

Alexander Schütz

Komplexität von IT-Architekturen

Konzeptualisierung, Quantifizierung,
Planung und Kontrolle



Springer Gabler

Komplexität von IT-Architekturen

Alexander Schütz

Komplexität von IT-Architekturen

Konzeptualisierung, Quantifizierung,
Planung und Kontrolle

Mit einem Geleitwort von Prof. Dr. Peter Buxmann



Springer Gabler

Alexander Schütz
Darmstadt, Deutschland

Dissertation, Technische Universität Darmstadt, 2016

D 17

ISBN 978-3-658-18225-0 ISBN 978-3-658-18226-7 (eBook)
DOI 10.1007/978-3-658-18226-7

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

Springer Gabler

© Springer Fachmedien Wiesbaden GmbH 2017

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Gabler ist Teil von Springer Nature

Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Geleitwort

Ein sich jederzeit in Bewegung befindliches Umfeld stellt Unternehmen immer wieder vor neue Herausforderungen. Wurde IT in ihren Anfangszeiten als Mittel zum Zweck gesehen, so stellt eine gut aufgestellte IT mittlerweile häufig einen Wettbewerbsvorteil gegenüber der Konkurrenz dar. Lag der klassische Fokus bei der Steuerung der Unternehmens-IT auf einer reinen Ausrichtung auf die Steigerung der Effizienz, so erfordert die moderne Rolle der IT in Unternehmen ein Umdenken. Zunehmend IT-affine Kunden und ein fortschreitender Digitalisierungsbedarf der Geschäftsprozesse der Unternehmen treffen häufig auf historisch gewachsene, verbaute IT-Landschaften, an denen Anpassungen häufig nur mit relativ großem Aufwand durchführbar sind.

Bestehende Ansätze in der Forschung setzen den Fokus bei dem Management der IT stark auf Standardisierung. Die damit verbundene Tendenz zu Rigidität bei IT-Entscheidungen kann dabei die Zielsetzung der Erreichung einer flexiblen und an neue Anforderungen gut anpassbaren IT gefährden. Herr Schütz lenkt mit seiner Arbeit den Fokus weg von einer rein auf die Standardisierung gerichteten Perspektive hin zu einer integrierten Perspektive auf das Management der Komplexität der IT-Architektur. Die Arbeit liefert Erkenntnisse, die wesentlich zum Verständnis von Komplexität in IT-Architekturen und deren Entstehung beitragen. Neben der Einführung einer Konzeptualisierung von Komplexität in IT-Architekturen wird in der vorliegenden Arbeit eine einheitliche, in allen Architekturebenen anwendbare Form der Quantifizierung von Komplexität vorgestellt. Die vorgenommene Unterscheidung in erforderliche und unnötige Komplexität erlaubt Herrn Schütz zudem, die Ursachen der Entstehung vermeidbarer Komplexität sowie die hierbei beteiligten Mechanismen zu untersuchen. Im Rahmen einer Design Science-Studie führt Herr Schütz darüberhinaus Designprinzipien ein, die beschreiben, wie ein Informationssystem zum Management von Komplexität in IT-Architekturen ausgestaltet sein sollte.

Alexander Schütz leistet mit seiner Arbeit einen wichtigen Beitrag zur Forschung in der Wirtschaftsinformatik und speziell für das Management von IT-Architekturen. Aufgrund des Praxisbezugs der in der Arbeit behandelten Forschungsfragen sowie der engen Zusammenarbeit mit Unternehmen, konnte bereits eine Anwendbarkeit der Forschungsergebnisse in der Praxis gezeigt werden. In diesem Sinne wünsche ich der Arbeit eine weite Verbreitung in Forschung und Praxis.

Prof. Dr. Peter Buxmann

Vorwort

Die vorliegende Dissertation ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Wirtschaftsinformatik | Software Business & Information Management an der Technischen Universität Darmstadt entstanden.

Bereits im Zuge meiner praktischen Erfahrungen vor der Rückkehr an die TU Darmstadt kam ich regelmäßig in Kontakt mit historisch gewachsenen, hoch komplexen IT-Architekturen, welche die zuständigen Mitarbeiter vor diverse Probleme bei dem Management der IT stellten. Da ich dies als hoch interessante Problemstellung empfand, freute ich mich sehr darüber, dass Prof. Dr. Peter Buxmann mir 2012 die Gelegenheit bot, an seinem Fachgebiet in diesem Themenfeld zu forschen. Die folgenden vier Jahre am Lehrstuhl waren eine schöne Zeit, in der ich sehr viele tolle Menschen kennenlernen durfte.

Mein besonderer Dank gilt Prof. Dr. Peter Buxmann und Prof. Dr. Thomas Widjaja für die Betreuung und Unterstützung während meiner Zeit am Lehrstuhl. Ein herzlicher Dank gilt Prof. Dr. Alexander Benlian für die Übernahme des Korreferats. Zudem danke ich Prof. Dr. Robert Wayne Gregory, Dr. Christian Schmidt, Martin und Ulli für die tolle Zusammenarbeit und all die wertvollen Gespräche, die wir in den vier Jahren geführt haben. Außerdem möchte ich mich auch bei den Praxispartnern bedanken, deren Erfahrungen und Feedback zu den Ergebnissen dieser Arbeit beigetragen haben.

Weiter möchte ich mich bei Adrian, Amina, André, Anton, Christoph, Francesco, Helena, Hendrik, Jasmin, Jin, Katrin, Markus, Maggi, Martin, Nicole, Nihal, Rabea, Ruth, Sven, Stefan, Thorsten und Tobias für die schöne gemeinsame Zeit am Fachgebiet bedanken. Vielen Dank an die fleißigen Probeleser der einzelnen Kapitel der Dissertation.

Ein besonderer Dank gilt meinen Eltern Brigitte und Klaus und meinem Bruder Marcus für die Unterstützung in den letzten Jahren. Schließlich möchte ich mich bei meiner Lebensgefährtin Karen bedanken, die mir in allen – noch so stressigen – Phasen der Entstehung der Arbeit stets zur Seite stand und mich immer unterstützt und motiviert hat.

Alexander Schütz

Inhaltsübersicht

Inhalt	XI
Abbildungsverzeichnis.....	XV
Tabellenverzeichnis	XIX
Abkürzungsverzeichnis.....	XXI
1 Einleitung.....	1
2 Grundlagen.....	9
3 Strukturelle Komplexität in Unternehmens-/IT-Architekturen	29
4 Qualitative Untersuchung der Ursachen und Auswirkungen von Komplexität in IT-Architekturen	51
5 Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen	107
6 Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen	147
7 Fazit und Implikationen	209
Literatur.....	221
Anhang.....	239

Inhalt

Inhaltsübersicht	IX
Abbildungsverzeichnis	XV
Tabellenverzeichnis	XIX
Abkürzungsverzeichnis	XXI
1 Einleitung.....	1
1.1 Relevanz der Arbeit.....	2
1.1.1 Praktische Relevanz	2
1.1.2 Wissenschaftliche Relevanz	4
1.2 Struktur und Aufbau der Arbeit	6
2 Grundlagen.....	9
2.1 Systemtheorie.....	9
2.1.1 Allgemeiner Systembegriff	9
2.1.2 Wurzeln der modernen Systemtheorie	11
2.1.3 Komplexe adaptive Systeme	13
2.2 Architekturen.....	16
2.2.1 Allgemeiner Architekturbegriff.....	16
2.2.2 Der Architekturbegriff im Kontext von Informationssystemen	17
2.2.3 Unternehmensarchitektur	18
2.2.4 IT-Architektur	21
2.2.5 Architekturmanagement	22
2.2.6 Die Rolle des Architekten	27
2.3 Theoretischer Bezugsrahmen der Arbeit.....	28
3 Strukturelle Komplexität in Unternehmens-/IT-Architekturen	29
3.1 Strukturierte Literaturrecherche zu Komplexität in der IS-Forschung	29
3.1.1 Vorgehen bei der strukturierten Literaturrecherche	29
3.1.2 Literaturanalyse	32
3.1.3 Forschungsagenda	41
3.2 Systemtheoretische Konzeptualisierung von Komplexität in Unternehmensarchitekturen.....	43
3.2.1 Unternehmens-/IT-Architektur als System	44
3.2.2 Verwendung des Konzeptes „Komplexität“ in der IS-Literatur	45

3.2.3	Vorschlag einer Konzeptualisierung von Komplexität in Unternehmensarchitekturen	47
3.3	Zusammenfassung und Diskussion der Ergebnisse	48
3.3.1	Theoretische Implikationen.....	49
3.3.2	Praktische Implikationen.....	49
4	Qualitative Untersuchung der Ursachen und Auswirkungen von Komplexität in IT-Architekturen	51
4.1	Motivation	51
4.2	Theoretische Grundlagen und zugrunde liegende Annahmen	52
4.2.1	IT-Architekturen als künstliche Systeme	52
4.2.2	Zusammenhang zwischen Geschäftsprozess- und IT-Architektur.....	53
4.2.3	Einführung der behandelten Untersuchungsfragen	56
4.3	Methodisches Vorgehen.....	58
4.3.1	Design der Studie	58
4.3.2	Datenerhebung	60
4.3.3	Auswertung der Daten.....	62
4.4	Ergebnisse der qualitativen Untersuchung.....	64
4.4.1	Trigger von Veränderungen der IT-Architektur.....	65
4.4.2	Einfluss von Anforderungen auf die „minimale IT-Komplexität“.....	69
4.4.3	Mechanismen, die ein Abweichen von der „minimalen IT-Komplexität“ erklären.....	75
4.4.4	Durch strukturelle Komplexität beeinflusste Zieleigenschaften des Systems	90
4.4.5	Darstellung der Ergebnisse anhand einer Auswahl von Projekten	98
4.5	Zusammenfassung und Diskussion der Ergebnisse	101
4.5.1	Theoretische Implikationen.....	103
4.5.2	Praktische Implikationen.....	103
4.5.3	Limitationen und weiterer Forschungsbedarf	104
5	Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen.....	107
5.1	Motivation	107
5.2	Verwandte Literatur	108
5.3	Methodisches Vorgehen.....	109
5.4	Anforderungen an ein Maß zur Quantifizierung von Heterogenität	111
5.4.1	Zusätzliche Charakteristiken	112
5.4.2	Effekt kleiner Gruppen von Elementen.....	113
5.4.3	Verschiebung in der Klassifikation	114
5.4.4	Effekt proportionaler Änderungen	115
5.5	Analyse häufig verwendeter Konzentrationsmaße.....	117
5.5.1	Herfindahl-Hirschman-Index	118
5.5.2	Horvath-Index	120

