzellente Buch von Jez Humble und David Farley *Continuous Delivery* [Humble & Farley 2010].

Manche dieser Dinge wären sowieso passiert, doch es besteht kein Zweifel daran, dass XP hier eine sehr wichtige Rolle gespielt hat.

Kent Beck war schon, bevor er XP entwickelte, ein Unit-Test-Pionier. Er gründete mit der Veröffentlichung von SUnit, einer Implementierung für Smalltalk-Sprachen, die xUnit-Familie, eine Sammlung von automatisierten Testframeworks mit offenem Quellcode. Seitdem wurden xUnit-Ausführungen für viele andere Sprachen entwickelt wie zum Beispiel jUnit für Java und Test::Unit für Ruby. Unit Testing ist heutzutage nicht einfach nur ein gelöstes Problem, sondern wie testgetriebene Entwicklung (*Test Driven Development*, TDD) auch auf dem Weg in den Mainstream.

Automatisiertes Akzeptanztesten beinhaltet:

1. Die Spezifikation des zu erwartenden Produktverhaltens – häufig in einer Form geschrieben, die von Nicht-Entwicklern gelesen (und sogar geschrieben) werden kann
2. Die technische Möglichkeit, ohne menschliches Zutun mit dem Produkt zu interagieren, oft durch einen (manchmal simulierten) Webbrowser
3. Den Kleber zwischen diesen beiden Dingen – z.B. Programmiersprachenunterstützung

Zusätzlich hat es eine große Menge Innovationen rund um das Thema gegeben, die zu ganzen Communitys für die verschiedenen Frameworks und Techniken zur Spezifikation geführt haben. Manche, wie *Behavior-Driven Development* (BDD), sind zu eigenen Methoden geworden.

Quelloffene, verteilte Versionskontrollsysteme (*distributed version control systems*, DVCS) wie *git* erlauben es großen Projekten wie Linux, dass Hunderte (wenn nicht Tausende) Entwickler daran arbeiten – wahrlich eine riesige, gemeinschaftliche Unternehmung. Diese Werkzeuge sind mittlerweile das Fundament für mehrere Lösungen von Entwicklungsproblemen. Immer häufiger – und das ist wundervoll für Hobby-Programmierer wie mich – sind diese in die Hosting-Plattformen integriert; meine neueste Kreation zu veröffentlichen kann so einfach sein wie ein schnelles »*git push*« in der Kommandozeile. *Continuous Delivery* (*Kontinuierliche Auslieferung*) führt die Automatisierung von Codemanagement, Testen und Auslieferung ins Extreme; Firmen wie Amazon liefern heutzutage so häufig aus, dass die durchschnittliche Zeit zwischen den Lieferungen in Sekunden gemessen wird!

Mit Arbeitsabläufen, die nur Stunden oder Tage dauern, müssen Arbeitspakete in XP sehr klein und testbar sein. Da *funktionale Anforderungen*, *Use Cases* und *Features* als zu groß empfunden wurden, machte XP die *User Story* populär. Dies ist eine Anforderung, die in einem einzelnen Satz auf eine Karte geschrieben wird – häufig stereotypisch (»Als <Benutzer> möchte ich <Etwas> damit ich <Nutzen>«) – und als Einstieg für eine Konversation dient. Diese User Stories

wurden auch zu einem viel erforschten Objekt – ganze Bücher wurden zu diesem Thema geschrieben.

13.1.3 Scrum

Sie haben vielleicht bemerkt, dass ich nicht auf die Planungsaspekte von XP eingegangen bin. Dafür gibt es einen guten Grund: Das *Planning Game* von XP wurde zu einem sehr großen Teil von Scrum verdrängt. Mittlerweile ist Agile für viele Menschen gleichbedeutend mit Scrum. Dabei ist es egal, ob mit oder ohne die technischen Praktiken von XP².

Scrum ist ein Mitte der 1990er-Jahre (wie XP entstand es vor dem Agilen Manifest) entwickeltes, iteratives und inkrementelles Prozessframework. Sein Prozess ist in Abbildung 13–3 dargestellt.

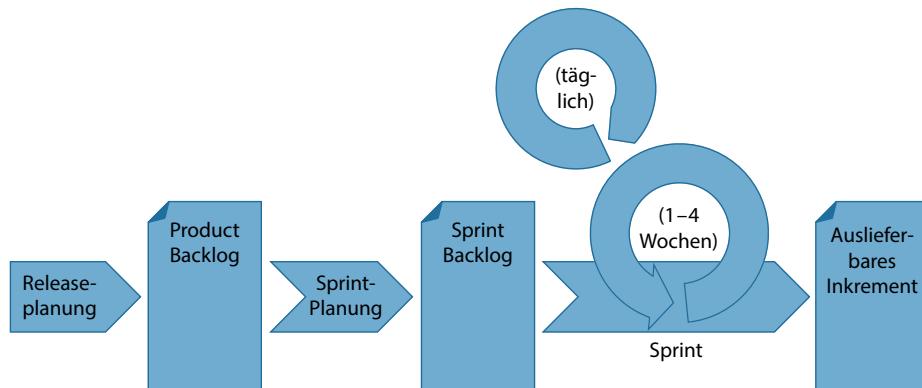


Abb. 13–3 Der Scrum-Prozess

Praktischerweise lässt sich Scrum (in etwa) in drei Dreiergruppen beschreiben³:

1. Drei Rollen
Product Owner (PO), Scrum Master (SM) und Team
 2. Drei Ereignisse
die Sprint-Planung (*Sprint Planning*) vor dem Sprint, das tägliche *Daily Scrum* während des Sprints und zwei Meetings, das *Sprint-Review* und die *Sprint-Retrospektive*, um den Sprint abzuschließen.
 3. Drei Schlüsselartefakte
das *Product Backlog*, das *Sprint Backlog* und das *auslieferbare Inkrement*⁴
-
2. Nicht völlig zu Unrecht hat Martin Fowler für Scrum ohne XP-Praktiken den bekannten Begriff »Flaccid Scrum« (schlaffes Scrum) geschaffen.
Siehe <http://martinfowler.com/bliki/FlaccidScrum.html>.
 3. Dass die Kanban-Werte auch in Dreiergruppen daherkommen, ist Zufall!
 4. Anm. d. Übersetzer: Im Original steht, dass die drei Scrum-Artefakte ein *Burndown-Chart* beinhalten. Dies wurde mit dem Scrum-Guide 2012 zum *auslieferbaren Inkrement* geändert.

Die Rolle des Scrum Master besteht darin, dem Team und dem Product Owner die Entwicklung und die Verbesserung zu erleichtern. Der Product Owner repräsentiert das zu entwickelnde Produkt, oft als Stellvertreter für den Kunden. Im Zentrum des Ganzen – das wurde bewusst so designt – steht das Team. Beachten Sie, dass der Scrum Master und der Product Owner auf einem schmalen Grat zwischen Schutz (um dem Team den Raum zu geben, produktiv zu sein) und Isolation (wo Zusammenarbeit entweder nur intern oder vermittelt stattfindet) wandern.

Die drei Ereignisse sollen regelmäßig stattfinden, wie ein Uhrwerk. Das Sprint-Planungs-Meeting (Sprint Planning) entscheidet über den Inhalt des nächsten Inkrement, d.h. die Arbeit, die im nächsten Sprint geliefert werden soll. Ein Sprint dauert typischerweise zwei Wochen. Das Daily Scrum ist ein Meeting im Stehen (Kap. 1 erwähnt diesen Punkt kurz). Im Sprint-Review wird die abgeschlossene Arbeit (in einer Demo für die Stakeholder) und die geplante, aber nicht fertiggestellte Arbeit inspiziert. Die Sprint-Retrospektive blickt zurück auf den abgeschlossenen Sprint, um herauszufinden, was gut funktioniert hat und was verbessert werden kann.

Der Rhythmus von Scrum ist gleichzeitig Stärke und (bewusst) größte Quelle für Herausforderung. Der schwierige Teil ist, die richtige Menge an Arbeit zu bestimmen und diese dann zu liefern, sodass sie genau in die *Timebox* des Sprints passt. Bei zu wenig oder zu viel Arbeit ist das Ergebnis das gleiche: Verschwendungen und Frustration innerhalb und außerhalb des Teams.

Das Product Backlog und das Sprint Backlog beschreiben die noch zu erledigende Arbeit auf verschiedenen Ebenen; das auslieferbare Inkrement stellt den aktuellen Fortschritt dar; Burndown-Charts (ein Verwandter des kumulativen Flussdiagramms) visualisieren den Fortschritt im Sprint oder bis zum nächsten Release. Typischerweise werden diese Artefakte gut sichtbar zur Verfügung gestellt.

Viele dieser Elemente könnten als pragmatische Lösungen zu häufigen Problemen gesehen werden. Haben Sie mehrere Kunden, weit entfernt vom Team? Besorgen Sie sich einen hochverfügbaren Product Owner. Brauchen Sie jemanden, der passende Praktiken und Verhaltensweisen modelliert und leichter implementierbar macht? Holen Sie sich einen Scrum Master. Sind Ihre Feedbackzyklen zu langsam für die Häufigkeit, mit der sich die Umgebung ändert? Planen Sie die Arbeit in kurzen Timeboxen und treffen Sie sich täglich.

Hier wurden die Hauptelemente von Scrum einzeln beschrieben. Die Väter des Frameworks erwarten von den Anwendern allerdings, dass alle Teile gleichzeitig stattfinden. Das ist nicht einfach. Und es soll auch nicht einfach sein; um Ken Schwaber zu zitieren: »Scrum ist schwierig und disruptiv.«⁵ Damit Scrum effektiv funktionieren kann, werden signifikante Änderungen von der Organisation gefordert.

5. »Scrum is hard and disruptive.« – <http://www.controlchaos.com/storage/scrum-articles/Scrum%20Is%20Hard%20and%20Disruptive.pdf>.

13.2 Kanban und Agile

13.2.1 Kompatibilität

Oft werden wir gefragt: »Ist Kanban Agile?« Für jeden, der Kanban als die »Beginne mit dem, was du gerade tust«-Methode versteht, ist das eine etwas komische Frage. Trotzdem verdient die Frage eine respektvolle Antwort.

Benutzt man »Agile« als Adjektiv, lohnt es sich, etwas tiefer zu bohren, um herauszufinden, was es mit dieser Frage wirklich auf sich hat:

- Sind die Werte von Kanban und die des Agilen Manifests kompatibel? Ja, absolut! Es gibt eine gemeinsame Basis, nicht nur für einen Vergleich, sondern auch für eine Integration.
- Kann Kanban dabei helfen, »Agilität zu verbessern« oder Dinge »agiler« zu machen, einen bestehenden Softwareentwicklungsprozess – explizit Agile oder sonstwie – komplett konsistent mit den Werten und Prinzipien des Agilen Manifests zu machen? Die Antwort auf diese Fragen ist nicht nur ein sich wiederholendes »Ja«, es ist sogar genau das, wofür die Methode anfangs entwickelt wurde.
- Ist Kanban im Sinne des Manifests ein *Weg, Software zu entwickeln?* Nein. Es ist wahrscheinlich Haarspaltereи, aber in diesem Sinne wäre es falsch, Kanban als agile Methode zu beschreiben. Es ist nicht im geringsten ein Entwicklungsprozess (oder irgendeine andere Art von Lieferprozess); in der Definition der Methode gibt es nichts, was sie mit Software verbindet. Kanban ist eine Managementmethode, die sehr breit auf kreative Wissensarbeit, mit besonderem Fokus auf organisatorische Änderung, angewendet werden kann.
- Ist Kanban Teil der agilen Bewegung? Die Identität der Kanban-Community ergibt innerhalb wie außerhalb der agilen Bewegung Sinn (beachten Sie, dass beide Communitys eine ähnliche Beziehung zu Lean haben). Manche Menschen scheint diese Zweideutigkeit zu stören, aber eigentlich ist sie sogar hilfreich und notwendig. Ideen haben jeweils innerhalb ihrer Communitys ausreichend Raum, um zu wachsen und zu gedeihen, und wenn die Zeit reif ist, gibt es einfache Möglichkeiten der gegenseitigen Befruchtung. Für Dienstleister (Trainer, Coaches, Berater usw.) kann die Fähigkeit, aus verschiedenen Identitäten wählen zu können, sehr hilfreich sein.

Auf der Ebene von Methoden und Praktiken:

- Funktioniert Kanban mit iterativen Methoden, speziell Scrum? Ja, und im Fall von Scrum hat die Kombination sogar einen eigenen Namen – *Scrumban* (wird in Kap. 16 beschrieben). Kanban funktioniert innerhalb von Scrum, wo es hauptsächlich Verbesserung auf Teamebene vorantreibt, und außerhalb von Scrum, wo es einem bewusst teamzentrierten Framework dabei hilft, die Herausforderungen von Skalierung zu adressieren.

- Bedeutet Kanban nicht, dass man Iterationen und andere Bestandteile von Scrum abschafft? Dies ist ein ernst zu nehmendes Missverständnis. Kanban ist die »*Beginne mit dem, was du gerade tust*«-Methode; wir wären die Ersten, die Sie davor warnen würden, Teile Ihres aktuellen Prozesses unkontrolliert zu streichen. Allerdings wäre es nicht ehrlich von uns, wenn wir so tun würden, als würde das Streben nach Arbeitsfluss nicht irgendwann Ihre Selbstverpflichtung zu Timeboxen, Story Points und Ähnlichem auf den Prüfstand stellen. Wie Sie und Ihre Organisation damit umgehen, liegt in Ihrem Ermessen.
- Stellen agile Bewegung, Methoden und Praktiken wichtige Modelle für Kanban-Anwender dar? Auf jeden Fall – es ist nahezu undenkbar, dass ein effektiver Kanban-Anwender aus dem Bereich der Softwareentwicklung ohne tiefes Verständnis und Respekt für Agile sehr weit kommen würde.

13.2.2 Wann ist Kanban zu benutzen?

Eine Frage bleibt: Wann und wie kann Kanban in einem agilen Kontext helfen? Weitere Fragen zu Ihrer Situation können hilfreich sein, diese Frage zu beantworten:

- Ist das agile Prinzip von *nachhaltiger Geschwindigkeit* immer noch nur ein anzustrebendes Ziel? Sind Menschen immer noch überlastet in einem Prozess, der nicht so recht zu passen scheint? Könnten **Transparenz**, wie sie Kanban mitbringt (Kap. 1), und **Balance** (Kap. 2) Entlastung bringen?
- Ist Ihre **Kooperation** (Kap. 3) nur nach innen gerichtet? Leidet der **Kundenfokus** (Kap. 4) unter einer überbehütenden Vermittlung rund um das Team oder einem exzessiven internen Fokus auf die Technologie, das Produkt und das Team?
- Schaffen es team- und prozesszentrierte Herangehensweisen nicht, eine notwendige Steigerung des **Arbeitsflusses** (Kap. 5) von Anfang bis Ende der Prozesse zu erzielen? Führen lokale Steigerungen vielleicht zu einer Verschlechterung an anderer Stelle?
- Werden **Führung** (Kap. 6), **Verständnis** (Kap. 7) und **Vereinbarung** (Kap. 8) nicht ausreichend wertgeschätzt? Wird **Respekt** (Kap. 9) zu oft und zu bereitwillig vergessen, wenn die großen Entscheidungen gefällt werden?

Die Art, wie Organisationen ticken, ist schwer zu ändern. Agile Einführungen stehen immer wieder vor dieser Herausforderung und es sollte keine Überraschung sein, wenn solche Probleme auftauchen. Es ist egal, ob Sie gerade erst planen, diesen Weg einzuschlagen, oder schon ein gutes Stück unterwegs sind, die »*Beginne mit dem, was du gerade tust*«-Methode ist da, um zu helfen, nicht um zu urteilen.

Wenn der Großteil Ihrer Organisation bereit ist, die Änderungen durchzuführen, die für eine *schwierige und disruptive* Einführung von Agile nötig sind, und Sie sich des Erfolgs sicher sein können, brauchen Sie Kanban womöglich nicht.

Für Organisationen, die nicht bereit sind, dieses Risiko einzugehen, bietet Kanban einen alternativen Weg zu Agilität. Es ist eine Reise der Koevolution mit offenem Ende, auf der bessere Wege, Dinge zu tun, auf ihre Entdeckung warten.

13.3 Das Modell Agile

Lassen Sie sich nicht durch die zwangsläufig prozesszentrierte Behandlung von FDD, XP und Scrum verwirren. Mit jeder einzelnen dieser Methoden können Sie iterative Entwicklung erzielen und immer noch Folgendes feststellen:

- Menschen dienen dem Prozess und nicht andersherum.
- Das Produkt wird durch die lauteste, interne Stimme getrieben und nicht durch die mit der Zeit sichtbar werdenden Bedürfnisse echter Kunden.
- Viel Arbeit wird fertiggestellt, ohne dass das Endprodukt jemals wirklich eingesetzt wird.
- Richtungswechsel können nicht in Betracht gezogen, geschweige denn im Plan untergebracht werden.

Ohne Unterstützung hilft Ihnen ein Prozess nur zu einem gewissen Teil weiter. Wenn es das Ziel ist, agil zu sein, müssen die agilen Werte selbstverständlich sein. Abgesehen von technischen Einzelheiten, wenn es weder für das Team noch für den Kunden funktioniert, versuchen Sie etwas anderes. Nennen Sie es aber nicht »Agile«, wenn die agilen Werte nicht berücksichtigt werden.

»Agile« hat das Spiel der Softwareentwicklung verändert und tut das auch immer noch. Es hat der evolutionären Auslieferung einen rechtmäßigen Existenzgrund gegeben und dabei dem plangetriebenen Stil des Projektmanagements, das häufig nur schlecht mit Unsicherheit umgehen konnte, die Kontrolle über weite Teile der Softwareentwicklungsbranche abringen können. Kein anständiger Kanban-Anwender kann es sich leisten, Agile zu ignorieren.

14 TPS und Lean

Als Taiichi Ohno, leitender Vizepräsident bei Toyota, 1978 in den Ruhestand ging, veröffentlichte er ein Buch, in dem er das *Toyota-Produktionssystem* (TPS) beschrieb (siehe [Ohno 1998]). Er erläuterte in seinem Buch eine beachtliche Anzahl von Konzepten und Werkzeugen, die außerhalb von Japan kaum bekannt waren, wie zum Beispiel:

- *Just-in-Time (JIT)*

Die radikale Idee, dass das richtige Material, die richtigen Teile und Baugruppen dann, wenn sie gebraucht werden, dort ankommen, wo sie gebraucht werden, und das in der kleinsten möglichen Stückzahl

- *Autonomation*

»Automation mit menschlichen Zügen«, das Frühwarnsystem der Produktionslinie

- *Das Andon-System*

Ein visuelles Signal, um Probleme anzuzeigen, und eine Möglichkeit für den gewöhnlichen Arbeiter, das Fließband zu stoppen (»Stop the line«)

- *Die 5-Why-Methode (Five Whys, 5W)*

Eine Technik zur Durchführung von Fehler-Ursachen-Analysen

- *Kanban*

Das auf Karten basierende System, durch das die Just-in-Time-Produktion gemanagt wird

Durch seine immer stärker werdende Position im globalen Automarkt wuchs das Interesse an Toyota in der folgenden Dekade. 1998 kam dann endlich eine englische Übersetzung seines bahnbrechenden Buches auf den Markt. Zwei Dinge werden Sie überraschen, wenn Sie es lesen: die brillante Einfachheit des Systems und seiner Werkzeuge und Ohnos bemerkenswerte Willensstärke und sein Eifer, ein System zu entwickeln, um die Vision der Firmengründer, der Toyoda-Familie, zu realisieren.

Es war nicht so, dass wir alle Ohno lasen. Populär wurden die ungewöhnlichen, von Toyota verwendeten Methoden, die später als *Lean Production* bekannt werden sollten, durch das Buch *The Machine That Changed the World*

(1991 veröffentlicht [Womack, Jones & Roos 2007]) von James P. Womack, Daniel T. Jones und Daniel Roos. Ein paar Jahre später folgte *Lean Thinking* (1996 veröffentlicht [Womack & Jones 2003]) von Womack und Jones. Japan hatte Toyota; die Welt hatte Lean.

Eine Welle japanischer Wörter fand Eingang in die Lexika:

- *Kaizen*
Kontinuierliche Verbesserung durch inkrementelle Änderungen
- *Kaikaku*
Radikale Veränderung
- *Heijunka*
Nivellierung der Produktion; Arbeitsaufträge werden auf der Produktionslinie gemischt, statt ähnliche Arbeit in größeren Batches zu produzieren
- *Poka-yoke* (oder *Baka Yoke*)
Fehlervermeidung
- *Gemba, gembutsu, genjitsu*
Die »drei Realen«: der reale Ort, an dem Arbeit erledigt wird; das reale Ding; reale Fakten
- *Hoshin Kanri*
Strategische Planung und die Umsetzung

Und die Liste geht noch weiter. Im Rückblick haben die frühen Jahre von Lean Toyotas Werkzeuge und Techniken gut für den westlichen Konsum verpackt. Die Ansätze scheiterten jedoch daran, Einsichten in das darunterliegende Denken zu geben, aus dem die Werkzeuge und Techniken entstanden sind. Das änderte sich, als Autoren wie John Shook, Mike Rother und Steven J. Spear ihren Fokus auf Toyotas Auffassung von Management richteten.

14.1 Drei Lean-Werkzeuge

Eine detaillierte Beschreibung der Funktionsweise einer Produktionslinie bei Toyota würde den Rahmen dieses Buches sprengen. Es folgt ein einfacher Aufbau, der *kanban*, *heijunka* und *andon* verwendet und genügend Details beinhaltet, um einen Vergleich zu den in Teil I beschriebenen Arten von Kanban-Systemen zu ermöglichen:

- Kein Teil oder Material wird bereitgestellt und nichts – weder Baugruppen noch fertige Produkte – wird gebaut ohne entsprechendes *kanban*.
- In der Auslieferung am Ende der Produktionslinie werden *kanban* einzeln¹ aus der *Heijunka*-Kiste gezogen. Die Fächer in dieser Kiste (oder Regal) orga-

1. »Kanban sind wie Schafe.« Ein Kanban, zwei Kanban, ...

nisieren die *kanban* in zwei Dimensionen, eine Achse ist die Tageszeit, die andere die Produktart. Durch die bewusste Auswahl oder physische Begrenzung wird die Produktion der einzelnen Produktarten über den Tag verteilt.

- Die Arbeitsstation (vielleicht eine Zelle mit mehreren Mitarbeitern) am Anfang der Produktionslinie produziert und packt nach den Spezifikationen auf dem *kanban* zusammen. Fällt der Vorrat von Teilen unter eine Nachfüllgrenze, werden *kanban* (Karten) zum vorgelagerten Bereich geschickt.
- Und so geht der Prozess weiter, Bauteile bewegen sich in der Produktionslinie weiter nach unten, *kanban* bewegen sich nach oben, bis zum Wareneingang, wo Teile und Material von externen Zulieferern bestellt werden müssen ...
- ... bis jemand die *Andon*-Kordel zieht, weil er ein Problem erkannt hat; Alarmsignale leuchten auf und die Linie kommt zum Stehen. Nach einer Untersuchung und Korrektur des Problems geht die Produktion weiter.

Die Beschreibung ist eher einfach gehalten, doch gibt es genügend Aspekte, um die hervorstechenden Charakteristiken des Systems zu benennen:

- Alle drei Werkzeuge (*kanban*, *heijunka* und *andon*) sind Beispiele des *visuellen Managements*.
- Alle Arten von Lagerbeständen sind begrenzt. Weder Grundmaterialien noch WIP (die Unterbaugruppen oder teilweise zusammengesetzte Produkte) werden nachgefüllt, bis entsprechende Mengen flussabwärts gezogen worden sind.
- Obwohl man denken könnte, dass es effizienter wäre, in der Produktionslinie große Batches von gleichen Dingen herstellen zu lassen, wird das nicht gemacht. Stattdessen wird über den Tag verteilt eine Reihe von verschiedenen Produkten produziert.
- Arbeiter an der Produktionslinie würden eher *das Fließband stoppen* (für jeden), als Arbeit mit minderwertiger Qualität weiterlaufen zu lassen.
- Dieses System und die Kanban-Boards aus Teil I arbeiten sehr unterschiedlich. Bei der Produktionslinie werden *kanban* flussaufwärts geschickt, um zu signalisieren, dass es eine zu erfüllende Anforderung gibt. Auf unseren Boards repräsentieren die Karten Arbeitspakete, die flussabwärts fließen; Signale werden hier durch die Differenz zwischen der Anzahl von in Bearbeitung befindlicher Arbeitspakete im jeweiligen Status und dem korrespondierenden WIP-Limit ausgelöst.
- Sowohl die *Heijunka-Kiste* als auch unsere Kanban-Boards ermöglichen es, den Mix aus Arbeit zu managen.

Es erscheint abwegig, Arbeit in dieser bewusst schwierigen und scheinbar ineffizienten Art und Weise zu strukturieren und zusätzlich die Arbeiter zu ermächtigen, alles zu jeder Zeit zum Halten bringen zu können! Offensichtlich muss es etwas Spezielles in der Kultur des Unternehmens geben, damit dies funktioniert. Aber warum sollten sie sich dazu entscheiden, die Dinge so zu tun?

14.2 TPS und Lean im richtigen Licht

Um diese Frage zu beantworten, müssen Sie TPS als ein großartiges Beispiel des Systemdenkens verstehen.

Es startet mit einer Vision, einem *Nordstern (true north)*, der die Richtung für die Veränderung vorgibt:

- *Single-Piece-Flow, nacheinander, auf Anfrage, ohne Fehler; 100% wertschöpfende Aktivitäten und Sicherheit für die Menschen, die sie ausführen*

Die Technologie existiert noch nicht, um wirtschaftlich die gesamte Produktionslinie in Batchgrößen von eins (was Single-Piece-Flow bedeutet) laufen zu lassen. Das Streben nach dieser vielleicht unmöglichen Vision hat Toyota in seinem Überlebenskampf im Nachkriegsjapan – in dem ein Mangel an Land, Platz für Fabriken, Fabrikanlagen und Material herrschte – zu der globalen Marktführerposition angetrieben, die das Unternehmen heute einnimmt.

Die Werkzeuge unterstützen einen oder beide der folgenden Zwecke:

1. Die Anforderungen von Kunden befriedigen, so schnell und gleichmäßig wie (derzeit) möglich und mit dem kleinstmöglichen Lagerbestand.
2. Die Firma zu entwickeln, sodass sie ihrer Vision näherkommt, indem die Fähigkeiten der gesamten Belegschaft genutzt werden, um den Arbeitsfluss gleichmäßiger zu machen, den Lagerbestand zu reduzieren, Fehler zu vermeiden, andere Formen von Verschwendungen zu eliminieren und (nicht zuletzt) neue Produkte zu entwickeln, die Kunden wirklich wollen und die profitabel und nachhaltig produziert werden können.

Die zwei Säulen *Just-in-Time* und *Respekt für Menschen* repräsentieren diese Unterziele in Kurzform.

Oft wird dieser entscheidende Punkt übersehen: Die Säulen und Werkzeuge können erst dann richtig gedeutet werden, wenn man versteht, dass Toyotas Streben nach Perfektion eine generationsübergreifende Herausforderung ist. Toyota arbeitet nicht nur daran, Autos zu bauen, sondern auch an der Entwicklung der Fähigkeiten der Firma, ihre eigene Vision zu erfüllen.

Wenn diese Art des Denkens außer Acht gelassen wird, können die Werkzeuge von Lean oberflächlich wirken. Ohne die Werkzeuge kann es sogar noch schlimmer sein – zu oft hört man, dass Lean reduziert wird auf einen kurzfristigen Fokus auf Verschwendungen (möglicherweise, um Kostensenkungen zu verschleiern) oder kontinuierliche Verbesserung (wichtig, aber schwer isoliert aufrechtzuhalten). Die Herausforderung der Lean-Bewegung ist es sicherzustellen, dass das Denken mit den Werkzeugen verbunden ist, sodass diese im angemessenen Kontext angewendet werden können.

14.3 Verbesserungen bei Lean

Vieles von diesem Lean-Denken findet sich innerhalb der folgenden fünf Verbesserungsschritte:

1. **Identifiziere Wert**
aus Sicht des Kunden.
2. **Identifiziere den Wertstrom**
– die wertschöpfenden Schritte im Prozess – und strebe danach, alles Nicht-wertschöpfende zu eliminieren.
3. **Erzeuge Arbeitsfluss**
durch das Entfernen von Verzögerungen zwischen diesen wertschöpfenden Schritten und strebe dabei nach Gleichmäßigkeit.
4. **Führe Pull ein,**
sodass Arbeit vom vorgelagerten Prozess nur dann zum nachgelagerten fließt, wenn dort Bedarf besteht und letztendlich beim Kunden.
5. **Identifiziere Verschwendungen**
und entferne dann Hindernisse für einen gleichmäßigen Arbeitsfluss, verringere Verzögerungen, reduziere Inventar, eliminiere Mängel an der Quelle usw.

Diese Schritte werden oft als die *Lean-Prinzipien* bezeichnet (vielleicht eine nicht ganz zutreffende Bezeichnung, was soll's). Wiederholt angewendet haben wir einen Verbesserungszyklus, der vergleichbar mit dem POOGI-Kreislauf der Engpasstheorie (Kap. 12) ist. Er ersetzt den speziellen POOGI-Fokus auf Engpässe (die üblicherweise als Begrenzung des Durchsatzes verstanden werden) durch Aufmerksamkeit auf Verzögerungen, Gleichmäßigkeit und WIP.

Innerhalb dieses Verbesserungszyklus bevorzugt Lean (und davor Toyota) PDCA (Kap. 3) als Rahmen für inkrementelle Veränderungen. Größere Veränderungen können in einem *A3-Report* (benannt nach der Blattgröße) dokumentiert und geplant werden und über die Zeit im Kontext einer Art Mentor/Mentee-Beziehung weiterentwickelt werden. Kleinere Änderungen können durch verschiedene Arten strukturierter Dialoge gemanagt werden, bekannt als *Katas* (ein weiteres japanisches Wort, dieses Mal nicht von Toyota, sondern aus den Kampfkünsten; es bedeutet eine genau festgelegte Abfolge von Bewegungen).

Schauen wir etwas genauer auf die Schritte 2 und 5 der Lean-Prinzipien. Ohno identifizierte *sieben Verschwendungen* oder nichtwertschöpfende Aktivitäten:

1. **Materialbewegungen**
Eine Quelle für Verzögerungen, Kosten und Risiko durch Verlust oder Schäden
2. **Bestände**
Materialien, Work in Progress (WIP), beendete, aber nicht gelieferte Arbeit – Verschwendungen durch entstandene Verzögerungen und die Kosten, sie zu finanzieren

3. Bewegungen

Schaden für Menschen und Gerät, die durch den Produktionsprozess verursacht werden

4. Wartezeiten

Zeit, die Arbeit in einem inaktiven Status verbringt

5. Überbearbeitung

Mehr Arbeit leisten, als laut Spezifikation nötig ist

6. Überproduktion

Mehr produzieren, als für die derzeitige Nachfrage notwendig ist

7. Korrekturen und Fehler

Effektive Kapazität wird verschwendet, um minderwertige Arbeit auf den erforderlichen Standard zu bringen

Als Ganzes sind die Lean-Prinzipien, die Verschwendungen, die visuellen Managementwerkzeuge (*Kanban*, *Heijunka*, *Andon* etc.) und die Managementpraktiken (*Kaizen*, *A3*, *Hoshin Kanri* etc.) weit davon entfernt, oberflächlich zu sein. Es gibt nur sehr wenige Systeme, die so gut dokumentiert sind. Man muss Zeit und Mühe investieren, um vollkommen wertschätzen zu können, welche geistige Leistung in das System hineingeflossen ist und welche Theorie es untermauert. Kurz gesagt, Lean ist ein wichtiger Wissensfundus.

14.4 Lean Product Development

Die bisher beschriebene Lean-Denkweise lässt noch eine signifikante Lücke offen. Offen gesagt, wird sich durch Verbesserung allein Toyotas Nordstern nicht erreichen lassen. »Auf Anfrage« und »Sicherheit für Menschen« brauchen mehr als nur eine exzellente Produktion – beide brauchen bereitwillige Kunden, gute Produkte in der Pipeline und nachhaltige Profite.

Lean Product Development (LPD) versucht, diese Lücke zu füllen. Im Vergleich zur *schlanken Produktion* (*Lean Manufacturing*) ist die schlanke Produktentwicklung noch ziemlich jung und wächst in einer Geschwindigkeit, die eine beträchtliche Diversität zulässt. Verschiedene Herangehensweisen an LPD werden hier anhand ihrer Autoren vorgestellt. Dies soll keinesfalls eine komplette Übersicht sein, eher eine Liste einiger LPD-Schlüsselfiguren, die die Entwicklung von Kanban beeinflusst haben:

- Donald G. Reinertsen führt Lean zurück zu seinen ersten Prinzipien in der Warteschlangentheorie und Ökonomie und ermöglicht damit, die Ideen und Techniken von Lean Manufacturing und Lean Product Development außerhalb ihrer ursprünglichen Domäne viel effektiver einzusetzen. Der Einfluss von Dons Buch *The Principles of Product Development Flow: Second Generation Lean Product Development* [Reinertsen 2009] geht weit über die Lean-

Community hinaus. Die Kanban-Methode würde wahrscheinlich ohne Dons persönliches Interesse an David Andersons Arbeit nicht existieren; *Serviceklassen* und *Verzögerungskosten* sind zwei Kanban-Konzepte, die ihm zuzuschreiben sind.

- Michael N. Kennedy ist der Autor von *Product Development for the Lean Enterprise* und *Ready, Set, Dominate*. Das »Set« im letzten Titel bezieht sich auf *Set-Based Learning*, wodurch ein wesentlicher Unterschied zur Art, wie Toyota die Produktion und die Produktentwicklung managt (mit TPS bzw. TPD), unterstrichen wird. Im Gegensatz zum Ansatz von Single-Piece-Flow, bei dem ein Teil einzeln nach dem anderen behandelt wird, versucht der set-basierte Ansatz eine parallele Suche nach Lösungen, die eine angemessene Kombination von technischer Möglichkeit, Kosteneffektivität und Attraktivität darstellen. In der Softwareentwicklung, wo physische Beschränkungen üblicherweise unwichtig sind, wollen wir stattdessen *Zusagen hinauszögern* und nicht so sehr den Lösungsraum erforschen, sondern mehrere Möglichkeiten erlauben und uns damit Flexibilität zu geringen oder keinen Kosten erkaufen.
- Peter Middleton und James Sutton waren nicht die Ersten, die die Möglichkeit, Lean-Prinzipien auf die Softwareentwicklung anzuwenden, in Betracht zogen; genauso wenig waren sie die Ersten, die die unpassende Verwendung von Massenproduktion als Metapher für kreative Wissensarbeit anprangerten. Nichtsdestotrotz ist ihr Buch *Lean Software Strategies* [Middleton & Sutton 2005] für seine Verpflichtung auf Kundenbedürfnissen lobenswert. Sie erklären darin, dass ein tiefes Verständnis der beständigsten Kundenbedürfnisse (die sie als *Werte* bezeichnen) nicht nur zu fruchtbaren Beziehungen, sondern auch zu nachhaltigeren Produkten und Produktarchitekturen führt.
- Zu guter Letzt sind da noch Mary und Tom Poppendieck, die die Lean-Prinzipien, Verschwendungen und andere Konzepte aus Lean Manufacturing nahmen und daraus eine für die Softwareentwicklung passendere Menge von Prinzipien erarbeiteten (siehe z.B. [Poppendieck & Poppendieck 2006]). Zweifelsohne hatten ihre Bücher² einen großen Einfluss auf Agile und traditionelleres Denken, lange bevor sich LPD als eigene Disziplin etablierte.

Man kann davon ausgehen, dass LPD sich weiterhin an pragmatischen Linien entwickeln wird, die Folgendes beinhalten:

1. Von Toyota und anderen führenden Firmen entlehnte Praktiken, die ausreichend von ihren Ursprungunternehmen abgrenzbar sind, sodass sie auch in anderen Bereichen vertrauensvoll eingesetzt werden können.
2. Ich möchte zu der persönlichen Anmerkung zu einflussreichen Büchern und Linienmanagern Dr. Peter Lappo hinzufügen, der das Alleinstellungsmerkmal hat, sowohl mein Linienmanager als auch der meiner Frau gewesen zu sein (nicht gleichzeitig). Er lieh mir ein Exemplar des ersten Buches der Poppendiecks kurz nach der Veröffentlichung 2003.

2. Praktiken, die in bestimmten Kontexten entwickelt wurden und dort gut funktionieren. Einige von ihnen werden sich als breiter einsetzbar herausstellen und andere wiederum nicht.
3. Eine starke theoretische Untermauerung, von der manches aus den vorher erwähnten Praktiken entnommen wurde, anderes aus anderen Bereichen importiert wurde und der Rest aus Prinzipien hergeleitet und in realen Projekten getestet wurde.

Dieser Prozess hilft nicht nur der Produktentwicklung, sondern auch dem Lean Manufacturing, der agilen Bewegung und Kanban.

14.5 Lean Startup

The Lean Startup von Eric Ries bringt die Produktentwicklung in Bereiche extremer Ungewissheit, wo grundlegende Dinge wie Kunden, Geschäftsmodelle und selbst die Grundzüge des Produkts zu großen Teilen unbekannt sind [Ries 2011]. In diesem Bereich sind Fehler schwer zu vermeiden, doch sollten sie nur schmerhaft und nicht katastrophal sein (ich habe noch den breiten ökonomischen Schaden durch das Platzen der Dotcom-Blase in den späten Neunzigern im Kopf). Genau wie die schlanke Produktion hat Lean Startup zwei wesentliche Ansätze für Veränderung.

Sein kontinuierlicher, inkrementeller Modus (vergleichbar mit *Kaizen*) ist um einen experimentellen Verbesserungskreislauf herum organisiert, der *Build/Measure/Learn* genannt wird und starke Ähnlichkeit mit PDCA hat:

- **Baue (Build)**
das kleinstmögliche Produktinkrement, das eine Hypothese testen kann.
- **Messe (Measure)**
die Auswirkungen des Inkrements in der realen Welt.
- **Lerne (Learn)**
aus den Ergebnissen, verändere deine Annahmen oder baue auf ihnen auf.

Dieser Ansatz passt sehr gut auf webbasierte Services, wo Techniken wie *Continuous Delivery* und *A/B-Tests* es erlauben, ein Produkt sehr schnell zu entwickeln.

Geht der kontinuierlichen Evolution der Dampf aus, wird etwas Radikaleres benötigt. Das *Kaikaku* des Lean Startup ist der *Pivot*. Dieser oft missbrauchte Begriff wird häufig verwendet, um das »Wegwerfen einer ursprünglichen Idee und neu anfangen« zu beschreiben. Eigentlich soll es aber auf eine bedeutende »Kursänderung« aufmerksam machen, die aus dem Ungültigwerden einer bestimmten, grundlegenden Annahme resultiert.

Mein besonderes Interesse gilt dem, was Lean Startup und die verwandte Disziplin *Lean UX* (das »UX« steht für *User Experience*) mit Feedbackzyklen

machen. Das ist nicht nur ein philosophischer Standpunkt – aufgrund von Erfahrungen aus erster Hand, aus einem »digitalen Musterbeispiel« der britischen Regierung, weiß ich, dass man nicht für ein Start-up-Unternehmen arbeiten muss, um von solchen Ansätzen zu profitieren.

Wie Sie aus dem vorherigen Kapitel noch wissen, hat XP den Fokus von Feedbackzyklen nach »rechts« verschoben, sodass diese sich nicht auf die Anforderungen oder das Design konzentrierten, sondern auf den Code. Lean Startup fügt Feedbackzyklen hinzu, die sich sogar noch weiter rechts befinden, um zu jeder Zeit die Interaktionen zwischen Produkt und seinen Kunden zu prüfen und so Annahmen über Kunden und ihre Bedürfnisse zu validieren (oder auch nicht), die die Entwicklung des Produkts vorantreiben.

Lean Startup setzt einen Trend, der weggeht von *Anforderungen*, das grundlegende Rohmaterial von traditionellen Projekten. XP führte *User Stories* ein als »Platzhalter für Konversationen«. Lean Startup ersetzt diese mit *Hypothesen*, die validiert werden müssen. Wenn Sie sich derart verpflichten, Ihre Annahmen zu testen, dann ändert sich Ihre Beziehung zu Work in Progress wirklich grundlegend.

14.6 Lean/Agile-Hybriden

Kein Zweifel, dass Lean den agilen Methoden viel beibringen kann, andersherum gilt das genauso. Es wird jedoch alles etwas schwerfällig, wenn man immer sagen muss: »Ich habe X von Lean und Y von agilen Praktiken.« Also warum nicht einen Hybriden aus beiden erschaffen, der agile Praktiken mit den Skalierungsansätzen von Lean kombiniert?

Das ist nicht ganz risikolos. Ohne die Werte sind agile Praktiken nur bedingt Agile. Lean-Werkzeuge ohne Lean-Denken? Das gleiche Problem. Und je größer und starrer der hybride Prozess, desto weniger können wir sicher sein, dass er in das Unternehmen passt, in das er eingeführt wird.

Noch ist es zu früh, um zu sagen, ob Lean/Agile-Hybriden von der Stange (SAFe ist das bekanntestes Beispiel dazu) den gleichen kulturellen Einfluss erreichen werden wie ihre Vorläufer. In der Zwischenzeit bevorzuge ich es, Lean und Agile einzeln die Ehre zu erweisen, die sie absolut verdient haben. Ich habe die Hoffnung, dass ihre vernünftige und wertschätzende Anwendung das ist, was mit »Lean/Agile« gemeint ist. Ich hoffe natürlich, dass sich darin auch eine gesunde Dosis Kanban finden lässt!

14.7 Kanban und Lean

Lassen Sie uns den Spieß umdrehen und einen Blick aus der Perspektive von Lean auf die Kanban-Methode werfen.

Kanban besitzt explizit:

- Visuelle Pull-Systeme, die in Teil I durch die Werte **Transparenz** und **Balance** dargestellt werden
- Respekt für Menschen (inklusive vielem von dem, was mit dem Konzept bei Toyota verbunden ist) in der Form von **Kooperation**, **Führung**, **Verständnis** und **Respekt**

Kanban bindet Toyotas Nordstern nicht mit ein und überlässt es damit jeder Organisation selbst, ihre eigenen Fitnesskriterien zu definieren. Nichtsdestotrotz passen die Werte **Kundenfokus**, **Arbeitsfluss** und **Respekt** sehr gut zum Nordstern Toyotas und jeder für sich bietet gute Möglichkeiten zum Messen.

Aus der Sicht von Lean könnte Kanban als Ergebnis des folgenden Prozesses beschrieben werden:

1. Zwei Lean-Werkzeuge nehmen – *Kanban* und *Heijunka* – und diese radikal für die kreative Wissensarbeit uminterpretieren
2. Feststellen, dass die Implementierung dieser beiden Tools einen Einfluss auf die Organisation haben kann
3. Aus erfolgreichen Implementierungen einen Satz von Prinzipien und Praktiken identifizieren
4. Eine neue Sammlung des Wissens um diese Prinzipien und Praktiken herum entstehen lassen

Diese Version der Geschichte ignoriert die Rollen, die TOC und agile Praktiken in der Entwicklung von Kanban gespielt haben. Dennoch gilt, dass Kanban die gleiche Orientierung wie Lean hat, auf einem Teil davon aufbaut und sich durch die Praktik »unter Verwendung von Modellen« weiterhin mit Begeisterung darauf bezieht.

Als Methode und Community erhält sich Kanban jedoch seine Unverwechselbarkeit:

- Wir neigen nicht dazu, den Lean-Jargon zu übernehmen. Das gilt vor allem – nicht ohne Ironie – für die japanischen Begriffe. Dem Erbe von Lean und TPS wird Anerkennung und Interesse entgegengebracht, doch wir bemühen uns, wo immer möglich, eine klare Sprache zu verwenden.
- Wir fühlen uns nicht wohl dabei, Produktion als Metapher für kreative Wissensarbeit zu verwenden. Genauso wollen wir nicht hauptsächlich für *Verschwendung* und *Kaizen* in Erinnerung bleiben; so wichtig diese auch sind, bei kreativer Wissensarbeit geht es genauso viel um die Maximierung zukünftiger Chancen wie um die Optimierung des derzeitigen Prozesses.

Auch nach all dem sind wir aber noch lange nicht fertig mit Lean. Das nächste Kapitel hat Don Reinertsen sehr viel zu verdanken; und auch Teil III (Implementierung) ist von Lean beeinflusst.

15 Ökonomische Ansätze zum Arbeitsfluss

Dieses Kapitel erklärt einige wichtige, ökonomische Konzepte in Kanban:

- *Verzögerungskosten (Cost of Delay, CoD)*
werden mit Don Reinertsen und Lean Product Development in Verbindung gebracht, sie sind aber genauso relevant für Servicetätigkeiten.
- *Haltekosten (Cost of Carry)*
werden in vielen traditionellen Industrien, im Einzelhandel und im Bankwesen (mein Hintergrund) verwendet.
- *Real Options*
sind in der Kanban und Agile Community von Chris Matts und Olav Maasen [Maassen, Matts & Geary 2013] bekannt gemacht worden. Nebenbei bedienen wir uns einiger Elemente der Informationstheorie, die von Douglas Hubbard [Hubbard 2010] beschrieben wurden.

Diese neuen Werkzeuge erfordern keine besonderen Fähigkeiten zur ökonomischen quantitativen Analyse. So ein Ansatz wäre auch sehr schlecht zu skalieren, wenn sich der Prozess in Richtung **Arbeitsfluss** entwickelt (Kap. 5) und die Arbeitspakete zunehmend kleiner werden. Vielmehr helfen die Werkzeuge uns, Prinzipien und Regeln zu entwickeln, die zu einer schnellen und effektiven Entscheidungsfindung führen – um es in Dons Worten zu sagen, ein *Rahmenwerk für Entscheidungen*.

Bevor wir jedoch einen Blick auf diese neuen Werkzeuge werfen, untersuchen wir die Wirtschaftlichkeit von *Projekten*, dem traditionellen Vehikel zur Produktentwicklung. Wir werden feststellen, dass die Wirtschaftlichkeit von Projekten unter bestimmten Voraussetzungen, die repräsentativ für ein breites Spektrum von Produktentwicklungsprojekten sind, sehr viel geringer ist, als es zuerst den Anschein hat.

15.1 ROI und die Pareto-Diät¹

So nah und doch so fern! Ihr Projekt schien so nah dran an der Genehmigung, doch nun wurde eine weitere Runde massiver Änderungen gefordert. Ihr jetziger Plan zeigt, dass Sie ein festes Projektteam für ein Jahr beschäftigen werden. Die Auslieferung wird am Ende dieses Jahres sein und einen Return of Investment (ROI) von 25 % mit sich bringen.

Es wurde Ihnen nahegelegt, dass Sie den Gesamtumfang und die Kosten des Projekts reduzieren, indem Sie Features weglassen und früher Nutzen erzielen. Es gilt dabei allerdings, einen Kompromiss zu machen: Die Fixkosten des Projekts werden einen größeren Teil des Ganzen ausmachen. Sie sind damit einverstanden, die Auswirkungen der Änderungen auf den ROI des Projekts auszuarbeiten.

Einfachheitshalber nehmen wir für diese Übung zwei Parameter an:

1. Die *Fixkosten* (Kosten, die nicht mit dem Projektumfang variieren) betragen deftige 10 % des derzeitigen Gesamtbetrags. Ausgedrückt in Aufwand ist das mehr als ein Monat. Diese Kosten entstehen selbst dann, wenn das überarbeitete Projekt nur die kleinstmögliche Änderung ausliefern würde.
2. 60 % des Werts werden in den ersten 40 % der Zeit und des Aufwands generiert. Noch weiter heruntergebrochen bilden die Erträge auf Featureebene eine Pareto-Verteilung (eine eher flache, verglichen mit der 80-20-Regel, die man mit Pareto verbindet).²

Außerdem müssen wir eine häufige Eigenschaft vieler Projekte hervorheben:

3. Außerhalb eines kleinen Kerns können Features weggelassen werden, ohne dass dies die Logik des Projekts schwerwiegend beeinträchtigt.

Der Wert, der vom Projekt geliefert werden kann, ist in Abbildung 15–1 als Funktion des Projektumfangs dargestellt. Man glaubt, dass das komplette Projekt 125 % seiner Kosten wert ist.

1. Ich verdanke »Pareto-Diät« (Pareto Diet) Apple. Ich wollte eigentlich »Pareto dist« (für »Pareto distribution«) twittern und bekam diese sehr treffende Autokorrektur geschenkt.

2. Siehe <https://de.wikipedia.org/wiki/Pareto-Verteilung>.

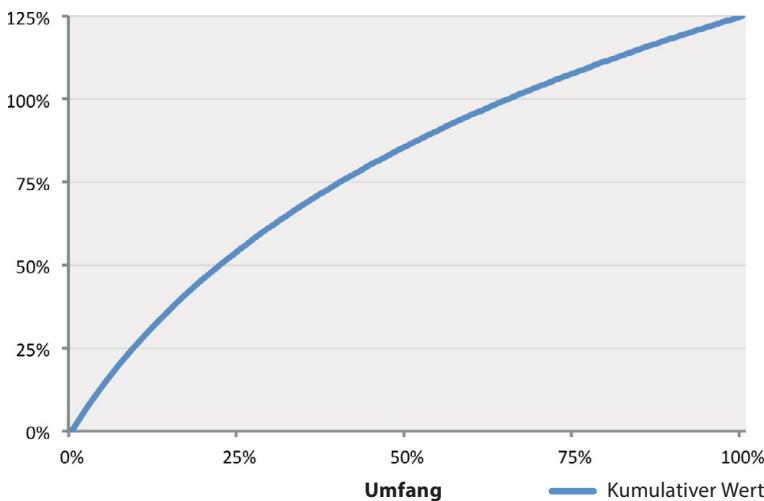


Abb. 15-1 Der Projektwert als Funktion des Umfangs

Reduzieren wir den Umfang des Projekts, werden weniger der Features mit einem geringeren Wert geliefert (vernünftigerweise würden wir den Dingen mit hohem Wert Priorität einräumen), sodass die Linie bei größerem Umfang nicht so steil wird.

Nun fügen wir noch eine Linie für die kumulativen Kosten hinzu, wie in Abbildung 15-2 gezeigt wird. Wegen der fixen Kosten beginnt diese Linie oberhalb des Nullpunkts; natürlicher wachsen die Kosten danach mit der Zeit. Bei 100 % des ursprünglich geplanten Umfangs (erreicht nach einem Jahr) entspricht die Lücke zwischen Wert und Kosten den 25 % ROI des Projekts.

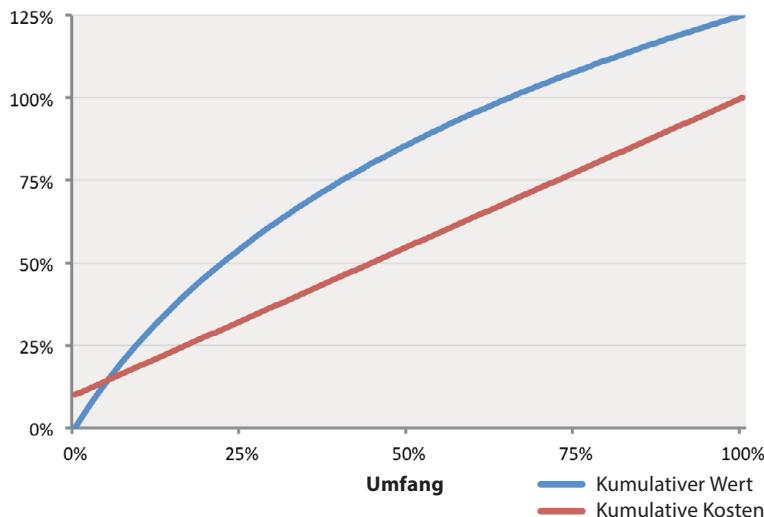


Abb. 15-2 Wert und Kosten als Funktion des Umfangs

Wir stellen fest:

- Bei einem sehr kleinen Umfang ist das Projekt weniger wert, als es kostet. In diesem Beispiel kreuzen sich Kosten und Wert bei etwa 5 % des Umfangs oder 18 Tagen Laufzeit; an diesem Rand der Skala können Projekte Geld verlieren.
- Unter einem Jahr und bei etwa 30 % des ursprünglichen Umfangs (109 Tage) übersteigen die absoluten Gewinne (die Differenz zwischen Kosten und Wert) sogar die erwarteten am Ende des Jahres. Am größten ist der absolute Gewinn bei etwa 60 % des Umfangs (219 Tage). Der ROI (der Gewinn dividiert durch die Kosten) beträgt hier 49 %, fast das Doppelte des ursprünglichen Plans.

Die Verdopplung unseres ROI wäre nicht schlecht. Doch es geht noch besser. Zeichnen wir ihn wie in Abbildung 15–3 ein, sehen wir, dass er seinen Höhepunkt hat, wenn der Umfang auf 26 % des ursprünglichen Umfangs (95 Tage) limitiert ist. Dort bringt uns das Projekt einen ROI von 68 %.

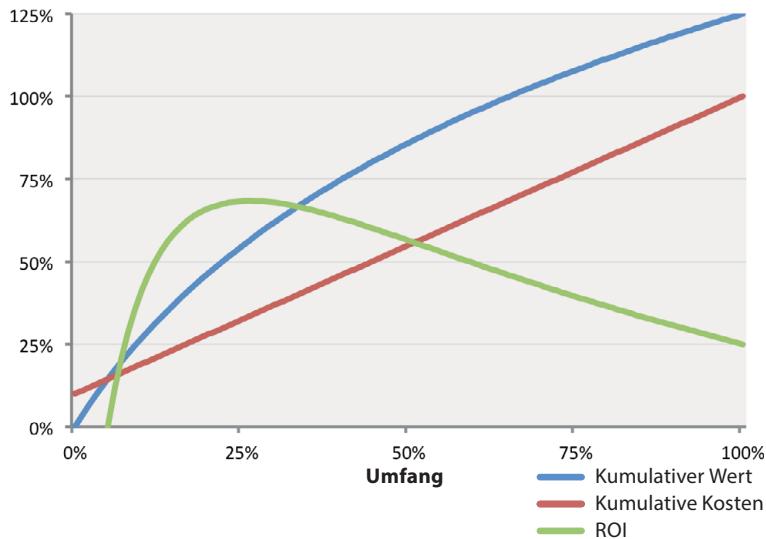


Abb. 15–3 Der ROI hat seinen Höhepunkt, wenn der Umfang limitiert ist.

Das klingt ein wenig absurd! Ist das glaubwürdig?

Quantitativ würde ich sogar die ursprünglichen 25 % ROI mit Vorsicht genießen; es ist das Verhältnis zwischen zwei ungewissen Zahlen. Würde ich demnach 68 % ernst nehmen? Sagen wir mal so, ich würde meine Karriere nicht darauf verwetten.

Qualitativ ist es dagegen robust genug. Sie können den ursprünglichen ROI oder die Dauer ändern, die Form der Wertverteilung oder der fixen Kosten würde zum selben Ergebnis gelangen: Sind Sie bezüglich des Umfangs flexibel, wird eine Kürzung wahrscheinlich die Gewinne verbessern. Zudem ist die Wahl der Para-

meter eher konservativ; der Effekt ist sogar stärker, wenn die Fixkosten nicht so hoch sind oder die Form der kumulativen Wertkurve ausgeprägter ist.

Darüber hinaus werden die wirklichen Kosten lang laufender Projekte durch diese ROI-basierte Betrachtung tatsächlich unterbewertet. Nehmen Sie Folgendes an:

- Manche Features bringen einen zusätzlichen Gewinn, wenn sie früher geliefert werden (wir sind von der oft schlechten Annahme ausgegangen, dass ROI auf Featureebene komplett zeitunabhängig ist). Verspätete Auslieferungen verursachen eine Art von *Opportunitätskosten*, die wir *Verzögerungskosten* (Cost of Delay, CoD) nennen.
- Wie jede finanzielle Struktur, die es mit sich bringt, für zukünftige Gewinne vorab Verbindlichkeiten einzugehen, haben Projekte *Haltekosten*; Projekte stellen einen *Lagerbestand* dar, der auf irgendeine Art finanziert werden muss. Zusätzlich könnten das Geld, die Menschen und andere durch das Projekt gebundene Güter vielleicht bessere Gewinne erwirtschaften, wenn sie an anderer Stelle eingesetzt würden (wieder einmal *Opportunitätskosten*).
- Während die Zeit vergeht und sich die Welt um das Projekt herum ändert, steigt der Aufwand, um alles im Lot zu halten. In diesem Sinne *verschwenden* Projekte *Vermögen*, ein vergängliches Gut.
- Eine frühe Implementierung mancher Features führt eventuell zur Erkenntnis, dass diese oder andere geplante Features anders oder gar nicht hätten entwickelt werden sollen. Sich früh auf den Umfang des Projekts festzulegen, zieht demnach Kosten nach sich. Genauso kostet es etwas, die Möglichkeit für Feedback auszuschließen, und auch das Offenhalten von Optionen hat einen Wert.

Schon bevor wir diese versteckten Kosten ergründen, sollte klar sein, dass dies bereits den Begriff *Projekt* infrage stellt. Bei Bedingungen mit flexiblem Umfang, bei denen das Timing für einzelne Features wichtig sein könnte, oder wenn es negative Konsequenzen durch verfrühte Verpflichtung gibt, sollten wir eigentlich eher über eine Aggregation sprechen und nicht über ein unzerlegbares Ganzes.

Treffender ausgedrückt ist es ein *Batch*; und in diesem Fall ein Batch, der viel zu groß ist. Vielleicht kann um den Batch als Ganzes herum ein einheitlicher Business Case konstruiert werden; vielleicht gibt es einen knorriegen, von Abhängigkeiten beherrschten Kern, der ein fähiges Projektmanagement absolut verdient hat; es ist und bleibt selbst dann ein wenig absurd, das gesamte Unterfangen vollkommen gleich zu behandeln.

Verwenden wir die Sprache von Batches. Ihre ideale Batchgröße³ ist vermutlich viel kleiner, als Sie denken. Wahrscheinlich können Sie es sich erlauben, Ihre Batches um ein Vielfaches zu reduzieren, ohne erst auf eine Reduzierung der

3. Ich achte hier sehr darauf, nicht den technischen Begriff ökonomische Batchgröße zu verwenden. Diese wird typischerweise mit einer Formel berechnet, die nur einen Teil unserer Sorgen in die Rechnung mit aufnimmt; nichtsdestotrotz ist sie eine nützliche Orientierungshilfe.

Transaktionskosten (die Fixkosten pro Batch) zu warten. Tatsächlich ist es sogar so, dass kleinere Batches die wirklichen Transaktionskosten viel einfacher sichtbar machen. Einmal offengelegt stehen sie unausweichlich im Fokus für Verbesserung. Und in der Zwischenzeit gibt es bessere Erträge!

Dies ist eine sehr hilfreiche Art, nicht nur auf das einzelne Projekt zu schauen, sondern auf das gesamte Portfolio. Wir sehen, dass Gespräche über Umfänge eigentlich Gespräche über Batchgrößen sind; das sind Dinge, die wir sichtbar machen können und sollten (**Transparenz**). Reduzierungen von Batchgrößen (unter Umständen aufgrund von Regeln) sind Reduzierungen des WIP, die Auswirkungen auf **Balance** und **Arbeitsfluss** haben. Es geht hier um riesige Vorteile für Kunden, Mitarbeiter und die Organisation, die niemals allein durch einen Fokus auf die Projektdurchführung erlangt werden können.

15.2 Verzögerungskosten

Verzögerungskosten (*Cost of Delay*, CoD) sind eine elegante Art, die Zeitabhängigkeit von Wert zu verstehen, und ein guter Ratgeber, um Entscheidungen zu planen. Ich bin das erste Mal auf *Verzögerungskosten* durch Don Reinertsens Buch *Principles of Product Development Flow* [Reinertsen 2009] und David Andersons Verwendung des Konzepts für *Serviceklassen* in Kanban (siehe Kap. 2) gestoßen.

Meine Behandlung von Verzögerungskosten beginnt nicht in Form von Quantifizierung, sondern von Erkennung. Die Sprache, mit der wir die Dringlichkeit eines Arbeitspaketes beschreiben, sagt uns viel mehr über Verzögerungskosten, als wir manchmal denken. Die folgenden Beispiele stellen keine technischen Begriffe heraus, sondern Sprachsignale (mit einigen versteckten Nachrichten):

1. »Unser Festplattenspeicherproblem wird *zunehmend dringender*.« (Mit jedem Tag Verzögerung steigt die Wahrscheinlichkeit eines Ausfalls.)
2. »Wir sind vom Markt – wir müssen das *jetzt* beheben!« (Jede Stunde Verzögerung kostet uns Hunderttausende Dollar, unverhältnismäßig mehr im Vergleich zu den Kosten, das Festplattenspeicherproblem zu beheben.)
3. »*Nächsten Freitag* wirst du dem Vorstand erklären, wie dieses Desaster passieren konnte.« (Nicht nur Geld, sondern auch Ansehen steht auf dem Spiel; unvorbereitet in dieses Meeting zu gehen, wäre töricht.)
4. Einige Zeit vorher: »Wir sollten anfangen, über das Projekt für die Festplattenüberwachung nachzudenken.« (Es scheint irgendwie wichtig zu sein, aber wie wichtig ist es?)

Diese Beispiele zeigen unterschiedliche Arten von Dringlichkeit – sie sind qualitativ unterschiedlich. Tatsächlich repräsentieren diese Sprachsignale Muster, die so gebräuchlich sind, dass wir ihnen schon Namen gegeben haben:

1. »Zunehmend dringender«, »X \$ Wert pro Jahr«, »wiederkehrende Einsparungen«, »bedeutender Wachstumsschritt«, »Wettbewerbsvorteil«, »Priorität« – dies ist die Sprache von *Standard*-Arbeitstypen. Hier ist die relative Dringlichkeit zwischen den Arbeitspaketen der Schlüssel für die Entscheidungsfindung.
2. »Jetzt«, »auf der Stelle«, »lass alles stehen und liegen« – dies ist die Sprache der *beschleunigten (Expedite)* Arbeitspakete. Es ist wichtig, dass wir diese sofort erkennen; hier ist keine Zeit für eine Kosten-Nutzen-Analyse! Retrospektiv schauen wir allerdings noch einmal zurück und lernen (es gibt ein Beispiel in Kap. 5, das zeigt, wie wir es geschafft haben, unser Supportaufkommen zu reduzieren).
3. »Nächsten Freitag«, »am Ende des Quartals«, »saisonale Chance« – offensichtlich sprechen wir hier von Terminen. In den wirklichen Fällen, in denen eine zu frühe Lieferung nur einen kleinen Mehrwert bringt, verglichen mit der hohen Strafe, wenn zu spät geliefert wird, sprechen wir von Arbeit mit *festem Liefetermin (Fixed Date)*. Die Herausforderungen hier bestehen darin, zu entscheiden, ob die Arbeit es überhaupt wert ist, gemacht zu werden, sie zum passenden Zeitpunkt zu beginnen und das Planungsrisiko unter angemessener Kontrolle zu halten, bis die Arbeit erledigt ist.
4. Schlussendlich die Trickkategorie, für die es fast unsinnig ist, die Dringlichkeit der Arbeit zu beschreiben; dies sind die Arbeitspakete mit *unbestimmbaren Kosten*. Das können die kleinen Verbesserungen sein, die wir immer wieder nach hinten schieben (zusammengefasst sind sie unverzichtbar), die Marktexperimente, die einen massiven oder gar keinen Ertrag liefern, oder die Risikominderung, die einem eventuell den Tag retten kann.

In der Theorie ist die quantitative Seite der Verzögerungskosten auf all diese Arbeitstypen gleichermaßen anwendbar. Geübte Anwender sind jedoch schlau genug, sich mit dem Arbeitstyp unnötige Arbeit zu ersparen. Sie benutzen den Typ hauptsächlich für zwei Dinge:

1. Um die Begründung für Arbeitspakete mit festem Liefetermin zu überprüfen. Ist er einmal ausgewählt, gilt die Annahme, dass sie pünktlich geliefert werden. Das Problem wird dann zu einem Risikomanagementproblem.
2. Für die Reihenfolge der Standardtickets.

Über diese beiden Typen wählen wir Arbeitspakete aus und bringen Sie in eine Reihenfolge, sodass die Verzögerungskosten effektiv minimiert werden. Die beiden anderen Typen werden im Grunde als Overhead behandelt; hier sind nicht so sehr die einzelnen Tickets wichtig, sondern ihr Gesamtanteil im Vergleich zum Ganzen.

Wirtschaftlich streben wir danach, *Opportunitätskosten* zu senken. Eine theoretisch optimale Auslieferungssequenz zu finden, ist sehr schwierig, doch auch hier ist der geübte Anwender wieder schlau genug, es erst gar nicht zu versuchen.

Lange Listen von Arbeitspaketen benötigen eine lange Zeit, sie auszuliefern; neue Arbeit wird eintreffen und Prioritäten werden sich in der Zwischenzeit ändern. Optimale Reihenfolgen sind nicht sehr nützlich, wenn sich die Ziele immer wieder verschieben. Was wir brauchen, ist eine geeignete Strategie, eine *Heuristik*.

Eine einfache und sehr effektive Heuristik ist der Versuch, die Verzögerungskosten schnellstmöglich, in kürzester Zeit, zu senken. Kurzfristig erreicht man dies, indem man die Arbeitspakete mit den höchsten Verzögerungskosten geteilt durch ihre geschätzte Dauer priorisiert. Die Reederei Maersk gibt *Verzögerungskosten geteilt durch die Dauer* (eine finanzielle Durchsatzberechnung) das nette Kürzel »CD3«; die daraus resultierende *Abfertigungsdisziplin* (die Entscheidungsregeln, mit denen wir die Warteschlange der Arbeit managen) ist bekannt als *Weighted Shortest Job First* (WSJF, deutsch: den gewichteten kürzesten Job zuerst).

Der paarweise Vergleich zwischen Arbeitspaketen unter Verwendung dieser Abfertigungsdisziplin sieht folgendermaßen aus:

- Haben die zwei Einheiten ähnliche Verzögerungskosten? Wähle diejenige, die früher geliefert werden kann.
- Sind beide in Bezug auf (verbleibende) Lieferzeit vergleichbar? Wähle die Einheit mit den höchsten Verzögerungskosten.
- Gibt es keinen klaren Gewinner bei den vorherigen Regeln, vergleiche die CD3-Verhältnisse.
- Falls eine Stichwahl nötig ist, könnten Sie positive und negative Risiken berücksichtigen. Betrachten Sie die Wahrscheinlichkeiten eines Riesenerfolgs, eines spektakulären Fehlschlags oder einer längeren Verzögerung. Marginale Unterschiede sind hier kaum von Belang; werfen Sie eine Münze, wenn nötig.

Diese Entscheidungsregeln funktionieren, egal, ob wir ein detailliertes Geschäftsmodell für unser Produkt haben (bei dem die Verzögerungskosten zum Beispiel als Verlust von *Profiten über den Lebenszyklus* gemessen werden) oder einfach T-Shirt-Größen verwenden (SML für niedrigere, mittlere oder hohe Verzögerungskosten und für kurze, mittlere oder lange Dauer; M/S schlägt zum Beispiel S/M). Denken Sie jedoch daran, dass Unterschiede von Größenordnungen in den Verzögerungskosten bei den größeren Einheiten lauern können. Machen Sie nicht den Fehler anzunehmen, dass große Einheiten die Mühe nicht wert sind.

Etwas anspruchsvoller ist das Vorgehen, die folgende Liste von Arbeitspaketen in eine angemessene Reihenfolge zu bringen:

- Einheit 1
Dauer von 10 Tagen, der Wert ist (glauben wir) eine Steigerung der jährlichen Verkäufe von 5 %, wenn wir es schaffen, unseren Mitbewerbern zuvorkommen. Jeder Tag zählt.

- Einheit 2

Dauer von 10 Tagen, der Wert ist eine Steigerung der jährlichen Verkäufe von 7%.

- Einheit 3 (die günstigste)

Dauer von 20 Tagen, ihr Wert ist eine Steigerung der jährlichen Verkäufe in Höhe von 14%.

Trotz ihres niedrigen, absoluten Werts ist die Entscheidung, Einheit 1 als Erstes einzuplanen, gerechtfertigt, weil ihre Verzögerungskosten am höchsten sind. Eine Verspätung von zehn oder gar zwanzig Tagen könnte zu einem Verlust der Chance führen; die anderen können so lange warten.

Die Entscheidung, Einheit 2 vor Einheit 3 zu platzieren, lässt eher Raum für Diskussion. Aus den wenigen verfügbaren Informationen schlussfolgern wir, dass beide die Verzögerungskosten in ungefähr derselben Rate senken. Die Erfahrung hat uns gelehrt, dass die Einheit mit der längeren Dauer sehr viel wahrscheinlicher unter unerwarteten Verzögerungen leiden wird. Also treffen wir eine risikobasierte Entscheidung.

Vielleicht überrascht es Sie, dass die Kosten in keiner der Entscheidungen eine Rolle spielten. Wir erledigen alle drei Einheiten früh genug und es wird insgesamt exakt die gleiche Menge von Geld ausgegeben. Viele Dinge werden sehr viel einfacher, wenn erst einmal ein Grobbudget vorhanden ist und der Entscheidungsprozess frei auf die erzielbaren Erträge fokussiert werden kann.

15.2.1 Die Kosten von Warteschlangen

Verzögerungskosten sind auch sehr hilfreich bei folgender Berechnung: Wenn Sie wissen,

- wie viel Zeit in Ihrem System üblicherweise durch Warteschlangen verloren geht,
- wie hoch die durchschnittlichen Verzögerungskosten pro Arbeitspaket sind und
- wie viele Einheiten sich durchschnittlich in Warteschlangen befinden,

kann eine schnelle Multiplikation diese Warteschlangen mit einem Preisschild versehen. Auf dieser Basis sind Sie dann in der Lage, die Kosten für Investitionen in die Kapazität (zum Beispiel Mitarbeiter hinzufügen) oder Prozessverbesserungen (um die Warteschlangen zu reduzieren) zu untersuchen.

15.3 Haltekosten

Stellen Sie sich vor, einem Investor wurde die einmalige und spannende Gelegenheit gegeben, die Projekte Ihres Unternehmens zu finanzieren. Eine interessante Vereinbarung wurde getroffen, die die Interessen von Investor und Sponsor gleichermaßen berücksichtigt:

- Der Investor stellt zu Beginn jedes Projekts Geld in Form eines Darlehens zur Verfügung, das die zu erwartenden Kosten deckt.
- Am Ende des Projekts ist das Darlehen mit Zinsen zurückzuzahlen. Die Höhe der Rückzahlung orientiert sich allerdings am Erfolg des Projekts:
 - Ist es wie geplant mit allen erhofften Erfolgen fertiggestellt worden, wird der gesamte ausstehende Betrag (Kapital plus Zinsen) an den Investor zurückgezahlt.
 - Ist das Projekt verspätet oder hat anderweitig enttäuschende Ergebnisse geliefert, fällt die Rückzahlung deutlich geringer aus. Dies spiegelt die zusätzlichen Finanzierungskosten und unsicheren Zukunftsaussichten wider.
 - Fehlgeschlagene Projekte bringen gar keine Rückzahlung; die Investition ist für immer verloren.

Schlüpfen Sie in die Rolle unseres hypothetischen Investors. Wie hoch müssten die zu erwartenden Zinsen sein, damit sich solch eine Investition für Sie lohnt?

Schauen Sie sich irgendein typisches Unternehmensprojektporfolio an und Ihnen wird schnell klar, dass diese Investitionen für den Investor erst rentabel werden, wenn die Zinsen sehr hoch ausfallen. Ihre Kreditkarte ist im Vergleich dazu günstig! Im Kreditmarkt sind hohe Zinsraten kein Merkmal für hohe Qualität, sondern das Gegenteil – wir bewegen uns in den Gefilden von hochverzinslichen Risikoanleihen und notleidenden Krediten, mit Raten von 30 %, 40 % und mehr.^{4, 5}

Faktoren, die jede Rechnung beeinflussen, die solche Investitionen mit sich bringen, beinhalten Folgendes:

- Projektdauer (und Dauer des Darlehens) – längere Projekte müssen höhere Zinsen zahlen
- Die Wahrscheinlichkeit von Erfolg oder Scheitern des Projekts – geringere Gewinne (also weniger Zinsen) dort, wo eher ein gutes Ergebnis zu erwarten ist; höhere Gewinne (und mehr Zinsen) bei risikoreicheren Projekten
- Die Fähigkeit, bei Beinaheausfällen etwas von Wert zu retten
- Die eigenen Finanzierungskosten des Investors (Geld ist nicht kostenlos)

-
4. Vereinfacht ausgedrückt, damit der Investor eine effektive Rate von r_i für ein einjähriges Darlehen mit einer erwarteten Fehlerrate von f erhält, ergibt sich die zu zahlende Rate r_c durch

$$r_c = (1 + r_i)/(1-f) - 1$$

Für eine nicht untypische Fehlerrate von 25 % und einer effektiven Rate für den offensichtlich risikofreudigen Investor von 5 % (was nach den Finanzierungskosten nur wenig Raum, wenn überhaupt, für Profite zulässt) erhalten wir eine r_c von 40 %.

5. Vergleichen Sie diese Raten mit den vorgegebenen 5 % der UK-Notenbank. Das sagt mir, dass Geld als kostenlos betrachtet wird und dass Projektdauer und Risiko ökonomisch kaum Bedeutung haben. Vielleicht müssen öffentliche Projekte gar nicht erfolgreich sein – es reicht, wenn sie die Wirtschaft ankurbeln; zum Zeitpunkt ihres Scheiterns sind ihre Sponsoren längst von der Bühne verschwunden.

Diese Faktoren arbeiten nicht unabhängig voneinander. Insbesondere die Projektdauer hat einen großen Einfluss auf die anderen Faktoren. Langlaufende Projekte sind überproportional risikoreicher und teurer und im Falle des Scheiterns ist die Wahrscheinlichkeit, etwas von Wert zu retten, sehr gering. In unserem Gedankenexperiment sollte man von langen Projekten nicht nur erwarten können, dass der absolute Zinsertrag höher ist, sondern auch, dass sie höhere Zinssätze bieten. Kombiniert man höhere Sätze und Aufzinsung über längere Zeit, ist der Gesamteffekt deutlich nicht linear (und deshalb sehr kostspielig).

Haltekosten bemessen die Kosten des Inventars, das die Organisation vorhält. Im Grunde sind sie eine Funktion aus Durchlaufzeit und den Projektgesamtkosten, eventuell verwenden sie feste Raten, die durch die Finanzierungsfunktion spezifiziert werden. Selbst auf diesem grundlegenden Niveau bieten Haltekosten dem Anwender einen einfachen Weg, die finanziellen Vorteile von Verbesserungen auf Portfolioebene zu quantifizieren. Sie können auch dort angewendet werden, wo sich Verzögerungskosten nur schwer bestimmen lassen.

Wenn wir diese Konzepte kombinieren und realistisch hohe Zinssätze verwenden, können wir noch differenzierter urteilen. Sind die angenommenen Verzögerungskosten die risikobereinigten Haltekosten wert? So begründe ich meine starke Präferenz für Arbeitspakete von kurzer Dauer – es ist nicht so, dass Arbeitspakete mit langer Dauer nie Priorität erhalten sollten, sie sollten jedoch eine überproportional wertvolle Chance bieten.

15.4 Optionen

Real Options greift die Idee eines Optioneninstrumentes (eine wichtige Art von Finanzderivaten) aus der Welt der Banken auf und wendet es auf den Bereich der Projektevaluation an. Basiert der Business Case eines Projekts auf einem beobachtbaren Marktpreis, beeinflussen Veränderungen im Preis oder seiner Volatilität die Durchführbarkeit des Projekts. Dies spiegelt die Preisgestaltung von traditionellen Finanzoptionen wider, die abhängig ist vom Preis des zugrunde liegenden Werts, dessen Schwankungsanfälligkeit und der Dauer der Option. Dazu kommen noch Zinssätze und andere externe Faktoren.

Gute Beispiele für die Anwendung von Real Options im großen Stil findet man im Energiesektor. Die Rentabilität von Projekten, bei denen Ölkraftwerke oder Erdölvorkommen erschlossen werden, hängt stark vom Erdölpreis ab. Hohe Preise lassen die Investition in ein Kraftwerk schlecht aussehen (die Option, eines zu bauen, wird weniger wert sein), wohingegen die Erschließung von Erdölfeldern sehr attraktiv scheint (die Option, ein Feld zu erschließen, wird wertvoller zu kaufen oder zu besitzen sein). Interessanterweise steigen jedoch beide Optionen im Wert, wenn die Volatilität der Energiepreise steigt. Selbst wenn die aktuellen Preise unattraktiv erscheinen, erhöhen Schwankungen die Chance, dass während der Laufzeit der Option günstige Bedingungen entstehen.

In beiden Beispielen kaufen Firmen die Optionen, um die Projekte durchzuführen. Sie kaufen Lizenzen und Land; sie treffen Vereinbarungen mit den Regierungen, Aufsichtsbehörden und Lieferanten, sowohl stromaufwärts als auch stromabwärts im Prozess. Jede dieser Handlungen kostet echtes Geld, doch nichts davon verpflichtet die Firma, irgendetwas zu bauen oder Erdöl zu fördern. Sie haben sich das Recht ohne jedwede Verpflichtung erkaufte; sie haben die Verpflichtung aufgeschoben. Vertraglich oder praktisch behalten diese Optionen aber nicht für immer ihre Gültigkeit – Vereinbarungen gelten grundsätzlich nicht unbegrenzt und Lizenzen haben explizite Ablaufdaten.

Die meisten von uns können sich nicht auf einfach beobachtbare Marktpreise berufen, mit denen wir den Wert unserer Optionen abschätzen können. Es ist jedoch nicht alles verloren! Qualitativ ist Optionendenken sehr gut auf kreative Wissensarbeit anwendbar und selbst einige der quantitativen Eigenschaften lassen sich überraschend gut anwenden.

Chris Matts und Olav Maasen haben drei Prinzipien aus dem Optionendenken extrahiert:

- Optionen haben Wert.
- Optionen verfallen.
- Binde dich nie frühzeitig, es sei denn, du weißt warum.

15.4.1 Optionen haben einen Wert

Philosophisch gesehen ist dies selbstverständlich. Aus ökonomischer Sicht ist der Wert der Information, die einige Optionen generieren können, besonders interessant. Zum Beispiel:

Projekt X wird als hochriskant erachtet. Es wird 100.000 \$ kosten. Es könnte 1.100.000 \$ einbringen. Aufgrund unseres derzeitigen Wissens in Bezug auf den Markt geben wir diesem Szenario aber nur eine Wahrscheinlichkeit von 10 %. Wir rechnen also aus, dass der zu erwartende Wert sein wird:

$$-100.000 \$ + 10\% \times 1.100.000 \$ = 10.000 \$$$

Projekt Y kostet 5.000 \$ für eine Marktuntersuchung, die uns mit Sicherheit sagen könnte, ob Projekt X gemacht werden sollte. Wie viel ist diese Kombination wirklich wert? Angenommen wir gehen mit unserem perfekten Wissen rational vor:

$$- 5.000 \$ + 10\% \times (1.100.000 \$ - 100.000 \$) = 95.000 \$$$

Ein ganz schöner Unterschied. Fast eine zehnfache Verbesserung!

Dies ist ein erfundenes Beispiel und wir werden wahrscheinlich selten solche Rechnungen aufstellen. Nichtsdestotrotz ist es klug, nach günstigen Wegen zu suchen, um Fragen wie »Können wir das machen?«, »Werden sie es mögen?« und »Brauchen wir das wirklich?« zu beantworten. Sind die Verzögerungskosten

dazu auch noch hoch genug, kann es durchaus sinnvoll sein, mehrere Ansätze parallel zu verfolgen, sodass in kürzester Zeit eine akzeptable Lösung gefunden werden kann.

15.4.2 Optionen verfallen

Wann verfällt die Option, ein neues Weihnachtsprodukt zu liefern? Lange vor dem 25. Dezember, das ist sicher! Mit wenigen Worten werden wir daran erinnert, dass termingetriebene Arbeit mehr bedeutet, als nur Termine zu halten; es geht darum, das meiste aus Chancen herauszuholen. Das bedeutet, zum richtigen Zeitpunkt zu starten oder es sein zu lassen – und ist das Zeitfenster klein und die Chance auf Profit nicht groß genug, ist es womöglich die richtige Entscheidung, die Chance vorbeiziehen zu lassen.