5.5.3	Shannon-Entropie.....	123
5.5.4	Zusammenfassung der Analyse der vorgestellten Maße.....	126
5.6	Richness und Evenness – Zwei Maße aus der Biologie.....	126
5.6.1	Richness	128
5.6.2	Evenness.....	128
5.7	Vorschlag für ein Maß zur Quantifizierung von Komplexität.....	130
5.7.1	Zuordnung der identifizierten Maßzahlen.....	130
5.7.2	Ein auf der Shannon-Entropie basierendes Maß	131
5.8	Anwendung des Maßes	132
5.8.1	Fall 1: Unternehmen aus der Finanzbranche	133
5.8.2	Fall 2: Unternehmen aus dem Transportsektor	136
5.8.3	Fall 3: Unternehmen aus der Versicherungsbranche.....	138
5.9	Zusammenfassung und Diskussion der Ergebnisse	143
5.9.1	Theoretische Implikationen.....	144
5.9.2	Praktische Implikationen	144
5.9.3	Limitationen und weiterer Forschungsbedarf	145
6	Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen	147
6.1	Motivation	147
6.2	Methodische Grundlagen	148
6.2.1	Artefakt.....	149
6.2.2	Design Science Research	150
6.2.3	Action Research	151
6.2.4	Action Design Research	152
6.3	Definition der betrachteten Problemklasse „Planung und Kontrolle von Komplexität in IT-Architekturen“.....	155
6.4	Grundlegendes Vorgehen im Rahmen des Forschungsprojektes.....	156
6.5	Erstes ADR-Projekt mit Bank A	159
6.5.1	Vorgehen	159
6.5.2	Ergebnisse des ersten ADR-Projektes.....	164
6.6	Zweites ADR-Projekt mit Bank B	176
6.6.1	Vorgehen	176
6.6.2	Ergebnisse des zweiten ADR-Projektes.....	178
6.7	Erweiterung der Ergebnisse des ersten ADR-Projektes.....	187
6.7.1	Vorgehen	187
6.7.2	Ergebnisse des dritten ADR-Projektes	188
6.8	Zusammenfassung und Diskussion der Ergebnisse	201
6.8.1	Theoretische Implikationen.....	201

6.8.2	Praktische Implikationen	205
6.8.3	Limitationen und weiterer Forschungsbedarf	208
7	Fazit und Implikationen	209
7.1	Zusammenfassung	209
7.2	Wissenschaftlicher Beitrag der Arbeit	211
7.3	Beitrag der Arbeit für die Unternehmenspraxis	214
7.4	Weitere Forschungsmöglichkeiten	217
Literatur		221
Anhang		239

Abbildungsverzeichnis

Abbildung 1:	Struktur und Aufbau der Dissertation	6
Abbildung 2:	Funktionales Konzept nach Ropohl (2009)	10
Abbildung 3:	Struktureles Konzept nach Ropohl (2009)	10
Abbildung 4:	Hierarchisches Konzept nach Ropohl (2009)	11
Abbildung 5:	Reiz-Reaktions-Schema	13
Abbildung 6:	Abgrenzung Unternehmens-/IT-Architektur nach Schütz et al. (2013a)	22
Abbildung 7:	Theoretischer Bezugsrahmen der Arbeit	28
Abbildung 8:	Strukturierte Literaturübersichten nach vom Brocke et al. (2009)	30
Abbildung 9:	Taxonomie einer Literaturübersicht nach Cooper (1988) – Darstellung in Anlehnung an vom Brocke et al. (2009)	30
Abbildung 10:	Verteilung der Publikationsorgane nach Häufigkeit	32
Abbildung 11:	Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen .	48
Abbildung 12:	Schema der Veränderung der IT-Architektur	52
Abbildung 13:	Komplexität der Geschäftsprozess- und IT-Architektur	53
Abbildung 14:	Güte einer Transformation in einer spezifischen Entscheidungssituation	54
Abbildung 15:	Beispiele der Veränderung der strukturellen Komplexität	55
Abbildung 16:	Verortung der Untersuchungsfragen	57
Abbildung 17:	Übersicht über den Coding-Prozess	62
Abbildung 18:	Struktur der Ergebnisse der qualitativen Untersuchung	64
Abbildung 19:	Vorschau auf die identifizierten Kategorien	65
Abbildung 20:	Einordnung des Einflusses von Anforderungen auf $K_{ITA,min}$ in den Untersuchungskontext	70
Abbildung 21:	Einflüsse auf $K_{ITA,min}$	70
Abbildung 22:	Einordnung der Mechanismen – die $\Delta K_{ITA,unnötig}$ erklären – in den Untersuchungskontext	75
Abbildung 23:	Einordnung der Architekturmaßnahmen in den Untersuchungskontext	86

Abbildung 24: Einordnung der Zieleigenschaften der IT-Architektur in den Untersuchungskontext.....	91
Abbildung 25: Vermuteter Zusammenhang der Heterogenität der IT-Architektur und der IT-Flexibilität.....	92
Abbildung 26: Ausprägungen für die Projekte A und B.....	98
Abbildung 27: Ausprägungen für Projekt C	99
Abbildung 28: Ausprägungen für Projekt D	100
Abbildung 29: Ausprägungen für Projekt E.....	100
Abbildung 30: Überblick über die Ergebnisse der qualitativen Untersuchung.....	101
Abbildung 31: Heterogenität eingesetzter DBs anhand der absoluten Häufigkeiten.....	111
Abbildung 32: Zusätzliche Charakteristiken.....	112
Abbildung 33: Eliminieren einer Charakteristik	113
Abbildung 34: Effekt kleiner Gruppen von Elementen	114
Abbildung 35: Verschiebung in der Klassifikation.....	115
Abbildung 36: Effekt proportionaler Änderungen.....	116
Abbildung 37: Zusätzliche Charakteristiken – HHI.....	119
Abbildung 38: Zusätzliche Charakteristiken – HOR	121
Abbildung 39: Verlauf der Funktionen 1-HOR, 1-HHI sowie EM für alle möglichen Verteilungen der Elemente zweier Charakteristiken	122
Abbildung 40: Zusätzliche Charakteristiken – EM.....	124
Abbildung 41: Gewichtungen der Anteile	125
Abbildung 42: Veränderung der Komplexität bei Durchführung der Transformation (1) ..	142
Abbildung 43: Veränderung der Komplexität bei Durchführung der Transformation (2) ..	142
Abbildung 44: Phasen der ADR-Methode nach Sein et al. (2011).....	153
Abbildung 45: Entwicklung der DPs im Rahmen des dreistufigen Forschungsprojektes ...	157
Abbildung 46: Verwendete „IT-Dominant BIE“	162
Abbildung 47: Konzeptualisierung von Komplexität in IT-Architekturen.....	165
Abbildung 48: Verortung der Designprinzipien in IT-COM	167
Abbildung 49: Zeitreihenanalyse in IT-COM.....	169
Abbildung 50: „IT-Dominant BIE“ des zweiten ADR-Projektes	177
Abbildung 51: Designprinzipien IT-COM_2 in der Architekturansicht.....	179
Abbildung 52: Detailansicht der Kennzahlen in IT-COM_2.....	180

Abbildung 53: Zusätzliche Visualisierungen der Kennzahlen in IT-COM_2.....	182
Abbildung 54: Simulation von Änderungen in IT-COM_2.....	184
Abbildung 55: Darstellung der Simulationsergebnisse in IT-COM_2	185
Abbildung 56: Darstellung von Modularität in IT-COM_3.....	192
Abbildung 57: Zweite Form der Visualisierung der Modularität in IT-COM_3	193
Abbildung 58: Ausprägung und Darstellung der Kennzahlen in IT-COM_3.....	194
Abbildung 59: Kennzahlen und Drill-Down in IT-COM_3	196
Abbildung 60: Visualisierung der Kennzahlen in IT-COM_3.....	197
Abbildung 61: Auswahl der Dimensionen des Scatter Plots in IT-COM_3	197
Abbildung 62: Beispielhafter Scatter Plot in IT-COM_3	198
Abbildung 63: Detailinformationen in IT-COM_3.....	198
Abbildung 64: Simulation in IT-COM_3.....	200
Abbildung 65: Beiträge des Forschungsprojektes.....	202

Tabellenverzeichnis

Tabelle 1:	Charakteristiken komplexer adaptiver Systeme	15
Tabelle 2:	Ebenen einer Unternehmensarchitektur.....	19
Tabelle 3:	Ausgewählte Definitionen des Begriffs Unternehmensarchitektur	20
Tabelle 4:	IS-Forschungsbeiträge im EA-Kontext mit Standardisierungs-Linse	26
Tabelle 5:	Konzeptmatrix – Konzeptualisierung von Komplexität	46
Tabelle 6:	Übersicht der befragten Experten der qualitativen Untersuchung.....	61
Tabelle 7:	Anforderungen an ein Maß zur Quantifizierung von Heterogenität.....	117
Tabelle 8:	Effekt kleiner Gruppen von Elementen – HHI	120
Tabelle 9:	Effekt kleiner Gruppen von Elementen – HOR.....	122
Tabelle 10:	Effekt kleiner Gruppen von Elementen – EM	125
Tabelle 11:	Ergebnisse der Analyse der betrachteten Maße.....	126
Tabelle 12:	Evenness bei einer dominanten sowie n-1 gleichverteilten Charakteristiken ..	129
Tabelle 13:	Verteilung und Komplexitätsmaße für die Datenbankmanagementsysteme.....	134
Tabelle 14:	Verteilung und Komplexitätsmaße für die Betriebssysteme	134
Tabelle 15:	Komplexitätsmaßzahlen für eine ausgewählte Domäne der Bank	135
Tabelle 16:	Verteilung der Schnittstellen und Maßzahlen der betrachteten Domäne 8	136
Tabelle 17:	Errechnete Maßzahlen für die ausgewählten Anwendungen der Domäne 8	137
Tabelle 18:	Zieldefinition sowie Elemente und Unterscheidungskriterien	139
Tabelle 19:	Interpretation und Beispiele für die Veränderung der Maßzahlen	140
Tabelle 20:	Berechnete Maßzahlen vor und nach der geplanten Transformation	141
Tabelle 21:	Phasen und Schritte der ADR-Methode	154
Tabelle 22:	Organisation der Stufen des Forschungsprojektes.....	157
Tabelle 23:	Verknüpfung und Zwischenergebnisse der Teilprojekte.....	158
Tabelle 24:	Forschungsschritte der ersten und zweiten Phase der ADR-Methode.....	160
Tabelle 25:	Forschungsschritte der dritten und vierten Phase der ADR-Methode	161
Tabelle 26:	Überblick über die durchgeführten BIE-Zyklen des ersten ADR-Projektes	163

Tabelle 27: Initiale abstrakte Designprinzipien	165
Tabelle 28: Instanzierte Designprinzipien des ersten ADR-Projektes	174
Tabelle 29: Aus dem ersten ADR-Projekt resultierende abstrakte Designprinzipien	175
Tabelle 30: Überblick über die durchgeführten BIE-Zyklen des zweiten ADR-Projektes .	177
Tabelle 31: Instanzierte Designprinzipien im Rahmen des zweiten ADR-Projektes	186
Tabelle 32: Aus dem zweiten ADR-Projekt resultierende abstrakte Designprinzipien.....	187
Tabelle 33: Überblick über die durchgeführten BIE-Zyklen des dritten ADR-Projektes ...	188
Tabelle 34: Instanzierte Designprinzipien im Rahmen des dritten ADR-Projektes	200
Tabelle 35: Aus den Forschungsergebnissen abgeleitete abstrakte Designprinzipien	201
Tabelle 36: Aus dem Forschungsprojekt abgeleitete abstrakte Designprinzipien (1)	203
Tabelle 37: Aus dem Forschungsprojekt abgeleitete abstrakte Designprinzipien (2)	204
Tabelle 38: Implikationen der BIE und IT-COM für die Banken A bzw. B	206

Abkürzungsverzeichnis

Abkürzung	Bedeutung
AA	Anwendungsarchitektur
ACM	Association for Computing Machinery
ADR	Action Design Research
AIS	Association for Information Systems
BIE	Building, Intervention and Evaluation
CAR	Canonical Action Research
CAS	Complex Adaptive System
CIO	Chief Information Officer
DA	Datenarchitektur
DB	Datenbanksystem
DP	Designprinzip
EA	Enterprise Architecture
EAM	Enterprise Architecture Management
EAP	Enterprise Architecture Planning
ECIS	European Conference on Information Systems
EM	Entropie-Maß
FE	Funktionale Einheit
FEAF	Federal Enterprise Architecture Framework
FF	Forschungsfrage
GPA	Geschäftsprozessarchitektur
GQM	Goal Question Metric
HHI	Herfindahl-Hirschmann-Index
HOR	Horvath-Index
IA	Infrastrukturarchitektur
IAF	Integrated Architecture Framework
ICIS	International Conference on Information Systems
IDV	Individuelle Datenverarbeitung
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers
IMS	Information Management System
IS	Information Systems
ISO	International Organization for Standardization
IT	Informationstechnologie
ITA	IT-Architektur
KPI	Key Performance Indicator
MIS	Management Information Systems
OS	Operating System
PC	Personal Computer
TOGAF	The Open Group Architecture Framework
UG	Unternehmensgruppe
VOC	Volatile organic compounds
VWL	Volkswirtschaftslehre
WI	Wirtschaftsinformatik

1 Einleitung

1884 erwarb Sarah Winchester, die Witwe des Gewehrfabrikanten William Wirt Winchester, ein sich noch im Bau befindliches Haus in San Jose (Kalifornien, USA), das zu diesem Zeitpunkt sechs Räume zählte. Sarah Winchester entschied sich, die bestehenden Baupläne zu verwerfen und traf sich stattdessen jeden Tag mit dem Vorarbeiter, um diesem zu erklären, welche Arbeiten durch die bis zu 22 Zimmermänner verrichtet werden sollten (Ross et al. 2006). Insgesamt befand sich das Gebäude – auch bekannt als „Winchester House“ – 38 Jahre im Bau. Durch die kontinuierliche Erweiterung des Hauses (welches heute u. a. 160 Räume, einen Ballsaal und über 1.000 Fenster zählt) und regelmäßige Umbaumaßnahmen an bereits bestehenden Räumen entstand eine Vielzahl eigenartiger architektonischer Facetten. Beispiele hierfür sind eine Treppe, die sieben Stufen hinab und unmittelbar wieder elf Stufen hinauf führt, oder Türen, hinter denen sich das Mauerwerk befindet. Ein Erdbeben im Jahr 1906 schloss Sarah Winchester in ihrem Schlafzimmer ein und es dauerte Stunden, bis ihre Bediensteten sie auffinden und befreien konnten. Dieser unglückliche Umstand war das Ergebnis der im Laufe der Jahre entstandenen Komplexität der Architektur des Gebäudes (Schütz et al. 2013a; Winchester Mystery House 2015).

Die Komplexität von Architekturen wird auch zunehmend zum Gegenstand der Diskussion im Bereich der Informationstechnologie (IT). Eine 2014 unter mehr als 1100 CIOs und IT-Managern aus Unternehmen unterschiedlicher Größe und Branchen durchgeführte Befragung zeigt, dass die Komplexität der IT-Landschaft eine der drei wichtigsten Herausforderungen von CIOs in Unternehmen darstellt (Capgemini 2014).¹ Über die Jahre gewachsene IT-Landschaften, bestehend aus einer Vielzahl unterschiedlicher Lösungen und diverser eng miteinander verflochtener Altsysteme (sogenannte Legacy-Systeme), stellen in vielen Unternehmen die Ausgangssituation für die Umsetzung neuer Anforderungen dar (Ross et al. 2006). Zu traditionell häufig angeführten Ursachen eines fortwährenden Wachstums der IT-Landschaft zählen u. a. die Umsetzung neuer Geschäftsanforderungen zur Unterstützung neuer Produkte, die Erschließung neuer Märkte oder anorganisches Unternehmenswachstum (Mattern et al. 2003; Schmidt & Buxmann 2011). Analog zu dem angeführten Beispiel der Entstehung von Komplexität im Falle des Winchester House, wird auch bei IT-Landschaften das Fehlen eines zentralen Planes zur Weiterentwicklung der IT-Architektur als eine Ursache der Entstehung von Komplexität angesehen (Ahlemann et al. 2012; Ross et al. 2006). Hinzu kommt, dass im Zuge der Implementierung neuer Anforderungen ein zunehmender Wettbewerbs- und Zeitdruck häufig zu kurzfristigen Umsetzungen führt, ohne eine vollständige In-

¹ Speziell im Bankenumfeld bewegen sich die Komplexität und die Beherrschbarkeit der IT zunehmend in den Fokus der Aufsicht, da IT-Risiken – als operationelle Risiken – wesentliche Risiken der Institute darstellen (BaFin 2012).

tegration der neuen Systeme in die bestehenden Umgebungen oder eine Ablösung der Altsysteme sicherzustellen (Mattern et al. 2003). Die Folgen für die Unternehmen werden dabei u. a. in einem zunehmenden Verlust an Transparenz, steigenden Kosten sowie operationellen Risiken des Betriebes, aber auch in der erschwerten Änderbarkeit einer komplexen IT-Landschaft gesehen (Ahlemann et al. 2012; Mattern et al. 2003; Ross et al. 2006). Als Strategie des Architekturmanagements, mit diesen Problemen umzugehen, legt die bestehende Literatur den Fokus meist auf die IT-Standardisierung. Diese zielt dabei primär auf die Erreichung eines kosteneffizienten Ressourceneinsatzes durch die Realisierung einer möglichst homogenen IT-Landschaft ab (Boh & Yellin 2006; Bradley et al. 2012; Richardson et al. 1990; Ross et al. 2006; Ross & Westerman 2004; Schmidt & Buxmann 2011).

In der vorliegenden Arbeit wird mit der Komplexität eine die bestehenden Ansätze – wie den der Standardisierung – komplementierende Perspektive auf IT-Architekturen eingeführt, welche insbesondere die Zielgröße der IT-Flexibilität stärker berücksichtigt. Hierzu wird zunächst literaturbasiert eine Konzeptualisierung von Komplexität in IT-Architekturen vorgeschlagen. Im Anschluss werden mittels einer Fallstudie mit einem Unternehmen aus der chemischen Industrie die Ursachen und Auswirkungen von Komplexität untersucht. Um das Konzept der Komplexität messbar zu machen, wird darüber hinaus eine Vorgehensweise zur Quantifizierung von Komplexität in IT-Architekturen entwickelt. Hierzu wird auf etablierte Konzentrationsmaße aus der Volkswirtschaftslehre und auf Konzepte aus der Biologie zurückgegriffen. Auf Basis eines Forschungsprojektes mit zwei Banken werden schließlich Designprinzipien erarbeitet, die beschreiben, wie ein Informationssystem ausgestaltet sein sollte, welches IT-Architekten bei dem Management komplexer IT-Architekturen unterstützt.

1.1 Relevanz der Arbeit

In den folgenden beiden Abschnitten werden sowohl die praktische als auch die wissenschaftliche Relevanz der in der vorliegenden Dissertation behandelten Forschungsfragen diskutiert.

1.1.1 Praktische Relevanz

Wie bereits angeführt, wird die Beherrschbarkeit komplexer IT-Landschaften zunehmend zu einem Problem der unternehmerischen Praxis (Capgemini 2014; Mattern et al. 2003). In einem Interview von Network World (2011) mit Peter Leukert, dem ehemaligen CIO der Commerzbank, zum Thema „Getting a handle on complexity“ macht dieser deutlich, dass eine zunehmende Komplexität die wesentlichen Zielgrößen einer Unternehmens-IT gefährdet:

„Complexity increases cost and decreases flexibility—often in unforeseen ways—and also tends to decrease stability. If you run IT, those are three of your most important KPIs. So all three are adversely affected by complexity.“ (Network World 2011)

Um ein Management der Komplexität der IT-Landschaft etablieren zu können, bedarf es zunächst eines klaren Verständnisses, was Komplexität bedeutet. Das Management einer IT-

Landschaft beschränkt sich dabei nicht auf eine einzelne Ebene einer IT-Architektur² (wie z. B. die Anwendungsarchitektur), sondern es muss die gesamte IT-Architektur (d. h. die Daten-, Anwendungs- und Infrastrukturarchitektur) des Unternehmens berücksichtigen (Ross 2003). Aus Managementperspektive ist es dabei problematisch, in jeder der Architekturebenen individuelle und von den übrigen Ebenen losgelöste Definitionen des Konzeptes „Komplexität“ vorzufinden. Zum einen erschwert ein unterschiedliches Verständnis die Greifbarkeit und die Diskussion der Problemstellung über die Grenzen der einzelnen Architekturebenen hinaus. Zum anderen sind auf unterschiedlichen Konzepten basierende Kennzahlen zur Steuerung für das Management nur schwer vergleich- und nachvollziehbar. Für die Praxis bedarf es daher eines einheitlichen Verständnisses von Komplexität, welches auf die IT-Architektur als Ganzes anwendbar ist, aber dennoch den spezifischen Anforderungen der einzelnen Architekturebenen gerecht wird. Ein weiterer Aspekt ist die bereits angedeutete Messbarkeit des Konzeptes Komplexität. Nur durch eine Messbarkeit des Grades an Komplexität kann Handlungsbedarf erkannt und die Wirkung von Maßnahmen beurteilt werden.

IT-Architekten und IT-Manager benötigen darüber hinaus ein klares Verständnis dafür, wie Komplexität entsteht, d. h. was die Ursachen von Eingriffen in die IT-Architektur sind, wie diese Eingriffe zu einer Veränderung der Komplexität führen und welche Mechanismen das Ausmaß der entstehenden Komplexität beeinflussen. Nur durch ein besseres Verständnis an dieser Stelle, können die erforderlichen organisatorischen Rahmenbedingungen geschaffen und die IT-Architektur zielgerichtet weiterentwickelt werden. Aufgrund des teilweise immensen Ausmaßes bestehender IT-Architekturen benötigen IT-Architekten und IT-Manager neben diesem grundlegenden Verständnis zudem operative Unterstützung bei der Planung und Kontrolle der Komplexität der IT-Architektur. Hier müssen Entscheider z. B. in die Lage versetzt werden, die Auswirkungen von Architekturentscheidungen im Vorfeld der Durchführung einer Maßnahme überprüfen zu können.