Manchmal investieren wir mit dem Wissen in diese Optionen, dass sie bald verfallen werden. Eine Webdomäne zu registrieren, kostet uns zum Beispiel nur sehr wenig; Sie haben immer die Kontrolle, wenn es darum geht, die Registrierung zu erneuern, und es raubt Ihnen nicht den Schlaf, wenn die Gelegenheiten, sie zu nutzen, verstreichen.

15.4.3 Binde dich nie frühzeitig, es sei denn, du weißt warum

Das klingt schon wieder nach einer Binsenweisheit, aber es ist eine, die es wert ist, wiederholt zu werden. Wie anders würden Projekte aussehen, wenn jemand bei jedem Feature die Frage stellen würde:

*Was müsste wahr sein, damit diese Option fantastisch aussieht?*⁶

In den Fällen, wo die Antwort zu dieser Frage unbekannt ist, eröffnet sich eine neue Ebene von Sondierungsmöglichkeiten.

Typischerweise bieten Projekte die perfekten Mechanismen, um die Frage zu vermeiden. Die Entscheidung zu Umfang, Dauer und Kosten ist schon längst entschieden worden, bevor die Frage überhaupt gestellt, geschweige denn beantwortet werden kann. Das Konzept ist so angelegt, dass Änderungen an einem dieser Punkte Drama und Stress erzeugen!

Stellen Sie dies einem auf Optionen basierenden Ansatz gegenüber. Statt eines vorher festgelegten Projektbacklogs haben wir ein Portfolio, bestehend aus Optionen. Ein wachsender Pool aus nicht verbindlichen Ideen, die vielleicht verwirklicht werden oder vielleicht auch nicht. Optionen werden ausgeführt – sprich Arbeit wird gezogen –, wenn sie im Vergleich zu allen anderen Alternativen die wertvollsten Informationen generieren.

6. »What would have to be true for this option to look fantastic?« – Roger L. Martin, in: [Lafley & Martin 2013]

Optionendenken hat auch interessante Auswirkungen auf das Risikomanagement. Außerhalb des Entwicklungsprozesses verbleiben nicht ausgeführte Optionen in den Händen derjenigen, die diese am besten managen können, zum Beispiel bei denen, die mit dem Wissen über den Markt die Chancen maximieren können. Ist die Option allerdings einmal innerhalb des Systems, ändert sich das Risiko in Bezug auf Art und Verantwortung – es gilt, Erwartungen zu erfüllen. Es steckt mehr hinter dem Akt des Ziehens von Arbeit, als man auf den ersten Blick sieht – es schafft eine Verpflichtung und transferiert Risiken. Dies sollte nicht auf die leichte Schulter genommen werden.

15.5 Alles zusammengeführt

- Setzen Sie Ihr Projektporfolio auf Diät – reduzieren Sie Batchgrößen offensiv in Richtung des derzeitigen Ideals.
- Behandeln Sie nicht jede Arbeit gleich. Verstehen und klassifizieren Sie Arbeit auf jeder Ebene nach ihrem Dringlichkeitsprofil; steuern Sie den Gesamtmix der Arbeit.
- Lernen Sie innerhalb jeder Dringlichkeitsklasse, Arbeit mit hohen Verzögerungskosten oder hohem Informationswert zu erkennen. Führen Sie Abfertigungsdisziplinen ein, die diesen Durchsatz erhöhen.
- Machen Sie Ihre Optionen sichtbar; wenn Sie nicht sicher sind, dass sie »fantastisch« sind, schaffen Sie weitere Sondierungsmöglichkeiten, die zusätzliche Informationen generieren, statt Optionen verfrüht zu ziehen.
- Egal, ob durch Verzögerungskosten oder Haltekosten, erfassen Sie die Kosten Ihres WIP. Finden Sie Wege, diese auf jeder Ebene zu kontrollieren, und erwarten Sie, dass sie weiter sinken.
- Streben Sie kontinuierlich danach, Transaktionskosten zu senken. Machen Sie es jedoch nicht zu einer Voraussetzung für die vorherigen Schritte. Sie *können* es sich leisten, schneller unterwegs zu sein!

16 Die Kanban-Methode

Die vorherigen 15 Kapitel zeigen die Kanban-Methode aus verschiedenen Perspektiven. Teil I ist ein ungewöhnlicher, auf die Werte gerichteter Blick ins Innere von Kanban. Bis hierhin hat Teil II die fortschreitende Beziehung zwischen Kanban und verschiedenen anderen Wissensgebieten erkundet, namentlich Systemdenken, die Engpasstheorie, Agile und Lean.

Aus einer etwas persönlichen Sicht gibt dieses Kapitel nun eine Referenz für die Schlüsselemente und Ressourcen der Kanban-Methode und seiner Community.

16.1 Eine sehr kurze Historie

Die frühen Jahre sahen wie folgt aus:

■ 2004:

David J. Anderson und Dragos Dumitriu arbeiten bei Microsoft gemeinsam an dem, was später die XIT Case-Study werden sollte. Sie wurde später in Kapitel 4 des »blauen Buches« [Anderson 2010] beschrieben und in diesem Buch in den Kapiteln 8 und 12 referenziert.

■ 2005:

XIT wird durch Beiträge von Don Reinertsen mit wenigen Änderungen als Kanban-System aufgesetzt.

■ 2006:

David wechselt zu Corbis und entwickelt die Methode weiter.

■ 2007:

Die Kanban-Methode hat bei einem Treffen am Rande der Agile 2007 Konferenz ihren ersten öffentlichen Auftritt. Aufgrund des großen Interesses der Delegierten von überall auf der Welt wird die Yahoo-basierte Diskussionsgruppe *kanbandev*¹ ins Leben gerufen (Yahoo gehört zu den frühen Kanban-Umsetzern).

1. <https://groups.yahoo.com/neo/groups/kanbandev/info>

■ 2009:

In Miami, Florida, wird die erste Konferenz der Kanban-Community abgehalten, die *Lean & Kanban 2009*. Hier wird die *Limited WIP Society*² gegründet, eine informelle Dachorganisation für lokale Treffen und andere Community-Ereignisse und -Ressourcen.

■ 2010:

David publiziert das »blaue Buch«, *Kanban: Successful Evolutionary Change for Your Technology Business* [Anderson 2010].

■ 2011:

Das erste *Kanban Leadership Retreat*, eine kleine »Unkonferenz«, wird in Reykjavík mit erfahrenen Anwendern, führenden Mitgliedern der Community und Geschäftspartnern abgehalten. Unter dem Schirm der Lean Kanban University (LNU)³ entsteht das, was später zum Accredited-Kanban-Training-(AKT-)Programm werden wird, und ein Programm für individuelle Coaches, die Kanban Coaching Professionals (KCP). (Ich war zwar bei der Konferenz in Reykjavík, aber da zu der Zeit mein Geschäft weder aus Training noch aus Coaching bestand, blieb ich den Gesprächen fern. Allerdings wurde ich bald involviert und übernahm neben anderen Dingen die Verantwortung für das Curriculum der LNU.)

Passend zu einer evolutionären Methode haben sich die Dinge seitdem weiterentwickelt. Die Definition der Methode hat mehrere Revisionen erfahren. Zum Zeitpunkt der Veröffentlichung dieses Buches werden an acht Orten weltweit jährliche Konferenzen veranstaltet. Das Leadership Retreat wird weiterhin ein- bis zweimal pro Jahr abgehalten und *kanbandev* ist auf mehr als 2500 Mitglieder angewachsen.

David veröffentlichte 2012 in Form einer Sammlung von Blogposts mit dem Namen *Lessons in Agile Management: On the Road to Kanban* einen Rückblick auf diesen Zeitraum [Anderson 2012]. Das Buch ist erstaunlich umfangreich und beinhaltet rund 150 Artikel aus Davids Blog über agiles Management aus zwölf Jahren.

Die ältesten Artikel des Buches wurden vor der Veröffentlichung von Davids erstem Buch *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results* geschrieben. David selbst beschreibt diesen ersten Versuch als eher theoretisch denn praktisch. Erzählen Sie es niemandem, aber ich bin noch nicht dazu gekommen, dieses erste Buch zu lesen.

2. <http://limitedwipsociety.ning.com/>

3. <http://edu.leankanban.com/>

16.2 Grundprinzipien

Sie beinhalten die Philosophie der Kanban-Methode bezüglich Veränderung:

- GP1:** Beginne mit dem, was du gerade tust.
- GP2:** Vereinbare, dass evolutionäre Veränderung verfolgt wird.
- GP3:** Respektiere initial bestehende Prozesse, Rollen, Verantwortlichkeiten und Jobtitel.
- GP4:** Ermutige dazu, Führung auf jeder Ebene der Organisation zu zeigen – vom einzelnen Mitarbeiter bis zum höheren Management.

Die Formulierung des GP2 ist seit 2010 leicht vereinfacht worden, indem das Wort »inkrementell« aus dem ursprünglichen Wortlaut entfernt wurde.

Die ersten drei Grundprinzipien beschreiben eine unverkennbar evolutionäre Herangehensweise, das unaufhörliche Streben nach Anpassungsfähigkeit und Zweckfitness. Genauso charakteristisch ist der Rat, es zu vermeiden, die »Felsbrocken« der Rollen, Verantwortlichkeiten und Jobtitel zu früh herauszufordern.

GP4 wurde nach der Veröffentlichung des »blauen Buches« hinzugefügt. Manchmal lassen wir es weg. Nicht, weil wir das Prinzip nicht mögen, sondern weil es sich nicht so explizit um evolutionäre Veränderung dreht. Alle vier sind allerdings eng verwandt: Ich gruppieren die Werte, die mit den ersten drei Prinzipien korrespondieren – **Verständnis, Vereinbarung, Respekt** (Kap. 7, 8 und 9) –, unter dem Begriff *Führungsdisziplinen*, sodass sie deutlich mit dem vierten, **Führung** (Kap. 6), verknüpft sind.

16.3 Kernpraktiken

Die Grundprinzipien beschreiben, wie Veränderung angegangen werden sollte; die Kernpraktiken beschreiben, wie wir sie kontinuierlich provozieren:

- KP1:** Visualisiere.
- KP2:** Limitiere die Menge paralleler Arbeit (Work in Progress, WIP).
- KP3:** Manage den Arbeitsfluss.
- KP4:** Mache Prozessregeln explizit.
- KP5:** Implementiere Feedbackzyklen.
- KP6:** Erziele Verbesserung kooperativ, entwickle experimentell (unter Verwendung von Modellen und der wissenschaftlichen Methode).

Häufig haben wir beim Lesen dieser Praktiken »mit Kanban-Systemen oder um Kanban-Systeme herum« im Hinterkopf. Es gibt jedoch Momente, in denen es hilfreich ist, die Formulierungen sehr breit zu interpretieren. Beispiele hierfür finden Sie weiter hinten in diesem Kapitel.

Kapitel 1 gruppieren KP1, KP4 und KP5 unter dem einzelnen Wert **Transparenz**. KP2 und KP6 entsprechen **Balance** und **Kooperation** (Kap. 2 und 3). KP3 erstreckt sich über zwei Werte, **Kundenfokus** und **Arbeitsfluss** (Kap. 4 und 5; außerdem Kap. 15).

Die Formulierungen der Kernpraktiken KP1 und KP4 sind seit 2010 vereinfacht worden. KP5 ist später hinzugefügt worden.

Es existiert Literatur (meist älteren Datums), die die Kernpraktiken »**Kern-eigenschaften**« nennt. Die Bezeichnung ist sinnvoll, wenn man im Speziellen meint, wie sie zuerst in freier Wildbahn wahrgenommen wurden, aber sie scheint mittlerweile veraltet und überholt und sollte am besten vermieden werden.

16.4 Kontextualisiertes Kanban

Einige Anwendungen Kanbans sind so verbreitet, dass sie ihren eigenen Namen haben. Drei von ihnen werden hier vorgestellt: *Personal Kanban*, *Portfolio-Kanban* und *Scrumban*.

16.4.1 Personal Kanban

Jim Benson und Tonianne DeMaria Barry beschreiben in *Personal Kanban: Mapping Work | Navigating Life* [Benson & DeMaria Barry 2011], wie Kanban auf die eigene, persönliche Arbeitslast angewendet werden kann.

Sie reduzieren Kanban auf die beiden Praktiken, die am relevantesten dafür sind, die »richtige Arbeit zur richtigen Zeit auszuwählen«:

1. Visualisiere deine Arbeit.
2. Limitiere die Menge paralleler Arbeit.

Ich bin ein großer Fan dieses Buches; drei Generationen meiner Familie haben es schon gelesen! Obwohl es nur zwei Praktiken bestimmt und sich nicht um die organisatorischen Auswirkungen kümmert, erforscht es **Transparenz**, **Balance** und **Arbeitsfluss** sehr tiefgehend (beschreibt diese Werte aber nicht explizit).

Fügen Sie **Kooperation** hinzu und Sie haben den Anfang dessen, was ich zögerlich »Team-Kanban« nenne. Es fehlen fünf der neun Werte, aber um fair zu sein: Es gibt immer noch ziemlich viel darüber zu schreiben. Aus Schweden (das Land gehört zu den frühen Anwendern Kanbans) kommt eine gute Einführung von Marc Hammarberg und Joakim Sundén, die sich *Kanban in Action* nennt [Hammarberg & Sundén 2014]. Wenn Sie in einem Team sind, das gerade mit Kanban beginnt, ist schon allein das Kapitel »Kanban Pitfalls« den Preis des Buches wert.

16.4.2 Portfolio-Kanban

Die Kanban-Methode ist schon aufgrund ihres Designs offen für eine Menge Interpretationen und einfallsreiche Anwendungen. Die Kernpraktiken mit »Finde Wege, um ...« einzuleiten, wie in Kapitel 2 vorgeschlagen, ist ein gutes Vorgehen, um Menschen dazu zu bringen, kreativer über die adressierbaren, organisatorischen Probleme nachzudenken.

Versuchen wir das mit dem Projektportfolio. *Beginne mit dem, was du gerade tust* und

- Finde Wege, um das Projektportfolio visuell zu organisieren. Schneide nach Geschäftsinitiative, Kunde, Team; in der Granularität ganzer Projekte, Releases, Features oder Experimente.
- Finde Wege, um die Menge paralleler Arbeit zu begrenzen, von der Teamebene aufwärts und von der Portfolioebene abwärts. Begrenze die Batchgröße nach Features, Zeit und Geld. *Stop starting and start finishing!*
- Finde Wege, um das Portfolio auf gleichmäßigen Arbeitsfluss und Pünktlichkeit hin zu managen. Verstehe die Verzögerungskosten aktiver und zukünftiger Arbeit und entwickle die Disziplin, entsprechend zu priorisieren. Manage die Produktlinien und Dienste von Anfang bis Ende und stelle entsprechend Budget und Mitarbeiter zur Verfügung.
- Finde Wege, um das Entscheidungsrahmenwerk über das Portfolio zu erfassen, zu teilen, zu übertragen und weiter zu entwickeln.
- Finde Wege, wie Entscheider, Teams und Kunden partizipieren können, sodass Signale erkannt werden und entsprechend gehandelt werden kann.
- Finde Wege, um verschiedene Teile der Organisation zusammenzubringen, damit sie häufige Probleme gemeinsam angehen. Teile Erfolge und bau auf ihnen auf.

Portfolio-Kanban ist die Anwendung dieses Denkens und nicht nur die Verwendung eines Klebezettels pro Projekt. Organisationen unterscheiden sich stark voneinander und Veränderungen der Regeln des Portfoliomagements benötigen häufig Wochen oder Monate, um ihren Weg durch das System zu bahnen. Deswegen ist jede angewandte Praktik zu jedem Zeitpunkt hoch kontextspezifisch.

Teil III dieses Buches ist auf das Portfoliomangement anwendbar, aber nicht nur dort. Ich kann keine anderen Bücher empfehlen, aber Sie finden in meinem Blog [Burrows o.J.] und in denen von Pawel Brodzinski [Brodzinski o.J.] und Ian Carroll [Carroll o.J.] relevante Materialien. Details erfahren Sie im Literaturverzeichnis im Anhang dieses Buches.

16.4.3 Scrumban

Scrumban ist ein Name, der von Cory Ladas dafür geprägt wurde, wenn das, *was du gerade tust*, Scrum ist und man Kanban anwendet.

Ich muss noch einmal auf die Warnungen aus Kapitel 13 hinweisen: Kanban bedeutet nicht, dass Sie alle agilen Vorgehensweisen wegwerfen; es ist vielmehr ein transformativer Prozess, der Zeit, Umsicht, Vorsicht und Kooperation benötigt.

Dieser Ablauf ist typisch:

- Das Team, das schon einen gewissen Grad an Visualisierung praktiziert, organisiert seine Arbeit entsprechend deren Fertigstellung. Das geht über »Code fertig«, »demonstrierbar« oder »potenziell auslieferbar« hinaus, um Akzeptanztests, Auslieferung und Validierung durch den Kunden abzudecken.
- Standup-Meetings werden mehr und mehr um das Board herum organisiert.
- Das Team limitiert den Work in Progress bereits durch den Sprint-Mechanismus und schenkt der begonnenen, aber noch nicht fertiggestellten Menge der Arbeit die meiste Aufmerksamkeit. Als Resultat sehen sie, dass die Arbeitspakete schneller fertiggestellt werden. Sofort oder nachdem sie erkennen, dass das Board gut funktioniert, könnten explizite WIP-Limits eingeführt werden.
- Nachdem Arbeit früher und sichtbarer fertiggestellt wird, befasst man sich mehr mit den späteren Prozessschritten. Dinge, die einer kontinuierlichen Lieferung (*Continuous Delivery*) im Weg stehen, werden adressiert. Der Charakter des Sprints verändert sich dadurch, dass Releases unabhängig davon geplant werden (falls überhaupt Planung erforderlich ist).
- Nachdem die Releases von der Sprint-Planung entkoppelt wurden, lässt das System nun einfach verschiedene Arbeitstypen und Geschwindigkeiten zu. Das Team befasst sich mit den Verzögerungskosten der einzelnen Einheiten und der Gesamtzusammenstellung. Veränderungen während des Sprints werden viel leichter handhabbar; eventuell werden Serviceklassen angeboten.
- Der Rhythmus der Sprint-Planung setzt sich fort, aber das Meeting selbst wird einfacher. Die richtige Menge an Arbeit für den Sprint abzuschätzen, wird weniger wichtig; es reicht, wenn sichergestellt ist, dass ausreichend Arbeit mit ausreichend hohem Wert und Qualität vorhanden ist und dass die risikoreichsten Einheiten identifiziert und dort heruntergebrochen wurden, wo es nötig ist.
- Indem sichtbar gemacht wurde, dass eine Validierung durch den Kunden erforderlich ist, tauchen neue Feedbackzyklen auf.

Verschiedene Einstellungen gegenüber den Scrum-Praktiken und -Rollen werden natürlich zu unterschiedlichen Ergebnissen führen. Es ist nicht unüblich, dass Teams die Art von Änderungen vollziehen, die hier beschrieben wurden, und sich trotzdem noch mit Scrum identifizieren. Das ist unsererseits völlig in Ordnung – es ist »Kanban mit« und nicht »Kanban gegen«.

Das Team, mit dem ich momentan als Interims-Entwicklungsleiter arbeite, ist schon weit vorangeschritten. Der Planungsrhythmus ist immer noch da und ich habe es auch nicht sonderlich eilig, dass er verschwindet. Das Highlight meiner zwei Wochen ist das »Show and Tell«, das Sprint-Review. Das ist ein lebhaftes Meeting, in dem das Entwicklungsteam zahlenmäßig übertroffen wird durch Kundenrepräsentanten und Beobachter von außerhalb (als »digitaler Leuchtturm«, eines von wenigen Pionierprojekten, das für die Online-Dienstleistung von britischen Regierungsbehörden für Bürger zuständig ist, bekommen wir häufig Besucher aus anderen Abteilungen und Behörden, die daran interessiert sind, wie wir arbeiten). Wir überprüfen nicht nur den Fortschritt und zeigen, was wir kürzlich gebaut haben, wir sehen uns auch häufig passende Videos dazu an, wie unsere Endnutzer mit dem Livesystem oder den Prototypen interagieren. Diese sind extrem motivierend – manchmal sogar bewegend – und der Effekt wird dadurch gesteigert, dass wir die Erfahrung gemeinsam machen.

16.5 Unterstützende Konzepte und Werkzeuge

Diese Konzepte und Werkzeuge könnte man – oder auch nicht – als Erste-Klasse-Komponenten der Kanban-Methode ansehen (ich tue es), auf jeden Fall sind sie wichtig und hilfreich genug, um sie hier vorzustellen.

Themen, die an anderer Stelle in diesem Buch behandelt werden:

- Die neun Werte (Kap. 1–9)
- Die drei Agenden (Kap. 10)
- Die Kanban-Linse (Kap. 10)

Vollständige Verweise auf die folgenden Kanban-Komponenten finden sich im Literaturverzeichnis:

- Die *Kanban-Katas*, die ursprünglich von Håkan Forss [Håkan 2012] stammen und durch Mike Rother's *Toyota Kata* [Rother 2009] inspiriert wurden. Sie geben uns einige Muster für Schlüssel-Feedbackzyklen.
- Das Kanban-Tiefen-Assessment-Tool (auch bekannt als »Wie tief ist Ihr Kanban?«), das Produkt des Kanban Leadership Retreat von 2012 [Anderson 2012a]. Meine Beziehung zu diesem Werkzeug ist etwas kompliziert:
 - Ich war im Raum, als es kreiert wurde (die Retreats sind eine fantastische Quelle für so etwas). Wir versuchten gerade, Håkans Frage »Sind die Praktiken in der richtigen Reihenfolge« und Davids Frage »Machen wir Kanban oder nicht?« zu beantworten. Wir kamen zu dem Schluss, dass »Wie tief?« eine viel bessere Frage ist.
 - Die Werte aus Teil I sind teilweise durch mein Bedauern, dass wir uns auf Kosten der Grundprinzipien auf Kanbans Kernpraktiken fokussiert hatten, entstanden⁴.

- Wenn ich meine philosophischen Bedenken zur Seite wische, finde ich das Werkzeug in der Anwendung sehr hilfreich. Eine alternative, wertbasierte Version ist in Kapitel 23 beschrieben.
- Kanban *Flight Levels*, die ursprünglich von Klaus Leopold stammen [Leopold 2014]. Sie beschreiben Muster der Skalierung und nicht der Implementierungstiefe. Zusammen mit den Werten und der Kanban-Linse hat Klaus' Modell die Entstehung der Agenden ausgelöst.

16.6 Implementierungshilfe: **STATIK**

Methoden kommen häufig ohne viel Hilfestellung, wie sie initial eingeführt werden sollten. Sie denken vielleicht, dass die »*Beginne mit dem, was du gerade tust*«-Methode ganz gut ohne Implementierungshilfe zurechtkommt, aber tatsächlich verfolgen wir den »Ansatz des Systemdenkens, um Kanban einzuführen«. Der englische Name *Systems Thinking Approach to Introducing Kanban* hat ein schönes Akronym, **STATIK**, das wir in Zukunft verwenden werden.

Die Erweiterung des ersten Grundprinzips in Kapitel 7 übernimmt einiges von **STATIK**:

GP1 (erweitert): Beginne mit dem, was du gerade tust, und verstehe,

- den Zweck des Systems,
- wie es dem Kunden dient,
- wie es für diejenigen innerhalb des Systems funktioniert,
- wie es Kunden unzufrieden macht und Mitarbeiter frustriert,
- wie es sicher verändert werden kann.

Ich glaube, dass **STATIK** wirklich als eine Kernkomponente der Kanban-Methode betrachtet werden sollte – mit der gleichen Wichtigkeit wie jede andere in den vorherigen Kapiteln – und ich bedauere, dass es noch nicht bekannter ist. Ich hoffe, dass Teil III dieses Buches etwas dagegen tut, indem es den Missstand adressiert.

Auf einem hohen Abstraktionsniveau lauten die sechs Schritte von **STATIK**:

1. Verstehe die Quellen der Unzufriedenheit.
 2. Analysiere Anforderungen und Fähigkeiten.
 3. Modelliere den Workflow.
 4. Entdecke Serviceklassen.
 5. Designe Kanban-Systeme.
 6. Führe es ein.
-
4. Das erinnert mich an Keith Sawyers Buch *Group Genius: The Creative Power of Collaboration* [Sawyer 2008], in dem er sagt, dass jegliche kreative Arbeit kooperativ ist, selbst wenn sie eine Einzelleistung zu sein scheint. Es verwundert nicht, dass unsere global verteilte Community massiv in Treffen von Angesicht zu Angesicht investiert. Ich danke meinem Freund Markus Andreazak für die Leseempfehlung.

Dies ist die grundlegende Variante, die in einer normalen zweitägigen Schulung gelehrt wird. Aber genau wie POOGI-0 (Kap. 12) den POOGI mit einem Schritt zur Vorbereitung verankert, beinhaltet *STATIK-0*:

0. Verstehe den Zweck des Systems.

Mit oder ohne diesen Schritt null ist STATIK ein gutes Mittel, um Veränderungen auch nach der initialen Einführung einen Rahmen zu geben. Es verstärkt die Disziplin, Veränderungen auf sinnvollen Bedürfnissen statt auf hauptsächlich technischen Überlegungen (Boarddesign, WIP-Limits usw.) zu stützen.

Wir beginnen gerade damit, *Reverse STATIK* zu dokumentieren. Das ist ein Arbeitsname für eine Verbesserungstechnik, die bei Schritt fünf beginnt (mit dem aktuellen Design der Kanban-Systeme) und rückwärts durch die vorherigen Schritte geht, bis das Bedürfnis nach Veränderung identifiziert wurde. Diese Veränderung wird dann auf konventionellem Weg vorwärts bis Schritt fünf oder sechs gebracht⁵.

5. Ich habe kürzlich »Reverse STATIK« in einem Blogpost im DJAA-Blog beschrieben:
»Reinvigorating an existing Kanban implementation with STATIK.«
<http://www.djaa.com/reinvigorating-existing-kanban-implementation-statik>.

17 Kleinere Modelle

Dieses Kapitel beschreibt einige Modelle, die die Konzepte aus Teil I unterstützen und einen hilfreichen Hintergrund für Teil III darstellen:

- *Littles Gesetz*, eine wunderschöne, einfache Formel mit einer hübschen visuellen Interpretation (und die ist auch eine gute Entschuldigung, um zum Cumulative Flow Diagramm zurückzukommen)
- Das *Satir-Modell für Veränderung*, eine mächtige Beschreibung der Veränderungsprozesse, die von Virginia Satir aufgestellt wurde
- Zwei Coaching-Modelle, das sehr nützliche Denkwerkzeug *GROW* und *Toyotas A3* (das bereits in Kap. 14 genannt wurde)
- Über einen Abstecher in das *Pyramidenprinzip* zu Jeff Andersons *Lean Change Canvas* [Anderson 2013]
- Verschiedene Modelle für die Prozessbegleitung, inklusive einiger Spiele
- Zwei Modelle für Führung und Kooperation, *T-förmige Führung* und *Triaden*

17.1 Zwei, die davongekommen sind

17.1.1 Littles Gesetz

Für einige ist es sicher überraschend, dass erst jetzt die wirkliche Einführung des Gesetzes von Little kommt. Warum habe ich es in Kapitel 2 (**Balance**) nur als Fußnote genannt? Und in Kapitel 5 (**Arbeitsfluss**) überhaupt nicht?

Es ist nicht so, dass ich Angst vor Mathematik habe (ich habe einen Abschluss darin). Noch mache ich mir Sorgen, dass es zu schwer zu erklären sei (ist es nicht). Ich bin nur skeptisch, wenn jemand behauptet, dass er durch wissenschaftliche Prinzipien bewiesen hat, dass seine Methode ohne Zweifel die beste ist, und ich werde nicht den gleichen Fehler machen. Ich lasse die Mathematik lieber Ihnen dabei helfen, das zu verdichten, was Sie bereits wissen.

Normalerweise schreibt man Littles Gesetz so auf:

$$L = \lambda W$$

Dabei gilt:

- L ist die langfristig durchschnittliche Anzahl von Arbeitseinheiten (die in der Warteschlangentheorie »Kunden« genannt werden) in einem *stabilen System* (wobei die strikte Definition dessen hier nicht notwendig ist).
- λ ist die langfristig durchschnittliche Ankunftsrate von Arbeitseinheiten
- W ist die durchschnittliche Zeit, die Arbeitseinheiten im System verbleiben.

Wir ersetzen die Symbole gerne mit Wörtern und verwenden den Strich über dem Wort, um den Durchschnitt kenntlich zu machen.

$$\overline{WIP} = \overline{\text{Lieferrate}} \times \overline{\text{Durchlaufzeit}}$$

Wenn wir die »Lieferrate« durch »Ankunftsrate« austauschen, werden wir daran erinnert, dass alle Arbeit, die in schwarzen Löchern verschwindet oder die sich aus dem Nichts materialisiert (Wissensarbeit kann manchmal so sein), berücksichtigt werden muss. Wir lassen den Strich über dem WIP auch manchmal mit der Begründung weg, dass in einem WIP-limitierten System der Work in Progress effektiv konstant ist (dieses Manöver erzeugt ein wenig Unwohlsein bei meinem inneren Mathematiker, aber man kann damit schon etwas klarmachen).

Ich wünschte, Littles Gesetz wäre das eine Stück der Warteschlangentheorie, das jeder Manager kennen würde. Wenn Sie nur eine Sache von diesem Gesetz behalten, dann sollte es die sein, dass die drei Quantitäten eng miteinander verbunden sind. Es gibt kein Entkommen vor der Mathematik: Man kann WIP, Durchlaufzeit oder Lieferrate absolut nicht ändern, ohne dass nicht in mindestens einer der beiden anderen eine entsprechende Veränderung passiert. Das sind gute Nachrichten und es erklärt, wie viele Systeminterventionen tatsächlich funktionieren.

Die Formel wird in Präsentationen zu Kanban typischerweise so arrangiert:

$$\overline{\text{Durchlaufzeit}} = \frac{\overline{WIP}}{\overline{\text{Lieferrate}}}$$

Wenn wir zum Beispiel die Durchlaufzeiten reduzieren wollen, hilft Littles Gesetz, die drei Angriffsstrategien zu organisieren:

1. Wir können den direkten Weg nehmen und Verzögerungsquellen entfernen. Das könnte im Lean-Stil über das Entfernen nicht wertbeitragender Aktivitäten passieren oder im Agile-Stil, indem wir Übergaben durch etwas Kooperativeres ersetzen. Littles Gesetz besagt, dass zu beobachten sein müsste, dass entweder WIP sinkt oder die Lieferrate steigt. Die Erfahrung zeigt, dass wir wahrscheinlich beides sehen.
2. Wir können einen indirekten Weg nehmen und Work in Progress limitieren. Solange der begleitende Verlust der Lieferrate nicht zu groß ist (das aggressive Absenken von WIP bringt dieses Risiko mit sich, es ist aber auch mög-

lich, dass wir stattdessen einen doppelten Nutzen erzielen), sagt Littles Gesetz, dass die durchschnittliche Durchlaufzeit sinken muss.

3. Wir können den Weg nehmen, den wir vielleicht Engpasstheorie-Route nennen. Wir steigern die Lieferrate, indem wir Personen oder andere Fähigkeiten hinzufügen oder Dinge wie Nacharbeit und Fehlerlast entfernen, die unproduktiv Kapazität fressen. Littles Gesetz erinnert uns daran, dass wir diese Dinge so implementieren müssen, dass die zusätzliche Kapazität nicht sofort durch eine entsprechende Steigerung des WIP absorbiert wird (und das ist manchmal einfacher gesagt als getan).

Arrangieren wir die Formel ein letztes Mal, sodass die Lieferrate im Fokus steht:

$$\overline{\text{Lieferrate}} = \frac{\overline{\text{WIP}}}{\overline{\text{Durchlaufzeit}}}$$

Weil die Lieferrate nach Definition eine Quantität über die Zeit ist, genau wie die rechte Seite dieser Gleichung, hat Littles Gesetz eine nette geometrische Interpretation. Dazu überlagern wir das Cumulative Fluss Diagramm (CFD), das wir zuerst in Kapitel 1 betrachtet haben, mit einigen Dreiecken (Abb. 17–1).

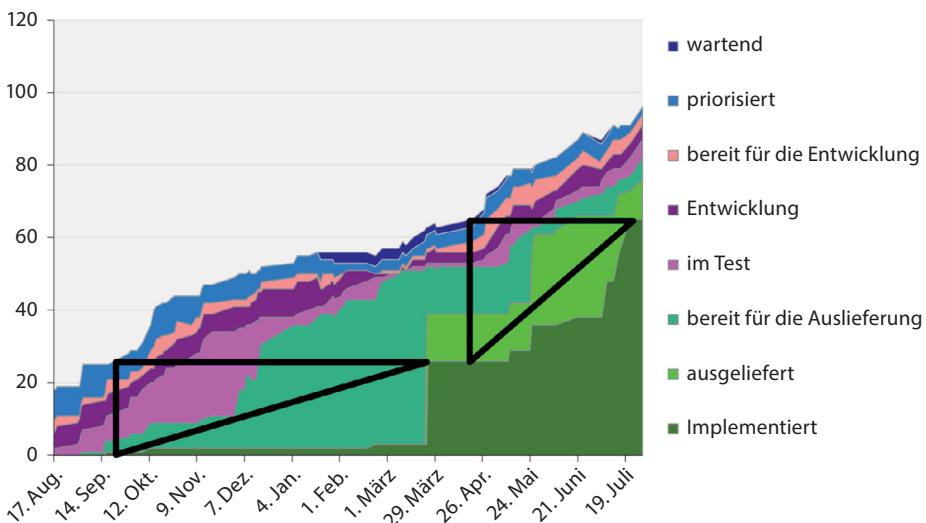


Abb. 17–1 Eine geometrische Interpretation des Gesetzes von Little

Aus Übersichtlichkeitsgründen habe ich den »Vorgeschlagen«-Status (»Proposed«) aus dieser Version des Diagramms entfernt, das den Prozess von »Priorisiert« bis zu »Implementiert« visualisiert. Keine Angst, ich mochte nicht: Littles Gesetz kann auf jeden Teil des Prozesses angewendet werden, solange wir genau die Ankunft und das Verlassen der Arbeitseinheiten definieren und keine Dinge zwischendurch verlorengehen.

Die vertikalen Linien der Dreiecke sind Messungen des WIP zu einem bestimmten Zeitpunkt. Die steigende Seite (die Hypotenuse) ist die Lieferrate und die horizontale Seite gibt einen Hinweis darauf, wie viel Zeit benötigt wurde, um diese Menge an WIP in einen gelieferten Zustand zu überführen. Dies sind nicht die langfristigen Durchschnittswerte des Gesetzes und das CFD gibt keine Auskunft, welche Dinge wann geliefert wurden, aber es sind trotzdem nützliche visuelle Orientierungshilfen.

Man sieht, dass das rechte Dreieck eine stärkere Steigung und eine kürzere horizontale Seite hat als das linke. Höherer Durchsatz und kürzere Durchlaufzeiten: Das ist wirklicher Fortschritt! Ich hätte hier gerne auch weniger WIP gesehen, aber man erkennt, dass ein großer Teil des WIP in den »Bereit für die Auslieferung«- und »Implementiert«-Spalten liegt und dass der WIP der vorhergehenden Spalten unter Kontrolle war. Wie in Kapitel 4 (**Kundenfokus**) beschrieben, wussten wir, dass wir in diesen späteren Zuständen Probleme hatten. Wir hatten auch schon Schritte unternommen, um die darunterliegenden Ursachen zu adressieren.

17.1.2 Das Satir-Modell für Veränderung

In Kapitel 7 wurde die J-Kurve eingeführt. Eine besondere Analyse des Effektes der J-Kurve ist der Verdienst von Virginia Satir, einer anerkannten Familientherapeutin und Autorin, die mit Gerald M. Weinberg zusammenarbeitete. Weinberg hat uns allen einen guten Dienst erwiesen, indem er das Modell von Satir in die Softwareindustrie einführte.

Statt es einfach nur hinzunehmen, dass Veränderung Dinge häufig schlechter macht, bevor sie besser werden (das haben die meisten von uns wahrscheinlich schon mal erfahren), hilft Satirs Modell uns dabei, über die psychologischen und soziologischen Auswirkungen der Veränderung nachzudenken.

Satir beschreibt Veränderung in fünf Schritten wie in Abbildung 17-2 gezeigt.

1. Der *späte Status quo*, die Zeit vor der (unerwarteten) Einführung eines *fremden Elements*.
2. Ein Zeitraum des *Widerstands*, das fremde Element wurde eingeführt, aber wir versuchen, am Status quo festzuhalten. Unsere gesteigerte Aufmerksamkeit kann die Leistungsfähigkeit vielleicht steigern, es ist aber nicht nachhaltig.
3. Wir begeben uns ins *Chaos*; wir gehen nicht angemessen mit dem fremden Element um und der Status quo ist nicht länger haltbar.
4. *Integration*: Der Groschen fällt, die Teile fügen sich plötzlich zusammen, eine *transformierende Idee* erlaubt es, dass das fremde Element angenommen wird.
5. Wir haben einen *neuen Status quo*, wenn dieser Integrationsprozess abgeschlossen wurde.



Abb. 17-2 Das Satir-Modell für Veränderung

Natürlich sind nicht alle fremden Elemente so, dass sie einen positiven Effekt erzielen können, und der neue Status quo ist nicht notwendigerweise besser als der alte. Diesbezüglich ist die Idee der Zustände in Satirs Beschreibungsmodell realistischer und hilfreicher als die J-Kurve.

17.2 Denkwerkzeuge und Coaching-Modelle

Diese Modelle unterstützen die Art von Mentorkonversationen, die in Kapitel 8 (**Vereinbarung**) beschrieben wurden. Sie sind auch als Denkwerkzeuge sehr nützlich – als Wege, um unsere Gedanken zu organisieren, zu testen und zu verfeinern, bevor wir sie mit anderen teilen.

17.2.1 GROW

Ich bin zum ersten Mal durch John Whitmores großartiges Buch *Coaching for Performance* [Whitmore 2010] mit dem GROW-Modell in Berührung gekommen. GROW ist eine simple Strukturierung, die eine Coaching-Konversation, geschrieben oder gesprochen, steuern kann:

■ Goal (Ziel)

- Was möchtest du, dass passiert?¹
- Wie wird »fertig« aussehen?
- (Vertiefung in das genannte Ziel) Welche Treiber existieren dafür?

1. Danke, Bob Marshall, für diese Referenz.

■ Reality (Realität)

- Was passiert im Moment? Wie interpretierst du die verfügbaren Daten?
- Wie sind die Auswirkungen? Wie könnte man das quantifizieren?
- Kannst du wahrscheinliche Ursachen ausmachen?

■ Options (Optionen)

- Was könntest du tun? Was solltest du tun?
- Wie sind ähnliche Dinge in der Vergangenheit erreicht worden?
- Wer könnte helfen?

■ Will or Way Forward (Willen oder Weg voran)

- Was wirst du tun? Was kommt zuerst? Warum?
- Auf wen hat das Auswirkungen? Gibt es andere Hürden?
- Wie adressiert dein Plan die Ursachen? Wo trifft er seine Ziele?

Jetzt suche ich in anderen Modellen nach dieser Struktur, inklusive des nächsten Modells A3.

17.2.2 A3

Wie in Kapitel 14 angemerkt, strukturiert Toyota seine veränderungsbezogenen Konversationen zwischen Mentor und Mentee rund um ein A3, einen Vorschlag, der so begrenzt wird, dass er auf ein einzelnes Blatt Papier der Größe DIN A3² passt. Es gibt keine direkte Vorlage für ein A3 selbst, aber Vorschläge enthalten typischerweise die folgenden Elemente:

- Etwas Kontext, vielleicht die Visualisierung des aktuellen Prozesses, vielleicht eine qualitative oder quantitative Analyse der aktuellen *Bedingung*
- Eine Aussage darüber, wie Dinge sein sollten – die *Zielbedingung*
- Eine Liste möglicher *Gegenmaßnahmen*, Ideen, die die Dinge, die wir an der aktuellen Bedingung nicht mögen, abmildern und die das System zur (oder zumindest in die Richtung der) Zielbedingung führen
- Ein Plan, der eine Übersicht darüber gibt, wie die gewählte Gegenmaßnahme (oder Gegenmaßnahmen) implementiert werden wird

A3 und GROW korrespondieren sehr stark miteinander, wobei das G und R bei A3 umgedreht sind. So wie bei GROW liegt der Fokus auf Konversation; man sollte erwarten, dass ein A3 sorgsame Überprüfung und deutliche Überarbeitung als Resultat des Mentoring-Prozesses durchläuft.