Wie die in der vorliegenden Arbeit dargestellte Literaturrecherche zeigt, existieren im Themenfeld der Komplexität bislang vornehmlich Forschungsergebnisse in spezifischen Teilbereichen der IT, wie z. B. der Daten- oder Anwendungsarchitektur. Dem Management der Komplexität der IT-Architektur als Ganzes wurde in der Literatur des Information Systems (IS) Research bislang wenig Beachtung geschenkt. Demzufolge kann die Praxis hier nur sehr begrenzt auf wissenschaftlich fundiertes Wissen zurückgreifen. Die Studien der vorliegenden Arbeit leisten hierzu einen Beitrag, indem sie für die Praxis anwendbare Ergebnisse herleiten.

Die in der Arbeit eingeführte Konzeptualisierung von Komplexität dient der Praxis als einheitliche Begriffsdefinition. Durch diese wird ein über die Architekturebenen konsistentes Verständnis des Konzeptes sichergestellt. Darüber hinaus stellt die Konzeptualisierung die Basis für die Erstellung von Kennzahlen zur Messung von Komplexität dar. Durch die in der vorliegenden Arbeit konzipierte Vorgehensweise zur Quantifizierung von Komplexität wird es IT-Managern und IT-Architekten ermöglicht, Kennzahlen zur Planung und Steuerung der Komplexität einer IT-Architektur zu definieren. Nach der vorgeschlagenen Vorgehensweise

² Der Begriff der IT-Architektur und die einzelnen Ebenen einer IT-Architektur werden in Abschnitt 2.2.4 erläutert.

abgeleitete Kennzahlen folgen einer einheitlichen Berechnungsvorschrift, was deren Verständnis für unterschiedliche Kontexte erleichtert. Gleichzeitig ist die Form der Quantifizierung auf alle Ebenen der IT-Architektur anwendbar. Die durchgeführte qualitative Studie unterstützt IT-Manager und IT-Architekten bei dem Verständnis der Ursachen und der Auswirkungen von Komplexität, wodurch die Identifikation potenzieller Handlungsfelder und relevanter Stellhebel zur Vermeidung übermäßiger Komplexität erleichtert wird. Die im Rahmen der Arbeit abgeleiteten Designprinzipien können in der Praxis dazu verwendet werden, Informationssysteme zu implementieren, die IT-Manager und IT-Architekten bei dem Management der Komplexität ihrer IT-Architektur unterstützen.

1.1.2 Wissenschaftliche Relevanz

Das Management der Komplexität wird in der Literatur als eine wesentliche Aufgabe des Architekturmanagements ausgemacht (Guillemette & Paré 2012; Ross et al. 2006; Schmidt & Buxmann 2011). Ungeachtet dessen existieren in der wissenschaftlichen Literatur noch keine Arbeiten, die sich explizit dem Konzept der Komplexität in Unternehmens-/IT-Architekturen sowie dessen Management gewidmet haben.

Die im Zuge der Arbeit durchgeführte Literaturrecherche zeigt auf, dass Komplexität in allen Ebenen einer Unternehmensarchitektur (d. h. der Geschäftsprozess-, Anwendungs-, Daten- und Infrastrukturarchitektur) eine Rolle spielt (Banker et al. 1998; Banker & Slaughter 2000; Hanseth et al. 2006; Lee & Xia 2002; Xia & Lee 2005; Xue et al. 2012). Die Ergebnisse wissenschaftlicher Untersuchungen in einzelnen Ebenen der Architektur basieren jedoch meist auf einem spezifischen Verständnis von Komplexität. Dies führt dazu, dass diese Ergebnisse nur schwer auf den Kontext anderer Teilbereiche der Architektur übertragen werden können. Durch die Erarbeitung einer Konzeptualisierung von Komplexität soll ein einheitliches Verständnis eingeführt und eine Überprüfung der Übertragbarkeit von Forschungsergebnissen erleichtert werden. Hierzu wird in der Arbeit eine systemtheoretische Perspektive eingenommen und der Fokus auf die strukturelle Komplexität der IT-Architektur gelegt. Die erste in der vorliegenden Dissertation bearbeitete Forschungsfrage lautet demzufolge:

Forschungsfrage 1: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen konzeptualisiert werden?

Die praxisorientierte Literatur führt eine Reihe von Ursachen einer zunehmenden Komplexität der IT-Architekturen an (Mattern et al. 2003). Die wissenschaftliche Literatur liefert bislang jedoch keine Erkenntnisse, was die Entstehung von Komplexität verursacht. Weiterhin existieren keine Untersuchungen, die beleuchten, welche Mechanismen das Ausmaß einer entstehenden Komplexität beeinflussen. Um hierzu ein tieferes Verständnis des Phänomens der Komplexität in IT-Architekturen zu erlangen, wird die folgende Forschungsfrage bearbeitet:

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Die Auswirkungen von Komplexität auf Zielgrößen der Unternehmen, wie Effizienz, Flexibilität oder Risiken, wurden in der Literatur bislang lediglich innerhalb einzelner Architektur-

ebenen untersucht (vgl. hierzu Midha 2008).³ Eine wissenschaftliche Arbeit, welche die Auswirkungen struktureller Komplexität der IT-Architektur als Ganzes näher untersucht, konnte hingegen nicht identifiziert werden.⁴ Die dritte Forschungsfrage lautet demnach:

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

Neben der fehlenden Konzeptualisierung wurde in der IS-Literatur noch kein Ansatz zur Quantifizierung von Komplexität vorgeschlagen, der auf alle Architekturebenen anwendbar ist. Im Zuge der vierten Forschungsfrage sollen Ansätze aus anderen (z. B. der Biologie oder der Volkswirtschaftslehre) oder angrenzenden (wie dem Softwareengineering) Forschungsfeldern hinsichtlich deren Übertragbarkeit auf den Kontext der Unternehmens- und IT-Architekturen untersucht werden:

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen quantifiziert werden?

In der Literatur wird betont, dass im Rahmen der Weiterentwicklung der IT-Landschaft sowohl die neu verfügbaren Technologien als auch die vorhandene IT-Landschaft des Unternehmens berücksichtigt werden müssen (Lee & Xia 2002; Xia & Lee 2005). Unterstützung des Managements bei dieser Problemstellung liefert die aktuelle IS-Literatur jedoch lediglich zu dem ersten Aspekt (vgl. hierzu Adomavicius et al. 2008).⁵ Zu dem zweiten von Xia und Lee angeführten Punkt, der auf eine Transparenz des Eingriffs in die bestehende IT-Landschaft sowie die mit diesem Eingriff zusammenhängenden Folgen abzielt, liefert die aktuelle Literatur hingegen keine Lösungsansätze. Guillemette & Paré (2012) führen das Management der Komplexität der IT-Architektur als eine der fundamentalen Aufgaben der IT-Abteilungen in Unternehmen an, wobei dem sogenannten „Architecture Builder“ die folgende Aufgabe zukommt:

“[...] to build and manage an IT infrastructure that supports business processes and reduces architectural complexity.” (Guillemette & Paré 2012, S. 532).

Die IS-Literatur benennt demnach das Management der Komplexität explizit als Aufgabe der IT-Abteilungen. Es existiert jedoch bislang keine theoretische Grundlage dafür, wie dieses Management erfolgen soll. Durch die vorliegende Dissertation werden IT-Architekten durch die Ableitung präskriptiven Wissens aus einem mehrjährigen Forschungsprojekt bei dieser Aufgabe unterstützt und gleichzeitig die Wissensbasis um weitere theoretische Erkenntnisse über das Management – speziell die Planung und die Kontrolle – komplexer IT-Architekturen erweitert. Die fünfte Forschungsfrage der vorliegenden Arbeit lautet wie folgt:

³ Midha (2008) zeigt beispielsweise, dass ein Zusammenhang zwischen der strukturellen Komplexität von Anwendungen und der Anzahl an Fehlern und der Wartbarkeit der Anwendung besteht.

⁴ Da es sich bei den untersuchten Zielgrößen um Eigenschaften des Systems “IT-Architektur” handelt, wird in diesem Kontext der Begriff der “Zieleigenschaften” verwendet (vgl. hierzu Alderson & Doyle 2010).

⁵ Adomavicius et al. (2008) entwickeln Artefakte zur Erstellung von Prognosen über die künftige Entwicklung von Technologien, die für eine IT-Investition in Frage kommen.

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

1.2 Struktur und Aufbau der Arbeit

Die Struktur der Arbeit ist in Abbildung 1 veranschaulicht. Nach dieser Einleitung werden in Kapitel 2 „Grundlagen“ die theoretischen und begrifflichen Grundlagen der Arbeit eingeführt. Der Fokus liegt hierbei auf der Systemtheorie (Unterkapitel 2.1) – als theoretische Perspektive der Arbeit – und dem Themenbereich der (Unternehmens-)Architekturen (Unterkapitel 2.2) – als spezifische inhaltliche Domäne, der die Beiträge der Arbeit zuzuordnen sind.

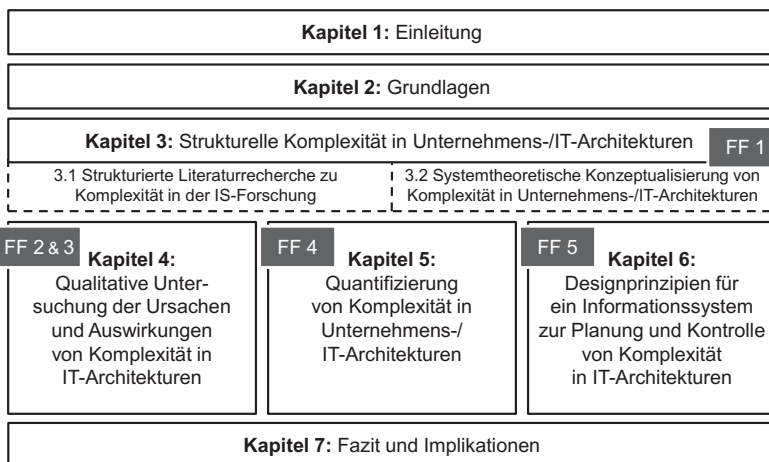


Abbildung 1: Struktur und Aufbau der Dissertation

Kapitel 3 „Strukturelle Komplexität in Unternehmens-/IT-Architekturen“ ist in zwei Teile untergliedert. In Unterkapitel 3.1 wird zunächst der Stand der IS-Forschung zum Thema der „IT-Komplexität“ mittels einer strukturierten Literaturrecherche untersucht und auf deren Grundlage die Forschungsagenda der vorliegenden Arbeit abgeleitet. Anhand der aus der strukturierten Literaturrecherche identifizierten Quellen wird in Unterkapitel 3.2 anschließend die erste Forschungsfrage (FF 1) durch die Erarbeitung einer systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen beantwortet. Diese systemtheoretische Konzeptualisierung bildet die Basis für die darauf folgenden drei Kapitel.

In Kapitel 4 „Qualitative Untersuchung der Ursachen und Auswirkungen von Komplexität in IT-Architekturen“ werden anhand einer qualitativ-empirischen Fallstudie die Ursachen der Entstehung von Komplexität in IT-Architekturen (FF 2) sowie die hierdurch für die Organisation entstehenden Auswirkungen auf Zieleigenschaften wie IT-Effizienz und IT-Flexibilität (FF 3) untersucht. Neben den im Kontext des betrachteten Falles identifizierten Ursachen und Auswirkungen führt die Studie zudem zu einem tieferen Verständnis des Phänomens der

Komplexität in IT-Architekturen. Hierzu wird die durchgeführte Studie in Unterkapitel 4.1 zunächst motiviert und in Unterkapitel 4.2 die der Studie zugrunde liegenden Annahmen eingeführt. Im Anschluss werden in Unterkapitel 4.3 das methodische Vorgehen vorgestellt und in Unterkapitel 4.4 die Ergebnisse der qualitativen Fallstudie präsentiert. Das Kapitel schließt in Unterkapitel 4.5 mit einer Zusammenfassung und Diskussion der Ergebnisse.

Die vierte Forschungsfrage (FF 4) wird in Kapitel 5 „Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen“ beantwortet. Auf Grundlage der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung von Komplexität wird eine holistische Vorgehensweise zur Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen vorgeschlagen, die auf der Shannon-Entropie – einem gut erforschten Maß aus der Informationstheorie – basiert. Nach einer Motivation der Forschungsfrage in Unterkapitel 5.1 und der Diskussion verwandter Literatur in Unterkapitel 5.2 wird in Unterkapitel 5.3 das sogenannte „Goal Question Metric Paradigma“, eine systematische Vorgehensweise zur Ableitung von Kennzahlen, vorgestellt. In Unterkapitel 5.4 werden anschließend Anforderungen an das zu erarbeitende Maß eingeführt, die in Unterkapitel 5.5 zur Überprüfung von bekannten Konzepten aus dem Feld der Konzentrationsmessung angewendet werden. In Unterkapitel 5.6 werden diese durch verwandte Konzepte aus der Biologie komplementiert und anschließend in Unterkapitel 5.7 ein auf den gewonnenen Erkenntnissen basierendes Maß zur Quantifizierung von Komplexität vorgeschlagen. Die Anwendbarkeit des Maßes wird in Unterkapitel 5.8 anhand der Architekturinformationen von drei Praxispartnern aufgezeigt. In Unterkapitel 5.9 werden die Ergebnisse schließlich diskutiert.

In Kapitel 6 „Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen“ wird ein mehrjähriges Forschungsprojekt mit zwei Banken vorgestellt, das sich auf die fünfte Forschungsfrage (FF 5) bezieht. Im Rahmen des Forschungsprojektes wurden zwei Prototypen für ein Komplexitätsmanagement von IT-Architekturen erarbeitet und abstrakte Designprinzipien für Informationssysteme zur Planung und Kontrolle von Komplexität in IT-Architekturen abgeleitet. Nach einer Motivation der Themenstellung in Unterkapitel 6.1 werden in Unterkapitel 6.2 die methodischen Grundlagen eingeführt. Die im Zuge des Forschungsprojektes bearbeitete Problemklasse wird in Unterkapitel 6.3 und das grundlegende Vorgehen in Unterkapitel 6.4 vorgestellt. In den darauf folgenden Unterkapiteln 6.5 bis 6.7 werden die Ergebnisse der einzelnen Teilprojekte dargestellt und anschließend in Unterkapitel 6.8 im Hinblick auf deren theoretische und praktische Implikationen diskutiert.

Die vorliegende Dissertation schließt mit Kapitel 7 „Fazit und Implikationen“. Nach einer Zusammenfassung der wesentlichen Ergebnisse in Unterkapitel 7.1 werden in den Unterkapiteln 7.2 und 7.3 die theoretischen Beiträge der Arbeit sowie Handlungsempfehlungen für die Praxis diskutiert. In Unterkapitel 7.4 werden schließlich Möglichkeiten weiterer Forschung im behandelten Themenfeld aufgezeigt.

2 Grundlagen

In diesem Kapitel werden die theoretischen und begrifflichen Grundlagen der vorliegenden Arbeit vorgestellt. Hierzu wird in Unterkapitel 2.1 zunächst die *Systemtheorie* als theoretische Linse der Arbeit diskutiert. Nach der Einführung der relevanten Grundlagen im Bereich der *Architekturen* in Unterkapitel 2.2 werden in Unterkapitel 2.3 die zuvor eingeführten Konzepte zu dem theoretischen Bezugsrahmen der Arbeit zusammengeführt.

2.1 Systemtheorie

Wesentliche theoretische Grundlage der vorliegenden Arbeit stellt die Systemtheorie dar. Dieses Unterkapitel ist in drei Abschnitte aufgeteilt. In Abschnitt 2.1.1 werden zunächst der *allgemeine Systembegriff* sowie die drei Systemkonzepte nach Ropohl (2009) dargestellt. Nach einem kurzen Abriss der Entwicklung hin zur *modernen Systemtheorie* in Abschnitt 2.1.2 wird in Abschnitt 2.1.3 die Theorie *komplexer adaptiver Systeme* eingeführt.

2.1.1 Allgemeiner Systembegriff

Der Systembegriff geht auf das altgriechische σύστημα [sýstēma] zurück, was soviel bedeutet wie „aus mehreren Teilen zusammengesetztes und gegliedertes Ganzes“. In der wissenschaftlichen Literatur existiert eine Reihe von Definitionen für den Begriff „System“. Ein weit verbreitetes Systemverständnis, welches den Fokus auf die System-Elemente und die Beziehungen zwischen diesen legt, findet sich dabei in einigen Variationen. Van Gigch (1991) definiert ein System wie folgt:

„A system is an assembly or set of related elements.“ (Van Gigch 1991, S. 30)

In eine sehr ähnliche Richtung geht die Definition von Hall & Fagen (1969), die jedoch auch Beziehungen zwischen den Attributen (oder Eigenschaften) von Elementen bzw. Objekten berücksichtigt:

„A system is a set of objects together with relationships between the objects and between their attributes.“ (Hall & Fagen 1969, S. 31)

Der Standard ISO/IEC 15288:2008 (Systems and software engineering – System life cycle processes) betont neben miteinander in Beziehung stehenden bzw. interagierenden Elementen zudem auch den Systemzweck:

„[A system is a] combination of interacting elements organized to achieve one or more stated purposes.“ (ISO/IEC 2008, S. 6)

Die vorliegende Arbeit folgt der Definition nach Hall & Fagen (1969). Mögliche Beispiele für Systeme im Sinne des gewählten Systembegriffs sind sehr vielseitig und reichen von Wirtschafts-, Öko-, Umwelt- und Nervensystemen über Betriebssysteme bis hin zu Waffensystemen.

Ropohl (2009) diskutiert in seiner Arbeit drei Systemkonzepte, die je nach Systemverständnis der wissenschaftlichen Autoren in den Vordergrund rücken, wobei nach Ropohl (2009) alle drei Konzepte durch den Systembegriff abgedeckt werden sollten. Bei diesen Konzepten handelt es sich um das *funktionale*, das *strukturele* sowie das *hierarchische* Systemkonzept.

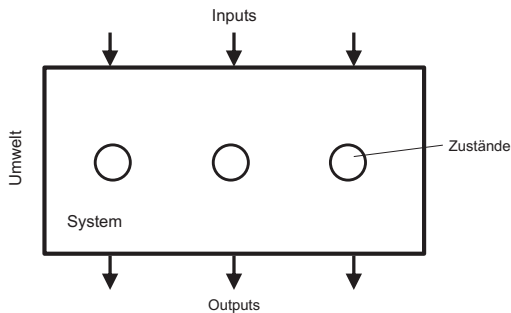


Abbildung 2: Funktionales Konzept nach Ropohl (2009)

Das funktionale Systemkonzept (vgl. Abbildung 2) abstrahiert vom inneren Aufbau des Systems und fokussiert auf das Verhalten des Systems in seiner Umwelt. Hierbei wird das System – bis auf die es beschreibenden Zustände – als „black box“ betrachtet, die bestimmte Eigenschaften aufweist. Zu diesen Eigenschaften zählen neben den bereits angeführten, das System beschreibenden Zuständen, die Eingangsgrößen (Inputs) und die Ausgangsgrößen (Outputs) des Systems. Wie das System aus den Eingangs- die entsprechenden Ausgangsgrößen generiert, wird durch das funktionale Konzept ausgeblendet.

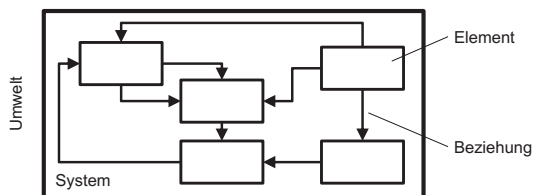


Abbildung 3: Struktureles Konzept nach Ropohl (2009)

Bei dem strukturalen Konzept (vgl. Abbildung 3) steht der Aufbau des Systems „[...] als eine Ganzheit miteinander verknüpfter Elemente [...]“ (Ropohl 2009, S. 75) im Fokus. Unterschiedliche Kombinationen und Verknüpfungen der gegebenen Systemelemente können zu einer Vielzahl von Systemeigenschaften führen. Das strukturele Konzept folgt daher dem Grundsatz, dass die Systemelemente nicht unabhängig von ihrem Kontext – d. h. den übrigen Systemelementen sowie der Verknüpfungen zwischen diesen – betrachtet werden sollten.

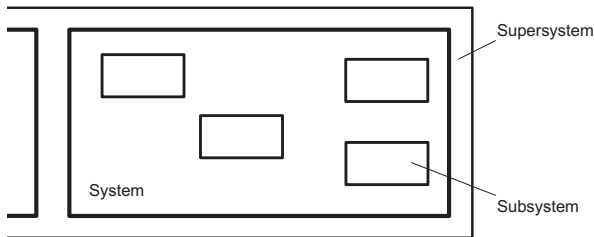


Abbildung 4: Hierarchisches Konzept nach Ropohl (2009)

Das hierarchische Konzept (vgl. Abbildung 4) betrachtet die Elemente eines Systems ebenfalls als Systeme, sogenannte Subsysteme. Nach Simon (1962) können Systeme in weitere Subsysteme untergliedert werden, bis eine unterste Ebene elementarer Subsysteme erreicht wird.⁶ Gleichzeitig kann das System aber auch als Element eines übergeordneten Systems, dem sogenannten Supersystem, verstanden werden. Aus einer systemanalytischen Perspektive unterstützt das hierarchische Konzept in zweierlei Hinsicht. Ein „Hineinzoomen“ in eine tiefere Hierarchieebene erlaubt ein besseres Verständnis und eine detailliertere Erklärung des Systems, wohingegen ein „Herauszoomen“ zu einem Erkenntnisgewinn hinsichtlich der Systembedeutung führt (Mesarovic & Macko 1969).

Es ist ersichtlich, dass sich die drei angeführten Systemkonzepte nicht ausschließen, sondern einander vielmehr komplementieren. Je nach Perspektive auf den Untersuchungsgegenstand erlauben die Systemkonzepte dem Betrachter einen unterschiedlichen Erkenntnisgewinn. Im Zuge dieser Arbeit sind primär die Perspektiven auf die Struktur der Elemente auf Systemebene – das strukturelle Konzept – sowie auf die hierarchische Betrachtung mit Blick auf die Zuordnung der Subsysteme zu den darüberliegenden Systemen – das hierarchische Konzept – im Fokus. Der in dieser Arbeit zugrunde liegende Systembegriff nach Hall & Fagen (1969) erwähnt die angeführten Systemkonzepte in ihrer Ganzheit zwar nicht explizit, ist jedoch auf diese anwendbar.