Es ist wichtig, »Gegenmaßnahme« statt »Lösung« zu verwenden; es wird als sehr wichtig angesehen, dass mehrere Gegenmaßnahmen in Betracht gezogen werden. Wird das weggelassen, ist das ein Indikator dafür, dass der Mentee vielleicht auf eine bestimmte Lösung fixiert ist, möglicherweise auf Kosten einer tat-

2. DIN-A3-Papier ist 297mm × 420mm groß.

sächlichen Lösung des Problems. Es hilft auch gegen eine Form des faulen Denkens, bei dem das Problem als die Abwesenheit einer präferierten Lösung ausgedrückt wird.

17.2.3 Exkurs: Wie ein Profi überprüfen!

Ich habe in meinem Leben schon viele Projektvorschläge gelesen, manche nicht größer als ein A3-Blatt (einseitig oder zwei A4, eine Handvoll PowerPoint-Folien vielleicht), manche viel größer. Unabhängig von ihrem Umfang waren die frühen Versionen der meisten ziemlich schrecklich.

Das ist so deprimierend, weil das Überprüfen und die Nachbearbeitung erlernbare Fähigkeiten sind. Es ist sogar so erlernbar, dass die Entschuldigung »Niemand kann effektiv seine eigene Arbeit kritisieren« ziemlich lahm ist.

Wenn Sie regelmäßig Vorschläge oder andere strukturierte Dokumente lesen oder schreiben, sollten Sie *Das Pyramidenprinzip* von Barbara Minto lesen. Sie werden eine Liste mit Aufzählungszeichen nie wieder im selben Licht sehen! Ich kann nichts strukturiert Geschriebenes ansehen, ohne instinktiv die MECE-Überprüfungen durchzuführen (MECE steht für »Mutually Exclusive, Collectively Exhaustive«³) und zu überprüfen, ob es so organisiert und zusammengefasst ist, dass die maximale Wirkung erzielt wird.

Ich suche in Vorschlägen und Anträgen nach:

- Einer Kontextbeschreibung, einer Beschreibung, was eigentlich verändert wird
- Warum dieser Antrag wichtig ist – Ziele
- Für wen wir das tun, die Kunden sollten dabei klar identifiziert werden – alle und nicht nur ihre Stellvertreter (Product Owner, Projektmanager und Ähnliche)
- Einer Aussage über die Kundenbedürfnisse, die befriedigt werden sollen
- Einer Vielzahl von Optionen (der Schutz gegen Festlegung und faules Denken)
- Einer kurzen Darstellung der gewählten Lösung

Der Trick besteht darin, diese Teile einzeln zu prüfen (und je deutlicher die Gliederung des Vorgestellten ist, desto besser können Mintos Ansätze verwendet werden) und danach wieder zusammenzuführen. Bestehen Sie darauf, dass

- der Kontext irgendwie berücksichtigt wird,
- die Bedürfnisse aus den verschiedenen Kundenperspektiven real sind,
- die Optionen die Bedürfnisse sinnvoll befriedigen, sodass es auch für die Kunden akzeptabel ist,
- die gewählte Option (oder Menge von Optionen) das Gesamtziel erfüllt,
- der Plan das liefert, was benötigt wird, und auch wirklich durchführbar ist.

3. Anmerkung der Übersetzer: »Mutually Exclusive, Collectively Exhaustive« steht für »Gegenseitig ausschließend, insgesamt erschöpfend«.

Falls Anträge und Vorschläge kurz gehalten werden können – und Toyota scheint zu beweisen, dass das möglich ist –, können diese Überprüfungen in wenigen Minuten erfolgen.

17.2.4 Der Lean Change Canvas

Jeff Anderson hat eine Veränderungsmethode entwickelt, die um einen *Lean Change Canvas* (Abb. 17–3) herum gebaut ist. Diese Visualisierung enthält alle Elemente, die ich in einem Antrag oder einem Vorschlag zu sehen erwarte, und eventuell noch andere Dinge darüber hinaus. Statt ein Dokument zu schreiben oder ein A3 zu zeichnen, kann eine Leinwand effektiv dazu verwendet werden, Beiträge in Klebezettelform zu organisieren.

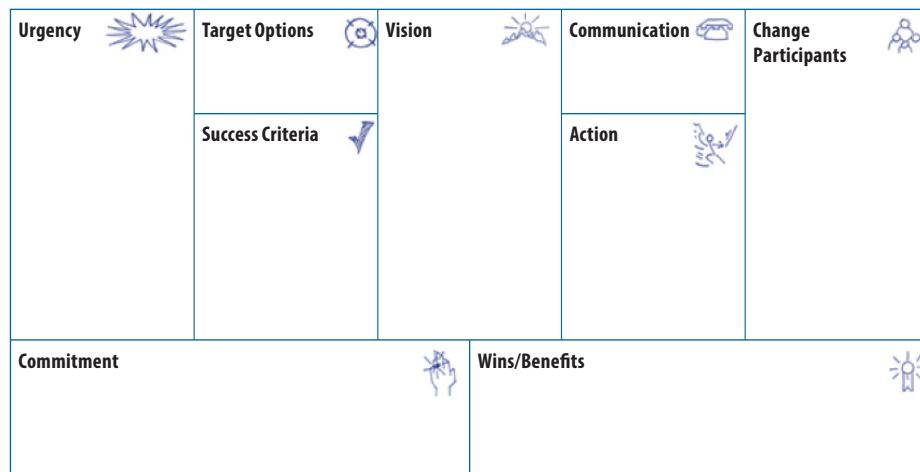


Abb. 17–3 Ein Lean Change Canvas (Dank-an-agileconsulting.blogspot.com)

Aktionen können dann auf ein Hilfs-Kanban-Board zur Ausführung übertragen werden. Die Methode ist in Jeffs Buch *The Lean Change Method* [Anderson 2013] beschrieben.

17.3 Gruppenmoderation und Spiele

17.3.1 Kaners Moderationsmodell

In *Facilitator's Guide to Participatory Decision-Making* [Kaner 2007] beschreibt Samuel Kaner die Rolle des Moderators im Prozess mit dem Modell, das in Abbildung 17–4 gezeigt wird.

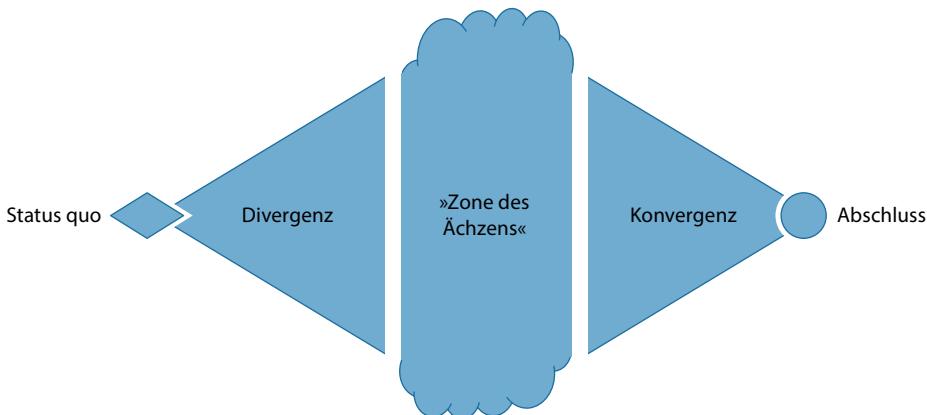


Abb. 17-4 Das Kaner-Modell für Gruppenmoderation

In diesem Modell begleitet der Moderator eine Gruppe von Menschen durch drei Stufen:

1. Die *Divergenz* stellt sicher, dass ein reichlicher Vorrat an Rohmaterialien vorhanden ist – durch die Generierung von Ideen, die Identifikation von Problemen usw.
2. Dann kommt der unbequeme Teil, die »Zone des Ächzens« (»Groan Zone«), in der das Rohmaterial gesiebt und analysiert wird. Nachdem die Divergenzphase voller Energie und Kreativität war, liegt das Unangenehme in dieser Stufe darin, dass ein klarer Weg vorwärts fehlt.
3. Schlussendlich kommt die *Konvergenz*, in der die Gedanken besser organisiert werden und man sich auf Ergebnisse einigen kann.

Ein fähiger Moderator schafft den sicheren Raum, der die Diversität von Beiträgen in der Divergenzphase fördert, unterstützt die Gruppe in den unangenehmen Momenten (und vermeidet vorsichtig die *verfrühte Konvergenz*, die einen bedeutenden Fortschritt vermeidet) und verwendet angemessene Übungen und Werkzeuge, um die Konversation in Richtung wertvoller Ergebnisse zu lenken.

Auf Basis dieses Modells beschreibt Jean Tabaka in *Collaboration Explained, Facilitation Skills for Software Project Leaders* [Tabaka 2006] eine Menge Meetingformate und Werkzeuge für Kooperation, die in Entwicklungsteams anwendbar sind. Es ist ein toller Leitfaden für die Dynamik von Meetings und anderen kooperativen Zusammentreffen und enthält viele Ratschläge für diejenigen, die sich vielleicht in einer moderierenden Rolle wiederfinden.

17.3.2 Serious Games

Jane McGonigal beschreibt in ihrem bahnbrechenden Buch *Reality is Broken: Why Games Make Us Better and How They Can Change the World* vier grundlegende Eigenschaften von Spielen:

- Ein klares Ziel
- Eine definierte Menge Regeln (bekannte Grenzen)
- Feedback (Festhalten des Spielstands)
- Optionale Teilnahme

Es ist verblüffend, bis zu welchem Grad diese Eigenschaften viel von dem beschreiben (oder beschreiben würden, wenn die Dinge nur besser wären), was innerhalb von Unternehmen und Organisationen passiert. Daniel Mezick legt in *The Culture Game* [Mezick 2012] nahe, dass Organisationen einfach zu verändern sind, wenn man sich ansieht, was sie entsprechend diesen Regeln durchführen. Mezick verwendet dieses Modell, um sowohl agile Methoden und Praktiken zu erklären als auch um einen sozialen Prozess organisatorischer Einführung zu beschreiben, den er *Tribal Learning* nennt.

Der generische Ausdruck *Serious Games* bezeichnet die bewusste Verwendung von Spielen (von denen viele McGonigals Kriterien entsprechen), um einen ernsten Zweck zu erreichen. Die Agile-Community hat einige davon aufgegriffen und viele weitere geschaffen (auf <http://tastycupcakes.org> finden sich viele Anregungen).

Ein einfaches und sehr nützliches Spiel ist *Punkt-Wahl (dot-voting)*, das in Teil III wieder auftaucht (ich verwende es sehr häufig). Das Spiel verwendet die uralte Idee der Mehr-Stimmen-Wahl, die einfach ein Wahlprozess ist, in dem die Teilnehmer jeweils eine vorher bestimmte Menge Stimmen haben, die sie für verschiedene Dinge vergeben können – Features, Verbesserungen, Diskussionsthemen usw. –, die geordnet werden müssen. In der heutigen Zeit repräsentieren Klebezettel diese Dinge und die Stimmen werden durch das Aufkleben von Punkten abgegeben.

In *Innovation Games* kombiniert mein Freund Luke Hohmann die *Serious Games* mit einem Moderationsmodell – *Ideas Into Action™* –, das auf dem Kaner-Modell aufbaut [Hohmann 2006]. Luke beschreibt Spiele, die sehr gut auf dem Papier funktionieren und die sich auch sehr effektiv ins Internet übersetzen lassen. Einige Spiele lassen sich deutlich besser mit elektronischer Unterstützung spielen. Ich ziehe beispielsweise die Onlineversion des Portfoliopriorisierungsspiels *Buy a Feature* der analogen Variante vor. Bei anderen Spielen ist der Spielbarkeitsunterschied marginal, aber der Moderator profitiert davon, die Chat-Verläufe analysieren zu können, Berichte über die Ergebnisse generieren zu können usw.

Zusammen mit Lukes Firma Conteneo, Inc. haben wir ein *Kanban-Knowsy*-Spiel gebaut, das in Abbildung 17–5 zu sehen ist. In Knowsy-Spielen erraten die Teilnehmer die Toplisten der anderen. Diese Listen sind gerankte Listen von fünf

Dingen, die aus einer größeren Menge Auswahlmöglichkeiten gewählt wurden. In der Kanban-Version erraten die Mitspieler die Toplisten der anderen zu den Prinzipien, Praktiken und (am beliebtesten) den Werten der Kanban-Methode. Dadurch entdecken sie, was dem Team insgesamt wichtig ist, wie gut sich das Team untereinander kennt und wie gut sie in dieselbe Richtung orientiert sind.



Abb. 17–5 Kanban Knowsy

Ein Spiel, das sowohl auf dem Papier als auch online gut funktioniert, ist das *Schnellboot* (siehe Abb. 17–6). In Teil III empfehle ich es Gruppen, damit sie ihre Hindernisse visualisieren. Hier kann ohne eine explizite Wahl eine Gewichtung vorgenommen werden (die vertikale Tiefe der »Anker« wird verwendet, um darzustellen, wie schwerwiegend das Hindernis ist), außerdem ist es möglich, sie horizontal anzuordnen, typischerweise werden damit die unterschiedlichen Phasen des Workflows repräsentiert.

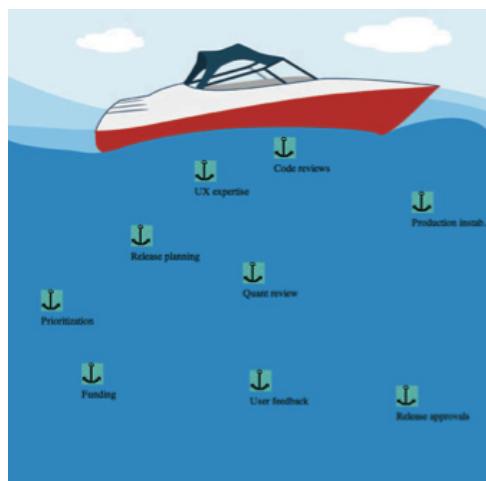


Abb. 17–6 Schnellboot (online)

17.4 Modelle für kooperative Führung: Triaden und T-Formen

Die *Triade* ist ein sehr einfaches Modell für **Kooperation** und kooperative **Führung**, das schon an erstaunlich vielen Orten bewusst verwendet worden ist. Dank *Tribal Leadership: Leveraging Natural Groups to Build a Thriving Organization*, einem Buch von Dave Logan, John King und Halee Fischer-Wright [Logan, King & Fischer-Wright 2008], verstehen wir auch, wie wir das Modell im Berufs- und gesellschaftlichen Leben verwenden können. Einige Kirchen haben Triaden in der Form von *Gebets-Drillingen* (meine Frau Sharon ist Mitglied in mehreren gewesen); sogar der KGB hat dieses Modell angewendet!

Eine Triade verbindet drei Personen, die über einen gemeinsamen Zweck verbunden sind. Manchmal entsteht eine Triade dadurch, dass eine Person zwei vorher unbekannte Personen sich gegenseitig vorstellt; manchmal werden sie gebildet, um eine bestimmte Aufgabe zu erledigen. Effektive Triaden folgen zwei Regeln:

1. Jedes Mitglied ist zu einem gewissen Grad verantwortlich für die Beziehung zwischen den anderen beiden Mitgliedern, er oder sie stärkt die Beziehung.
2. Wachstum entsteht nicht dadurch, dass Triaden zu Vierergruppen erweitert werden, sondern dadurch, dass zusätzliche Triaden gebildet werden, zu denen ein oder zwei Mitglieder bestehender Triaden gehören. Es entstehen also Netzwerke.

Ich bin eine Person, die einer »Networking-Veranstaltung« mit Furcht entgegenguckt, und das Triaden-Modell ist ungefähr die einzige Form des Netzwerkens, die für mich gut funktioniert. Ich habe mir zum Ziel gesetzt, Personen gegenseitig vorzustellen, von denen ich weiß, dass sie gemeinsame Interessen haben. Das ist allein schon lohnend, aber ich erlebe häufig doppelten und dreifachen Nutzen über eine ertragreiche Kooperation oder weiteres gegenseitiges Vorstellen.

Triaden drücken kooperative Führung aus, wenn sie bewusst dazu verwendet werden, Wissen zu teilen, Möglichkeiten zu schaffen und Brücken zwischen verschiedenen Teilen einer Organisation zu bauen. Ich habe Mitarbeiter, die wir kurz nach ihrem Abschluss rekrutiert haben, dazu ermutigt, langfristig stabile Triaden zu bilden und sich gegenseitig aus diesen Triaden heraus zu helfen, die eigenen Netzwerke zu erweitern. Ich habe sie aber auch schon verwendet, um kurzfristig spezifische Probleme zu lösen.

Morten Hansen beschreibt das *T-förmige Management* [Hansen 2009], das den *T-förmigen Mitarbeitern* zumindest ein wenig ähnelt, die ich im Vorwort des Buches angesprochen habe. Seine T-förmigen Manager fördern Kooperation auf zwei sehr unterschiedliche Arten:

1. Sehr stark wie in Kapitel 3 beschrieben, also enge Zusammenarbeit innerhalb ihres Teils der Organisation.
2. Die Nachteile der Kooperation adressierend, die am Ende von Kapitel 3 genannt werden, d.h. »disziplinierte« Kooperation über die Grenzen hinweg zum Rest der Organisation.

Der Schlüssel zu Hansens Modell besteht darin, dass die zweite Art der Kooperation zweckorientiert und effektiv sein muss; es geht nicht um das Netzwerken an sich und es wird erwartet, dass Resultate geliefert werden, die in einem gesunden Verhältnis zum investierten Aufwand stehen. Undisziplinierte Kooperation könnte schlimmer sein als überhaupt keine Kooperation.

Beide Modelle sind komplett kompatibel mit Kanbans Art von Führungsverhalten *auf jeder Ebene*. Triaden müssen überhaupt keine organisatorischen Grenzen respektieren und die T-Formen können überall hervorgehen. Wir alle können es.

Teil III

Implementierung

Teil III ist nach dem *Systems Thinking Approach to Introducing Kanban*, oder abgekürzt *STATIK*, gestaltet. David Anderson räumte dieser Herangehensweise einen doppelten Slot auf der Lean Software & Systems Konferenz 2012 in Boston ein¹. Sie bildet das Rückgrat der meisten Grundlagentrainings zu Kanban.

Die Kapitel dazu lauten:

18. Quellen der Unzufriedenheit erkennen
 - Interne und externe Frustrationen aus mehreren Perspektiven aufnehmen
 - Quellen von Variabilität identifizieren
19. Anforderungen und Leistungsfähigkeit analysieren
 - Arbeitstypen identifizieren, Muster der Anforderungen
 - Beschreiben und versuchen zu quantifizieren, wie (und wie gut) die Anforderungen erfüllt werden
20. Den Workflow modellieren
 - Den Prozess der Wissenserarbeitung visualisieren und überarbeiten
21. Serviceklassen finden
 - Im Rahmen der Leistungsfähigkeit des Systems das Dienstleistungsangebot den Kundenerwartungen anpassen
22. Kanban-Systeme gestalten
 - Visualisierung, Work-in-Progress-Limits und Prozessregeln

1. Ein Video von Davids Vortrag ist hier zu finden <http://vimeo.com/46272041>.

23. Ein Kanban-System einführen

- Planung und Formen
- Veränderung durch das System ziehen

Es ist ziemlich irreführend, diesen Prozess so zu präsentieren, als passierte er strikt sequenziell. STATIK neigt dazu, hochgradig iterativ und explorativ zu sein, und das soll auch so sein. Dadurch helfen wir nicht nur dem Kanban-System zu einer glatten Einführung, wir modellieren und beschleunigen auch die evolutive Veränderung der Kanban-Methode.

Da STATIK iterativ ausgelegt ist, können wir es auf mindestens zwei Arten verwenden:

1. Eine neue Kanban-Initiative starten
2. Eine bestehende Implementierung überarbeiten und aktualisieren

STATIK hilft dabei, die Kanban-Implementierung (wieder) in Verbindung mit den Bedürfnissen der Organisation zu setzen. Mit Anspielung auf die *Agenden für Veränderung* aus Kapitel 10 bezeichne ich dies als das *Formen der Agenda*. Es ist einer der befriedigendsten Aspekte meiner Arbeit innerhalb von Organisationen. Häufig korrigiert dies einen vorherigen Fehler – es ist zu einfach, die Benutzung eines Kanban-Boards als Praktik zu kopieren, ohne dabei die größere Bedeutung hinter der Praktik zu verstehen.

Egal, ob man sich dafür entscheidet, einen externen Experten einzuladen oder nicht, man sollte STATIK zu einer Gruppenaktivität machen, statt es in einer Soloaktivität durchzuziehen. STATIK erfordert, dass man mehrere Perspektiven erfasst, und es ist viel besser, diese in einem Raum repräsentiert zu haben, als sie zu erraten. Es alleine zu versuchen, wäre mindestens eine verschwendete Möglichkeit, potenziell sogar schädlich. Wie sonst sollte man ohne **Kooperation** **Verständnis** aufbauen und **Vereinbarung** erreichen? Wo ist der **Respekt**?

18 Quellen der Unzufriedenheit erkennen

Jede Art von gewollter Veränderung benötigt zwei wesentliche Informationen:

1. Den Umfang – eine Grenze um das, *was wir gerade tun* und auf das die Veränderung ausgerichtet ist – das »Was« der Veränderung
2. Ihr Ziel – ein Ausdruck dessen, was wir hoffen, durch die Veränderung zu erreichen, relativ zur aktuellen Situation – das »Warum« der Veränderung

Am Anfang ist es unwahrscheinlich, dass beide Dinge genauer bekannt sind. Das sollte Ihnen nicht übermäßig Sorgen bereiten – es ist viel besser, mit einer Untersuchung des Problemraums zu beginnen, als zu versuchen, verfrüht zu Lösungen zu kommen. Und wir sollten realistisch sein: Umfang und Ziele werden häufig mehr durch das bestimmt, was die Menschen für organisatorisch umsetzbar halten, als durch das, was notwendig ist. Als Change Agents finden wir das häufig sehr frustrierend, aber es ist in Ordnung: Beginnen wir mit dem Möglichen; das Unmögliche können wir dann später noch machen.

Wir beginnen mit den Quellen der Unzufriedenheit, weil sie uns sehr schnell zu etwas viel Positiverem führen: einer Menge von Dingen, die Menschen erreichen möchten. Wenn wir uns die Mühe machen, wirklich herauszufinden, warum diese Unzufriedenheiten ein Problem für Menschen darstellen, werden wichtige Punkte wie der Umfang und die Unterstützung für die Initiative häufig viel klarer.

Das ist jetzt nicht der Zeitpunkt, sich abzuschotten. Sie werden mit Menschen sprechen müssen! Seien Sie bereit, vor die Tür zu treten und sie zu treffen oder sie hereinzuholen; machen Sie sie zum Teil der Konversation.

18.1 Zwei Perspektiven

Selbst mit einer nur groben Idee über den Umfang können wir ziemlich einfach zwei unterschiedliche Perspektiven identifizieren:

1. Die Perspektive derer, die im System arbeiten – ihr Verständnis des Systems aus erster Hand und ihr Eindruck über die Außenwahrnehmung des Systems

2. Die Perspektive derer außerhalb des Systems (Kunden, höheres Management, Dienstleister für verwandte Dienste) – wie es ihnen hilft, ihre allgemeinen Bedürfnisse zu befriedigen (und was diese sein könnten), und ihr Eindruck, wie ihre direkten Bedürfnisse bedient werden

Weil wir die Menschen dazu bringen, sowohl nach innen als auch nach außen zu sehen, ist es kein Desaster, wenn wir die Grenzen des Umfangs falsch gesetzt haben. Überall, wo wir diese Grenze setzen, lernen wir viel dadurch, dass wir diese beiden Perspektiven abgleichen und die wichtigen Unterschiede berücksichtigen. Meinungen zur Position und Art der Grenze können auch wichtige Dinge aufdecken.

18.2 Zwei Fragen

Vieles von dem, was wir in diesem Kapitel erreichen wollen, kann auf zwei Schlüsselfragen reduziert werden:

1. Aus deiner persönlichen Perspektive und von dem, was du von anderen innerhalb oder außerhalb wahrnimmst, **was sind die Hauptquellen der Unzufriedenheit mit dem System?** In anderen Worten: Welche (und wessen) Bedürfnisse werden nicht erfüllt?
2. **Welche Quellen von Variabilität und fehlender Vorhersagbarkeit würdest du hervorheben?** In anderen Worten: Was frustriert dich und das ganze System bei deinem Versuch, wertvolle Dinge qualitativ hochwertig und zeitnah zu liefern?

Beide Fragen drehen sich um Bedürfnisse und die aktuelle Fähigkeit des Systems, diese zu befriedigen. Die Fragen in solche Worte zu fassen, bietet uns eine gute Möglichkeit, einige Treiber für Veränderungen zu ermitteln und die Veränderung zu etwas Persönlichem zu machen.

Beachten Sie, wie die zweite Frage, die hauptsächlich (aber nicht exklusiv) auf interne Teilnehmer abzielt, offensichtlich in Beziehung mit dem Wert **Arbeitsfluss** steht. Hier kann es hilfreich sein, eine kleine Pause einzulegen und darüber nachzudenken, welche anderen Werte diesen beiden Fragen zugrunde liegen. Vielleicht auch, um sie explizit zu machen. Für mich sind **Verständnis, Respekt und Kundenfokus** hier sehr naheliegend; in Ihrem speziellen Kontext sehen Sie vielleicht andere.

18.3 Formate

Meiner Erfahrung nach funktionieren die folgenden Formate gut, manchmal in Kombination miteinander:

- Hinausgehen und mit den Personen individuell sprechen und, wenn sich größere Zusammenhänge ergeben, immer weitere Runden drehen.
- Die Menschen in einem großen Workshop zusammenbringen, die Fragen direkt stellen und sie auch informell bearbeiten.
- In einem Workshop oder einer virtuellen Umgebung arbeiten und dabei durch unterstützende Übungen oder Spiele den Einzelnen und Gruppen dabei helfen, eine Menge unterschiedlicher Antworten zu finden und zu ordnen.

Jedes Format hat seine Vor- und Nachteile in Bezug auf die Menge der generierten Informationen, die Qualität der Exploration und den Umfang des geschaffenen, gemeinsamen Verständnisses. Persönliche und Organisationsvorlieben haben natürlich auch eine Auswirkung.

Meine Brot-und-Butter-Technik, die ich bereits in zahllosen Gruppentreffen verwendet habe, funktioniert so:

■ Ein Klebezettel pro Problem:

Die einzelnen Teilnehmer schreiben ihre Unzufriedenheiten auf. Ein kurzer Satz mit einem dicken Filzschreiber pro Klebezettel, jeder Klebezettel nennt (meistens anonym) eine einzelne Unzufriedenheit.

Variante: Dies in kleinen Diskussionsgruppen durchführen.

■ Sammeln und in Cluster zusammenfassen:

Die Klebezettel werden eingesammelt und gegliedert, entweder direkt beim Anbringen auf der Wand oder nachträglich. Duplikate werden übereinander gestapelt; andere eng verwandte Dinge werden zusammengefasst.

■ Die Cluster benennen:

Für jeden Cluster wird ein Name ausgewählt, der den Inhalt repräsentiert. Falls kein guter Name gefunden werden kann, denkt man noch einmal über die Zusammenfassung nach und versucht es noch einmal. Die Namen werden auf Klebezetteln mit einer sich absetzenden Farbe dargestellt.

■ Dot-Voting (optional):

Jeder bekommt eine festgelegte Anzahl Stimmen – typischerweise drei bis fünf –, die durch das Anbringen von Punkten auf einem Klebezettel vergeben werden. Man kann gemeinsam festlegen, ob diese Stimmen für ganze Cluster oder einzelne Probleme vergeben werden, und meistens ist es erlaubt, dass eine Person mehrere Stimmen für dasselbe Problem abgibt. Die Probleme werden dann nach der Anzahl der Stimmen in eine Rangliste sortiert.

- Das *Schnellboot*-Spiel aus Kapitel 17 ist eine gute Alternative. Statt die Unzufriedenheiten durch die Punktwahl nach Wichtigkeit zu ordnen, müssen

die Teilnehmer in Gruppen von fünf bis acht Personen zusammenarbeiten, um durch die vertikale Platzierung der »Anker« die relativen Auswirkungen zu visualisieren. Daraus können Gespräche entstehen, die die Gruppe vielleicht zusammenfassend wiedergeben möchte, wenn sie ihre Ergebnisse präsentiert. Auch eine horizontale Organisation der Anker ist möglich, entweder wird das vordefiniert oder durch die Gruppe entschieden.

Manchmal müssen verschiedene Techniken in Kombination eingesetzt werden. Denken Sie immer daran, dass interne Workshops, die zum Nutzen für einen einzelnen Bereich organisiert werden (Schulungen fallen typischerweise in diese Kategorie), zwangsläufig auf Wahrnehmungen zurückgreifen müssen, die im besten Fall aus zweiter Hand kommen und im schlimmsten Fall wilde Spekulation oder Wunschdenken sind. Sie sollten nicht ungeprüft bleiben.

Eine weitere Option ist es, sich auf die Meinung einiger Schlüsselpersonen zu verlassen. Aber egal, wie scharfsinnig oder einflussreich diese Menschen auch sind, es wird dadurch eine wichtige Möglichkeit verpasst, ein gemeinsames Verständnis aufzubauen. Ich rate von dieser Vorgehensweise ab, wenn sie nicht als Vorbereitung für eine größere Übung gedacht ist.

Machen Sie allerdings reichlich Gebrauch von unterstützenden Informationen, die vielleicht bereits existieren. In größeren Organisationen ist es häufig möglich, an Dinge wie z.B. Befragungen zum Arbeitsklima, Arbeit von externen Beratern aus jüngster Zeit oder Resultate aus Teamretrospektiven zu gelangen. Was auch immer Sie finden, stellen Sie sicher, dass Sie die richtige Breite von Perspektiven berücksichtigen; seien Sie vorsichtig mit frühzeitigen Schlussfolgerungen; und seien Sie bereit, tiefer einzusteigen, wo es notwendig ist.

18.4 Organisiere und erforsche

Ein guter Prozessbegleiter weiß, dass er diese informationsgenerierenden Übungen nicht zu früh auf eine zu kleine Auswahl von Ergebnissen konvergieren lassen darf. Irgendwann müssen Sie allerdings die bestehenden Ergebnisse gliedern und ein Gefühl für die Schlüsselthemen und Prioritäten vermitteln. Drücken Sie diese vorsichtig aus. Bleiben Sie bei der Sprache, die verwendet wurde, um die Unzufriedenheiten und Frustrationen zu beschreiben, und beschreiben Sie keine Lösungen.

Achten Sie auch auf Lösungen, die sich als Probleme verkleiden – »Unser Problem ist, dass wir X nicht haben« (wobei X eine Lösung ist). Als Prozessbegleiter ist es essenziell, dass Sie hinter diese Äußerungen blicken und zu den Unzufriedenheiten und Frustrationen vordringen, die die Menschen persönlich fühlen.

Es gibt zwei sehr wichtige Lösungen, die häufig als verkleidete Bedürfnisse daherkommen. Sie laden zu einer reflexartigen Antwort ein, die ich ganz sicher nicht empfehlen würde:

- »Wir benötigen besser definierte Prozesse.« (Sagen Sie hier nicht: »Ja, genau das machen wir mit Kanban.«)
- »Wir benötigen besser definierte Rollen.« (Sagen Sie hier nicht: »Ich würde wirklich nicht empfehlen, mit diesen zu starten«, und bitte verwenden Sie nicht Kapitel 9 als Begründung!)

Falls Sie sich selbst in der Gefahr sehen, so unbedacht oder nachlässig zu antworten, nehmen Sie den Hinweis ernst, dass der Problemraum tiefer erforscht werden sollte.

Einige aufrichtig empfundene Unzufriedenheiten sind es vielleicht wert, weiter erforscht zu werden, für den Fall, dass sich andere Sorgen dahinter verstecken. Einige bekannte Beispiele sind:

- Sorgen um Termine könnten Sorgen um Kommunikation, Koordination und Qualität verstecken.
- Sorgen um Personalbestand, Arbeitslast, Priorisierung und die gesamte Systemeffektivität neigen dazu, gegeneinander austauschbar zu sein.

Noch einmal: Seien Sie darauf bedacht, Bedürfnisse zu erforschen, ohne eine bestimmte Lösung anzunehmen. Diese Untersuchung zahlt sich aus, wenn die »Sorge hinter der Sorge« ein »Bedürfnis hinter dem Bedürfnis« hervorbringt. Vielleicht kommt das System der Bedürfnisbefriedigung aus dem einfachen Grund nicht nach, weil diese bis jetzt nie klar genug artikuliert wurden.

18.5 Folgemaßnahmen

Schlussendlich: Teilen der Ergebnisse, Feedback einholen und verfeinern. Wenn Sie diesen Teil von STATIK in einem Workshop durchführen, sollte vieles oder alles durch ein bewusstes Gestalten des Workshops an diesem Tag stattfinden. Andere Formate benötigen wahrscheinlich viel mehr Nachfolgeaktivitäten, vielleicht sogar mehrere Runden.

Was werden Sie erreicht haben?

- Geteiltes Verständnis und Vereinbarung bezüglich der Unzufriedenheiten. Das sind die Anfänge eines Plädoyers für die Veränderung, hinter der bereits eine breite Menge von Stakeholdern steht.
- Vermutlich entweder eine Art von Patenschaft oder zumindest eine gute Idee, wo diese zu suchen ist.

Und man hat jede Menge Rohdaten; die werden Sie benötigen.

19 Anforderungen und Leistungsfähigkeiten analysieren

Im vorherigen Kapitel ging es um Kontext und Wahrnehmung. Überlegungen zur Implementierung wurden nicht berücksichtigt. Dieses Kapitel handelt vom Sammeln einiger bestimmter qualitativer und quantitativer Fakten über den aktuellen Prozess, die das Design des Kanban-Systems beeinflussen.

Falls Sie das in Kooperation mit anderen machen können, werden Sie die Arbeit aus dem vorherigen Kapitel weiter optimieren. Falls das nicht möglich ist, müssen Sie Ihre Ergebnisse aus der Zuarbeit mit anderen teilen und von ihnen prüfen lassen.

Diese Analyse muss sowohl ein qualitatives als auch quantitatives Verständnis erzeugen.

■ Qualitativ:

- Das Verständnis der verschiedenen auftretenden Arbeitstypen hilft Ihnen, die verschiedenen Varianten des Arbeitsflusses zu identifizieren, die gesteuert werden müssen.
- Die Kenntnis der verschiedenen Quellen der Arbeit wird dabei helfen, sich auf die zu erwartende Aufgabenlast vorzubereiten, sie zu gestalten und zu bearbeiten.
- Das Verständnis dafür, warum Arbeit getan werden soll, wird Ihnen helfen, die verschiedenen Risikotypen zu verstehen, damit Sie angemessen mit ihnen umgehen können.

■ Quantitativ

- Die zu erwartende Menge der Arbeit zu kennen, hilft, eine verwendbare Granularitätsstufe auszuwählen, um die Arbeit zu visualisieren und zu überwachen.
- Durch Kenntnis der Lücke zwischen der aktuellen Lieferfähigkeit des Prozesses (gemessen in Durchsatz, Durchlaufzeiten, Vorhersagbarkeit etc.) und den Erwartungen der Kunden und der Organisation drumherum wird aufgezeigt, welche Arten von Verbesserungen notwendig sind.
- Quantitative Analysen der kürzlich fertiggestellten, der aktuell im Prozess befindlichen und der auf Abarbeitung wartenden Arbeit können uns viel-

leicht zeigen, wo Chancen für Verbesserungen am wahrscheinlichsten zu finden sind.

Wir folgen dem bekannten Muster und beginnen mit dem, was aus dem Prozess herauskommt, und arbeiten uns dann von dort aus rückwärts vor.

19.1 Wissen, was Sie wem liefern und warum

19.1.1 Was

Wenn ich Kanban zum ersten Mal in einer Organisation einführe, bitte ich manchmal meine Zuhörer darum aufzuschreiben, woran sie gerade arbeiten, und anschließend sammeln wir alles an einer Wand. Die Vielfältigkeit der Antworten ist enorm:

- »Projekte« oder »Programme« wie in »Wir liefern Projekte oder Programme«
- »Analyse« (oder der Name einer anderen Funktion oder Tätigkeit) mit der Bedeutung »Das ist das, was ich und meine direkten Kollegen machen«
- Beschreibungen von spezifischen, sehr detaillierten Aufgaben, die etwas zu einem bestimmten Liefergegenstand beitragen. Zum Beispiel »Währungstabelle erstellen« (oder auch einfach nur »Währungstabelle«), Dinge, die für sich selbst genommen nicht besonders viel Kundenwert schaffen
- Die Namen kundenspezifischer Arbeitspakete – Produkte, Features usw. – »GBP Swap curve«, »Monatsendbericht«, »Claims Service« zum Beispiel

Obwohl ich die Gespräche eher in Richtung auslieferbarer Einheiten und weg von Aktivitäten und einzelnen Aufgaben lenken würde, sind keine falschen Antworten dabei. Sie sagen uns alle etwas darüber, wie Arbeit organisiert ist, und über die unternehmensspezifische Sprache, in der sie beschrieben wird; und Arbeit organisieren ist das, was unser Kanban-Board tun wird. Tatsächlich können wir diese Klebezettel später für unser Boarddesign verwenden (vgl. Kap. 22).

Es schadet nichts, einen informellen Vorsprung bei der quantitativen Untersuchung zu haben. Nachdem wir identifiziert haben, was geliefert wird, können wir ungefähr herausfinden, wie groß die Einheiten sind, wie lange der Lieferprozess typischerweise dauert und in welcher Rate (pro Zeiteinheit) sie fertiggestellt werden.

Zwei Teams, zu denen ich gehörte, hätten ihr »Was« vielleicht so beschrieben:

1. »Wir unterstützen eine gewisse Anzahl kürzlich gebauter Anwendungen und arbeiten an zwei weiteren. Auf der Teamebene arbeiten wir hauptsächlich in Features (manchmal Bugfixes) mit dem Umfang von einigen Tagen Entwicklungszeit und einem Gesamtumfang von wenigen Tagen. Wir releases so häufig wie möglich in die Produktion, typischerweise mehrere Male pro Wo-

che, manchmal mehr als einmal pro Tag. Auf höherer Ebene denken wir in Businessinitiativen; wir denken immer weniger in Projekten.«

2. »Wir arbeiten an einem großen, global verteilten System mit einem global verteilten Team. Unsere Liefereinheiten werden »Deliverables« genannt und sie repräsentieren große Features, die nach Komponenten aufgeteilt sind und nach »Change Request« und »Region« organisiert sind. Wir rollen weltweit etwa alle sechs Wochen aus und geben nach Bedarf kleinere, regionale Releases zwischendurch heraus.«

19.1.2 Wem

»Wem« kann sich entweder auf eine Aktivität oder Funktion stromabwärts beziehen oder auch auf den Endkunden. Beide sind relevant, aber die zweiten sind meistens viel interessanter – Kunden geben einem am besten das »Warum«, zu dem wir gleich kommen werden.

In der Zwischenzeit sollten Sie sicherstellen, dass niemand das »Wem« mit »unter wessen Steuerung« verwechselt. Einigen Menschen fällt es schwer, sich von der Vorstellung zu lösen, dass sie Projekten oder Projektleitern oder Linienmanagern liefern; manchmal scheint diese Denkweise auch institutionalisiert zu sein.