2.1.2 Wurzeln der modernen Systemtheorie

Nachdem im vorangegangenen Abschnitt der Systembegriff sowie die verschiedenen Systemkonzepte eingeführt wurden, gibt dieser Abschnitt einen kurzen Abriss über die Entwicklung der noch relativ jungen Disziplin der Systemwissenschaft hin zur modernen Systemtheorie. Die Ursprünge letzterer sind im Wesentlichen in der allgemeinen Systemtheorie (Bertalanffy 1948; Bertalanffy 1968), der Kybernetik (Wiener 1961) sowie der Chaostheorie zu finden.

Der Biologe Ludwig von Bertalanffy beobachtete seinerzeit eine immer weiter fortschreitende Spezialisierung der einzelnen Forschungsdisziplinen, aus denen sich jeweils wiederum eine

⁶ Vgl. hierzu auch den von Simon (1962) verwendeten Begriff hierarchischer Systeme: „By a hierarchic system, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem. In most systems in nature, it is somewhat arbitrary as to where we leave off the partitioning, and what subsystems we take as elementary“ (Simon 1962, S. 468).

Vielzahl von Subdisziplinen herausbildete. Jede dieser Disziplinen und Subdisziplinen entwickelte sich zu einem eigenen „privaten Universum“ mit eigenen Methoden und Termini, was es zunehmend erschwerte, ein „*Wort von einem Kokon in einen anderen zu bekommen*“ (Bertalanffy 1968, S. 30). Gleichzeitig beobachtete von Bertalanffy jedoch, dass sich in verschiedenen Forschungsgebieten vergleichbare Problemstellungen und Konzepte herausgebildet hatten. In einer allgemeinen Systemtheorie sah von Bertalanffy eine Möglichkeit, die Zusammenarbeit einzelner Forschungsgebiete zu fördern und theoretische Konzepte zu identifizieren, die nicht auf Einzelwissenschaften beschränkt sind. Diese Konzepte sollten demnach auf diverse Systeme anwendbar sein. Ludwig von Bertalanffy diskutierte in seinen Arbeiten Konzepte wie geschlossene und offene Systeme, Selbstregulation und Gleichgewicht (Bertalanffy 1968). Das Interesse, ein besseres Verständnis für gemeinsame, grundlegende Konzepte verschiedener Disziplinen zu entwickeln, teilte von Bertalanffy mit einer Vielzahl von Forschern aus den unterschiedlichsten Forschungsgebieten. Im Rahmen von Symposien kam es zu einem Austausch von Wissenschaftlern aus verschiedensten Bereichen wie u. a. dem Ingenieurwesen, der Mathematik, der Volkswirtschaftslehre, der Biologie oder der Philosophie (Mesarovic 1964).

Auch der Mathematiker Norbert Wiener und der Psychiater William Ross Ashby trugen mit ihren Arbeiten zur Kybernetik – der zweiten Strömung hin zur modernen Systemtheorie – zu einem disziplinübergreifenden Erkenntnisgewinn bei (Ashby 1956; Wiener 1961). In seinem Buch „Cybernetics or Control and Communication in the Animal and the Machine“ zeigte Wiener u. a., dass sowohl in künstlichen Systemen als auch in lebendigen Organismen vergleichbare Regelungs- und Steuerungsvorgänge zu beobachten sind (Wiener 1961).

Ein parallel dazu entstandenes, verwandtes Gebiet ist das der Chaostheorie, die Phänomene einer Reihe nichtlinearer dynamischer Systeme beschreibt. Das Verhalten solcher Systeme wird als chaotisch bezeichnet, wenn dieses empfindlich auf eine Variation der Anfangsbedingungen reagiert, obwohl sich die ihm zugrunde liegenden Gesetzmäßigkeiten deterministisch verhalten. Ein Beispiel für ein solches, sensibel auf Anfangswerte reagierendes System ist das Wetter. Edward Lorenz etablierte an diesem Beispiel den Begriff des „Schmetterlingseffektes“⁷ und legte einen wichtigen Grundstein für die Entstehung der Chaostheorie (Gleick 1988; Lorenz 2000). Ein Beispiel für chaotisches Verhalten einfacher nichtlinearer Gleichungen ist die sogenannte logistische Gleichung⁸ (May 1976; Verhulst 1838).

Die noch relativ junge Theorie komplexer adaptiver Systeme wurde stark von Arbeiten des Mitte der 90er Jahren gegründeten Santa Fe Institutes (New Mexico) geprägt (vgl. hierzu Holland 1995; Holland 1998; Sneppen et al. 1995). Da diese Theorie wesentlich für das Ver-

⁷ Edward Lorenz diskutierte im Rahmen des 139. Treffens der American Association for the Advancement of Science den potenziellen Einfluss eines Flügelschlags eines Schmetterlings auf die Entstehung eines Tornados, um die Schwierigkeit in der Vorhersage von Wetterphänomenen zu illustrieren (Lorenz 2000).

⁸ Verhulst führte die logistische Gleichung 1838 als mathematisches Modell zur Darstellung der Entwicklung einer Population ein. Die logistische Gleichung der Form $x_{n+1} = rx_n - rx_n^2$ erzeugt bei Variation des konstanten Parameters r im Bereich $[2, 4]$ Folgen in der Umgebung einer zunehmenden Anzahl von Häufungspunkten. Mit steigendem Wert für r nimmt die Anzahl der Häufungspunkte zu. Gleichzeitig verkürzen sich die Intervalle mit gleicher Anzahl von Häufungspunkten bis die Folge zu chaotischem Verhalten übergeht.

ständnis der vorliegenden Arbeit und deren theoretischen Beitrag im Bereich der Systemtheorie ist, führt der folgende Abschnitt näher in die Annahmen und Charakteristiken, die dieser Theorie zugrunde liegen, ein.

2.1.3 Komplexe adaptive Systeme

Das Forschungsgebiet der komplexen adaptiven Systeme – im weiteren Verlauf der Arbeit abgekürzt mit „CAS“ (Complex Adaptive System) – stellt eine vergleichsweise neue Weise dar, über Systeme zu denken, die sich von den bestehenden Theorien deutlich abgrenzt. Entgegen mechanistischer Theorien, die eine Form zentraler Kontrollstruktur vorsehen, basiert die Theorie der CAS auf der Idee, dass Ordnung durch die Interaktion der Systemelemente emergiert (Benbya & McKelvey 2006). Gegenüber der Chaostheorie, die sich auf nicht voraussagbares Verhalten deterministischer Gleichungen fokussiert, beschäftigt sich die Theorie der CAS mit Ordnung und der Frage, wie diese Ordnung entsteht (Benbya & McKelvey 2006). Ein wesentlicher Fokus der CAS liegt dabei auf dem Zusammenspiel der Systeme mit ihrer Umwelt und der Koevolution, also des sich miteinander Weiterentwickelns, der Systeme und ihrer Umwelt (Anderson 1999; Choi et al. 2001).

Holland (1995) definiert CAS als Systeme, die aus einer großen Anzahl *interagierender, heterogener*⁹ „aktiver Elemente“ – sogenannter *Agenten* – bestehen. Das Verhalten der Agenten wird durch eine Menge an *Regeln* bestimmt. Diese Regeln haben typischerweise die Form eines Reiz-Reaktionsmusters: „wenn Reiz *s*, dann Reaktion *r*“ (vgl. Abbildung 5).

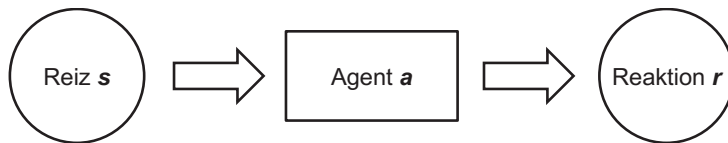


Abbildung 5: Reiz-Reaktions-Schema

Die Menge der Reiz-Reaktions-Regeln eines Agenten können durch die Reize, die der Agent empfangen kann, sowie die Reaktionen, die der Agent auf die Reize hin zeigt, beschrieben werden. Die Sequenzen, die aus den Reiz-Reaktions-Regeln gebildet werden können, beschreiben die möglichen Verhaltensmuster¹⁰ des Agenten. Basierend auf Erfahrungen, die Agenten sammeln, können sich die Agenten an ihre Umgebung anpassen, indem sie die den Verhaltensmustern zugrunde liegenden Regeln ändern. Diese Fähigkeit der Agenten wird auch als *Adaption und Lernen* bezeichnet (Holland 1995).

Das Verhalten des Gesamtsystems wird weder extern vorgegeben, noch wird es durch einen einzelnen Agenten bestimmt. Systemverhalten entsteht aus der Interaktion aller Agenten

⁹ Unterschiedlich in Gestalt und Fähigkeiten (vgl. Holland 1995, S. 6)

¹⁰ Kim & Kaplan (2006) verstehen diese Verhaltensmuster als kontextsensitive Strategien: „Actors, or populations of actors, interact with their environment and other actors within neighbourhoods, and employ a variety of context bound strategies that may be planned and purposeful or conditioned and reactive.“ (Kim & Kaplan 2006, S. 3)

(Choi et al. 2001). CAS durchlaufen dabei einen Prozess der *Selbstorganisation*, durch den neues Systemverhalten *emergiert*.

Die Agenten sind dabei nicht alle unmittelbar miteinander „verknüpft“, sondern interagieren nur mit einer Reihe von Agenten in ihrem Umfeld. Sowohl die Muster, nach denen die Agenten untereinander verknüpft sind, als auch die „Stärke“ der Verknüpfung sind dabei nicht statisch vorgegeben, sondern können sich im Zeitverlauf ändern. Durch regen Austausch und positive Erfahrungen bei der Interaktion (z. B. Wertschätzung der Zusammenarbeit) können sich die Verbindungen verstärken, sich nach einer einmaligen Interaktion jedoch auch wieder auflösen. Gleichmaßen können durch den Bedarf eines Austausches neue Verbindungen entstehen. Hierdurch ist eine Dynamik in der Struktur des Systems (im Sinne einer Netzwerktopologie) zu beobachten (Merali 2006). An dieser Stelle kommt auch das Konzept der beschränkten Rationalität („bounded rationality“; vgl. hierzu auch March & Simon (1958)) zum Tragen. Kein Agent hat vollständige Informationen über den Zustand des gesamten Systems zu einem bestimmten Zeitpunkt. Agenten agieren auf Basis einer beschränkten Informationsmenge über einen unmittelbaren Ausschnitt ihrer aktuell vorzufindenden Umgebung inklusive der Agenten im direkten Umfeld (vgl. Kim & Kaplan 2006). Demnach emergiert der Zustand des gesamten Systems stets aus dem lokalen Verhalten der Agenten, welches wiederum aus deren Erfahrungen und den mit diesen verbundenen Verhaltensmustern resultiert (Merali 2006). Evolution und Lerneffekte einzelner Agenten haben dabei Einfluss auf das Verhalten, die Evolution und die Lerneffekte anderer, mit diesen interagierenden Agenten. Dies führt zur sogenannten *Koevolution* der Agenten sowie der Koevolution der Agenten und des Systems (Anderson 1999). Die wesentlichen Charakteristiken der CAS sind in Tabelle 1 zusammengefasst.

Beispiele für CAS sind u. a. das zentrale Nervensystem (Agenten: Neuronen), das Immunsystem (Agenten: Antikörper), ein Unternehmen (Agenten: Mitarbeiter) eine Ökonomie (Agenten: Unternehmen) oder auch ein Ökosystem (Agenten: Lebewesen). Die Zeitdauer bis zu einer Änderung des CAS – basierend auf den Adaptionen- und Lernprozessen – bewegt sich im Falle der Nervensysteme im Bereich von Sekunden bis Stunden. Bei Immunsystemen liegt diese bei Stunden bis Tagen. Im Falle von Unternehmen bei Monaten bis Jahren (Holland 1995).

Neben den hier angeführten Beispielen aus der Biologie und der Ökonomie wird das Konzept CAS auch zunehmend auf den IS-Kontext angewandt. So untersucht Van Aardt (2004) die Entwicklung von Open Source Software als CAS und Hanseth & Lyytinen (2010) schlagen in ihrer Arbeit eine auf CAS basierende Designtheorie für die dynamische Komplexität in der Gestalt von Informationsinfrastrukturen am Beispiel des Internets vor. Benbya & McKelvey (2006) gehen einen Schritt weiter, indem sie jedes Informationssystem als CAS betrachten. Allen & Varga (2006) führen hingegen an, dass IT-Systeme – als Hard- und Software ein wesentlicher Teil der Informationssysteme – nicht die agentenhaften Eigenschaften eines CAS aufzeigen und sich somit auch nicht – im Sinne einer Koevolution – an sich ändernde Realitäten anpassen. Auf diesen Punkt geht die vorliegende Arbeit in Unterabschnitt 3.1.2.5 weiter ein.

Tabelle 1: Charakteristiken komplexer adaptiver Systeme

Charakteristik	Beschreibung	Literatur
Heterogene Agenten	CAS bestehen aus heterogenen Agenten, die sowohl untereinander als auch mit ihrer Umgebung interagieren und ihr Verhalten auf Grundlage vorheriger Erfahrungen anpassen können. Aufgrund der Möglichkeit kontinuierlicher Anpassung der lokal agierenden Agenten innerhalb des Systems, kann aus dem Verhalten der Agenten nicht auf das Verhalten des Gesamtsystems geschlossen werden.	Benbya & McKelvey (2006) Holland (1995)
Interaktion	Das Verhalten der Agenten kann sowohl durch die anderen Agenten als auch das System beeinflusst werden. Die Agenten agieren entsprechend ihrer Beziehungen innerhalb des Systems und ihrer Regelmenge.	Benbya & McKelvey (2006) Holland (1995)
Adaption und Lernen	Das Verhalten der Agenten wird vorgegeben durch Schemata. Ein Schema ist als eine Abfolge von Regeln, die aufgrund des durch den Agenten wahrgenommenen Zustandes der Umwelt anzuwenden ist, zu verstehen. Agenten können über geteilte als auch individuelle Schemata verfügen. Das Bilden neuer oder Anpassen bestehender Schemata aufgrund von Erfahrungen (z. B. einer sich ändernden Umweltsituation) oder der Variation bestehender Regeln wird als Adaption und Lernen bezeichnet.	Anderson (1999) Holland (1995)
Selbstorganisation und Emergenz	(Neues) Systemverhalten entsteht aus der Interaktion der Agenten: Entgegen der Kontrolltheorien aus der Kybernetik bestimmt weder ein einzelner Agent das Verhalten des Systems, noch werden die neu entstehenden Strukturen, Eigenschaften und Verhaltensweisen des Systems extern vorgegeben. Vielmehr durchlaufen CAS einen Prozess der Selbstorganisation in dessen Rahmen neues Systemverhalten durch die interagierenden Agenten entsteht (emergiert).	Anderson (1999) Benbya & McKelvey (2006) Choi et al. (2001) Holland (1998)
Koevolution	„Ein CAS und seine Umwelt interagieren und bilden dabei dynamische emergierende Realitäten“ (Choi et al. 2001, S. 356): Da Agenten – in Anlehnung an das Konzept der beschränkten Rationalität („bounded rationality“, vgl. hierzu auch March & Simon (1958)) – nicht die vollständigen Konsequenzen ihres Verhaltens für das Gesamtsystem beurteilen können, optimieren sie ihr Verhalten hinsichtlich ihrer eigenen Ziele, nicht jedoch hinsichtlich der des Gesamtsystems. Evolution und Lerneffekte einzelner Agenten haben Einfluss auf das Verhalten, die Evolution und die Lerneffekte anderer, mit diesen interagierenden Agenten. Dies führt zur sogenannten Koevolution der Agenten. Die gleiche Logik gilt für die Koevolution von interagierenden Mengen von Agenten sowie die des gesamten CAS mit dessen Umwelt.	Anderson (1999) Choi et al. (2001) March & Simon (1958) Merali (2006)

2.2 Architekturen

In diesem Unterkapitel wird zunächst der *allgemeine Architekturbegriff* eingeführt (Abschnitt 2.2.1) und darauf folgend auf den *Architekturbegriff im Kontext der Informationssysteme* eingegangen (Abschnitt 2.2.2). Im Anschluss daran werden in den Abschnitten 2.2.3 und 2.2.4 die Konzepte der *Unternehmensarchitektur* sowie der *IT-Architektur* diskutiert und voneinander abgegrenzt. Das *Architekturmanagement* – als eine Managementdisziplin in Unternehmen – wird in Abschnitt 2.2.5 dargestellt, bevor in Abschnitt 2.2.6 die *Rolle des Architekten* diskutiert wird.

2.2.1 Allgemeiner Architekturbegriff

Etymologisch geht der Begriff „Architektur“ auf das aus dem Lateinischen stammende Wort „*architectura*“ zurück. Dieses ist aus den griechischen Wörtern „*ἀρχή*“ (gesprochen [arché]; Bedeutung: „Anfang“, „Ursprung“, „Grundlage“ oder „das Erste“) und „*τέχνη*“ (gesprochen [techné]; Bedeutung: „Kunst“ oder „Handwerk“) zusammengesetzt. Es lässt sich demnach als „das Erste Handwerk“ oder „die Erste Kunst“ übersetzen (Pfeifer 1997; Seebold & Kluge 2011). Die heutige Bedeutung des Begriffs Architektur hängt von dem Kontext sowie dem Aufgabenfeld des „Architekten“¹¹ ab.

Bei Betrachtung des aus zehn Büchern bestehenden Werkes des römischen Architekten und Ingenieurs Marcus Vitruvius Pollio aus dem 1. Jahrhundert v. Chr. wird bereits der starke Bezug des klassischen Architekturbegriffs zum Städtebau sowie zur Gestaltung von Bauwerken wie Tempeln und sonstigen öffentlichen und privaten Gebäuden deutlich (Vitruvius Pollio 2004). In der Literatur ab der Neuzeit findet der Architekturbegriff keine einheitliche Verwendung. Ruskin (1865) versteht Architektur als die Kunst der Erbauung von Bauwerken:

“Architecture is the art which so disposes and adorns the edifices raised by man for whatsoever uses, that the sight of them contribute to his mental health, power and pleasure.” (Ruskin 1865, S. 7)

Browne (1658) legt bei seiner Verwendung des Architekturbegriffs den Fokus auf den Erstellungsprozess von Gebäuden bzw. Städten. Eine weitere Verwendung des Architekturbegriffs zeigt sich in der Bezeichnung der Gesamtheit von Bauten eines Stiles, eines Volkes bzw. einer Epoche. John Ruskin nennt in seinem Werk „*Stones of Venice*“ u. a. den byzantinischen, den romanischen und den gotischen Baustil (Ruskin 1851).

Mittlerweile hat der Architekturbegriff Einzug in die verschiedensten wissenschaftlichen Domänen gefunden und findet dort u. a. zur Beschreibung einer bestimmten Konstruktion oder Struktur Anwendung. So wird z. B. in der Biologie und der Chemie bei der Untersuchung molekularer Strukturen häufig auch von einer molekularen Architektur gesprochen (vgl. z.B. Ferreira et al. 2004; Pauling 1946; Woese et al. 1990).

¹¹ Der Begriff „Architekt“, lat. *architectus*, entstand aus dem altgriechischen *αρχιτέκτων* (gesprochen [architék-ton]). Letzteres setzt sich zusammen aus „*αρχι*“ (gesprochen [archi]) – was so viel bedeutet wie „erster“, „oberster“ oder „höchster“ – sowie „*τέκτων*“ (gesprochen [téktion]) – „Handwerker, Zimmermann“.

2.2.2 *Der Architekturbegriff im Kontext von Informationssystemen*

Für die Errichtung von Bauwerken werden bestimmte Elemente, wie unterschiedliche Baustoffe und Materialien, Werkzeuge, Baupläne, Skizzen und Vorgehensweisen benötigt. Analog zur klassischen Baukunst ist auch die Architektur von Informationssystemen¹² charakterisiert durch eine Vielzahl von miteinander in Beziehung stehenden Elementen und den dazugehörigen Prozesslandkarten und Dokumentationen (Matthes 2011). So überrascht es nicht, dass der Architekturbegriff auf Strukturen in den Feldern der Informatik und der Wirtschaftsinformatik übertragen wurde (Aier et al. 2008; Espinosa et al. 2011; Kruchten et al. 2006). Auch Begriffe aus der klassischen Architektur – wie dem Städtebau – haben mittlerweile Einzug in die Sprache der Architektur von Informationssystemen gefunden. Ein Beispiel hierfür ist der Begriff der Bebauungsplanung, der im Kontext der Informationssysteme zur Dokumentation der bestehenden (As-Is) sowie der geplanten (To-Be) IT-Landschaft Verwendung findet (Aier et al. 2012; Niemann 2005; Saat et al. 2009).

Wie bei der Einführung des allgemeinen Architekturbegriffs deutlich wurde, kann dieser unterschiedliche Perspektiven umfassen. Schmidt (2009) unterscheidet in seiner Arbeit drei Dimensionen des Architekturbegriffs:

1. Die ontologische Begriffsdimension ist auf das Objekt selbst fokussiert (z. B. ein Gebäude oder die grundlegende Struktur eines Systems).
2. Die pragmatische Begriffsdimension ist auf den (Erstellungs-)Prozess in Form von Handlungsvorschriften oder Prinzipien fokussiert. (z. B. ein Bauplan eines Hauses oder eine Menge an Prinzipien zur Entwicklung bzw. Weiterentwicklung eines Systems)
3. Die deskriptive Begriffsdimension ist auf die Beschreibung eines Objektes anhand von Modellen fokussiert¹³. Im Sinne der Modelltheorie handelt es sich hierbei um zweckorientierte Abbildungen der Struktur eines Systems (Schmidt 2009).