Die bereits genannten Teams hätten ihr »Wem« folgendermaßen beschreiben können:

1. »Wir releasesen selbst in die Produktionsumgebung und führen auch unseren eigenen Support durch. Unsere direkten Kunden im Geschäftsfeld nutzen die Systeme, um Risikoanalysen auf dem Energiemarkt durchzuführen, Handelshinweise zu generieren und den Kunden des Unternehmens auf täglicher Basis Berichte zu liefern.«
2. »Serverseitig stellen wir unseren regionalen Supportteams installierbare Applikationen zur Verfügung. Die Teams führen das Produktionsdeployment selbst durch (wir stehen zur Verfügung, falls wir beraten müssen). Auf der Clientseite werden die Desktop-Applikationen nach Bedarf aktualisiert (das ist größtenteils unsichtbar für die meisten Benutzer). In den meisten Ländern, in denen wir vertreten sind, haben wir Nutzer im Vertrieb, Handel, Betrieb, dem Risikocontrolling und dem Finanzcontrolling. Vertriebsmitarbeiter und Händler haben Kunden und Ansprechpartner überall auf der Welt; wir sind mit vielen von ihnen über elektronische Handelsplattformen und andere Netzwerke verbunden.«

Häufig gibt es mehrere »Wem«. In diesem Fall ist es wertvoll herauszufinden, ob sich der Workflow nach Kunden unterscheidet.

19.1.3 Warum

Wir haben bereits viel gelernt, aber das »Was« und das »Wem« sind eigentlich nur die Vorbereitung für die große Frage: »Warum?«

Beginnen wir diesmal mit möglichen Antworten von zwei Teams, die ich früher geführt habe:

1. »Unsere zahlenden Kunden verbrauchen Energie (Strom, Gas und Öl) in solchen großen Mengen, dass es sich für sie lohnt, ihre Risiken abzusichern, indem sie entweder einige Monate im Voraus kaufen oder durch den Kauf von Derivat-Verträgen. Unsere Werkzeuge helfen ihnen, die richtigen Mengen zur richtigen Zeit zu kaufen, zeitnahe Informationen sind also sehr wichtig. Wir arbeiten intensiv daran, sowohl unsere Marktdeckung zu verbreitern (Energiemarkte sind sehr fragmentiert), als auch daran, hinsichtlich der Ausgereiftheit und der Aktualität vor unserer Konkurrenz zu bleiben.«
2. »Der Bond-Handel besteht nur aus Risiken: Zinsrisiko, Kreditratenrisiko, Gegenparteirisiko usw. Man investiert in das Risiko und sichert es ab. Dabei geht es um schwindelerregende Geldmengen! Kein Handelsteilnehmer kann es sich leisten, lange vom Markt abwesend zu sein; besonders dann nicht, wenn wichtige Ankündigungen oder Angebote platziert werden. Unsere Nutzer benötigen eine schnelle und zuverlässige Plattform, die mit der immer weiter wachsenden Geschäftsvielfalt und dem Geschäftsvolumen umzugehen weiß, die die wichtigsten Institutionen abwickeln.«

Ich habe also in zwei Branchen gearbeitet, die sehr explizit auf Risiko fokussiert sind, und diese beiden Aussagen über das »Warum« sagen oder implizieren sehr viel über die Qualitäten – Aktualität, Zuverlässigkeit, Kapazität usw. –, die in den jeweiligen Produkten und Prozessen benötigt werden.

Für andere Teams in anderen Branchen ist es vielleicht deutlich schwieriger, zu einem nützlichen »Warum« zu kommen. Es lohnt sich aber – insbesondere dann, wenn ein Sinn für den Zweck in irgendeiner Weise fehlt.

Ausgerüstet mit einem Sinn für das »Warum« auf höchster Ebene können wir uns repräsentative Stichproben einzelner Lieferungen ansehen. Warum benötigt der Kunde sie? Welche Auswirkungen hat eine schnellere oder langsamere Lieferung auf sie? Können wir mit unserem Prozess und unserer Produktstrategie den Bedürfnissen der Kunden effektiv genug begegnen?

19.2 Quantitative Analyse

Wie detailliert Sie die quantitative Analyse benötigen, hängt von Ihrer spezifischen Situation ab; alles, was Sie bis jetzt getan haben, sollte Ihnen einen guten Eindruck darüber geben, welche Art von Analyse voraussichtlich wichtig werden wird.

Lassen Sie sich nicht durch das offensichtliche Fehlen von Daten abschrecken; Sie sollten weder diesen Schritt übergehen, noch sollten Sie jetzt mit einem aufwendigen Projekt beginnen, um diese Daten zu sammeln. Es sollte nicht lange dauern, um zumindest eine grobe Abschätzung für folgende Dinge zu bekommen:

- Wie häufig liefern wir?
- Wie viele Arbeitspakete sind in jeder Lieferung enthalten?
- Wie lange dauern Auslieferungen typischerweise von Anfang bis Ende? Wie variabel ist dies? Wie vorhersagbar?
- Wie sieht die typische Altersstruktur der enthaltenen Einheiten in einer typischen Lieferung aus? Man kann das Alter sowohl relativ zum ursprünglichen Anfragedatum der Einheiten messen als auch zu dem Zeitpunkt, an dem sie zum ersten Mal zugesagt wurden.
- Wie viele Arbeitspakete sind aktuell in Bearbeitung und wie sieht die Altersstruktur aus?
- Wie viele Pakete müssen erst noch begonnen werden und wie ist hier die Altersstruktur? Werden sie als *Backlog* betrachtet (sie warten auf Bearbeitung) oder als etwas Flexibleres (Ideen, Optionen etc., aus denen ausgewählt werden kann)?
- Wird die Situation schlechter (längere Zeiträume, mehr parallele Arbeit, weniger Durchsatz, mehr Fehler, unglücklichere Kunden) oder besser? Kann man das quantifizieren?

Wenn möglich, visualisieren Sie! Verlaufsdiagramme, wie in Abbildung 19–1 abgebildet, funktionieren sehr gut für Quantitäten, in denen Variabilität zwischen den Einheiten oder Stichproben auftritt, selbst wenn die Menge der genommenen Proben jeden Tag schwanken kann.

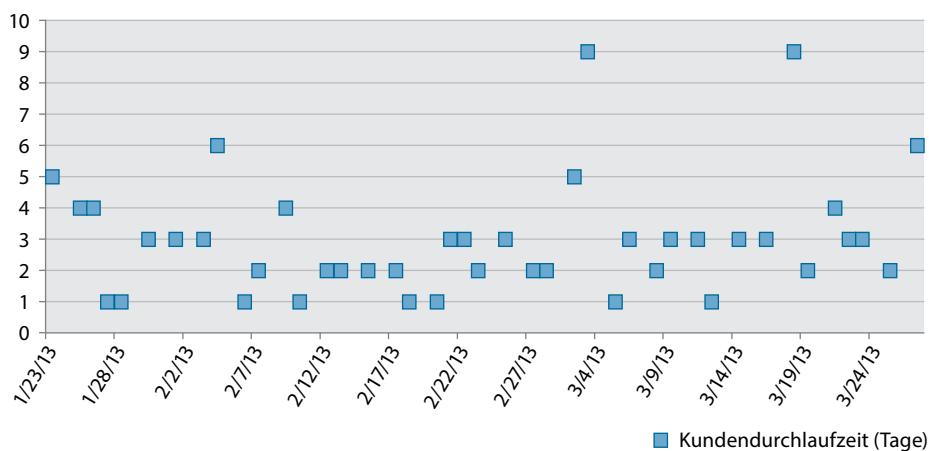


Abb. 19–1 Ein Verlaufsdiagramm (Run-Chart), das die Kundendurchlaufzeit zeigt

Histogramme fassen die Daten zusammen und geben ihre Verteilung wieder. Abbildung 19–2 basiert auf denselben Kundendurchlaufzeiten wie Abbildung 19–1. Es könnte vielleicht wichtig sein, dass dieses Diagramm zwei Spitzen aufweist (eine bei Tag 1, die andere bei Tag 3); vielleicht teilen sich die Daten naturgemäß in zwei individuelle Populationen auf.

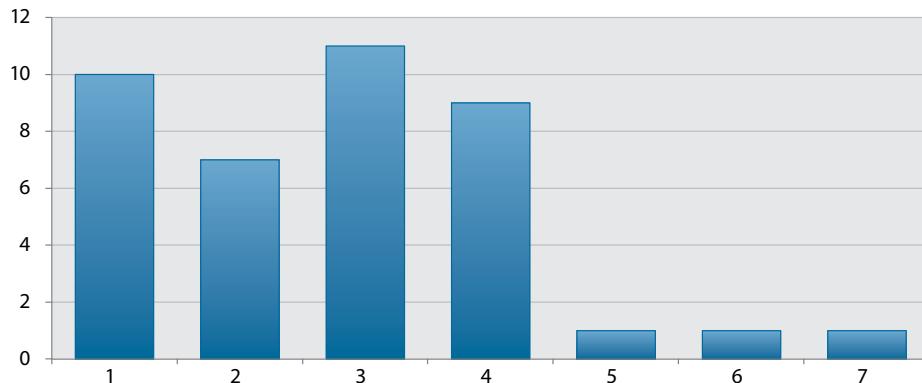


Abb. 19–2 Ein Histogramm mit denselben Durchlaufzeiten wie in Abbildung 19–1

Paretdiagramme gliedern kategorisierte Daten, so wie zum Beispiel Arbeitseinheiten nach Kundenart oder Fehler nach der Grundursache. Sie sind nützlich, wenn wir vor unserer Analyse priorisieren wollen. In Abbildung 19–3 sind Kunde A und Kunde B für mehr als 80 % der Arbeit verantwortlich; sich auf die Kunden C, D und E zu konzentrieren, wäre vielleicht nicht besonders ergebnisreich zu diesem Augenblick. Sie könnten versucht sein, diese Daten in einem Tortendiagramm zu zeigen, aber das würde die Rangfolge nicht annähernd so klar darstellen.

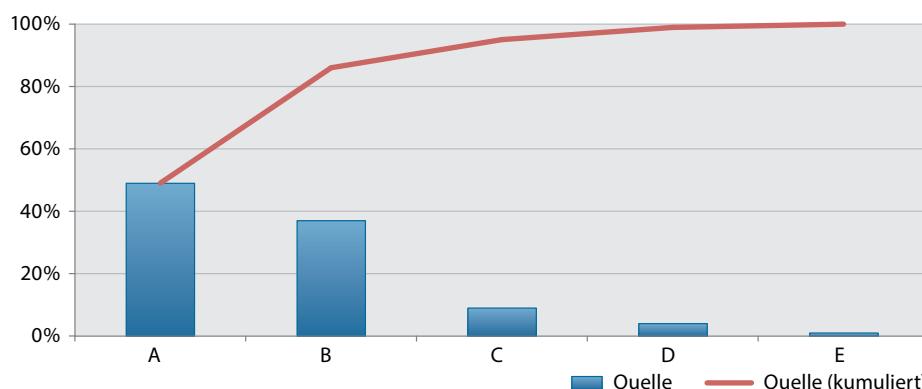


Abb. 19–3 Ein Paretdiagramm

Sogar in dieser frühen Phase kann ein Cumulative Flow Diagram (siehe Kap. 1 und 17) vielleicht infrage kommen. Der vordere Teil des CFD in Abbildung 19–4

wurde aus einem SharePoint-Auszug extrahiert, den ich bekommen konnte. Ich war überglücklich darüber, ihn zu finden – ich wäre sogar bereit gewesen, mich durch die Meetingprotokolle mehrerer Monate zu wühlen, wäre das notwendig gewesen. Am Ende hat es mich nicht mehr als ein paar Stunden Arbeit gekostet, um die Daten zu sammeln und eine fünfzigfache Steigerung des Durchsatzes zu visualisieren. Ganz sicher war das den Aufwand wert!

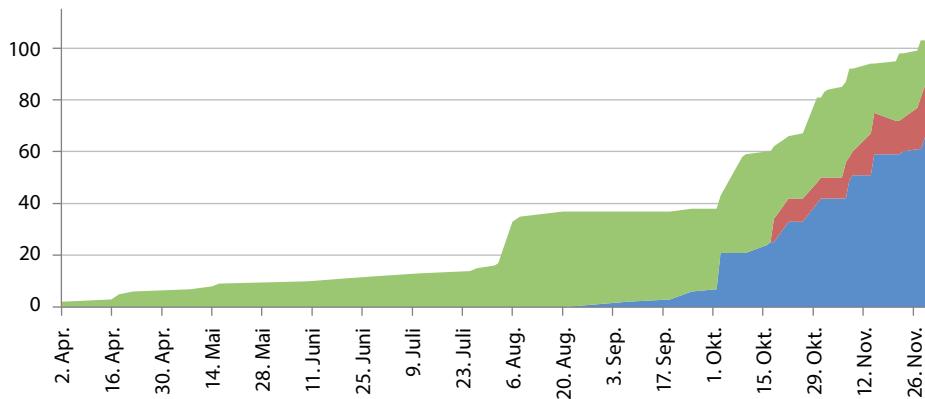


Abb. 19-4 Ein CFD aus verschiedenen Quellen gesammelt

19.2.1 Flusseffizienz

Eine Metrik, die einen großen Einfluss haben kann, ist die *Flusseffizienz*: das prozentual ausgedrückte Verhältnis der *Bearbeitungszeit* zur Gesamtdurchlaufzeit zwischen definierten Start- und Zielpunkten. Die Bearbeitungszeit ist die Gesamtzeit, in der aktiv an einer Arbeitseinheit gearbeitet wird, im Gegensatz zu folgenden Situationen:

- Sie befindet sich in einem explizit inaktiven Status wie das Warten in einer Warteschlange oder in einer Blockade (und die Einheit wartet auf die Behebung der Probleme).
- Sie befindet sich in einem angeblich aktiven Status, aber es wird nicht aktiv daran gearbeitet, weil die verantwortlichen Personen anders beansprucht sind (sie könnten an einer anderen Arbeitseinheit arbeiten oder auch etwas ganz anderes machen).

Die Bearbeitungszeit und die unterschiedlichen Arten von Wartezeiten sollten zusammen die Gesamtdurchlaufzeit ergeben. Denken Sie an die Qualifizierung, wenn Sie Auskunft über Durchlaufzeiten geben, sodass die Start- und Endpunkte eindeutig definiert sind.

Es ist nicht immer zulässig, zwischen unterschiedlichen Prozessen die Ist-Zahlen zu vergleichen (diese Metrik ist zum Beispiel sensibel in Bezug auf die Granu-

larität der Arbeitseinheiten, was wenig hilfreich ist), aber seien Sie nicht überrascht, wenn Sie Flusseffizienzen finden, die deutlich unterhalb von 100 % liegen, eventuell sogar am einsteligen Ende der Skala.

Eine niedrige Flusseffizienz sagt Ihnen, dass durch das Entfernen von Wartezeiten aus dem Prozess viel erreicht werden kann. Berechnen Sie die Flusseffizienz sobald wie möglich, damit Sie später eine Basis für Vergleiche haben.

Diese Definition der Flusseffizienz bewertet jegliche Bearbeitungszeit gleich. Man könnte sich auch dazu entschließen, zwischen Aktivitäten zu unterscheiden, die klar zur Wissensentdeckung gehören, und solchen, die das nicht tun.

19.3 Wie Arbeit ankommt

Nachdem wir uns angeschaut haben, was aus dem Prozess herauskommt, und einige Informationen darüber gesammelt haben, was wirklich darin passiert, sind wir nun bereit, uns anzusehen, wie Arbeit ankommt.

Für jede Art von Arbeit stellen sich folgende Fragen:

- Wie kommt sie an? Bittet man aktiv darum, wartet man auf das Ankommen, oder gibt es etwas dazwischen – Gespräche, Erkundungen oder Verhandlungen?
- In welcher Rate kommt sie an (so viele pro Tag, pro Woche oder pro Monat)?
- Kommt sie regelmäßig an (beispielsweise durch ein angesetztes Meeting) oder zufällig (beispielsweise durch Anrufe beim Helpdesk)?
- Kommt sie in größeren Batches oder Stößen an? Gibt es vorhersagbare saisonale Schwankungen?
- Über welche Schlüsselmetriken beurteilen Sie Erfolg? Wie gut passen die zu dem, was die Kunden geliefert bekommen?

Meine früheren Teams würden diese Fragen vielleicht so beantworten:

1. »Es gibt kein regelmäßiges Muster, wie Arbeit ankommt. Unsere größeren Projekte wurden durch Beschluss des Managementteams gestartet; der kleinere Kram kommt durch private Unterhaltungen oder über unsere gemeinsame Inbox. Wir zielen darauf ab, Supportanfragen innerhalb von 24 Stunden zu lösen (und unsere Leistung ist nach dieser Messung sehr gut); die Erwartungen bezüglich der Entwicklungsarbeit variieren entsprechend der Größe und wie dringend sie ist.«
2. »Neue Arbeit wird regional gesteuert, typischerweise durch regelmäßige Treffen der regionalen Manager oder der Businessanalysten. Größere Brocken Arbeit werden in zweiwöchentlichen globalen Telefonkonferenzen (eine für jeden Hauptgeschäftszweig) zusammen mit dem Managementteam und unseren Ansprechpartnern auf der Geschäftsseite nachverfolgt. Die Arbeit wird nach Releases geplant; manchmal rutscht etwas aus einem Release in das nächste, aber das geschieht nie ohne Warnung und Diskussion.«

19.4 Passt alles zusammen?

Nach dem Durchlesen dieses und des vorherigen Kapitels haben Sie wahrscheinlich ein gutes Gefühl für folgende Aspekte:

- Was produziert wird, in welchen Größen und wie häufig
- Was beauftragt wird, wie, wann und warum
- Wie viel Arbeit zwischendurch liegen bleibt und wie lange sie das tut
- Die Unzufriedenheiten derer außerhalb des Prozesses
- Die Frustration derer innerhalb des Prozesses

Was sagt Ihnen das persönlich? Besteht in Ihrem Team bereits ein gemeinsames Verständnis darüber, was sich ändern muss? Oder gibt es noch Dinge, die nicht zusammenpassen? Was machen Sie mit diesen Dingen?

Machen Sie sich keine Sorgen, wenn ein paar Unklarheiten, Unstimmigkeiten oder Lücken im Wissen bestehen – die Dinge werden viel klarer werden, wenn die Workshops und die Analysen abgeschlossen sind und damit begonnen wird, die Arbeit wirklich zu visualisieren. Das Wichtige ist, dass die Bühne vorbereitet wurde.

20 Den Workflow modellieren

Dieses Kapitel stellt drei verschiedene Herangehensweisen vor, um den Workflow zu modellieren, den unser Kanban-System unterstützen wird:

1. Grob aufzeichnen
2. Top-down-Dekomposition
3. Bottom-up-Organisation

Es muss daran erinnert werden, dass unser Ziel ein funktionierendes Kanban-System ist und kein statisches Modell. Am besten ist es, nicht zu sehr an den Ergebnissen dieser Übung festzuhalten – sie werden schnell ihren Wert verlieren, wenn das System beginnt, sich weiter zu entwickeln.

Um das Ganze möglichst einfach zu halten, gehe ich nur von einem Hauptworkflow aus. Wenn Sie mehr als einen haben, können Sie entweder alle nacheinander modellieren oder einen als Vorlage verwenden und die anderen nach diesem modellieren.

20.1 Grob aufzeichnen

Wir empfehlen keine bestimmte Notation hierfür. Es bleibt Ihnen überlassen, ob Sie lieber einen formalen Businessprozess verwenden wollen oder die Notation der Wertstromanalyse (*value stream mapping*, VSM), die unternehmensübliche Notation zum Zeichnen von Prozessen beibehalten oder etwas gänzlich Informelles verwenden. Wählen Sie ein Vorgehen, das keinen unnötigen Widerstand erzeugt oder in das Sie nicht mehr Zeit als notwendig investieren müssen.

Sie können es sich leisten, über jedes Detail hinwegzugehen, das keinen wesentlichen Einfluss auf die Entscheidungen hat, die die Mitarbeiter bezüglich der Auswahl und der Umsetzung der Arbeit treffen. Dieser Hinweis bezieht sich insbesondere auf funktionale Grenzen und Rollen. Sollte ihr Vorhandensein Menschen davon abhalten, gute Entscheidungen zu treffen? In einer idealen Welt tun sie das nicht und es ist weder die Aufgabe unserer Entwürfe noch die unseres resultierenden Kanban-Systems, diese Grenzen zu verstärken.

Abbildung 20–1 beschreibt in einer nichtstandardisierten Kurzschreibweise den initialen Prozess der XIT-Studie (die Fallstudie aus Kap. 4 des »blauen Buches« von David Anderson [Anderson 2010]).

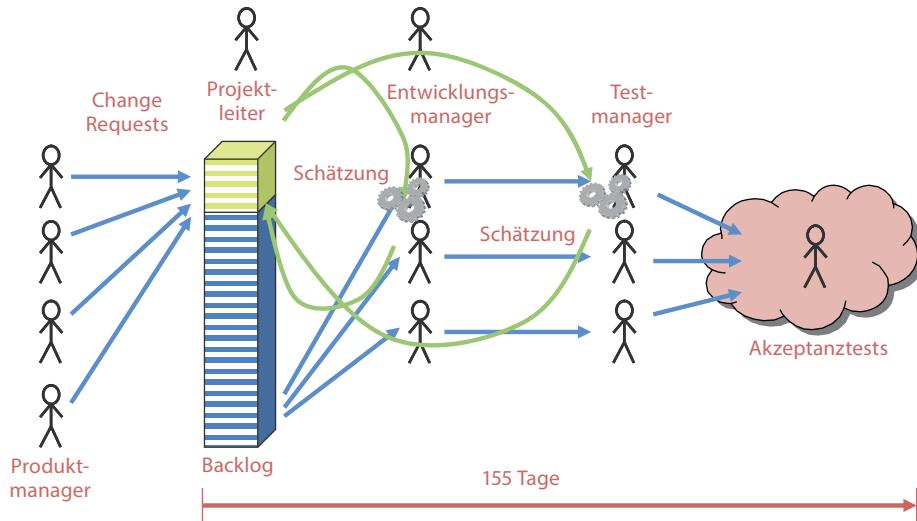


Abb. 20–1 Der ursprüngliche Prozess der XIT-Geschichte

Wenn Sie die Geschichte kennen, wissen Sie, dass das Diagramm versucht, die Unterbrechung der Entwickler und der Tester darzustellen, die durch die konstante Schätzung neuer Arbeitsaufträge und eine regelmäßige Neupriorisierung der riesigen Aufgabenliste verursacht wurde. Zehn Jahre später bin ich allerdings versucht, eine eher lineare Repräsentation der Reise der Arbeitseinheiten zu verwenden und mir dabei viel weniger Sorgen über die Rollen zu machen. Dafür stellt Abbildung 20–2 den Prozess ausreichend gut dar.

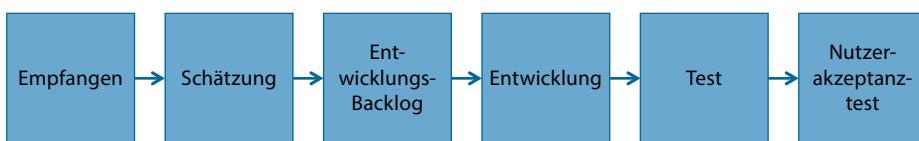


Abb. 20–2 Eine alternative Darstellung der ursprünglichen XIT-Geschichte

Fragen Sie sich auch, was nach »Nutzerakzeptanztest« passiert? Gute Frage!

20.2 Top-down-Dekomposition

»Top-down-Dekomposition« klingt bedeutsam, aber am Ende handelt es sich um das Niederschreiben einer sehr groben Antwort mit ein paar lenkenden Schritten und ein wenig Verfeinerung:

1. Identifizieren Sie Ihre *Commitment-Points*, Ihre Zusagepunkte (einer oder zwei wären typisch); das sind zum Beispiel Punkte, an denen Sie zum ersten Mal zusagen, etwas zu bauen oder eine Dienstleistung zu erbringen, und Sie in der Folge diese dann liefern oder anderweitig fertigstellen.
2. Vergeben Sie Kategorienamen für die Status oder Aktivitäten, die vor, zwischen und nach diesen Zusagepunkten liegen. »Backlog«, »Umsetzung« und »Implementierung« könnten für das XIT-Beispiel funktionieren.
3. Brechen Sie diese Namen so weit herunter, wie es notwendig scheint, sodass die Arbeitsrealität wiedergegeben wird. Zum Beispiel:
 - Backlog
 - Empfangen
 - Schätzen
 - Priorisiert
 - Umsetzung
 - Entwicklung
 - Test
 - Implementierung
 - Nutzerakzeptanz
 - Auslieferung
 - Fertig
4. Identifizieren Sie Warteschlangen, in denen Arbeit zwischen aktiven Status wartet, und verfeinern sie nach Bedarf. Diese Warteschlangen finden sich häufig bei Übergaben.

In diesem Beispiel sind »Empfangen« und »Priorisiert« Status, in denen gewartet wird. Es ist sehr wahrscheinlich, dass Arbeit auch zwischen »Entwicklung« und »Test« wartet, wenn unterschiedliche Menschen diese Aktivitäten durchführen. Ein »Entwicklung fertig«- oder ein »Bereit für den Test«-Status würde darauf hinweisen, dass Arbeit vor oder nach dieser Übergabe für gewöhnlich warten muss.

Eine hochinformelle Variante dieser Vorgehensweise ist es, mit »Zu bearbeiten«, »In Bearbeitung« und »Fertig« zu beginnen (vielleicht ist so etwas sogar bereits vorhanden) und dann zu fragen:

- Gibt es verschiedene Grade von »Zu bearbeiten«? Organisieren wir nach Aktivität oder Priorität?
- Was passiert in »In Bearbeitung«?
- Gibt es verschiedene Grade von »Fertig«? Wie finden wir heraus, was »wirklich fertig« ist?

20.3 Bottom-up-Organisation

Statt tiefer in die Unterscheidung zwischen Aktivitäten und Status der einzelnen Arbeitseinheiten einzutauchen, habe ich das Ganze mit dem allgemeinen Sammelbegriff »Kategorie« versehen. Das war ein bisschen hinterhältig, aber Absicht: Denken Sie daran, dass wir eine effektive Organisation der Arbeitseinheiten finden wollen.

Dieses Ziel ist die Grundlage dieser dritten Strategie. Statt den Workflow zu modellieren, warum organisieren wir nicht einfach die Dinge, die wir aktuell bearbeiten? Dieses Vorgehen kann gut funktionieren, ist aber davon abhängig, dass man genügend Arbeitspakete hat, um eine ausreichende Anzahl verschiedener Status zu repräsentieren.

Sie könnten das existierende Nachverfolgungssystem untersuchen, aber ein mehr sichtbarer (und sozialerer) Weg ist es jedoch, die Haftnotizen aus der Arbeit des vorhergehenden Kapitels zu nehmen und sie horizontal nach ihrem Grad der Fertigstellung zu arrangieren. Das bezieht sich nicht darauf, wie viel Zeit noch aufgebracht werden muss, sondern in welchem Zustand sich die Arbeit befindet (oder äquivalent die Aktivität, die vornehmlich durchgeführt werden muss, um Fortschritt bei der Arbeit zu erzielen). Fragen wie diese können hilfreich sein:

- Was benötigt dieses Stück Arbeit noch?
- Was benötigt es, bevor es mehr wie das andere ist?
- Was wird benötigt, damit wir es als fertiggestellt ansehen?
- Wie ist es hierhergekommen?

Jetzt geht es darum, den Gruppen von Arbeitspaketen in ähnlichen Zuständen der Fertigstellung Namen zu geben. Dann müssen die Kategorien gruppiert oder konsolidiert werden, bis man eine Konfiguration hat, die funktioniert.

Übrigens kann diese Technik einige interessante Fälle aufzeigen:

Frage: *Was benötigt dieses Stück Arbeit noch?*

Antwort: *Es wird eine Fehlerbehebung benötigt, bevor das Testen fortgesetzt werden kann*, oder

Antwort: *Es muss von Team X zurückkommen*

Kapitel 22 kommt auf diese Beispiele zurück; es ist wichtig, der Visualisierung von Nacharbeit und Abhängigkeiten besonders viel Aufmerksamkeit zu schenken, da sie zwei Hauptquellen für Verzögerungen und Frustration darstellen.

20.4 Fazit

Bevor wir weitermachen, sind hier noch einige gute Methoden angegeben, um Ihre bisherigen Ergebnisse zu prüfen und zu verbessern:

- Kombinieren Sie die verschiedenen Vorgehensweisen, um Ihr Modell zu validieren. Zum Beispiel:
 - Fertigen Sie eine Skizze des Top-down- oder des Bottom-up-Modells an.
 - Stellen Sie sicher, dass die tatsächlichen Arbeitseinheiten zu dem passen, was Sie grob aufgezeichnet haben oder in einem Top-down-Modell entwickelt haben. Verwenden Sie dann die »Was benötigt diese Arbeitseinheit noch?«-Fragen.
 - Denken Sie darüber nach, ob es hilfreich wäre, Kategorien zu gruppieren, zu konsolidieren oder aufzubrechen.
- Überprüfen Sie, ob Warteschlangen angemessen repräsentiert sind, und ob Sie wissen, wo Ihre Zusagepunkte hingehören.
- Schauen Sie sich an, wo die Unzufriedenheiten und Frustrationen herkommen, die in Kapitel 18 besprochen wurden.
- Identifizieren Sie die Arten der Wissensentdeckung, die mit jedem aktiven Status assoziiert sind.
- Versuchen Sie der funktionalen Organisation weniger Gewicht zu geben.
- Präsentieren Sie Ihr Modell anderen Personen.

Nichts davon sollte lange dauern. Denken Sie daran: Es sollte schnell skizziert sein!

21 Serviceklassen finden

Kapitel 2 bespricht die Serviceklassen kurz und beschreibt sie als Kategorisierungen, die mit den Kundenerwartungen und einer Sensibilität für die zeitliche Planung zu tun haben. Hier sind noch einmal die Beschreibungen der vier Kategorien, die so häufig vorkommen, dass wir ihnen Standardnamen gegeben haben:

■ *Beschleunigt/Expedite*

Sehr dringende Arbeitspakete, für die wir andere Arbeit unterbrechen, um uns sofort darum zu kümmern.

■ *Termingeschrieben (oder Fester Termin)*

Arbeit, deren Verzögerung nach einem bestimmten Datum zu einer erheblichen Strafe führt, die unverhältnismäßig zu jeglichem Gewinn ist, der aus verfrühter Lieferung entsteht. Ihre Risiken in der zeitlichen Planung werden aktiv gesteuert.

■ *Standard*

Dringlichkeitsgetriebene Arbeit, die in einer mit dem Kunden abgesprochenen Reihenfolge geliefert oder durch eine Systemregel sequenziert wird. Passende Regeln können so etwas einfaches sein wie *First in, First out* (FIFO) oder *die Teammitglieder wählen lassen* oder auf einem ökonomischen Modell wie Verzögerungskosten (Cost of Delay, siehe Kap. 15) basieren.

■ *Unbestimmbare Kosten*

Systemverbesserungen, Wartungsarbeiten und Aktualisierungen, Experimente zu Technologien oder zum Markt, Investitionen in Mitarbeiter – Arbeit, die mittel- und langfristig essenziell ist, deren direkter und kurzfristiger Geschäftswert sich jedoch schwierig messen lässt.

Für den Systemdesigner geht es bei den Serviceklassen darum, die Vielfalt mit einzubeziehen (im Gegensatz zum Verneinen oder sich selbst dafür überorganisieren), sodass die Vorhersagbarkeit verbessert werden kann. Ihre Bedeutung geht allerdings weit über die Systeminterna hinaus; Serviceklassen sind ein sichtbarer Teil des Dienstleistungsangebots und ein Mittel, um Kundenerwartungen und Systemfähigkeiten in Übereinstimmung zu bringen.

Diese Standardklassen dienen nur dazu, erst einmal beginnen zu können. Der Trick ist, qualitative Unterschiede zu identifizieren, die

1. einfach und unumstritten im Voraus erkannt werden können,
2. eine unterschiedliche Behandlung, zeitliche Planung und/oder Risiko-management intern benötigen,
3. vernünftige Erwartungen extern beeinflussen.

Die Konzepte der Serviceklassen und der Arbeitstypen (das »Was« in Kap. 19) überlappen sich offensichtlich gegenseitig. Die Abgrenzung wird interessant, wenn eine Entscheidung erforderlich wird, zum Beispiel:

- Ist dies »Support« (mit einer festgeschriebenen Serviceerwartung) oder eine »kleine Wartungsarbeit« (die in eine Warteschlange mit variabler Länge eingefügt wird, in der sie vielleicht niedrig priorisiert und erst später bearbeitet wird)?
- Wird es »bis Freitag« benötigt (und wird dementsprechend als termingebundenes Risiko gemanagt) oder »so bald wie möglich« (auf Wer-zuerst-kommt-Basis, zum Beispiel)?

Wer kann das schon sagen, ohne zu wissen, welche Bedürfnisse hinter einer Anfrage liegen? Einige Nachforschungen und Verhandlungen sind vielleicht angebracht, aber in vielen Fällen vertraut man den Kunden, die Entscheidung selbst zu treffen.

21.1 Entdecken, überprüfen

Schauen Sie sich die Arbeitstypen nacheinander an. Für jeden einzelnen stellen Sie die folgenden Fragen:

- Werden alle Arbeitspakete behandelt, als seien sie gleich dringend? Falls nicht, wie werden die dringenderen identifiziert? Ist irgendeine spezielle Sprache mit ihnen verbunden?
- Weisen die Daten der Vergangenheit darauf hin, dass verschiedene Leistungsfähigkeiten-Cluster das Gesamte ausmachen? Falls dem so ist, wie zuverlässig können die einzelnen Cluster durch Faktoren bestimmt werden, die einfach im Voraus erkannt werden können?
- Welche Auswahl wird den Kunden gegeben? In welchem Ausmaß sind Arbeitstypen untereinander austauschbar?
- Gibt es bereits *Service Level Agreements* (SLAs)? Sind sie effektiv?

Falls die Antworten zu diesen Fragen darauf hinweisen, dass tatsächlich mehrere Serviceklassen existieren, überprüfen Sie, ob diese auch in der Praxis sinnvoll sein werden:

- Wie wird internes Verhalten durch ihr Vorhandensein beeinflusst werden? Welche Regeln würden greifen und welche Auswirkungen hätte das?
- Wie könnte sich das Verhalten der Kunden verändern, wenn mehrere Serviceklassen angeboten werden? Würden sie davon profitieren?

Für eine Gegenprüfung schauen Sie sich die aktuelle Arbeitslast an:

- Welche Arbeitspakete bekommen (oder verdienten) eine besondere Behandlung?

21.2 Beispiele

Diese Beispiele sind typisch für die Breite des Spektrums, das schon in einer einzelnen Implementierung anzutreffen ist:

- Supportanfragen, deren Beantwortung innerhalb von 24 Stunden erwartet wird
- Aktualisierungen externer Schnittstellen:
 - Zu bestimmten Zeitpunkten ausgeliefert, um Geschäftskontinuität sicherzustellen
 - Als dringlichkeitsgetriebene Arbeitspakete behandelt, um Geschäftschancen zu wahren
 - Als Einheiten mit unbestimmbaren Kosten betrachtet, um der Systemgesundheit willen
- Wichtige Technologieaktualisierungen (beispielsweise eine Versionsaktualisierung der unterliegenden Datenbanktechnologie oder Sprachplattform):
 - Als eine Folge von Arbeitspaketen mit unbestimmbaren Kosten, die kontinuierlich im Hintergrund weiterwandern und ausgeliefert werden, wenn sie fertig sind, und zwar abseits jeder Deadline mit starken Auswirkungen
 - Als Hauptprojekt, das zum »letzten verantwortlichen Moment« durchgeführt wird (das schiebt unvermeidbar andere Projekte aus dem Weg)
- Anforderungen, bei denen schon im Voraus erkannt werden kann, dass sie eine Genehmigung durch Gruppen, wie die IT-Sicherheit oder operatives Risikocontrolling, erfordern, die ein Veto einlegen können.

21.3 In Richtung einer gesunden Arbeitszusammenstellung

So viel sollte klar sein: Sie können nicht planen, dass alle zu 100 % ihrer Zeit mit termingebundener Arbeit beschäftigt sind, und gleichzeitig erlauben, dass sie durch beschleunigte Arbeitspakete unterbrochen werden, und dann immer noch Vorhersagbarkeit erwarten. Sogar ohne Unterbrechungen ist eine hundertprozentige Auslastung mit Arbeit, die einen engen Zeitrahmen erfüllen muss, keine Garantie für Effizienz, sondern für extreme Unvorhersagbarkeit und Verzögerung – mal abgesehen von der Qual auf menschlicher Ebene. Systeme benötigen *Slack*, wenn sie Variabilität absorbieren sollen, statt sie zu akkumulieren, und Menschen benötigen ihn auch.

Sie fragen sich vielleicht, wie eine »sichere« (oder »menschliche«) Auslastung aussieht. 90 %? 80 %? Ich glaube, dass das Problem auf diese Weise aus der falschen Richtung angegangen wird. Aus meiner direkten Erfahrung als Manager und aus Erfahrungswerten, die David und ich bei unseren Kunden gesammelt haben, besteht die Arbeitslast in den meisten Organisationen nur zu 20 % aus wirklich termingebundenen Arbeiten; häufig kann es noch deutlich weniger sein. Nachdem wir vielleicht 10–20 % für Arbeit mit unbestimmbaren Kosten und einen realistischen Teil für beschleunigte und andere ungeplante Arbeit vorgesehen haben, sollte die größte Kategorie die dringlichkeitsgetriebene Arbeit für eine hohe Wertschöpfung sein.

Genau so sollte es sein, vorausgesetzt, man möchte den Fluss des Geschäftswertes maximieren. Glücklicherweise ist auch Vorhersagbarkeit erreichbar, wenn im System Arbeitspakete vorhanden sind, die gefahrlos verzögert werden können, um den Weg für zeitkritische Arbeit freizumachen.

Serviceklassen und andere Kategorisierungen ermöglichen eine breitgefasste Priorisierungsentscheidung, die unabhängig von potenziell umstrittenen Entscheidungen ist, die sich meistens nur um einzelne Dinge drehen. Diese Klassen explizit zu machen, gibt uns sowohl die Möglichkeit, diese an die allgemeineren Unternehmensprioritäten anzupassen, als auch sie mit Kundenbedürfnissen zusammenzubringen. Solange man kurzfristig flexibel zwischen den Kategorien abwägen kann, sollte es möglich sein, beide Interessengruppen zu befriedigen.

Für Ihre Workshops sind Fragen wie diese hilfreich, um die gesamte Zusammensetzung in die richtige Richtung zu lenken und Diskussionen über das Systemdesign zu fördern:

- Welcher Teil unseres Portfolios ist notwendigerweise termingetrieben? Gehen wir Terminzusagen unnötig ein und wenn ja, können wir diesen Anteil reduzieren? Berücksichtigen wir die bereitstehende Kapazität, wenn wir Dinge zeitlich planen? Sind wir in der Lage zu handeln, wenn Termine in Gefahr sind?
- Wie viel Kapazität müssen wir für die beschleunigte oder andere ungeplante Arbeit reservieren? Welche Schritte können wir unternehmen, um die Menge, die wir zu jedem Zeitpunkt bearbeiten, zu reduzieren und/oder zu limitieren?
- Bearbeiten wir eine ausreichende Menge an Arbeit mit unbestimmbaren Kosten? Welchen Raum geben wir den Mitarbeitern, um Wege zum besseren Arbeiten zu finden?
- Welche Kapazität bleibt für dringlichkeitsgetriebene Standardarbeit übrig? Bringen unsere Systeme die richtigen Arbeitspakete an den Anfang der Warteschlange?

Falls diese Fragen für Sie unbequem sind – vielleicht sollte das so sein? Könnte es sein, dass die Mitarbeiter überorganisiert sind – »projektisiert« oder in Silos gebunden? Wenn Ihrer Organisation die Fähigkeit fehlt, sich selbst zu organisieren, wenn und wo es am meisten benötigt wird, zahlt sie einen hohen Preis.

Ich habe in der Vergangenheit Kanbans Fähigkeit, mit Vielfältigkeit umzugehen, als eines der »Killer-Features« beschrieben. Aber wie STATIK Organisationen ermutigt, Vielfältigkeit einzubeziehen, ist noch viel mächtiger. Wenn es denn nur bekannter wäre!

22 Kanban-Systeme gestalten

Alles ist vorbereitet, jetzt kommen wir zum unterhaltsamen Teil – der Gestaltung des Boards, den WIP-Limits und dem Rest.

22.1 Umfang, Granularität und Status der Arbeitseinheiten

Über diese drei Parameter wird am besten gemeinsam entschieden. Wählen Sie den Umfang Ihres ersten Boards zu groß, die Granularität der Arbeitseinheiten zu klein und die Status so, dass die Einheiten nur sehr kurz in den Status verweilen, erhalten Sie vielleicht ein Boarddesign, auf dem zu viel los ist, um noch praktikabel zu sein. Ähnlich verhält es sich mit einem Board, das nur einen sehr engen Bereich abdeckt und auf dem sich große Arbeitseinheiten nur sehr langsam bewegen. Es wird nicht sehr hilfreich sein.

Die Wahl der Parameter muss zum Zweck und zur Zielgruppe des Boards passen. Ein Board, das zur Unterstützung der Verwaltung der Arbeitslast einzelner Entwickler gestaltet wurde, gehört wahrscheinlich nicht in das Büro des CIO. Ein Board auf Projektebene vielleicht schon eher. Ihre Möglichkeiten sind auch rein praktisch schon durch die physische Größe und den Ort des Boards beschränkt (das bezieht sich sogar auf elektronische Systeme).

Idealerweise möchte man, dass sich jedes Mitglied der Zielgruppe des Boards wenigstens mit ein paar Dingen auf dem Board identifiziert. Außerdem soll auf dem Board zwischen den Standup-Meetings sinnvolle Bewegung stattfinden. Wenn diese Ziele nicht erreichbar erscheinen, benötigen Sie vielleicht entweder mehrere Boards oder ein hierarchisches Boarddesign.

22.1.1 Aufeinanderfolgende Status

Für Abbildung 22-1 habe ich eine grobe Strukturierung eines Beispiels aus Kapitel 20 in ein einfaches, unkompliziertes Boarddesign umgesetzt: Arbeitseinheiten kommen in »Empfangen« an und bewegen sich von links nach rechts, bis sie »Fertig« erreichen. Zu einem späteren, passenden Zeitpunkt können sie dann vom Board entfernt und archiviert werden.

Backlog			Umsetzung		Implementierung		Fertig
Empfan-gen	Schät-zung	Priorisiert	Entwick-lung	Test	Akzep-tanz	Ausliefe-rung	

Abb. 22–1 Aufeinanderfolgende Status

Die Status haben noch eine gröbere Strukturierung (»Backlog«, »Umsetzung«, »Implementierung«), aber das muss keine Auswirkung auf die Verwendung des Systems haben.

22.1.2 Parallele Status

Es ist nicht immer möglich, die Status in eine strenge Sequenz von links nach rechts zu setzen. Dies ist besonders dann der Fall, wenn manche Statuswechsel im Hauptprozess parallel passieren. Zum Beispiel:

- Arbeit könnte optimistisch vorangezogen werden und es wäre erlaubt, dass Freigaben auch erst nach dem Beginn der Arbeit erfolgen können.
- Arbeit könnte durch Probleme bei der Technik, der Qualität oder geschäftsbedingt zu jedem Zeitpunkt des Prozesses blockiert werden; ein gewisser Anteil an Nacharbeit wäre zu erwarten.

Die zweckmäßigsten visuellen Techniken, um mit parallelen Zuständen umzugehen, sind häufig ticketgebunden:

- Fertigstellungen oder Freigaben werden auf den Tickets beispielsweise mithilfe einer Checkbox im Design des Tickets, die als Erinnerung dient, abgehakt.
 - Performance getestet
 - Durch den Nutzer abgenommen
- Ebenso können Tickets gekennzeichnet werden, die Zusatzaufgaben benötigen. Diese Zusatzaufgaben werden abgehakt, wenn sie fertiggestellt wurden.
 - UX-Design
 - Datenbankmigration

Wie in Abbildung 22–2 werden Blockaden deutlich durch das Anbringen von Klebezetteln in einer Warnfarbe am blockierten Ticket gekennzeichnet (Pink ist die übliche Farbwahl für diese »Blockadekleber«).

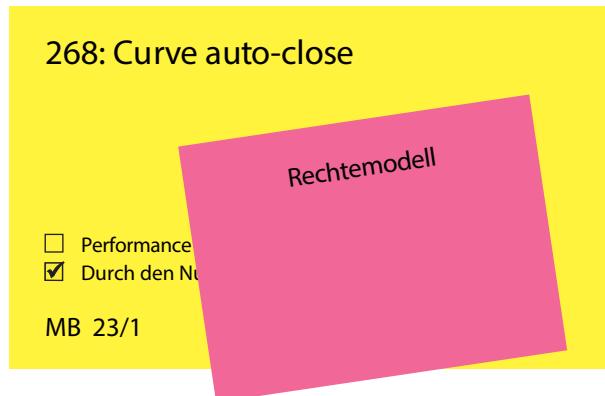


Abb. 22-2 Blockiert!

Statt die Statuswechsel zu visualisieren, können Sie diese auch unabhängig vom Board behandeln – durch Regeln wie »*Definition of Ready*« und »*Definition of Done*« auf Spaltenebene. Achten Sie darauf, dass eine Vereinbarung existiert, wie diese eingehalten werden.

22.1.3 Fehler

Die »Blockadekleber«-Technik wird sehr häufig dafür verwendet, um auf das Aufkommen von Fehlern hinzuweisen. Meine persönliche Meinung dazu ist die folgende:

- Ich mag es wirklich nicht sehen, wenn Arbeitseinheiten sich durch den Prozess vorwärts bewegen, von denen bekannt ist, dass sie fehlerhaft sind. Man hebt sich den Ärger nur für später auf, dann tritt der aber vervielfacht auf!
- Genauso unwohl wird mir, wenn Arbeitseinheiten, die Nacharbeit benötigen, auf dem Board zurückwandern, statt an ihrem aktuellen Platz blockiert zu werden. Häufig wird der wirkliche Zustand der Arbeit dadurch missinterpretiert, manchmal ist das sogar gefährlich.

Ich mache aber auch Ausnahmen:

- Fehler, die eher zum existierenden Produkt als zur fraglichen Arbeitseinheit gehören, kann man getrost als neue Arbeitsinhalte einbringen, sodass existierende Arbeitseinheiten unbehindert vorangehen können.
- Gelegentlich ist es wirtschaftlich sinnvoll, Arbeitseinheiten zu erlauben, trotz enthaltener, bekannter Fehler weiter zu gehen. Seien Sie aber vorsichtig: Greift man zu häufig auf dieses Verhalten zurück, nimmt die Qualität des Produkts oder angebotenen Dienstes schnell ab.
- Falls klar ist, dass die Arbeitseinheit nie für die Bearbeitung ausgewählt hätte werden dürfen, könnte ein Zurücksenden die angemessenste Vorgehensweise

sein. Das sollte aber nur sehr selten vorkommen und auch eine Untersuchung nach sich ziehen. Was ist falsch gelaufen? Wie konnte unser Prozess dieses Verhalten erlauben? Haben wir irgendwelche Überreste hinterlassen, die beseitigt werden müssen? Was haben wir nicht verstanden?

Auch hybride Lösungen könnten angebracht sein. Wenn zum Beispiel ein Fehler in einem Feature gefunden wurde, das bereits ausgeliefert wurde, sich aber noch in der Validierungs- oder Garantiephase befindet, kann das noch sichtbare Ticket mit einer Haftnotiz markiert werden und eine neue Arbeitseinheit für die Nacharbeit in das System eingebracht werden. Dies weist auf zwei wichtige Fakten hin: Wir haben ein fehlerhaftes Feature in der Produktivumgebung und die Fehlerbehebung bewegt sich vorwärts durch das System.

22.1.4 Abhängigkeiten

Dieser Abschnitt behandelt zwei Dinge: Abhängigkeiten zwischen Arbeitseinheiten und Arbeitseinheiten, die Zuarbeit von anderen Diensten benötigen. Zwei Haupttechniken sind anwendbar:

1. Die Arbeitseinheit mit der Abhängigkeit wird als blockiert markiert wie in Abbildung 22–3. Wenn die Situation eher vorhersehbar ist und keine Ausnahme darstellt, verwenden Sie eine geeignete Farbcodierung, vielleicht eine Farbe pro Dienst. Wenn eine Abhängigkeit zu einer anderen Arbeitseinheit auftritt, sollte dies durch Angabe des Namens (und Identifizierungsnummer, falls Sie die verwenden) klargestellt werden.

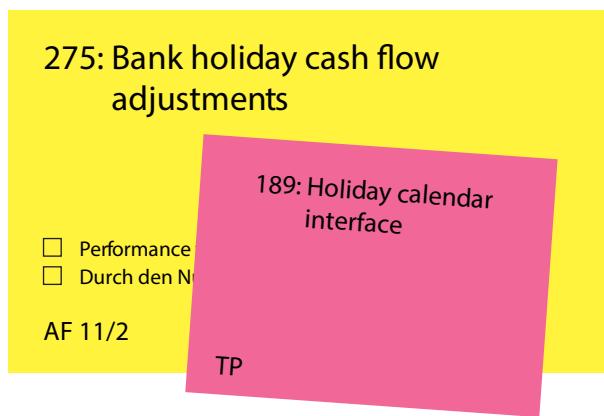


Abb. 22–3 Eine Arbeitseinheit, die von einer anderen abhängt.

2. Schieben Sie die Arbeit in einen Wartebereich, der vielleicht nach dem fraglichen Dienst benannt ist (zum Beispiel »Infrastruktur«).

Die erste Technik ist meist die bessere Wahl für Abhängigkeiten zwischen Arbeitseinheiten. Für Abhängigkeiten zwischen Diensten funktionieren beide. Die zweite Technik macht das Design des Boards etwas komplizierter und könnte nicht immer durchführbar sein. Aber mit ihr lässt sich einfach erkennen, wie viele Arbeitseinheiten sich gerade in diesem Zustand befinden, vielleicht kann damit sogar ein enthaltener Workflow nachverfolgt werden.

22.2 Andere Dimensionen

Je größer die Anzahl der zu visualisierenden Arbeitseinheiten ist, desto hilfreicher ist es, sie in weiteren Dimensionen zu organisieren, wie zum Beispiel:

- Arbeitstyp und/oder Serviceklasse
- Quelle
- Eine weniger permanente Kategorie, wie beispielsweise Initiative, Projekt, Epic oder Sprint
- Der »Besitzer« der Arbeitseinheit

Es gibt zwei Hauptmöglichkeiten, um diese Dimensionen visuell darzustellen:

1. Auf Ticketebene unter Verwendung einer Kombination aus Ticketfarbe, Anmerkungen oder Verzierungen. Zum Beispiel:
 - Eine Standardticketfarbe »Gelb« für »Standard«, also dringlichkeitsgetriebene Arbeit
 - Orange Tickets für termingetriebene Arbeitseinheiten, mit einem Vermerk zum Termin
 - Rote Tickets für beschleunigte Arbeitseinheiten (oder überhaupt kein Ticket, wenn sie kurzlebig genug sind)
 - Ein Vermerk auf dem Ticket mit dem Sprint, in den die Arbeit gezogen wurde (ich schreibe gerne die Sprint-Nummer auf ein kleines, farbiges Etikett, wobei ich für jeden Sprint eine andere Farbe verwende, damit sie aus etwas Entfernung erkannt werden können).
 - Die Tickets mit den Initialen des Verantwortlichen versehen oder einen beweglichen *Avatar* auf dem Ticket anbringen.
2. Linien hinzufügen, um horizontale *Schwimmbahnen* quer über das gesamte Board oder Teile davon abzugrenzen. Zum Beispiel:
 - Eine ständige Bahn, die nur für beschleunigte Tickets vorgesehen ist.
 - Bahnen, die für Teams oder einzelne Personen vorgesehen sind (ich rate davon ab, weil es wahrscheinlich die Selbstorganisation behindert).
 - Schwimmbahnen, die die Arbeitsinhalte nach Projekten, Initiativen oder Ähnlichem organisieren wie in Abbildung 22–4. Die Namen dieser Bahnen bestehen nur vorübergehend.

Kanal	Backlog	Auslieferung	Validierung	Fertig
A				
B				

Abb. 22–4 Zwei parallele Kanäle für Arbeit

22.3 Hierarchische Boards und das Aufbrechen/Zusammenführen-Muster

Schauen Sie sich das etwas komplexere Design in Abbildung 22–5 an.

Pipeline		Backlog	Implementierung			Auslieferung		Fertig
Ideen	Beschlossen	Epic	Bereit	Bauen	Fertig	Deployment	Validierung	

Abb. 22–5 Epics und Features

Sehen wir uns die Details an. Der vordere Teil des Boards (die linke Seite) ist in Abbildung 22–6 dargestellt. Er befasst sich mit großen *Epics*.

Pipeline			Backlog
Ideen		Beschlossen	Epic

Abb. 22–6 Eine Pipeline von Epics ...

Der hintere Teil des Boards (in Richtung der Mitte), wie in Abbildung 22–7 gezeigt, ermöglicht es, dass bis zu drei Epics in drei kleinere Arbeitseinheiten (Features oder User Stories) *aufgebrochen* werden.

Backlog	Implementierung			Auslieferung		Fertig
Epic	Bereit	Bauen	Fertig	Deployment	Validierung	

Abb. 22–7 ... Feature-weise ausgeliefert

Dieses spezielle Design geht von der Annahme aus, dass Features einzeln und unabhängig von anderen ausgeliefert und validiert werden können. Dort, wo das nicht der Fall ist, werden sie wieder zu Epics zusammengeführt, die dann als Ganzes geliefert werden können. Das Boarddesign in Abbildung 22–8 unterstützt beispielsweise das Aufbrechen/Zusammenführen-Muster.

Pipeline		Backlog	Implementierung			Auslieferung		Fertig
Ideen	Beschlossen	Epic	Bereit	Bauen	Fertig	Deployment	Validierung	

Abb. 22–8 Aufbrechen/Zusammenführen

22.4 Work in Progress limitieren

Ein echtes Kanban-System beinhaltet einen Mechanismus, um die Menge an paralleler Arbeit zu limitieren. Wieder einmal gibt es viele Wege, das zu erreichen.

- Numerische WIP-Limits, die jeweils die Menge an Work in Progress in einer einzelnen Spalte, mehreren nebeneinanderliegenden Spalten oder einer horizontalen Schwimmbahn limitieren.
- Physische Begrenzungen wie beispielsweise die Anzahl der verfügbaren Schwimmbahnen
- Eine limitierte Anzahl von Markern (bspw. persönliche Avatare), die sich im Umlauf befinden. Diese werden an den Tickets angebracht. Die Steuerung, die durch diesen Mechanismus erfolgen kann, wird geschwächt, wenn Tickets in Bearbeitung sein dürfen, ohne dass so eine Markierung auf ihnen angebracht ist; in diesem Fall wird nicht die gesamte Menge der Arbeit begrenzt, sondern nur die Menge, die als »aktiv« angesehen wird.
- Regeln, die Work in Progress beispielsweise pro Person oder Serviceklasse steuern.
- Ein externer Mechanismus, der neue Arbeit nur dann in das System lässt, wenn ausreichend Kapazität vorhanden ist. Scrum erreicht dies dadurch, dass die Menge der Arbeit in einem zeitbegrenzten Sprint durch das Implementierungssteam am Anfang des Sprints festgelegt wird.

Es gibt viele Gründe, die dafür sprechen, die verschiedenen Techniken gemeinsam zu verwenden, wie beispielsweise:

- Horizontale (bspw. Serviceklassen) und vertikale (Spalte/Status) Limits in Kombination miteinander

- Physische Begrenzungen (bspw. für Epics), die mit vertikalen Limits kombiniert werden (bspw. für die Menge an Features, die sich gerade parallel in der Entwicklung befinden).
- Sprints, die mit Limits über mehrere Spalten kombiniert werden, um Multi-tasking zu drosseln und um darauf zu drängen, dass Arbeitseinheiten, die nur einen halbfertigen Zustand erreicht haben (bspw. »Code geschrieben«), vorwärts bewegt werden.

Wenn das Zusammenspiel verschiedener, übergreifender Mechanismen ordentlich kalibriert wurde, fühlt sich keines der Limits als übermäßig begrenzend an, obwohl durch die Kombination der Work in Progress auf ein so geringes Niveau gesenkt werden kann, das vorher vielleicht unerreichbar erschien. Jedes Limit adressiert ein bestimmtes Symptom oder Verhalten und schaltet sich dann ein, wenn es gebraucht wird, und hilft dabei, den Gesamtfluss zu glätten.

Pipeline		Backlog	Implementierung			Auslieferung (4)		Fertig
Ideen	Beschlossen (5)	Epic	Bereit	Bauen (6)	Fertig	Deployment	Validierung	

Abb. 22–9 Übergreifende WIP-Limits

Abbildung 22–9 zeigt, wie das für unser Aufbrechen/Zusammenführen-Beispiel aussehen könnte. Hier existiert ein numerisches Limit von fünf Epics in der Spalte »Beschlossen«, eine physische Begrenzung auf drei Epics in der »Implementierung« und ein numerisches Limit von vier Epics für die »Auslieferung«. Die Anzahl der aktiv in Bearbeitung befindlichen Epics wird so auf sieben ($3+4$) begrenzt. Die Anzahl der angenommenen und noch nicht fertiggestellten Epics ist auf zwölf limitiert ($5+3+4$). Das Limit für sechs Features in der »Bauen«-Spalte greift über drei Epics und schreibt nicht vor, wo diese Features auftreten müssen.

Jedes dieser Limits hat eine andere Wirkung:

- Das Limit auf den »beschlossenen« Epics erfordert, dass eine Art Priorisierung stattfindet; falls neue Arbeitseinheiten ankommen, die eine ausreichend hohe Priorität haben, werden bereits wartende Arbeiten gegebenenfalls zurückgedrängt (das ist unter diesen Umständen keine schlechte Sache).
- Die Begrenzung der Epics in der Spalte »Implementierung« bringt sogar auf diesem Grad der Granularität ein Fluss-Verhalten wie »eins raus, eins rein« mit sich.
- Das Limit in »Bauen« soll dazu führen, dass Features und die betreffenden Epics schnell gebaut werden.

- Das Limit für Epics in der »Auslieferung« soll verhindern, dass die Epics länger als notwendig dort herumhängen. Allerdings wird vielleicht ein höheres Limit oder ein einzelnes Limit auf dem »Deployment« benötigt, wenn sich herausstellen sollte, dass wahrscheinlich der gesamte Prozess durch Probleme mit der Verfügbarkeit von Kunden blockiert wird.

22.5 Review

Mit all diesen Auswahlmöglichkeiten für das Design ist es eine gute Idee, das Boarddesign noch einmal zu betrachten, bevor Sie es als Ihr »initiales« Design erklären (denken Sie noch nicht einmal daran, dass das Ihr »endgültiges« Design ist!).

Beginnen Sie mit ein paar grundlegenden Überprüfungen:

- Wie sieht das Board aus, wenn es mit Arbeitseinheiten befüllt wurde? Organisiert es diese effektiv? Ist ausreichend Platz vorhanden?
- Wie viel Bewegung wird zwischen den Standup-Meetings sichtbar? Nicht ausreichend? Zu viel?
- Ist es verständlich für diejenigen, die es verwenden sollen? Ist es zu kompliziert? Zu sehr vereinfacht? Fragen Sie sie oder noch besser: Involvieren Sie sie.
- Stellt es unrealistische Forderungen oder nimmt es Veränderungen im Prozess, der Organisation oder in den Rollen an, für die noch nicht alle ihr Einverständnis gegeben haben? (Ich lehne radikale Veränderungen im System zu diesem Zeitpunkt nicht kategorisch ab, aber ich empfehle sie meistens nicht.)

Etwas abstrakter ausgedrückt adressiert ein gutes Design mehrere Bedürfnisse unterschiedlicher Art:

- Es macht **transparent**, was passiert und wie es passiert, verhilft zu besseren Entscheidungen und fördert Selbstorganisation und **Kooperation**. Können Sie sich diese Dinge (kollektiv) in diesem Design bildlich vorstellen?
- Es hilft dabei, **Balance** zu erreichen – zwischen Nachfrage und Angebot, über verschiedene Nachfragekategorien hinweg usw. Wird es das tun, selbst wenn es nur zögerlich geschieht?
- Es fördert sowohl Gedanken als auch Aktivität bezüglich des **Kundenfokus** und des **Arbeitsflusses**. Denken Sie an Kapitel 4 und 5 und versuchen Sie, Ihr Board als Denkwerkzeug zu verwenden.

Zuletzt, der wichtigste Punkt:

- In welcher Art und Weise gelingt es mit diesem Design, die besonderen Unzufriedenheiten und Frustrationen zu adressieren, die Sie ganz am Anfang dieses Prozesses gesammelt haben (Kap. 18)? Erwarten Sie nicht, dass alle sofort behoben werden – es könnte sogar besser sein, das gar nicht erst zu versuchen –,

aber wird es die Symptome oder Ursachen näher an die sichtbare Oberfläche bringen, als sie es jetzt sind?

Diese Überlegungen zur Überprüfung des Designs – insbesondere die letzte – sind nicht nur beim initialen Design des Boards anwendbar, sondern auch bei der Weiterentwicklung. Es ist besser, ein hässliches Board zu haben, dessen Veränderungen zielgerichtet sind, als ein wunderschönes Board, dessen Verfeinerungen nur zum eigenen Zweck dienen. Lernen Sie, durch das Board das repräsentierte System zu sehen; handeln Sie auf dem einen, um auf dem anderen eine Wirkung zu erzielen.

23 Ein Kanban-System einführen

Es ist durchaus hilfreich, eine Kanban-Implementierung als dreistufigen Prozess zu sehen:

1. Den Einsatz planen

Vorbereitung in Bezug auf die Teilnehmer, Veranstaltungsorte, Werkzeuge, unterstützende Materialien usw.

2. Die Agenda zusammenstellen

STATIK einsetzen mit dem expliziten Ziel, eine überzeugende Menge vereinbarter Prioritäten, Ziele und Aktivitäten zu entwickeln

3. Veränderung durch das System vorantreiben

Impuls für die Zukunft beibehalten und sicherstellen, dass der Fortschritt auch weiterhin sowohl sichtbar als auch sinnvoll ist

Dieser Aufbau kann unabhängig davon angewendet werden, ob das Ziel darin besteht, ein alleinstehendes Kanban-System aufzubauen, die Kanban-Methode zum ersten Mal einzuführen oder um eine neue Runde Veränderungen zu starten. Sie können es sogar retrospektiv verwenden. Es wird Ihnen helfen, konstruktiv über eine Implementierung nachzudenken, die eine engere Verbindung zur umgebenden Organisation benötigt.

Ich hoffe, dass ich zeigen kann, dass es kein Widerspruch ist, die Kanban-Methode wirkungsvoll einzuführen und gleichzeitig ihrer menschlichen Gesinnung treu zu bleiben. Gegen Ende dieses Kapitels überprüfen wir, welche Rolle die Werte für die Implementierung spielen können.

23.1 Den Einsatz planen

Wenn man einen unterstützenden Prozess durchführt, zahlt es sich wirklich aus, ihn vorher vorzubereiten. Als Prozessbegleiter sollte man sich diese Art von Fragen stellen, wenn man STATIK verwenden möchte:

1. Die Quellen der Unzufriedenheit erkennen (Kap. 18)
 - Haben Sie wenigstens eine grobe Vorstellung vom Umfang dieser Übung? Hat (oder benötigt) sie eine Patenschaft?
 - Wer sollte teilnehmen? Wer wird die Menschen repräsentieren, die innerhalb der angenommenen Systemgrenzen arbeiten? Wer repräsentiert die Kunden des Systems? Benötigt die umgebende Organisation ebenfalls einen Repräsentanten?
 - Welche Werkzeuge werden Sie verwenden, um eine gute und breite Menge von Antworten einzuholen und zu organisieren (Kap. 17)?
2. Anforderungen und Leistungsfähigkeiten analysieren (Kap. 19)
 - Sind Sie bereit, bei der Erfassung des »Was, wem und warum« zu helfen?
 - Werden Sie Daten im Voraus einholen (und führen Sie vielleicht schon selbst eine initiale, quantitative Analyse durch) oder warten Sie darauf, was andere Teilnehmer tun wollen? Wie können Sie dabei unterstützen?
 - Was wissen Sie bereits darüber, wie Arbeit in das System kommt? Haben Sie alle benötigten Personen im Raum, um das sorgfältig herauszufinden?
3. Den Workflow modellieren (Kap. 20)
 - Wie sieht Ihre präferierte Vorgehensweise aus (aufzeichnen, top-down, bottom-up)?
 - Sind Sie mit Prüfungsfragen ausgerüstet, um das Ergebnis zu inspizieren?
4. Serviceklassen finden (Kap. 21)
 - Ist es effektiver, sich den Serviceklassen als internes Werkzeug zur Organisation und Lieferplanung zu nähern oder als Weg, Kundenerwartungen zu erforschen? Wie bringen Sie diese beiden Aspekte zusammen?
 - Auf welcher Organisationsebene kann diese Übung die Zusammensetzung der Arbeit beeinflussen?
5. Kanban-Systeme gestalten (Kap. 22)
 - Wie bringen Sie die Konzepte ein und teilen mit, was an anderer Stelle funktioniert hat? (Diese Frage beantwortet sich von selbst, wenn der Workshop auch eine Schulung ist.)
 - Wie viel Zeit soll in die Verfeinerung von Designs fließen, bevor sie im praktischen Einsatz ausprobiert werden?
 - Physisch oder elektronisch? Welche einschränkenden Faktoren (physisch, geografisch, organisatorisch, Privatsphäre und Sicherheit oder featureabhängig) müssen berücksichtigt werden?

Als Letztes sollten Sie so grundlegende Dinge wie die Auswahl und Verfügbarkeit von Veranstaltungsorten nicht vergessen, genauso wie die Verfügbarkeit der Teilnehmer, Schreibmaterialien und sonstigen Ausstattung. Diese Dinge sind auch dann wichtig, wenn Sie keinen großen Workshop planen.

23.2 Die Agenda zusammenstellen: die drei P

Wir haben die Details nun geplant; jetzt ist der Zeitpunkt gekommen, einen Schritt zurückzutreten und darüber nachzudenken, in welchem Rahmen sich die Tätigkeiten bewegen sollen. Positionierung, Zweck (Purpose) und Priorität geben der Initiative eine grundlegende Form, auf die potenzielle Teilnehmer reagieren können.

23.2.1 Positionierung

Wie man mit der Organisation und den Menschen darin in Verbindung tritt, hängt vom Kontext und den persönlichen Präferenzen ab. Auch hier zahlt es sich wieder aus, sich darüber ein paar Gedanken zu machen. Sie müssen darüber nachdenken, wie Sie die Kanban-Methode in Bezug auf die Bedürfnisse der Organisation (soweit Sie davon Kenntnis haben) positionieren.

Hier ist eine Positionierung, die auf einigen bereits bekannten Elementen basiert, damit Sie anfangen können:

- Die Kanban-Methode wird als die menschliche »Beginne mit dem, was du gerade tust«-Herangehensweise für Veränderung beschrieben.
- Wir sehen uns kurz die Prinzipien, Praktiken und Werte an und denken darüber nach, wie sie in unserer Situation anwendbar sind.
- Wir betrachten *was wir gerade tun* – mit allen Schönheitsflecken – und verwenden dafür die *Kanban-Linse*. Diese Linse ermutigt uns zu folgender Erkenntnis:
 - Was wir tun – unsere Art von *kreativer Wissensarbeit* – ist serviceorientiert.
 - Das Erbringen von Dienstleistungen beinhaltet einen Workflow.
 - Ein Workflow beinhaltet eine Folge von Aktivitäten zur Wissenserarbeitung.
- Durch eine Folge von Übungen erreichen wir Folgendes:
 - Wir kartieren unseren Wissenserarbeitungsprozess.
 - Wir berücksichtigen, wie und warum Arbeit bei uns ankommt.
 - Wir rüsten uns entsprechend aus, dass wir Arbeit verfolgen können, wie sie durch und zwischen verschiedenen Diensten fließt.
- Um den Erfolg sicherzustellen, werden wir uns die Zeit für folgende Dinge nehmen:
 - Den Zweck und den Umfang des betrachteten Systems gemeinsam zu beschließen.
 - Quellen der Unzufriedenheiten aus verschiedenen Perspektiven zu identifizieren.
 - Aktivitäten zu priorisieren, die diese Unzufriedenheiten zu adressieren beginnen und die das Design und den Betrieb mit seinem ursprünglichen Zweck besser abstimmen.

23.2.2 Zweck (Purpose)

Erinnern Sie sich an STATIK-0 aus Kapitel 16? Planen Sie den nullten Schritt mit ein:

0. Den Zweck des Systems verstehen

Es scheint selbstverständlich zu sein, diesen Schritt vor dem ersten Schritt zu platzieren (Quellen der Unzufriedenheit erkennen). Ich empfehle aber, dass Sie beim »Was, wem und warum« von Schritt 2 (Anforderungen und Leistungsfähigkeit analysieren) etwas verweilen und es als Grundlage verwenden, um zu erforschen, wie effektiv das System seinen Zweck erfüllt.

Das bringt Sie in eine gute Position für eine Diskussion über Zweck-Fitness und Fitnesskriterien. Das sollte viel mehr sein als eine philosophische Diskussion nebenbei; Lücken und Erfolgskriterien zu identifizieren, wird dabei helfen, die folgenden Designaktivitäten zu fokussieren, und der Einführung einen zusätzlichen Schwung verleihen.

23.2.3 Prioritäten

Auf ähnliche Weise habe ich es als sehr kraftvoll empfunden, die Werte zu priorisieren und die wichtigsten drei oder vier zu identifizieren, um die dann eine eindringliche Handlungsaufforderung herumgebaut werden kann. Ich mache das mit der wenig kreativ benannten »Kanban-Werte-Übung«. Die Materialien dazu habe ich mit einer Open-Source-Lizenz auf Slideshare¹ veröffentlicht.

Die Teilnehmer bekommen eine Liste der neun Werte; zuerst bringen sie diese mit den Grundprinzipien und den Kernpraktiken zusammen. Die Erfahrung hat gezeigt, dass diese Reihenfolge am besten funktioniert und es viel einfacher ist, wenn die Prinzipien und Praktiken in umgekehrter Reihenfolge verwendet werden.

Ich erkläre gerne, dass es keine falschen Antworten gibt – aber es gibt eine kanonische Antwort! Wenn Sie meiner empfohlenen Zuordnung folgen und von unten nach oben arbeiten, sollten Sie **Führung, Respekt, Vereinbarung** und **Verständnis** bei den Prinzipien bekommen, dann **Kooperation, Transparenz** (zwei-mal), **Kundenfokus** und **Arbeitsfluss** (beide für KP4), **Balance** und (zum dritten Mal) **Transparenz** bei den Praktiken. Mit den Prinzipien zu beginnen, erlaubt es uns, vier Werte schon wegstreichen zu können, bevor wir zum schwierigeren Teil kommen.

Es folgen zwei Runden Abstimmung mit drei Punkten pro Person, um zu identifizieren, welche Werte

1. Sie können ein PDF unter <http://www.slideshare.net/asplake/kanban-values-exercise> herunterladen. Gerne stelle ich Ihnen auch die Quelldatei aus PowerPoint zur Verfügung. Beide sind unter einer Creative Commons Attribution-ShareAlike 4.0 International License lizenziert.

- bei den Teilnehmern am meisten auf persönlicher Ebene – aus welchen Gründen auch immer – Anklang finden und
- in der Implementierung initial besonders betont werden sollten.

Diskutieren Sie über die Resultate jeder Runde. Unterschiede zwischen den beiden können eklatant sein; kommen Sie in Ihrer Rolle nicht voreilig zu negativen Schlussfolgerungen (ich habe diesen Fehler schon gemacht).

Nehmen Sie am Ende die drei oder vier Werte mit den meisten Stimmen aus der zweiten Runde und versuchen Sie eine Art Geschichte um sie herum zu konstruieren. Die Kanban-Agenden können nützliche Führungsinstrumente sein, aber die Resultate sollten kontextspezifisch sein und von der Gruppe als ihre eigenen akzeptiert werden.

Das Ganze funktioniert am besten mit Gruppen zwischen fünf und acht Teilnehmern. Falls Sie mehr als acht Teilnehmer haben, können Sie verschiedene Resultate aus Untergruppen in einer Nachbesprechung oder Auswertung konsolidieren.

Zwischen zwei Abstimmungsrunden können Sie auch ein Kanban-Knowsys-Spiel einfügen (Kap. 17). Verwenden Sie es, um spielerisch und mit Spaß herauszufinden, wie gut die Teilnehmer sich gegenseitig zugehört haben und auch wie gut sie übereinstimmen. Sie können die Resultate der zweiten Runde dann in Kanban Knowsys »Discover Play« eingeben, um sie mit den drei vordefinierten Agenden zu vergleichen.

23.3 Veränderung durch das System vorantreiben

Sie haben die Planung und die Rahmengabe abgeschlossen; Sie haben den Workshop abgehalten oder neue Runden gedreht; Sie haben sich auf Prioritäten für die Veränderung geeinigt; Teams richten sich in ihren neuen Kanban-Systemen ein. Wie stellt man jetzt sicher, dass es nicht hier aufhört?

Glücklicherweise haben wir eine große Anzahl gudokumentierter Kanban-Implementierungen signifikanter Größe, aus denen einige gute Lehren gezogen werden können – zum Beispiel *The Kanban Kick Start Field Guide* von Christophe Achouiantz und Johan Nordin [Achouiantz & Nordin 2014], Yuval Yerets *Pull-based Change Management* [Yeret 2013] und Jeff Andersons *The Lean Change Method* [Anderson 2013] (wie in Kap. 17 erwähnt). Diese und zahllose kleinere Implementierungen sind sich über die Wichtigkeit einig, parallel zum Hauptliefersystem ein Hilfs-Pullsystem zu unterhalten. In diesem werden kleine Inkreme der Veränderung aus einer Art Backlog (vielleicht dargestellt auf einem *Change Canvas* oder in einer *Story Map*) gezogen und visuell während ihrer Implementierung gemanagt.

Es gibt ein paar interessante Unterschiede in der Gewichtung zwischen den Autoren, die herausgestellt werden sollten. Einige verwenden gerne das *Kanban-Tiefen-Assessment-Tool* (»Wie tief ist Ihr Kanban?«) oder Ähnliches, um die

Praktiken zu priorisieren, die implementiert oder verfeinert werden sollen; andere fokussieren auf die Unzufriedenheiten und Probleme und nutzen diese als Antreiber für Veränderung.

Die erste, praktikgetriebene Herangehensweise ist sinnvoll, wenn in einer begrenzten Zeit ausgerollt werden soll. Ich tendiere allerdings aus zwei Gründen zum letzteren, eher problemgetriebenen System: Mich macht Veränderung misstrauisch, die um ihrer selbst willen implementiert wird, statt bereits existierende Unzufriedenheiten zu adressieren, und ein Problemmanagementsystem ist einfach zu nützlich, um es zu ignorieren.

23.3.1 Veränderungskomitee identifizieren

Die neun Werte repräsentieren Praktiken und beschreiben Nutzen; wir können sie ohne Weiteres dafür einsetzen, die Probleme zu organisieren, die sie adressieren (der scheinbare Zwiespalt zwischen praktikgetriebener und problemgetriebener Veränderung muss also vielleicht doch gar keine große Sache sein). Gehen wir sie ein letztes Mal durch und identifizieren wir die Schlüsselcharakteristika effektiver Kanban-Implementierungen.

Sie können die Werte verwenden, um zu beurteilen, wo Sie gerade stehen, und mit ihnen Veränderungskomitee priorisieren, die in Ihrer Situation sinnvoll sind. Geben Sie Ihrem System eine Note von eins bis vier für jedes nummerierte Merkmal gemäß dieser Skala:

1. Unser System zeigt diesen Aspekt kaum, wenn überhaupt.
2. Unser System ist manchmal dazu in der Lage, diesen Aspekt zu zeigen.
3. Unser System zeigt diesen Aspekt meistens überzeugend.
4. Unser System weicht nur in Ausnahmefällen davon ab; wir bewältigen die Konsequenzen, wenn es doch dazu kommt.

Transparenz

1. Arbeitseinheiten sind visuell nach Typ, Status (in irgendeiner Wissenserarbeitungsaktivität, in einer Warteschlange oder in einem anderen Status), parallelen Arbeitsflüssen und Serviceklassen organisiert.
2. Es ist deutlich zu sehen, welche Arbeitseinheiten aus welchem Grund blockiert sind.
3. Soweit es relevant ist, ist klar, wer woran arbeitet.
4. Explizite Regeln fassen gemeinsame Erwartungen an die Auswahl von Arbeitseinheiten, Qualitätskriterien usw. zusammen.
5. Der Fortschritt der Arbeit und die Gesamteffektivität des Systems werden in verschiedenen festgelegten Zeitabständen überprüft, von wenigstens täglich (beispielsweise beim Standup-Meeting) bis vierteljährlich oder länger.

6. Es wird darauf geachtet, wie Fortschritt, Anforderungen und Fähigkeiten extern berichtet werden, sowohl zum Kunden als auch unternehmensweit.
7. Metriken haben eine klare Beziehung zum Zweck des Systems.

Noch bevor Prozessveränderungen implementiert werden, hat die Einführung des visuellen Managements häufig einen sofortigen Nutzen. Es macht sichtbar, welche Entscheidungen wo getroffen werden müssen (manchmal große Entscheidungen). Die anderen Aspekte der **Transparenz**, die hier genannt wurden, benötigen vielleicht mehr Zeit, bis sie greifen; sie sind aber nicht weniger wichtig bezüglich der organisatorischen Auswirkungen, die sie erzielen können.

Balance

1. Work in Progress (WIP) ist limitiert, sodass kein Individuum, keine Aktivität oder Teilstrom überlastet wird oder einen größeren Teil der verfügbaren Kapazität oder geteilten Ressource in Anspruch nimmt, als angemessen wäre.
2. Arbeit wird nur dann in und durch das System gezogen, wenn Kapazität verfügbar ist.
3. WIP-Limits gelten für jede Arbeit, die begonnen, aber noch nicht fertiggestellt wurde. Das schließt Arbeit mit ein, die zwischen Aktivitäten und Diensten wartet.
4. Das System bietet eine Auswahl von Risikoprofilen bezüglich der Zeitplanung (zum Beispiel die Unterscheidung in termingetriebene und dringlichkeitsgetriebene Arbeit) sowie Serviceklassen, die verwendet werden können, ohne große Gefahr zu laufen, das System aus dem Gleichgewicht zu bringen.
5. Wenn Kapazität zwischen konkurrierenden Quellen von Arbeit aufgeteilt wird, werden die Bedürfnisse aller Stakeholder und die Gesamtkapazität über mehrere verschiedene Zeiträume berücksichtigt.

Unerfahrene Kanban-Anwender sorgen sich anscheinend sehr über die Aspekte des Veränderungsmanagements, die die **Balance** betreffen. Dazu kann ich zwei Ratschläge geben:

- Denken Sie daran, dass mehr Arbeit fertiggestellt werden muss, als neu begonnen wird, wenn der WIP reduziert werden soll. Wenn es Mitarbeitern nicht klar ist, was zuerst fertiggestellt werden sollte, sollten Sie sich zunächst einmal auf dieses Problem konzentrieren. Organisieren Sie die Arbeit visuell und mithilfe von Regeln und stellen Sie durch effektive Feedbackzyklen sicher, dass sie organisiert bleibt.
- Denken Sie daran, dass das Work-in-Progress-Niveau ein Hebel und ein Symptom ist. Visualisieren Sie die Arbeit; bringen Sie die Ursachen an die Oberfläche; erwarten Sie, dass der WIP mit der Prozessverbesserung reduziert wird.

Kooperation

1. Verbesserungen werden als Experimente gestaltet und strukturiert sowie vi-suell gemanagt.
2. Andere Wissensgebiete werden als Modelle für Verbesserung verwendet (als Wege, Systeme zu betrachten; als Wege, Veränderung zu strukturieren; als technische oder Managementmethoden usw.).
3. Kooperation ist als Quelle von Leistung Verbesserungstreiber und als Gegenmittel für systembedingte Frustration anerkannt.
4. Das System verfolgt Zweck-Fitness und ist dabei offen für Veränderungen von innen (es ist selbstorganisiert).

Denken Sie daran:

- Kooperation ist nicht nur »Nett miteinander sein«; sie ist aber auch nicht nur auf das Problemlösen begrenzt.
- Eine gute Antwort auf Enttäuschung ist, darüber nachzudenken, wie man mit Kooperation der Enttäuschung hätte zuvorkommen können.

Kundenfokus

1. Der Lieferworkflow wird als Prozess der Wissenserarbeitung verstanden, in dem Bedürfnisse, Möglichkeiten und Fähigkeiten erforscht werden.
2. Stromaufwärts eines definierten Zusagepunktes werden Arbeitseinheiten als Optionen behandelt.
3. Stromabwärts der Auslieferung werden Arbeitseinheiten weiter gemanagt, bis ihr Nutzen in den Händen des Kunden validiert wurde.

Die Erfahrung zeigt, dass der letzte Punkt entscheidend ist. Validierung findet paradoixerweise am Ende des Auslieferungsprozesses statt, trotzdem ist es am ehesten dieser Aspekt, der einen kooperativen **Kundenfokus** als wichtigen Punkt im Gesamtprozess etabliert. Im Extrem des Lean Startup ist die Validierung der Motor des Prozesses.

Arbeitsfluss

1. Arbeitseinheiten sind größenmäßig so geschnitten und werden so ausgewählt, dass ein starker und verlässlicher Wertfluss erreicht wird.
2. Die Größe von Batches wird so geschnitten und die Auslieferung der Batches zeitlich so geplant, dass das gesamtwirtschaftliche Ergebnis maximiert wird (und nicht nur die Auslieferungskosten minimiert werden).
3. Arbeitseinheiten mit außerordentlichem Wert oder Risiko werden angemesen behandelt und gemanagt.
4. Das System liefert Arbeitseinheiten, die nicht außergewöhnlich sind, mit zuverlässiger Vorhersagbarkeit.

5. Von Anfang bis Ende gemessen dominiert die Zeit, die aktiv in die Wissenserarbeitung fließt, über die Zeit, die durch Verzögerung (Zeit in Warteschlangen, Multitasking, Blockaden) und sonstige Arten von Arbeit verloren geht.
6. Abhängigkeiten zwischen Arbeitseinheiten und zu anderen Diensten werden zeitnah erkannt und visualisiert.
7. Arbeitseinheiten können unabhängig von ihrer Aufnahme in das System für die Auslieferung eingeplant werden.

Wir setzen hohe Standards für den **Arbeitsfluss**. Die meisten Implementierungen werden es wahrscheinlich nicht schaffen, viele dieser Aspekte überzeugend darzustellen, bis sie ihre Verbesserungsanstrengungen auf Quellen der Unvorhersagbarkeit und Verzögerungen lenken und ernsthaft über die ökonomischen Realitäten ihrer Arbeit nachdenken.

Führung und die Führungsdisziplinen

1. Führung steht allen offen; Führungsverhalten, das Veränderung mit sich bringt, ist es besonders wert, gebührend gewürdigt zu werden.
2. Es gibt ein gemeinsames und anhaltendes Bekennen zur Veränderung, das auf einem sich entwickelnden Verständnis darüber beruht, was wir gerade tun, und der Ausrichtung auf den Zweck aus der Sicht aller Stakeholder.
3. Der Nachweis, dass Veränderung notwendig ist, bleibt immer präsent.
4. Veränderung ist sicher; die nachteiligen Risiken werden identifiziert und abgeschwächt (inklusive des Risikos, dass die Veränderung zu langsam oder überhaupt nicht stattfindet).
5. Die potenziellen Nutzen der Veränderung (Vorteilsrisiken) werden beobachtet und gefördert.
6. Veränderung wird durch Vereinbarung implementiert; die Veränderungspraktiken und die Fähigkeit zur Veränderung selbst stehen im Fokus für Verbesserung.
7. Respekt ist immer präsent; in Zeiten des Wandels wird nie unterschätzt, wie Menschen mit ihren aktuellen Rollen, Praktiken und Organisationsformen verbunden sind.

Lassen Sie bei Ihrer Selbstbeurteilung das Management von Änderungen, die von außerhalb kommen – durch das Topmanagement oder die Personalabteilung –, nicht außen vor. Begegnen Sie diesen mit **Verständnis, Vereinbarung und Respekt**. Das gibt Ihnen die Möglichkeit, in anderen Teilen der Organisation **Führung** zu demonstrieren.

23.3.2 Veränderung visualisieren

Die Beurteilung visualisieren

In Abbildung 23–1 wird das geometrische Mittel² der Noten jeder Kategorie in einem Netzdiagramm³ visualisiert. Hiermit bekommen wir eine alternative, wertbasierte Darstellung des Kanban-Tiefen-Assessment-Tools, die nicht bei Kanbans Praktiken aufhört.

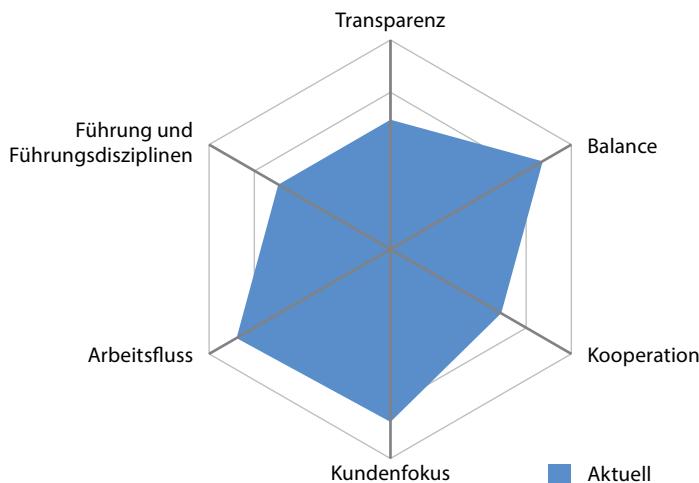


Abb. 23–1 Kanban-Tiefe – nach Werten

Zusätzlich kann es nützlich sein, den historischen Prozess oder die gewünschte Entwicklung zu visualisieren. Mein Freund Ruben Olsen hat mehrere Beurteilungen für dieselben Teams durchgeführt und die Resultate über die Zeit hinweg aufgezeichnet. Alternativ können Sie die Fragen des Assessments auch so beantworten, indem Sie beschreiben, wo Sie sich beispielsweise in drei Monaten befinden wollen. Abbildung 23–2 zeigt, wie diese Herangehensweise visualisiert werden könnte.

-
- 2. Im Vergleich zum einfachen arithmetischen Mittel verstärkt das geometrische Mittel den Effekt niedriger Werte.
 - 3. »Netzdiagramme« werden sie bei Excel genannt; Sie kennen sie vielleicht auch als Kiviat-Diagramme.

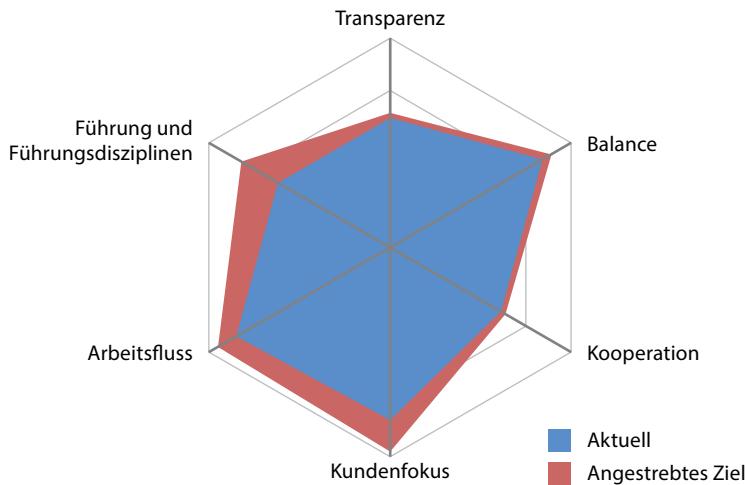


Abb. 23–2 Kanban-Tiefe – mit Entwicklungsverlauf

Veränderung visuell managen

Die Herausforderung besteht darin, konkrete Aktionen so zur Geltung zu bringen, dass ein Impuls erzeugt wird. Dies kann sich sehr nach einem Lieferprozess anfühlen und tatsächlich hilft es, das Ganze mit ähnlichen Werkzeugen zu managen.

Es ist mittlerweile kaum noch möglich, eine wichtige Kanban-Implementierung zu finden, die nicht irgendeine Art visuelles Managementsystem parallel zum Liefersystem unterhält. Diese Hilfssysteme sind für Veränderungen, Probleme, ungewöhnliche Abhängigkeiten usw. vorgesehen.

Ich habe das Design des Problemboards aus Abbildung 23–3 gemeinsam mit Kevin Murray 2012 bei Valtech entwickelt. Wir haben seitdem – gemeinsam und jeder für sich – verschiedene Varianten des Designs verwendet.

Problem	Wird gelöst	Gelöst ☺	
		Täglich	Geschlossen

Abb. 23–3 Das »Problemboard«

Wir betreiben dieses Board wie folgt:

- Jedermann darf neue Probleme in die Eingangsspalte auf der linken Seite hinzufügen.
- Nach täglicher Sichtung und Zuweisung an einen Besitzer bewegen sich die aktiven Probleme vertikal zwischen den Bereichen »Täglich« und »Wöchentlich« unter »Wird gelöst«, entsprechend der Zeit, die wir ihnen in der Diskussion einräumen wollen.
- Einige Zeit, nachdem sie als »Gelöst« erachtet wurden und wir sicher sind, dass sie nicht wieder auftreten werden oder entsprechende Entscheidungen getroffen wurden usw., werden die Probleme in die Spalte »Geschlossen« verschoben.

Ein eher veränderungsfokussiertes Design ist in Abbildung 23–4 dargestellt, das von Jeff Anderson stammt; es ergänzt seinen *Lean Change Canvas* (Kap. 17), kann aber auch unabhängig davon verwendet werden.

Auf Dringlichkeit einigen	Über Veränderung verhandeln	Einführung validieren	Performance verifizieren

Abb. 23–4 Ein Kanban-System für »validierte Veränderung«

Ich mag Jeffs Design sehr. Die beiden ersten Spalten betonen die Vereinbarung; die beiden letzteren betonen die Validierung (und demzufolge den Kundenfokus). Die Trennung der qualitativen Validierung von der quantitativen Verifizierung scheint auch sehr klug zu sein; typischerweise bestätigen Teams gerne Verhaltensänderungen, lange bevor es möglich ist, einen signifikanten Leistungsanstieg zu erkennen.

Nicht jede Veränderung ist gleich

Damals im Jahr 2010 oder 2011 (was uns bei Kanban schon als eine Ewigkeit vorkommt) hat mich mein Freund und langjähriger Kollege Patrick Steyaert zu dem Satz »Nicht jede Arbeit ist gleich« getrieben. Das ist ein Satz, den ich seitdem schon viele Male verwendet habe, auch in diesem Buch.

Mit »Nicht jede Veränderung ist gleich« erinnert Patrick uns an eine weitere nützliche Parallele zwischen dem Management von Lieferungen für Kunden einerseits und Systemveränderungen andererseits. Patrick verweist auf Tomas Kuhnes Buch *The Structure of Scientific Revolutions*⁴ und stellt zwei Achsen dar, über die Veränderung analysiert werden kann:

4. Patricks Post dazu <http://lean-adaptive.com/2014/06/03/not-all-change-is-alike/>; er bezieht sich auf Thomas Kuhne, *The Structure of Scientific Revolutions* [Kuhne 2012].

1. Beinhaltet die Veränderung eine Rekonstruktion der Organisationsfunde, indem es vorherrschendes Denken herausfordert (revolutionäre, *paradigmenbrechende* Veränderung), oder baut sie auf dem auf, was bereits existiert (kumulative, *paradigmenkonsistente* Veränderung)?
2. Ist die Veränderung von außen auferlegt (*reakтив*) oder intern initiiert (*proaktiv*)?

Die beiden Achsen fügen sich zu vier Quadranten zusammen, die in Abbildung 23–5 gezeigt werden. Patrick hat sie auch noch hilfreich beschriftet.



Abb. 23–5 Nicht jede Veränderung ist gleich.

Einige Beobachtungen:

- Bei Kanban geht es nicht darum, Feuerwehr zu spielen oder Existenzkrisen zu beheben, aber wenn Sie sich in so einer Situation befinden und darauf reagieren müssen, kann Kanban vielleicht helfen.
- Kanban einzuführen ist meistens eine proaktive Veränderung und es ist auf das Generieren weiterer proaktiver Veränderung ausgelegt.
- Ob Kanban als Ganzes, die individuellen Prinzipien, Praktiken und Techniken oder die Veränderungen, die es produziert, als paradigmakonsistent oder paradigmabrechend anzusehen sind, hängt sehr stark davon ab, wo Sie sich jetzt befinden. Ihre Organisation könnte vielleicht schon mit Pull-Systemen arbeiten oder sie könnte die Idee befremdlich finden.
- In der Sprache von Patricks Quadranten geht es bei Kanban nicht nur um Verbesserung; es dreht sich auch sehr viel um Zielsetzung.

Kanban schließt radikale Veränderung auch nicht aus. Es fordert allerdings, dass Sie verstehen, wo Sie sich gerade befinden, bevor Sie auf Basis von Vereinbarung und Respekt weitergehen. Wenn Sie das nicht können oder wollen, sind Sie – zumindest was Kanban angeht – alleine unterwegs.

23.4 Abschließende Gedanken

Es scheint mir angemessen, dass ich am Ende des letzten Kapitels die Arbeit anderer Menschen preise. Das Ganze ist eine sehr stark kooperative Leistung, die Arbeit einer Community, die für Ideen offen ist, Veränderung mit offenen Armen empfängt und die Führung bei anderen kultiviert.

Ich bin sehr glücklich, dass ich dieses Buch so schließen kann, wie ich es auch beginnen konnte, mit Werten. Es gab einige Skeptiker, als ich die neun Werte zum ersten Mal darstellte; sie sorgten sich (nicht ohne Grund), dass es nicht Aufgabe der Methode (oder mir) sei, die Werte ihrer Organisationen zu diktieren. Einige warnten mich, dass Werte fragile Dinge sind – man fasst sie an und sie verschwinden!

Als Platzhalter für Praktiken und Nutzen sowie als Werkzeug, um Gedanken, Geschichten, Prioritäten zu organisieren, haben sich die Werte als nicht bedrohlich erwiesen. Sie sind nützlich, um Denken anzuregen, und stellen sich sogar als ziemlich robust heraus. Allen, die diese Möglichkeiten früh erkannt haben, und Ihnen, die Sie jetzt daran teilhaben, möchte ich danken.

Anhang

A Demings 14 Punkte für gutes Management

1. Schaffe ein konstantes Unternehmensziel in Richtung einer ständigen Verbesserung von Produkt und Dienstleistung, um konkurrenzfähig zu werden, auf dem Markt zu bleiben und Arbeitsplätze zu sichern.
2. Wende die neue Philosophie an. Wir befinden uns in einer neuen wirtschaftlichen Ära. Das westliche Management muss für die Herausforderung aufwachen, seine Verantwortungen kennen und die Führung für die Veränderung übernehmen.
3. Beende die Abhängigkeit von Qualitätskontrollen. Beseitige die Notwendigkeit von Masseninspektion, indem Qualität von Anfang an in die Produkte eingebaut wird.
4. Beende die Praxis, Geschäfte auf der Grundlage des niedrigsten Preises zu machen. Minimiere stattdessen die Gesamtkosten. Suche für jedes Lieferstück einen einzelnen Lieferanten und gehe mit ihm eine langfristige Beziehung auf Basis von Loyalität und Vertrauen ein.
5. Verbessere ständig und immer die Systeme der Produktion und der Dienstleistung, um Qualität und Produktivität zu steigern und damit die Kosten zu senken.
6. Institutionalisiere das Lernen bei der Arbeit.
7. Sorge für richtiges Führungsverhalten (vgl. Punkt 12 und Kap. 8 in *Out of the Crisis* [Deming 2000a]). Das Ziel der Supervision besteht darin, Menschen, Maschinen und Werkzeugen zu helfen, bessere Arbeit zu machen. Die Supervision des Managements bedarf der Überarbeitung, wie auch die Supervision der Mitarbeiter in der Produktion.
8. Sorge für eine Atmosphäre ohne Angst, sodass jeder effektiv für das Unternehmen arbeiten kann (vgl. Kap. 4 in *Out of the Crisis* [Deming 2000a]).
9. Beseitige die Abgrenzungen zwischen Abteilungen. Mitarbeiter in der Forschung, dem Design, dem Vertrieb und der Produktion müssen als Team miteinander arbeiten, um Probleme in der Produktion und Nutzung zu erkennen, die mit beim Produkt oder der Dienstleistung auftreten können.

10. Schaffe Anweisungen, Plakate, Ermahnungen und Zielvorgaben für Mitarbeiter ab, die null Fehler und gesteigerte Produktivität verlangen. Solche Ermahnungen erzeugen nur ein feindliches Klima, da der Großteil der Gründe für geringe Qualität und geringe Produktivität im System liegen und damit außerhalb der Verantwortlichkeit der Mitarbeiter.
11. a. Beseitige Leistungsvorgaben in der Produktion. Ersetze sie durch Führung.
b. Beseitige *Management by Objective*. Beseitige Management nach Zahlen und zahlenmäßigen Zielen. Ersetze sie stattdessen durch Führung.
12. a. Beseitige Hindernisse, die dem Mitarbeiter das Recht nehmen, auf seine Arbeit stolz zu sein. Die Verantwortung des Vorgesetzten muss weg von reinen Zahlen in Richtung Qualität geändert werden.
b. Beseitige Hindernisse, die den Managern und Ingenieuren das Recht nehmen, auf ihre Arbeit stolz zu sein. Das heißt unter anderem, dass jährliche Beurteilungen und *Management by Objective* abgeschafft werden müssen.
13. Richte ein durchgreifendes Programm für Ausbildung und Selbstverbesserung ein.
14. Bringe jeden in der Firma dazu, diese Veränderung zu unterstützen. Die Umsetzung ist jedermanns Aufgabe.

B Danksagung

Nur dadurch, dass ich selbst ein Buch geschrieben habe, habe ich wirklich verstanden, warum der Abschnitt mit den Danksagungen in vielen meiner liebsten Bücher so lang ist. Vorne auf dem Cover steht zwar mein Name, aber ich wäre ohne die Ermutigung, Inspiration und direkte Unterstützung vieler Leute, von denen nicht alle hier aufgelistet werden können, nie so weit gekommen.

Als Erstes – und definitiv als Allererste – kommt meine Frau Sharon. Das Wort »wundervoll« kratzt nicht an der Oberfläche, um sie zu beschreiben! Trotz einiger beträchtlicher familiärer Herausforderungen, die in dem einen Jahr auftauchten, in dem ich das Buch geschrieben und herausgebracht habe, hat sie mich zum Schreiben angehalten, den Tagesbeginn um 4:30 Uhr erträglich gemacht und die Stellung gehalten, wenn ich physisch oder geistig abwesend war. In Liebe: danke.

Bei David J. Anderson and Associates und unserem Verlag Blue Hole Press muss ich David selbst danken, Janice Linden-Reed, Wes Harris, Irina Dzhambazova, Agnes Sellgren und unserem früheren Kollegen Dragos Dumitriu. Danke euch für euer individuelles und gemeinsames Engagement für dieses Projekt. Danke auch an unseren Lektor Vicki Rowland – ich hoffe, ich habe nicht allzu viel Ärger bereitet!

Aus der langen Liste der lobenswerten Kollegen, mit denen ich das Vergnügen hatte, zusammen in aktuellen Projekten zu arbeiten, möchte ich folgende namentlich erwähnen: Mark Dickinson, mein früherer Geschäftsführer bei Encore International; Kevin Murray von Valtech; Leigh Mortimer und Joanne Clarkson vom Department for Work and Pensions (DWP); Allon Lister und Steve Wood vom Government Digital Service (GDS). Danke für das Verständnis für was ich tat und für das Vertrauen, dass ich es tun konnte.

Diese netten Menschen haben wichtiges Feedback zu verschiedenen meiner Entwürfe gegeben: Greg Brougham, John Clapham, Kevin Murray, Klaus Leopold, Kelvyn Youngman, Steve Tendon, Wolfgang Wiedenroth und Markus Hippeli. Eure harte Arbeit und Expertise schätze ich sehr.

Ben Linders von InfoQ war so freundlich, eine Serie von Artikeln zu publizieren, die auf einem sehr frühen (und schrecklich dicht geschriebenen) Entwurf des

späteren Kapitels 10 basierten. Ich stehe in der Schuld des immer ehrlichen Luke Hohmann für seine Kommentare in jener entscheidenden Phase und natürlich, dass er später seine wertvolle Zeit für das Geleitwort dieses Buches investiert hat.

Dieses Buch hat an Vielseitigkeit gewonnen durch die kürzlichen Entwicklungen, die der Kanban-Community durch David Anderson, Jeff Anderson (nicht verwandt) und Patrick Steyaert beigesteuert wurden. Für ihre Grundlagenarbeit danke ich Don Reinertsen, Dave Snowden, Chris Matts und Olav Maassen.

Die Kanban-Community musste 2013 viel integrieren, es war ein gutes Jahr! Auch wenn einige Namen schon vorher genannt wurden, schätze ich das Feedback dieser Menschen zu den Materialien bezüglich der Werte, das in Blogposts, auf Konferenzen, Meetups, Diskussionsforen, privaten E-Mails und Ähnlichem entstand: Maria Alfredéen, Markus Andrezak, David J. Anderson, Dimitar Bakardzhiev, Corinna Baldauf, Jabe Bloom, Matthias Bohlen, Royd Brayshay, Paweł Brodzinski, Greg Brougham, Martin Burns, Tom Cagley, Andy Carmichael, Jose Casal, Chris Chan, Fred Engel, Thomas Epping, Zsolt Fabok, Alex Fedtke, Rob Ferguson, Eric Green, Ellen Grove, Torbjörn Gyllebring, Kurt Häusler, Hermanni Hyttiälä, Matthias Jouan, Sigi Kaltenecker, Liz Keogh, Klaus Leopold, Janice Linden-Reed, Simon Marcus, Gaetano Mazzanti, Chris McDermott, Marco Melas, John Miller, Rodolfo Moeller, Pierre Nies, Frode Odegard, Stephen Parry, Ajay Reddy, Arne Roock, Bernd Schiffer, Karl Scotland, David Shrimpton, Mattias Skarin, Dave Snowden, Patrick Steyaert, Jim Sutton, Jon Terry, Simon Thomas, Björn Tikkanen, Dave White, Eric Willeke und Yuval Yeret.

C Glossar

Abfertigungsdisziplin Die Menge der *Regeln*, die festlegen, wie *Arbeitseinheiten* ausgewählt werden. First in, First out (FIFO) und *Weighted Shortest Job First* (WSJF) sind zwei wichtige Beispiele.

Abhängigkeit *Arbeitseinheiten* hängen unter Umständen voneinander ab (in diesem Fall müssen sie vorsichtig sequenziellisiert werden, um eine Blockierung zu vermeiden) oder benötigen Zulieferungen anderer Dienstleister (in diesem Fall müssen sie aus Gründen der Verfügbarkeit umsichtig zeitlich geplant werden).

Agenda In diesem Buch ist eine *Agenda* (oder *Agenda zur Veränderung*) ein dringlicher Aufruf zum Handeln, der auf den organisatorischen Bedürfnissen basiert. Kapitel 10 beschreibt »drei Agenden«, die vordefiniert sind. Wie man um eine kleine Anzahl priorisierter *Werte* eine Agenda formt, ist in Kapitel 23 beschrieben.

Aktivität Im Kontext eines *Workflows* werden Aktivitäten an *Arbeitseinheiten* durchgeführt, die sich in den angemessenen *Status* (Zuständen) befinden; Aktivitäten bringen Arbeitseinheiten häufig von einem Zustand in einen anderen. Aktivitäten und ihre entsprechenden Zustände lassen sich typischerweise auf die Spalten eines *Kanban-Boards* abbilden.

Anpassungsfähigkeit Die Fähigkeit, auf Veränderungen in der Umgebung zu reagieren. Es ist zu beachten, dass sehr stabile Umgebungen Anpassungsfähigkeit nicht notwendigerweise begünstigen. Die Mitarbeiter in diesen Umgebungen tun sich dann besonders schwer, wenn sich die Bedingungen irgendwann ändern. Vgl. auch *Fitness*.

Arbeitseinheit Eine auslieferbare Einheit oder eine Komponente davon (zum Beispiel ein neues Produktfeature). Einzelaufgaben und Aktivitäten sind in diesem Sinne im Allgemeinen keine Arbeitseinheiten. Vgl. *Karte* und *Workflow*.

Avatar Eine bewegliche Markierung, die eine Person repräsentiert. Auf einem *Kanban-Board* können Avatare auf *Karten* platziert werden, um zu zeigen, wer an ihnen arbeitet; die Anzahl der verfügbaren Avatare zu limitieren, ist eine Technik, um Multitasking zu reduzieren.

Backlog Eine Liste von *Arbeitseinheiten*, die noch nicht begonnen wurden. Verwenden Sie diesen Begriff mit Vorsicht, falls damit eventuell eine stärkere Zusage impliziert wird, als wirklich gerechtfertigt wäre.

Batch, Batchgröße, Batchtransfer Eine Gruppe von *Arbeitseinheiten*, die sich zusammen durch das System (oder Teile davon) bewegen. Projekte implizieren häufig große Batchgrößen am Eingang und Ausgang des Systems und große Batchtransfers zwischen den verschiedenen definierten Status. Eine Strategie, bei der kleinere Batches in das System und daraus heraus bewegt werden, verkürzt häufig die *Durchlaufzeit* und senkt den *Work in Progress*. Kapitel 15 behandelt ökonomische Aspekte. Vgl. auch *Transaktionskosten*.

Bearbeitungszeit Die Nettozeit, in der an einer bestimmten *Arbeitseinheit* gearbeitet wird, im Gegensatz zu der Zeit, die diese Einheit in Warteschlangen, blockiert oder durch *Multitasking festgefahren* verbringt. Vgl. auch *Flusseffizienz*.

Beschleunigt, Expedite Beschreibt das Management von *Arbeitseinheiten*, die schnell auf Kosten anderer durch das System fortschreiten müssen. Vgl. *Serviceklassen*.

Blockiert, Blockade, blockierendes Problem Eine *Arbeitseinheit* wird als blockiert bezeichnet, wenn eine unnormale Bedingung sie am Vorwärtskommen hindert. Die unmittelbare Ursache wird häufig als Blockade bezeichnet (häufig ein Fehler oder eine Abhängigkeit) und auf der betreffenden Arbeitseinheit auf dem *Kanban-Board* visualisiert. Vgl. auch *festgefahren*.

Capability Review Ein Review zu Fortschritt und Leistung, das regelmäßig auf Abteilungsebene abgehalten wird.

CD3 Ein Akronym, das von Maersk geprägt wurde: *Cost of Delay* geteilt durch die Dauer, wobei die Dauer die verbleibende *Durchlaufzeit* ist. Die *Abfertigungsdiszplin*, die als *Weighted Shortest Job First* (WSJF) bekannt ist, versucht CD3 zu minimieren. Vgl. Kapitel 15.

Change Agent Ein Change Agent ist meistens ein Individuum, das dafür sorgt, dass Veränderung stattfindet.

Cost of Delay (CoD), Verzögerungskosten Eine Metrik für die Auswirkungen, die das Verzögern einer *Arbeitseinheit* verursacht, um vielleicht einer Einheit mit höheren Verzögerungskosten Vorfahrt einzuräumen. In der fortgeschrittenen Anwendung sind die Verzögerungskosten eine Funktion der Zeit und messen die Auswirkungen auf die Profite im Gesamtlebenszyklus des betroffenen Produkts oder Portfolios.

Cumulative Flow Diagram (CFD), kumulatives Flussdiagramm Ein Diagramm, das durch gestapelte Linien oder Flächen den *Durchsatz* und den *Work in Progress* für mehrere Aktivitäten zeigt. Vgl. Kapitel 17.

Cycle Time Dieser Begriff ist mit Vorsicht zu verwenden! Meistens ist hiermit die *Durchlaufzeit* durch den Teil des »Arbeitsablaufs« im Prozess gemeint (gemessen von dem Punkt, wo die Arbeit beginnt, bis die Einheit bereit zur Lieferung ist), es kann aber auch die *Kundendurchlaufzeit* oder (ganz etwas anderes) der Kehrwert der *Lieferrate* sein.

Dringlichkeitsgetrieben Vgl. *Serviceklasse*.

Durchlaufzeit Nicht näher definiert beschreibt es die Zeit, die eine *Arbeitseinheit* benötigt, um durch einen Prozess fortzuschreiten. Sie wird häufig synonym mit der spezifischeren *Kundendurchlaufzeit* verwendet. Vgl. *Cycle Time*.

Durchsatz Vgl. *Lieferrate*.

Engpass Eine *Aktivität*, deren *Durchsatz* den Durchsatz des Gesamtsystems begrenzt.

Feedback Der Informationsfluss stromaufwärts über ein Produkt oder einen Prozess.

Feedbackzyklus Meistens meint dies die absichtliche Einbeziehung von Feedback in das Design des Prozesses, damit das resultierende Produkt, der resultierende Dienst oder der Lieferprozess selbst kontrolliert und verbessert werden kann. Beachten Sie allerdings, dass nicht alle Feedbackzyklen per Design entstehen; einige sind vielleicht schwer zu identifizieren und sind nicht immer harmlos. Vgl. Kapitel 11.

Festgefahren (stalled) Eine *Arbeitseinheit* ist festgefahren, wenn sie brachliegt, weil das System keine Kapazität zur Bearbeitung hat. Vgl. auch *blockiert* und *verhungern*.

Fitness In seiner evolutionären Bedeutung beschreibt Fitness, wie gut sich etwas an die Umgebung angepasst hat. In Wettbewerbsumgebungen bedeutet relative Fitness einen Vorteil. Vgl. *Anpassungsfähigkeit*.

Fitnesskriterien Die Messkriterien, die zusammen *Fitness* indizieren.

Flusseffizienz Eine wichtige Metrik, die das Verhältnis zwischen *Bearbeitungszeit* und (*Kunden-*)*Durchlaufzeit* angibt. Die Flusseffizienz steigt, wenn *Arbeitseinheiten* weniger Verzögerungen unterliegen. Vgl. Kapitel 19.

Hierarchisches Board Ein *Kanban-Board*, das *Arbeitseinheiten* verschiedener Granularitätsstufen (z.B. Epics und User Stories) managt. Vgl. Kapitel 22.

Histogramm Eine Visualisierung einer Datenverteilung. Im Kontext von Kanban werden Histogramme insbesondere für *Durchlaufzeiten* verwendet.

Input Queue Eine Warteschlange, die am Anfang des Prozesses platziert wird und die *Arbeitseinheiten* beinhaltet, die bald bearbeitet werden, aber noch nicht begonnen wurden. Wird häufig als »Bereit«-Spalte visualisiert. Unter der Annahme, dass eine *Zusage* getroffen wird, zählen die Arbeitseinheiten in der Input Queue zu dem Gesamt-*Work-in-Progress* des Systems. Die Input Queue ist dementsprechend ein guter Ort, um ein *WIP-Limit* anzuwenden.

Kanban Ein überladener Begriff, der häufig verschiedene Bedeutungen zur gleichen Zeit trägt. In *Kanji* (chinesische Buchstaben) geschrieben, bedeutet er »Zeichen« oder »großes, sichtbares Brett«. In *Hiragana* (japanische Buchstaben) geschrieben, bedeutet er »Signalkarte« (sowohl Singular als auch Plural). In technischen Präsentationen zur Mechanik von *Kanban-Systemen* beziehen wir uns auf die letztere Bezeichnung. Informell werden die Verwendung von *Kanban-Systemen* und die *Kanban-Methode* damit gemeint.

Kanban-Board Eine visuelle Organisation von *Karten* (die *kanban*) in einem *Kanban-System*. Typischerweise sind Boards in vertikale *Spalten* und (optional) horizontale *Schwimmbahnen* unterteilt; weitere Dimensionen können durch Farben oder anderen Kartenattribute dargestellt werden. Karten bewegen sich nach rechts durch die Spalten, korrespondierend zu den repräsentierten *Arbeitseinheiten*, die sich durch das System bewegen. *Work-in-Progress-Limits* und andere *Regeln* können ebenfalls visuell dargestellt werden.

Kanban-Linse Ein Weg, um die Organisation, Arbeitsabläufe und Verbesserung von *kreativer Wissensarbeit* zu verstehen. Sein servicebasiertes Modell bietet eher eine alternative Sicht auf die Projektentwicklung und funktionale Organisation als konventionelle Modelle. Zusätzlich lässt sich damit die Einführungsmethodik der *Kanban-Methode STATIK* auf wenige Worte zusammenfassen. Vgl. Kapitel 10.

Kanban-Methode Eine evolutionäre Herangehensweise für Veränderung, die von David J. Anderson durch sechs Kernpraktiken und vier Grundprinzipien beschrieben wurde. Vgl. Kapitel 16.

Kanban-System Ein *Pull-System*, das implementiert wird, indem die Anzahl der Kanban-Karten limitiert wird, die sich im Umlauf befinden.

Kanban-System-Durchlaufzeit Die *Durchlaufzeit* durch den Teil des Systems, der über *Work-in-Progress-Limits* verwaltet wird.

Karte (oder Ticket) Eine visuelle Repräsentation einer *Arbeitseinheit*.

Kreative Wissensarbeit *Wissensarbeit*, die darauf fokussiert ist, Kundenbedürfnisse durch *Wissenserarbeitungsprozesse* und kreatives Problemlösen zu befriedigen. Viele Formen der Produktentwicklung und Dienstleistung sowohl innerhalb als auch außerhalb des Technologiesektors stimmen mit dieser Definition überein.

Kundendurchlaufzeit Eine bestimmte *Durchlaufzeit* – die von Arbeitseinheiten benötigte Zeit, um durch das System zu kommen –, wahrgenommen aus Kundensicht. Typischerweise wird diese von Anforderung bis Auslieferung gemessen.

Lieferrate Auch als *Durchsatz* bekannt (was manchmal ein finanzieller Messwert ist), ist die Lieferrate die Anzahl der vom System (oder von Teilen des Systems) gelieferten *Arbeitseinheiten* pro Zeiteinheit.

Limit Vgl. *Work-in-Progress-Limit*.

Littles Gesetz Ein wesentliches Ergebnis der Warteschlangentheorie, das die Ankunftsrate eines Prozesses (für die wir stellvertretend die *Lieferrate* verwenden), die *Durchlaufzeit* und den *Work in Progress* in Beziehung setzt. Das Gesetz ist nach John Little benannt, der 1961 den ersten Beweis dafür veröffentlichte. Vgl. Kapitel 17.

Modell Wie in der Einleitung von Teil III beschrieben, kann ein Modell: 1.) das Beispiel anderer sein, das kopiert werden kann; 2.) aus richtungsweisenden Mustern, Vorlagen oder Routinen bestehen, die den eigenen Handlungen oder dem Denken eine Struktur geben; 3.) ein Weg sein, die Welt basierend auf einem Satz definierter Annahmen zu verstehen; 4.) eine Menge von Ergebnissen, die als Konsequenz dieser Annahmen erwartet werden können. Die *Kanban-Methode* enthält eine konkrete Aufforderung an seine Anwender, Modelle zu verwenden, um evolutionäre Veränderung durch Kooperation und Experimentieren zu inspirieren, zu strukturieren und zu lenken.

Multitasking Ein Zustand, in dem ein Individuum oder Team mehr als eine *Arbeitseinheit* gleichzeitig bearbeitet. Die Gesamtmenge des *Work in Progress* (WIP) eines Systems zu kontrollieren, bedeutet, Multitasking zu kontrollieren; der Umkehrschluss ist nicht zwingend wahr.

Operations Review Ein Meeting auf Bereichsebene, typischerweise einmal pro Monat abgehalten, bei dem verschiedene Teams ihre Leistungsdaten, Fehlerberichte und Aktuelles zu Verbesserungen miteinander, (idealerweise) mit Repräsentanten der Kunden und mit der Restorganisation teilen. Vgl. auch *Service Delivery Review*.

Opportunitätskosten Der ökonomische Nutzen, der verfliegt, wenn zwischen Alternativen ausgewählt wird.

PDCA-Zyklus Der kanonische Verbesserungszyklus, der auch als Deming-Zyklus, Shewhart-Zyklus oder nur PDCA bekannt ist. Das Akronym steht für »Plan, Do, Check, Act«; manchmal heißt es auch PDSA für »Plan, Do, Study, Act«. Der Zyklus beschreibt ein (im wissenschaftlichen Sinn) experimentelles Vorgehen, um Verbesserungen zu strukturieren. Vgl. Kapitel 3, 11 und 14.

Personal Kanban Eine Anwendung von *Kanban* auf die Arbeitslast eines Einzelnen oder eines Teams. Im gleichnamigen Buch [Benson & DeMaria Barry 2011] stellen Jim Benson und Tonianne DeMaria Barry die beiden wichtigsten Praktiken heraus, um »*die richtige Arbeit zur richtigen Zeit auszuwählen*«: 1.) Arbeit visualisieren. 2.) Arbeit limitieren. Vgl. Kapitel 16.

Portfolio-Kanban Die Anwendung der *Kanban-Methode* auf das Management eines Projektportfolios.

Puffer Eine *Warteschlange*, die bewusst vor einem *Engpass* oder einer anderen kritischen Aktivität platziert wird, um eine kontinuierliche Versorgung mit Arbeit sicherzustellen.

Proto Kanban Eine frühe Kanban-Implementierung, die *Multitasking* adressiert, die aber noch nicht die Menge an *Work in Progress* (WIP) zwischen einzelnen Aktivitäten kontrolliert.

Pull-System Eine weit gefasste Kategorie von Arbeitsmanagementsystemen, in denen Arbeit nur gestartet wird – »gepullt« von flussaufwärts –, wenn Kapazität verfügbar ist. *Kanban-Systeme* sind Pull-Systeme; die Verfügbarkeit von Kapazität und die Möglichkeit, Arbeit zu ziehen, wird durch die Lücke zwischen dem aktuellen *Work in Progress* und dem korrespondierenden *Limit* angezeigt. Vgl. auch *Push*.

Push Arbeit in ein System oder eine Aktivität zu geben, ohne zu berücksichtigen, ob dafür verfügbare Kapazität vorhanden ist.

Queue, Warteschlange Ein Ort in einem *Workflow* (auf einem *Kanban-Board* typischerweise durch eine *Spalte* dargestellt), in dem *Arbeitseinheiten* vor einer folgenden *Aktivität* festgehalten werden.

Regel Eine explizite Beschreibung erwarteten Verhaltens oder Prozessbegrenzung. Regeln, die mit *Kanban-Systemen* verbunden werden, beinhalten »Definition of Ready« und »Definition of Done« auf Spalten-Ebene. *Work-in-Progress-Limits* werden auch als Regeln bezeichnet.

Release Das Ausliefern an den Kunden oder das Produkt, das ausgeliefert wird.

Replenishment, Replenishment-Meeting Das Befüllen der *Input Queue* und das Meeting, das dazu dient.

Run-Chart Ein Diagramm, das Vorkommnisse über die Zeit hinweg aufzeigt. Run-Charts werden üblicherweise für die Visualisierung von *Durchlaufzeiten* verwendet.

Safe-to-fail-Experiment Ein Experiment, das so gestaltet ist, dass nur sehr begrenzte Auswirkungen entstehen, falls es fehlschlägt.

Schwimmbahn Vgl. *Swimlane*.

Scrumban Die Anwendung der *Kanban-Methode* im Kontext einer existierenden Scrum-Implementierung. Umgangssprachlich gesagt ist es Kanban, wenn das, »was du gerade tust«, Scrum ist.

Selbstorganisation Die Fähigkeit des Systems, sich selbst, ohne Intervention von außen, zu ändern.

Service; Serviceorientierung Ein System, das geschaffen wurde, um den Verbrauchern einen Nutzen zu stiften; ein Paradigma, das auf Services, ihren Interaktionen und den für den Kunden generierten Resultaten basiert. Vgl. *Kanban-Linse*.

Service Delivery Review (oder System Capability Review) Ein wöchentlich oder zweiwöchentlich abgehaltenes Überprüfungs- oder Fortschritt-und-Leistungs-Meeting, das auf Abteilungsebene abgehalten wird. Vgl. *Operations Review*.

Serviceklassen Breit definierte Kundenerwartungen für verschiedene Untermengen der Gesamtarbeitslast. Sie beeinflussen die Auswahlentscheidungen, die innerhalb des Systems getroffen werden. Verschiedene Serviceklassen werden typischerweise mit qualitativ unterschiedlichen Risikoprofilen assoziiert, insbesondere bezüglich zeitlichen Planungsrisiken und Verzögerungskosten.

Es haben sich vier generische Serviceklassen etabliert: »Standard«, »Termingeschrieben«, »Beschleunigt« und »Unbestimmbare Kosten«. Ich beschreibe die ersten beiden häufig als »dringlichkeitsgetrieben« und »termingetrieben«.

Spalte, Spaltenlimit Vertikale Spalten organisieren die Arbeit auf einem *Kanban-Board* typischerweise nach *Status*, sodass sie sich nach rechts bewegen, wenn sie in Richtung Fertigstellung voranschreiten.

Spaltenlimits sind *Work-in-Progress-Limits* (*WIP-Limits*), die eine Beschränkung auf die Anzahl der *Arbeitseinheiten* legen, die sich zu jedem Zeitpunkt in einer Spalte befinden dürfen.

Standard Die Basis-Serviceklasse, die für *Arbeitseinheiten* vorgesehen ist, die weder *beschleunigt*, *termingeschrieben* oder *unbestimmt* sind.

Standup-Meeting Ein regelmäßiges (häufig tägliches) Treffen, das kurz genug ist (typischerweise bis zu 15 Minuten), dass alle Teilnehmer dabei stehen können. Vgl. Kapitel 1.

STATIK Ein Akronym für »Systems Thinking Approach to Implementing Kanban«, eine empfohlene Herangehensweise, um die *Kanban-Methode* einzuführen und voranzubringen. Wird in Teil III beschrieben.

Status Der Gesamtzustand einer *Arbeitseinheit*, aus dem folgt, wo im System sie sich befinden sollte und welche *Aktivität(en)* für sie stattfinden sollten.

Swimlane (Schwimmbahn) Auf einem *Kanban-Board* ist eine Swimlane eine horizontale Bahn, auf der *Karten* wandern. Swimlanes organisieren die Karten in verschiedenen Kategorien; Karten bewegen sich typischerweise nicht zwischen den Swimlanes (das impliziert, dass die Kategorien relativ stabil sind, wie zum Beispiel Arbeitseinheiten von der Größe Epic oder Projekt, Kunden oder *Serviceklassen*).

System Capability Review Vgl. *Service Delivery Review*.

Termingeschrieben Beschreibt *Arbeitseinheiten*, die kurz vor oder zu einem bestimmten Datum geliefert werden müssen. Typischerweise reagieren die *Verzögerungskosten* dieser Arbeitseinheiten sehr sensibel auch auf kleine Änderungen des Lieferdatums um dieses fixe Datum herum und ein gewisses aktives Management ist angebracht. Vgl. *Serviceklassen*.

Termingetrieben Vgl. *Serviceklassen*.

Ticket (oder Karte) Die visuelle Darstellung einer *Arbeitseinheit* auf einem *Kanban-Board*. Ziemlich häufig handelt es sich dabei um eine Karte (die z.B. von einem Magneten auf dem Board gehalten wird) oder eine Haftnotiz.

Transaktionskosten Die Gemeinkosten, die die Batchgröße beeinflussen. Die Reduzierung der wirklich festen Kosten jeder Transaktion (oder die Erkenntnis, dass der größere Teil der Transaktionskosten tatsächlich eine Funktion der Batchgröße ist) erlaubt es uns, die Transaktionen kleiner zu machen. Vgl. Kapitel 15.

Überlastung In der üblichen Verwendung bedeutet es, Menschen oder Prozessen mehr Arbeit zu geben, als sie effektiv oder menschlich akzeptabel leisten können. Enger definiert bedeutet es, mehr *Work in Progress (WIP)* im System (oder in einem Teil davon) zu halten, als nachhaltig wäre.

Unbestimmbare Kosten (Intangible) Beschreibt *Arbeitseinheiten*, deren kurzfristiger ökonomischer Wert schwer zu quantifizieren ist, deren Vorhandensein aber langfristig lebenswichtig für die Gesundheit und die Leistungsfähigkeit des Systems ist. Wird häufig für präventive Wartungsarbeiten, Experimente, Systemverbesserungen usw. angewendet. Vgl. *Serviceklassen*.

Unlimitierte (unendliche) Warteschlange Eine Warteschlange, die keine Begrenzung hat.

Verhungern Ein Zustand, in dem Mitarbeiter oder Aktivitäten keine Arbeit haben, weil der Arbeitsfluss stromaufwärts unzureichend ist. Vgl. auch *festgefahren*.

Weighted Shortest Job First (WSJF) Eine *Abfertigungsdisziplin*, die darauf abzielt, die Verzögerungskosten zu minimieren, indem *Arbeitseinheiten* vorgezogen werden, die die größten wirtschaftlichen Auswirkungen in Bezug zu ihrer verbleibenden Restzeit besitzen. Vgl. Kapitel 15.

Werte In diesem Buch werden Werte als Eigenschaften angesehen, für die es eine breite Vereinbarung gibt, dass sie erstrebenswert sind, die eine Art Richtung geben (weil generell »mehr davon besser ist«), die dazu dienen, hilfreiche Praktiken und Artefakte zu empfehlen, zu organisieren und zu repräsentieren. Die neun Werte, die in Teil I beschrieben sind, sind aus den Grundprinzipien und Kernpraktiken der *Kanban-Methode* abstrahiert worden. Andere Denkschulen und andere Unternehmenskulturen werden andere Werte betonen und herausstellen; Werte können deshalb auch dazu dienen, Vergleiche vorzunehmen und die passende Wahl zu treffen.

Wissensarbeit Arbeit, bei der es hauptsächlich darum geht, wie Wissen verwendet und entwickelt wird (nach Peter Drucker). Vgl. *creative Wissensarbeit*.

Wissenserarbeitungsprozess Eine Möglichkeit, den Prozess von *creative Wissensarbeit* zu verstehen. Wir erkennen dabei, wie wenig wir am Anfang des Prozesses wissen, und erhalten Hilfe, die unterschiedlichen Arten von Wissen zu identifizieren, die in jeder Phase des Prozesses erschlossen werden.

Workflow Die Folge von *Aktivitäten* oder (weitgehend äquivalent) *Status* der *Arbeitseinheiten*, die dazu führt, dass Produkte oder Dienste geliefert werden. Workflows überspannen häufig funktionale Strukturen einer Organisation, allerdings nicht immer optimal. Vgl. auch *Wissenserarbeitungsprozess*.

Work in Progress (WIP) Work in Progress beschreibt auf Systemebene die Arbeit, die bereits angefangen wurde, die das System aber noch nicht wieder verlassen hat. Wir interessieren uns in der *Kanban-Methode* auch dafür, wo sich der WIP gerade befindet, und messen und kontrollieren die Anzahl der *Arbeitseinheiten*, die einen bestimmten *Status* oder eine *Aktivität* besetzen. Wir können auch die Zuordnung des WIP in andere Kategorien wie zum Beispiel Kunden, Arbeitstypen oder *Serviceklassen* steuern. Vgl. Kapitel 2 und 21.

Work-in-Progress-Limit (WIP-Limit) Eine *Regel*, die die Menge an *WIP* begrenzt, die sich in einem bestimmten Teil des Systems befinden darf. Typischerweise wird sie als Zahl ausgedrückt – die maximale Anzahl von *Arbeitseinheiten*. WIP-limitierte Systeme sind *Pull-Systeme*. Minimale Limits können auch verwendet werden, um *Replenishment* auszulösen.

Zusage, Zusagepunkt (commitment, commitment point) *Arbeitseinheiten* sind zugesagt, wenn eine starke Erwartung mit dem Kunden geteilt wird, dass die Arbeit an den Einheiten jetzt beginnen oder fortgeführt werden soll.

Falls eine Zusage existiert, wird der Punkt, an dem der Übergang zwischen nicht zugesagtem und zugesagtem Status typischerweise passiert (vielleicht als Resultat eines *Replenishment-Meetings*), Zusagepunkt genannt. Auf einem *Kanban-Board* wird er durch eine Linie zwischen Spalten dargestellt.

Es kann auch ein zweiter Zusagepunkt später im Prozess existieren: Der Punkt, an dem entschieden wird, ob Arbeitseinheiten alleine oder in einem *Batch* freigegeben, geliefert oder bereitgestellt werden.

D Literatur

- [Achouiantz & Nordin 2014] Achouiantz, C., Nordin, J.: *The Kanban Kick Start Field Guide: Create the Capability to Evolve*, 2014;
<http://leanagileprojects.blogspot.co.uk/2013/11/the-kanban-kick-start-field-guide-v11.html>.
- [Ackoff 1991] Ackoff, R. L.: *Ackoff's Fables: Irreverent Reflections on Business and Bureaucracy*. Hoboken, NJ: John Wiley & Sons, 1991.
- [Agile Manifesto] Manifest für Agile Softwareentwicklung;
<http://www.agilemanifesto.org/iso/de>.
- [Anderson 2010] Anderson, D. J.: *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, WA: Blue Hole Press, 2010. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Kanban – Evolutionäres Change Management für IT-Organisationen* 2011 im dpunkt.verlag, Heidelberg erschienen.]
- [Anderson 2012] Anderson, D. J.: *Lessons in Agile Management: On the Road to Kanban*. Sequim, WA: Blue Hole Press, 2012.
- [Anderson 2012a] Anderson, D. J.: *How Deep is Your Kanban?* 2012;
<http://www.djaa.com/sites/ltdwip/DepthOfKanban.pdf>.
- [Anderson 2013] Anderson, J.: *The Lean Change Method: Managing Agile Transformation Using Kanban, Kotter, and Lean Startup Thinking*. LeanPub, 2013.
- [Beck 1999] Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 1999.
- [Benson & DeMaria Barry 2011] Benson, J., DeMaria Barry, T.: *Mapping Work | Navigating Life*. Seattle: Modus Cooperandi, 2011. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Personal Kanban – Visualisierung und Planung von Aufgaben, Projekten und Terminen mit dem Kanban-Board* 2013 im dpunkt.verlag, Heidelberg erschienen.]
- [Brodzinski o.J.] Brodzinski, P.: blog posts tagged »project portfolio«;
<http://brodzinski.com/tag/project-portfolio>.

- [Burrows o.J.] Burrows, M.: blog posts tagged »portfolio«;
<http://positiveincline.com/index.php/tag/portfolio-2>.
- [Carroll o.J.] Carroll, I.: blog posts tagged »portfolio management«;
<http://iancarroll.com/category/portfolio-management>.
- [Collins 2001] Collins, J.: *Good to Great*. New York: HarperBusiness, 2001.
- [Deming 2000a] Deming, E. W.: *Out of the Crisis*. Reprint, The Mit Press, 2000.
- [Fulmer & Keys 1998] Fulmer, R., Keys, J. B.: *A conversation with Peter Senge: New developments in organizational learning*. Organizational Dynamics, vol. 27, no. 2, p. 33 (Autumn 1998).
- [Gall 2003] Gall, J.: *The Systems Bible: The Beginner's Guide to Systems Large and Small*. 3rd ed., Walker, MN: General Systemantics, 2003.
- [Goldratt & Cox 2004] Goldratt, E. M., Cox, J.: *The Goal: A Process of Ongoing Improvement*. 3rd ed., North River Press, 2004.
- [Håkan 2012] Håkan, F.: *Kanban Kata*, 2012–13;
<http://hakanforss.wordpress.com/tag/kanban-kata>.
- [Hammarberg & Sundén 2014] Hammarberg, M., Sundén, J.: *Kanban in Action*. Shelter Island, NY: Manning Publications, 2014.
- [Hansen 2009] Hansen, M.: *Collaboration: How Leaders Avoid the Traps, Build Common Ground, and Reap Big Results*. Harvard Business School Press, 2009.
- [Hohmann 2006] Hohmann, L.: *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Addison-Wesley Professional, 2006.
- [Hubbard 2010] Hubbard, D. W.: *How to Measure Anything: Finding the Value of Intangibles in Business*. 2nd ed., Wiley, 2010.
- [Humble & Farley 2010] Humble, J., Farley, D.: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Signature Series (Fowler), 2010.
- [Kaner 2007] Kaner, S.: *Facilitator's Guide to Participatory Decision-Making*. 2nd ed., Jossey-Bass Business & Management, 2007.
- [Kuhne 2012] Kuhne, T.: *The Structure of Scientific Revolutions*. University of Chicago Press, 50th anniversary ed., 2012.
- [Lafley & Martin 2013] Lafley, A. G., Martin, R. L.: *Playing to Win, How Strategy Really Works*. Harvard Business Press, Boston, 2013.
- [Leopold 2014] Leopold, K.: *Kanban Flight Levels*, 2014;
www.klausleopold.com/kanban-flight-levels.
- [Leopold & Kaltenecker 2013] Leopold, K., Kaltenecker, S.: Kanban in der IT: Eine Kultur der kontinuierlichen Verbesserung schaffen. Hanser Verlag, München, 2013.

- [Logan, King & Fischer-Wright 2008] Logan, D., King, J., Fischer-Wright, H.: *Tribal Leadership: Leveraging Natural Groups to Build a Thriving Organization*. Harper Business, 2008.
- [Maassen, Matts & Geary 2013] Maassen, O., Matts, C., Geary, C.: *Commitment*. Hathaway te Brake Publications, 2013.
- [Meadows et al. 1972] Meadows, D. H., Meadows, D. L., Randers, J., Behrens, W. W.: *The Limits to Growth*. New York: Universe Books, 1972.
- [Mezick 2012] Mezick, D.: *The Culture Game: Tools for the Agile Manager*. FreeStanding Press, 2012.
- [Middleton & Sutton 2005] Middleton, P., Sutton, J.: *Lean Software Strategies: Proven Techniques for Managers and Developers*. New York: Productivity Press, 2005.
- [Ohno 1998] Ohno, T.: *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, 1998.
- [Poppendieck & Poppendieck 2006] Poppendieck, M., Poppendieck, T.: *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley Professional, 2006.
- [Reinertsen 2009] Reinertsen, D. G.: *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.
- [Ries 2011] Ries, E.: *The Lean Startup*. Crown Business, 2011.
- [Rother 2009] Rother, M.: *Toyota Kata: Managing People for Improvement, Adaptiveness and Superior Results*. McGraw-Hill, 2009.
- [Sawyer 2008] Sawyer, K.: *Group Genius: The Creative Power of Collaboration*. New York: Basic Books, 2008.
- [Seddon 2003] Seddon, J.: *Freedom from Command and Control: A Better Way to Make the Work Work*. Buckingham, UK: Vanguard Consulting Ltd., 2003.
- [Senge 2006] Senge, P.: *The Fifth Discipline: The Art and practice of the learning organization*. 2nd rev. ed., Random House, 2006. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Die fünfte Disziplin: Kunst und Praxis der lernenden Organisation* in der 11. Auflage 2011 im Verlag Schäffer-Poeschel erschienen.]
- [Snowden & Boone 2007] Snowden, D. J., Boone, M. E.: *A Leader's Framework for Decision Making*. Harvard Business Review, Nov. 2007.
- [Tabaka 2006] Tabaka, J.: *Collaboration Explained, Facilitation Skills for Software Project Leaders*. Addison-Wesley Professional, 2006.
- [Tendon & Müller 2013] Tendon, S., Müller, W.: *Tame the Flow*. 2013; leanpub.com/tame-the-flow.

- [Tendon & Müller 2015] Tendon, S., Müller, W.: *Hyper-Productive Knowledge Work Performance, The TameFlow Approach and Its Application to Scrum and Kanban.* J. Ross Publishing Inc., 2015.
- [Weinberg 1992] Weinberg, G. M.: *Quality Software Management, Volume 1: Systems Thinking.* New York: Dorset House, 1992.
- [Whitmore 2010] Whitmore, J.: *Coaching for Performance: GROWing Human Potential and Purpose: The Principles and Practice of Coaching and Leadership.* 4th ed., Nicholas Brealey Publishing, 2010.
- [Womack & Jones 2003] Womack J. P., Jones D. T.: *Lean Thinking: Banish Waste And Create Wealth In Your Corporation.* New edition, Simon & Schuster UK Ltd, 2003.
- [Womack, Jones & Roos 2007] Womack, J. P., Jones, D. T., Roos, D.: *The Machine That Changed the World: The Story of Lean Production – Toyota's Secret Weapon in the Global Car Wars That Is Now Revolutionizing World Industry.* Reprint, Free Press, 2007.
- [Yeret 2013] Yeret, Y.: *Pull-based Change Management, 2013-14;*
<http://yuvalyeret.com/2014/05/27/pull-based-change-management>

Weiterführende Literatur zu einzelnen Themen

Systemdenken (Kap. 11)

- [Deming 2000b] Deming, W. E.: *The New Economics.* 2nd ed., Cambridge, MA: MIT Press, 2000.
- [Gall 2003] Gall, J.: *The Systems Bible: The Beginner's Guide to Systems Large and Small.* 3rd ed., Walker, MN: General Systemantics, 2003.
- [Meadows 2008] Meadows, D. H.: *Thinking in Systems: A Primer.* White River Junction, VT: Chelsea Green, 2008.
- [Seddon 2003] Seddon, J.: *Freedom from Command and Control: A Better Way to Make the Work Work.* Buckingham, UK: Vanguard Consulting Ltd., 2003.
- [Weinberg 1992] Weinberg, G. M.: *Quality Software Management, Volume 1: Systems Thinking.* New York: Dorset House, 1992.

Lernen (Kap. 11)

- [Kahneman 2001] Kahneman, D.: *Thinking, Fast and Slow.* Farrar, Straus and Giroux, 2011.
- [Noonan 2007] Noonan, W. R.: *Discussing the Undiscussable: A Guide to Overcoming Defensive Routines in the Workplace.* Jossey-Bass Business & Management, 2007.

- [Patterson et al. 2007] Patterson, K., Grenny, J., Mcmillan, R., Switzler, A.: *Influencer: The Power to Change Anything*. McGraw-Hill Professional, 2007.
- [Patterson et al. 2011] Patterson, K., Grenny, J., Mcmillan, R., Switzler, A.: *Crucial Conversations: Tools for Talking When Stakes Are High*. 2nd ed., McGraw-Hill Professional, 2011.
- [Senge 2006] Senge, P.: *The Fifth Discipline: The Art and practice of the learning organization*. 2nd rev. ed., Random House, 2006. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Die fünfte Disziplin: Kunst und Praxis der lernenden Organisation* in der 11. Auflage 2011 im Verlag Schäffer-Poeschel erschienen.]
- [Shulz 2011] Shulz, K.: *Being Wrong: Adventures in the Margin of Error: The Meaning of Error in an Age of Certainty*. Portobello Books Ltd., 2011.

Engpasstheorie (TOC) (Kap. 12)

- [Goldratt & Cox 2004] Goldratt, E. M., Cox, J.: *The Goal: A Process of Ongoing Improvement*. 3rd ed., North River Press, 2004.
- [Tendon & Müller 2013] Tendon, S., Müller, W.: *Tame the Flow*. leanpub.com/tame-the-flow, 2013.
- [Youngman o.J.] Youngman, K.: *A Guide to Implementing the Theory of Constraints (TOC)*; www.dbrmfg.co.nz.

Agile (Kap. 13)

- [Agile Manifesto] *Manifest für Agile Softwareentwicklung*; <http://www.agilemanifesto.org/iso/de>.
- [Ambler o.J.] Ambler, S.: *Feature Driven Design (FDD) and Agile Modeling*; www.agilemodeling.com/essays/fdd.htm.
- [Beck 1999] Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 1999.
- [Humble & Farley 2010] Humble, J., Farley, D.: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Signature Series (Fowler), 2010.
- [Rubin 2012] Rubin, K. S.: *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 2012.
- [Schwaber & Sutherland o.J.] Schwaber, K., Sutherland, J.: *The Scrum Guide*; www.scrum.org/Scrum-Guide.
- [Sheridan 2013] Sheridan, R.: *Joy, Inc.: How We Built a Workplace People Love*. Portfolio, 2013.

TPS und Lean (Kap. 14)

- [Middleton & Sutton 2005] Middleton, P., Sutton, J.: *Lean Software Strategies: Proven Techniques for Managers and Developers*. Productivity Press, 2005.
- [Ohno 1998] Ohno, T.: *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, 1998.
- [Poppdieck & Poppdieck 2006] Poppdieck, M., Poppdieck, T.: *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley Professional, 2006.
- [Reinertsen 2009] Reinertsen, D. G.: *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.
- [Ries 2011] Ries, E.: *The Lean Startup*. Crown Business, 2011.
- [Rother 2009] Rother, M.: *Toyota Kata: Managing People for Improvement, Adaptiveness and Superior Results*. McGraw-Hill, 2009.
- [Spear 2010] Spear, S. J.: *The High-Velocity Edge: How Market Leaders Leverage Operational Excellence to Beat the Competition*. 2nd ed., McGraw-Hill, 2010.

Ökonomische Ansätze zum Arbeitsfluss (Kap. 15)

- [Arnold & Yüce 2013] Arnold, J. J., Yüce, Ö.: *Black Swan Farming using Cost of Delay*, 2013; <http://blackswanfarming.com/experience-report-maersk-line>.
- [Hubbard 2010] Hubbard, D. W.: *How to Measure Anything: Finding the Value of Intangibles in Business*. 2nd ed., Wiley, 2010.
- [Maassen, Matts & Geary 2013] Maassen, O., Matts, C., Geary, C.: *Commitment*. Hathaway te Brake Publications, 2013.
- [Reinertsen 2009] Reinertsen, D. G.: *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Die Kanban-Methode (Kap. 16)

- [Achouiantz 2012] Achouiantz, C.: *Assessing the Depth of a Kanban Implementation*, 2012; <http://www.slideshare.net/ChrisAch/depth-of-a-kanban-implementation>.
- [Anderson 2010] Anderson, D. J.: *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, WA: Blue Hole Press, 2010. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Kanban – Evolutionäres Change Management für IT-Organisationen* 2011 im dpunkt.verlag, Heidelberg erschienen.]
- [Anderson 2012] Anderson, D. J.: *Lessons in Agile Management: On the Road to Kanban*. Sequim, WA: Blue Hole Press, 2012.
- [Anderson 2012a] Anderson, D. J.: *How Deep is Your Kanban?* 2012; <http://www.djaa.com/sites/ltdwip/DepthOfKanban.pdf>.

- [Benson & DeMaria Barry 2011] Benson, J., DeMaria Barry, T.: *Mapping Work | Navigating Life*. Seattle: Modus Cooperandi, 2011. [Anm.: Die deutsche Übersetzung dieses Buches ist unter dem Titel *Personal Kanban – Visualisierung und Planung von Aufgaben, Projekten und Terminen mit dem Kanban-Board* 2013 im dpunkt.verlag, Heidelberg erschienen.]
- [Brodzinski o.J.] Brodzinski, P.: blog posts tagged »project portfolio«;
<http://brodzinski.com/tag/project-portfolio>.
- [Burrows o.J.] Burrows, M.: blog posts tagged »portfolio«;
<http://positiveincline.com/index.php/tag/portfolio-2>.
- [Carroll o.J.] Carroll, I.: blog posts tagged »portfolio management«;
<http://iancarroll.com/category/portfolio-management>.
- [Hammarberg & Sundén 2014] Hammarberg, M., Sundén, J.: *Kanban in Action*. Shelter Island, NY: Manning Publications, 2014.
- [Håkan 2012] Håkan, F.: *Kanban Kata*, 2012–13;
<http://hakanforss.wordpress.com/tag/kanban-kata/>.
- [Leopold 2014] Leopold, K.: *Kanban Flight Levels*, 2014;
www.klausleopold.com/kanban-flight-levels.

Kleinere Modelle (Kap. 17)

- [Anderson 2013] Anderson, J.: *The Lean Change Method: Managing Agile Transformation Using Kanban, Kotter, and Lean Startup Thinking*. LeanPub, 2013.
- [Hansen 2009] Hansen, M.: *Collaboration: How Leaders Avoid the Traps, Build Common Ground, and Reap Big Results*. Harvard Business School Press, 2009.
- [Hohmann 2006] Hohmann, L.: *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Addison-Wesley Professional, 2006.
- [Kaner 2007] Kaner, S.: *Facilitator's Guide to Participatory Decision-Making*. 2nd ed., Jossey-Bass Business & Management, 2007.
- [Logan, King & Fischer-Wright 2008] Logan, D., King, J., Fischer-Wright, H.: *Tribal Leadership: Leveraging Natural Groups to Build a Thriving Organization*. Harper Business, 2008.
- [Mezick 2012] Mezick, D.: *The Culture Game: Tools for the Agile Manager*. FreeStanding Press, 2012.
- [Shook 2008] Shook, J.: *Managing to Learn: Using The A3 Management Process to Solve Problems, Gain Agreement, Mentor, and Lead*. Lean Enterprise Institute, 2008.
- [Sobek II & Smalley 2008] Sobek II, D. K., Smalley, A.: *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*. Productivity Press, 2008.

[Tabaka 2006] Tabaka, J.: *Collaboration Explained, Facilitation Skills for Software Project Leaders*. Addison-Wesley Professional, 2006.

[TastyCupcakes] TastyCupcakes.org: *Fuel for Invention and Learning*;
<http://tastycupcakes.org>.

[Whitmore 2010] Whitmore, J.: *Coaching for Performance: GROWing Human Potential and Purpose: The Principles and Practice of Coaching and Leadership*. 4th ed., Nicholas Brealey Publishing, 2010.

Ein Kanban-System einführen (Kap. 23)

[Achouiantz & Nordin 2014] Achouiantz, C., Nordin, J.: *The Kanban Kick Start Field Guide: Create the Capability to Evolve*, 2014;
<http://leanagileprojects.blogspot.co.uk/2013/11/the-kanban-kick-start-field-guide-v11.html>.

[Anderson 2013] Anderson, J.: *The Lean Change Method: Managing Agile Transformation Using Kanban, Kotter, and Lean Startup Thinking*. LeanPub, 2013.

[Yeret 2013] Yeret, Y.: *Pull-based Change Management*, 2013-14;
<http://yuvalyeret.com/2014/05/27/pull-based-change-management>

Index

A

- Abfertigungsdisziplin 136
- Abhängigkeit 200
- Agenda 77
 - für Veränderung 77–78
 - Nachhaltigkeits-~ 79, 82
 - Serviceorientierungs-~ 79–80, 82
 - Überlebensfähigkeits-~ 79, 81–82
- Agile-Modell 118
- agile Softwareentwicklung, zwölf Prinzipien 109
- Agiles Manifest 109
- Akzeptanztesten, automatisiertes 113
- Analyse, quantitative 178
- Andon 119–121
- Anforderung 208
- Anpassungsfähigkeit 145
- Anti-Pattern 55
- Arbeitsfluss (Flow) 37–38, 42, 45, 49, 53, 75, 80, 205, 214
- Arbeitspaket 5, 192
- Arbeitstyp 192, 201
- aufbrechen/zusammenführen 203–204
- Auslastung 194
- auslieferbares Inkrement 114
- automatisiertes Akzeptanztesten 113
- Autonomation 119
- Avatar 201, 203
- A3 123, 158

B

- Balance 13, 15, 17, 20–21, 23, 29, 35, 48, 74, 79, 205, 213
- Batch 16, 133
 - Größe 122, 133

Baue (Build) 126

Bearbeitungszeit 181

Bedürfnisse 36

beschleunigt *siehe Expedite*

Board, hierarchisches 202

Bottom-up-Organisation 185, 188

Buffer-Management 101, 107

Build *siehe Baue*

C

- Capacity-Constrained Resource *siehe kapazitätsbegrenzte Ressource (CCR)*
- CCPM *siehe Critical-Chain-Projektmanagement*
- CCR *siehe kapazitätsbegrenzte Ressource (Capacity-Constrained Resource)*
- CD3 136
- CFD *siehe Cumulative Flow Diagram*
- Change
 - Agent 66–67
 - Canvas 211
 - Management 66
 - Team 69
- Chaos 156
- CoD *siehe Verzögerungskosten (Cost of Delay)*
- Commitment-Point *siehe Zusagepunkt*
- Continuous Delivery 148
- Cost of Delay *siehe Verzögerungskosten (CoD)*
- Critical-Chain-Projektmanagement (CCPM) 103, 107
- Cumulative Flow Diagram (CFD) 11, 180
- Cynefin-Framework 91

D

- Daily Scrum 114
DBR *siehe Drum-Buffer-Rope*
Deming-Zyklus 26
Denkprozess (Thinking Processes) 102–103, 107
Divergenz 161
Doppelschleifen-Lernen 94
Dot-Voting *siehe Punkt-Wahl*
Draufgängertum 56, 75
dringlichkeitsgetrieben 18
Drum-Buffer-Rope (DBR) 101, 107
Durchsatzrechnung (Throughput Accounting) 105

E

- echte variable Kosten (TVC) 105
Einmischung 58, 75
Engpass 100, 106
Theorie (TOC) 99, 102, 105
Epic 202
Expedite (beschleunigt) 19, 135, 191
Experiment 26
Extreme Programming (XP) 111

F

- Feature-Driven Development (FDD) 110
Feedback
Mechanismus 15
Zyklus 8, 10–11, 24
Fehler 199
Ursachen-Analyse 119
Fertigstellung, Zustand der ~ 188
fester Liefertermin 135
fester Termin *siehe auch termingetrieben* 19, 191
FFD *siehe Feature-Driven Development*
FIFO *siehe First in, First out*
First in, First out (FIFO) 19, 191
Fitness 63
Kriterium 128
Zweck~ 145
Five Whys *siehe 5-Why-Methode (5W)*
Fixkosten 130
Flaschenhals 100, 106
Fließband stoppen 121
Flight Level 150
Flow *siehe Arbeitsfluss*
Flusseffizienz 181
Fokussierungsschritt 100
fünf ~ 106

- Framework, Cynefin~ 91
Führung (Leadership) 6, 47–51, 53, 75, 80, 97, 215
kooperative 164
Führungsdisziplin 71, 75, 145, 215
fünf Fokussierungsschritte 106

G

- Gemba 120
Gesetz, Littles ~ 153
Gleichmäßigkeit 37
Goal, Reality, Option, Will (GROW) 157
GP *siehe Grundprinzip*
Groan Zone *siehe Zone des Ächzens*
GROW *siehe Goal, Reality, Option, Will*
Grundprinzip (GP) 1, 63, 71, 145
GP 1: Beginne mit dem, was du gerade tust. 1, 47, 55, 145
(erweitert): Beginne mit dem, was du gerade tust, und versteh. 60
GP 2: Vereinbare, dass evolutionäre Veränderung verfolgt wird. 1, 47, 63, 145
GP 3: Respektiere initial bestehende Prozesse, Rollen, Verantwortlichkeiten und Jobtitel. 1, 47, 71, 145
GP 4: Ermutige dazu, Führung auf jeder Ebene der Organisation zu zeigen – vom einzelnen Mitarbeiter bis zum höheren Management. 1, 48, 145

H

- Haltekosten (Cost of Carry) 129, 133, 137
Hebelpunkte 88
Heijunka 120–121
hierarchisches Board 202

I

- Inkrement
auslieferbares 114
Veränderungs~ 212
inkrementell 126
Input Queue *siehe auch Backlog* 10
iterativ 114

J

- JIT *siehe Just-in-Time*
J-Kurve 59, 64
Just-in-Time (JIT) 119, 122

K

- Kaikaku 120
Kaizen 120
Kanban 4, 119–120
 Kata 149
 Linse 80, 209
 Methode 143
 Personal ~ 146
 Portfolio~ 147
 System 208
 Team~ 146
 Tiefen-Assessment-Tool 149, 216
 Upstream~ 33–34
kapazitätsbegrenzte Ressource (CCR) 100
Karte 5
Kategorisierung, risikobasierte 18
Kausalität 91
Kernpraktik (KP) 1, 145
 KP 1: Visualisiere. 1, 4, 47, 145
 KP 2: Limitiere die Menge paralleler Arbeit (Work in Progress, WIP). 1, 13, 47, 145
 KP 3: Manage den Arbeitsfluss. 1, 31, 37, 47, 145
 (erweitert): Manage den Arbeitsfluss, um Gleichmäßigkeit, Pünktlichkeit, und gute ökonomische Ergebnisse zu erreichen, und nehme Kundenbedürfnisse vorweg. 31, 37
 KP 4: Mache Prozessregeln explizit. 2, 4, 7, 47, 145
 KP 5: Implementiere Feedbackzyklen. 2, 4, 8, 47, 145
 KP 6: Erzielle Verbesserung kooperativ, entwickle experimentell (unter Verwendung von Modellen und der wissenschaftlichen Methode). 2, 23, 47, 85, 145
Kette, kritische 104
komplexes adaptives System 92
Komplexität 87, 89
Konvergenz 161
Kooperation (Zusammenarbeit) 23–25, 27, 29–30, 35, 48, 74, 79, 97, 109, 205, 214
kooperative Führung 164

Kosten

- echte variable (TVC) 105
 Fix~ 130
 Halte~ 129, 133, 137
 Opportunitäts~ 133
 Transaktions~ 134
 unbestimmbare 19, 135, 191
 Verzögerungs~ (CoD) 19, 125, 129, 133–134

KP *siehe Kernpraktik*

kreative Wissensarbeit 33, 209

kritische Kette 104

Kundenerwartung 191

Kundenfokus 31, 33, 35, 45, 48, 53, 74, 80, 205, 214

L

Lagerbestand 133

Layers of Resistance *siehe Widerstandsschichten*

Leadership *siehe Führung*

Lean

- Change Canvas 160, 218
 Manufacturing 124
 Prinzip 123
 Product Development (LPD) 124
 Production 119
 Startup 126
 UX 126

Learn *siehe Lerne*

Leistungsfähigkeit 175, 208

Lerne (Learn) 126

lernen 93

 Doppelschleifen~ 94

lernende Organisation 87, 93, 96

Liefertermin, fester 135

Limited WIP Society 144

Littles Gesetz 153

LPD *siehe Lean Product Development*

M**Management**

- Portfolio~ 147
 visuelles 121, 217

Manifest, Agiles 109

Measure *siehe Messe*

Meeting 10

- Replenishment~ 10
 Standup~ 9, 39, 148

- Mentor 67
 Messe (Measure) 126
 Metrik 11
 Modell 85
 - Agile-~ 118
 - A3-~ 123, 158
 - GROW-~ 157
 - Neun-Schichten-~ 102
 - Satir-~ 156
 Muster 77
- N**
- Nachhaltigkeits-Agenda 79, 82
 Neun-Schichten-Modell 102
NIA *siehe nicht sofort verfügbare Ressource (Non-Instant Availability)*
 nicht sofort verfügbare Ressource (NIA) 100
Non-Instant Availability *siehe nicht sofort verfügbare Ressource (NIA)*
 Nordstern (True North) 122
 numerisches WIP-Limit 203
- O**
- Ohno, Taiichi 119
 Operations Review 10
 Opportunitätskosten 133
 Option 139
 Organisation, lernende 87, 93, 96
- P**
- parallele Arbeit *siehe auch Work in Progress (WIP)* 13
 Pareto 130
 Participatory Decision-Making 160
 Pattern, Anti-~ 55
PDCA *siehe Plan, Do, Check, Act*
PDSA *siehe Plan, Do, Study, Act*
 Personal Kanban 146
 Pivot 126
 Plan, Do, Check, Act (PDCA) 26, 123
 - Zyklus 26
 Plan, Do, Study, Act (PDSA) 26
PO *siehe Product Owner*
POOGI *siehe Process of Ongoing Improvement*
 Portfolio
 - Kanban 147
 - Management 147
 Positionierung 209
- Process of Ongoing Improvement (POOGI)
 - 100, 106, 123
- Product**
 - Backlog 114
 - Owner (PO) 114
- Projekt 129
 Prozessregel 7–8
 Prozess, Wissenserarbeitungs-~ 209
 Pull-System 13–14, 128
 Punkt-Wahl (Dot-Voting) 162, 171
Purpose *siehe Zweck*
 Pyramidenprinzip 159
- Q**
- quantitative Analyse 178
 Quelle der Unzufriedenheit 169, 208
- R**
- radikale Veränderung 219
 Real Option 129, 139
 Regulär *siehe auch Standard* 19
 Replenishment-Meeting 10
 Respekt 71, 73–75, 81, 122
 Return of Investment (ROI) 132
 Rhythmus von Scrum 115
 risikobasierte Kategorisierung 18
ROI *siehe Return of Investment*
 Rollen 71
 Rope 102, 107
- S**
- SAFe 127
 Satir-Modell 156
 Schnellboot 163
 Schwimmbahn (Swimlane) 201, 203
 Scrum 114
 - Master (SM) 114
 - Rhythmus 115
 - Scrumban 116, 148
 - Selbstorganisation 6
 - Selbstzufriedenheit 56, 75
 - Serious Game 162
 - Service Delivery Review 10
 - Service Level Agreement (SLA) 192
 - Serviceklasse 18–20, 125, 134, 191, 201, 208
 - Serviceorientierungs-Agenda 79–80, 82
 - Shewhart-Zyklus 26
 - Single-Piece-Flow 122
 - SLA *siehe Service Level Agreement*

- SM *siehe Scrum Master*
Softwareentwicklung, agile, zwölf Prinzipien 109
Sprint 114, 203
 Backlog 114
 Planning 114
 Planung 148
 Retrospektive 114
 Review 114, 149
Stakeholder 20
Standard 19, 191
Standup-Meeting 9, 39, 148
STATIK-0 151, 210
STATIK *siehe Systems Thinking Approach to Introducing Kanban*
Status 198
Swimlane *siehe Schwimmabahn*
System
 Capability Review 10
 Denken 87, 97
 Dynamik 88
 Kanban 208
 komplexes adaptives 92
 Pull-~ 128
Systems Thinking Approach to Introducing Kanban (STATIK) 150, 167, 207
- T**
Team-Kanban 146
termingetrieben *siehe auch fester Termin 18–19, 191*
Theory of Constraints (TOC) *siehe Engpasstheorie*
Thinking Processes *siehe Denkprozess*
Throughput Accounting *siehe Durchsatzrechnung*
Ticket 5
Timebox 115
TOC *siehe Engpasstheorie (Theory of Constraints)*
Top-down-Dekomposition 185–186
Totally Variable Costs *siehe echte variable Kosten (TVC)*
Toyota-Produktionssystem (TPS) 119
TPS *siehe Toyota-Produktionssystem*
Transaktionskosten 134
Transparenz 3, 5, 7–8, 12, 23–24, 29, 35, 48, 73, 79, 97, 212
Triade 164
Tribal Leadership 164
- True North *siehe Nordstern*
TVC *siehe echte variable Kosten (Totally Variable Costs)*
- U**
Überlebensfähigkeits-Agenda 79, 81–82
unbestimmbare Kosten 19, 135, 191
Unzufriedenheit, Quelle der 169, 208
Upstream-Kanban 33–34
User Experience (UX) 126
UX *siehe User Experience*
- V**
Value Stream Mapping *siehe Wertstromanalyse (VSM)*
Variabilität 170
Veränderung 8, 60, 64, 156, 216
 radikale 219
Veränderungssinkrement 212
Vereinbarung 63, 66, 69, 76, 81, 97
Verschwendung 123
Verständnis 55, 60, 75, 81, 97
Verzögerung 43
Verzögerungskosten (CoD) 19, 125, 129, 133–134
Verzögerungsquelle 154
Visualisierung 5
visuelles Management 121, 217
Vorhersagbarkeit 20
VSM *siehe Wertstromanalyse (Value Stream Mapping)*
- W**
Warteschlange 187
Warteschlangentheorie 124, 154
Weighted Shortest Job First (WSJF) 136
Wertstrom 123
Wertstromanalyse (VSM) 185
Widerstand 103
Widerstandsschichten (Layers of Resistance) 102
WIP *siehe Work in Progress*
Wissen 93
Wissensarbeit 13
 kreative 33, 209
Wissenserarbeitungsprozess 209
Work in Progress (WIP) 13, 203
 Limit 13–16, 203
Workflow 185, 208–209
WSJF *siehe Weighted Shortest Job First* 136

X

XIT 143, 186
XP *siehe Extreme Programming*

Z

Zone des Ächzens (Groan Zone) 161
Zufriedenheit 32
Zusagepunkt (Commitment-Point) 187
Zusammenarbeit *siehe Kooperation*
zusammenführen/aufbrechen 203–204
Zustand der Fertigstellung 188
Zweck (Purpose) 61, 209
 Fitness 145
zwölf Prinzipien agiler Software-
entwicklung 109

Zyklus

Deming-~ 26
Feedback-~ 8, 10–11, 24
PDCA-~ 26
Shewhart-~ 26

Ziffern

12 Prinzipien agiler Softwareentwicklung
 109
5 Fokussierungsschritte 106
5W *siehe 5-Why-Methode (Five Whys)*
5-Why-Methode (5W) 119
9-Schichten-Modell 102