Die Internationale Organisation für Normung (ISO), die Internationale Elektrotechnische Kommission (IEC) und das Institute of Electrical and Electronics Engineers (IEEE) führen in dem gemeinsam veröffentlichten Standard ISO/IEC/IEEE 42010:2011¹⁴ Definitionen der für die Beschreibung der Architektur von Systemen relevanten Begriffe an (ISO/IEC/IEEE 2011). Architektur wird hier definiert als:

¹² Krcmar (2009) folgend werden "Informationssysteme" als soziotechnische Systeme verstanden, bestehend aus menschlichen sowie maschinellen Komponenten. Die maschinellen Komponenten teilen sich wiederum in Anwendungs- und Hardware-Komponenten auf.

¹³ Das im Rahmen der vorliegenden Arbeit verwendete Systemkonzept ist eng verknüpft mit der Modelltheorie. So werden reale Sachverhalte (wie u. a. die in Abschnitt 2.1.3 angeführten Beispiele) meist unter Zuhilfenahme von Konzepten der Modelltheorie beschrieben. Nach der allgemeinen Modelltheorie (Stachowiak 1973) ist unter einem Modell eine vereinfachte (Verkürzungsmerkmal) Darstellung (Abbildungsmerkmal) eines Originals zu verstehen. Ein Modell erfüllt dabei eine Ersetzungsfunktion des Originals, wobei sich die Gestalt des Modells auf die durch den jeweiligen Ersteller als wesentlich erkannten Elemente beschränkt und zu einem bestimmten Zeitpunkt sowie für einen definierten Zweck erstellt wird (pragmatisches Merkmal).

¹⁴ Der ISO/IEC/IEEE 42010:2011 "Systems and Software engineering – Architecture description" ersetzt den vielzitierten IEEE Std. 1471-2000 "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems" (IEEE Architecture Working Group 2000), welcher durch die ISO und die IEC auch als ISO/IEC 42010:2007 geführt wurde.

„[...] *fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*“ (ISO/IEC/IEEE 2011, S. 2)

Nach dieser Definition besteht eine Architektur aus den Konzepten und Eigenschaften eines Systems, die sich sowohl am realen System selbst zeigen als auch an den (verschriftlichten) Prinzipien bezüglich der Ausgestaltung und Weiterentwicklung des Systems. Damit berücksichtigt diese Definition die ontologische und die pragmatische Begriffsdimension. Die deskriptive Perspektive wird im ISO/IEC/IEEE (2011) durch die Architekturdokumentation aufgegriffen. Die sogenannte „architecture description“ wird dabei definiert als:

„*work product used to express an architecture*“ (ISO/IEC/IEEE 2011, S. 2)

Die angeführten Arbeitsergebnisse umfassen u. a. die Identifikation der Systemakteure samt deren Interessen und Belangen, die Definition der für die jeweilige Architektur zugrunde liegenden Architekturperspektiven (sog. „architecture viewpoints“), Architekturmodelle für die verwendeten Architekturperspektiven, aber auch getroffene Architekturentscheidungen, deren Begründungen sowie die durch diese entstehenden Implikationen für das System und dessen Umfeld.

Die vorliegende Arbeit verwendet den Architekturbegriff im Sinne der ontologischen Begriffsdimension. In Anlehnung an den ISO/IEC/IEEE (2011) wird *Architektur* im Kontext von Informationssystemen aus einer systemtheoretischen Perspektive wie folgt definiert: *Eine Architektur ist die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht.*

2.2.3 Unternehmensarchitektur

Der zuvor eingeführte Architekturbegriff im Kontext von Informationssystemen unterteilt sich in der wissenschaftlichen Literatur – abhängig von dem betrachteten System – in zwei Forschungsströme. Im Forschungsfeld der Kerninformatik liegt der Fokus auf den Software-systemen und damit der Softwarearchitektur. Betrachtungsgegenstand sind hierbei die Struktur und die Organisation von Software-Komponenten und deren Schnittstellen (Kruchten et al. 2006; Parnas 1972). In der Wissenschaftsdisziplin der Wirtschaftsinformatik – die ihre Wurzeln sowohl in der Informatik als auch in den Wirtschaftswissenschaften hat – bzw. dem angelsächsischen Pendant¹⁵ der Information Systems wird eine ganzheitlichere Perspektive auf die sog. Unternehmensarchitektur (engl. „Enterprise Architecture“) angenommen. Der ganzheitliche Charakter zeigt sich anhand der über die reine Softwarearchitektur hinausgehenden Betrachtung der organisationalen und der technischen Ebene (vgl. Tabelle 2).

¹⁵ Frank et al. (2008) zeigen in ihrer Arbeit bestehende Unterschiede in den Forschungszielen sowie –methoden auf. Herrscht in der Wirtschaftsinformatik (WI) ein eher konstruktiver Ansatz vor, haben sich in der angelsächsischen Information Systems (IS) Community in erster Linie Ansätze behavioristischer Sozialforschung etabliert. In einer aktuellen Untersuchung der methodischen Schwerpunkte in der WI kommen Schreiner et al. (2015) zu dem Ergebnis, dass konstruktive Methoden in der WI weiterhin leicht überwiegen, die WI sich jedoch zunehmend den verhaltenswissenschaftlichen Methoden öffnet. Gleichzeitig etablieren sich konstruktive Ansätze in der IS-Forschung (Gregor & Hevner 2013).

Der Begriff der Unternehmensarchitektur ist sowohl in der Praxis als auch der Forschung weit verbreitet (vgl. z.B. Ahlemann et al. 2012; Aier et al. 2008; Ross et al. 2006), wird allerdings nicht einheitlich verwendet. Aier et al. (2008) diskutieren in ihrer Arbeit Publikationen mit Bezug zu Unternehmensarchitekturen und stellen dabei das unterschiedliche Verständnis des jeweiligen Gestaltungsgegenstandes der Unternehmensarchitektur sowie die verschiedenen Nutzungsformen in Literatur und Praxis heraus.

Der wesentliche Gestaltungsgegenstand etablierter Unternehmensarchitektur-Rahmenwerke¹⁶ wie TOGAF (The Open Group Architecture Framework; The Open Group 2011) oder EAP (Enterprise Architecture Planning; Spewak & Hill 1993) sind die vier Architekturebenen Geschäftsprozesse, Daten, Anwendungen und Infrastruktur (vgl. Tabelle 2).

Tabelle 2: Ebenen einer Unternehmensarchitektur

Architekturebene	Beschreibung	Quelle	Beispiel
Geschäftsprozess-architektur (GPA)	Ausgestaltung der Geschäftsprozesse, bestehend aus einer Menge von Aktivitäten, Aufgaben und Services	Ross et al. (2006), S. 48 Pulkkinen (2006), S. 2	Kreditprozess in einer Bank
Datenarchitektur (DA)	Struktur der logischen und physischen Datenbestände	The Open Group (2011), S. 29	Syntax und Semantik von Kundeninformationen in der Datenbank
Anwendungsarchitektur (AA)	Struktur der Anwendungen und deren Schnittstellen; Anwendungen verwenden Daten und werden durch Infrastrukturkomponenten unterstützt.	Ross et al. (2006), S. 48 The Open Group (2011), S. 23	Fachliche Anwendungen und deren Schnittstellen
Infrastrukturarchitektur (IA)	Eine Sammlung von Technologie-Komponenten (Hardware und/oder Software) die Services anbieten, um Anwendungen zu unterstützen	Pulkkinen (2006), S. 2 The Open Group (2011), S. 24, 34	Server (Hardware oder Software), Betriebssysteme, Datenbankmanagementsysteme

Wie Tabelle 3 zeigt, wird in der wissenschaftlichen Literatur eine vergleichbare Aufteilung vorgenommen. Die in den Arbeiten von Richardson et al. (1990) und Iyer & Gottlieb (2004) verwendeten Begriffsdefinitionen haben einen eher technischen Fokus, was sich daran zeigt, dass die Geschäftsprozesse keine Berücksichtigung finden. Die weiteren in Tabelle 3 aufgeführten Definitionen beziehen die Geschäftsprozesse mit ein und ähneln sich im Kern grundsätzlich.

¹⁶ Die vorliegende Arbeit verwendet die Definition eines Unternehmensarchitektur-Rahmenwerkes (engl. "Architecture Framework") nach The Open Group (2011): "An architecture framework is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks." (The Open Group 2011, S. 7)

Tabelle 3: Ausgewählte Definitionen des Begriffs Unternehmensarchitektur

Quelle	Definition	Berücksichtigte Perspektiven
Richardson et al. (1990), S. 386	“[Enterprise Architecture] defines and interrelates data, hardware, software, and communications resources, as well as the supporting organization required to maintain the overall physical structure required by the architecture.”	<ul style="list-style-type: none"> ▪ Daten ▪ Hardware ▪ Anwendungen ▪ Beziehungen ▪ Organisation für den Betrieb und die Wartung der Infrastruktur
Iyer & Gottlieb (2004), S. 587	“The decomposition of the enterprise into manageable parts, the definition of those parts, and the orchestration of the interaction among those parts constitutes the enterprise architecture.”	<ul style="list-style-type: none"> ▪ Dekomposition in handhabbare Abschnitte ▪ Interaktionen
Ross et al. (2006), S. 9	“Enterprise Architecture is the organizing logic for business processes and IT-infrastructure, reflecting the integration and standardization requirements of the company’s operating model. The enterprise architecture provides a long-term view of a company’s processes, systems, and technologies so that individual projects can built capabilities – not just fulfill immediate needs.”	<ul style="list-style-type: none"> ▪ Geschäftsprozesse ▪ Systeme (Informationen + Anwendungen) ▪ IT-Infrastruktur ▪ Potenziale
Jonkers et al. (2006), S. 64	“Architecture at the level of an entire organisation is commonly referred to as “enterprise architecture” (EA). It is a coherent whole of principles, methods and models that are used in the design and realisation of the enterprise’s organizational structure, business processes, information systems, and infrastructure. EA captures the essentials of the business, IT and its evolution.”	<ul style="list-style-type: none"> ▪ Prinzipien, Methoden und Modelle ▪ Organisationsstruktur ▪ Geschäftsprozesse ▪ Informationssysteme ▪ Infrastruktur
Tamm et al. (2011), S. 141	“Enterprise architecture (EA) is the definition and representation of a high-level view of an enterprise’s business processes and IT systems, their interrelationships, and the extent to which these processes and systems are shared by different parts of the enterprise.”	<ul style="list-style-type: none"> ▪ Geschäftsprozesse ▪ IT-Systeme ▪ Beziehungen
Schmidt & Buxmann (2011), S. 169	“The EIS [Enterprise Information System] can be defined as a firm’s information processing subsystem consisting of all information processing tasks, the applications used to perform these tasks as well as all the technical components necessary to run the applications. Just like with any other system, the fundamental organization of the EIS (as embodied in its components, their relationships to each other and to the environment) may be referred to as the system’s architecture.[...] In accordance with industry practices, we use the term EA for this.”	<ul style="list-style-type: none"> ▪ Informationsverarbeitende Aufgaben (Daten und Prozesse) ▪ Anwendungen ▪ Technische Komponenten ▪ Beziehungen
Bradley et al. (2012), S. 98	“Broadly defined, EA is the organising logic for an organisation’s IT infrastructure and business processes.”	<ul style="list-style-type: none"> ▪ Geschäftsprozesse ▪ IT-Infrastruktur

Obleich nicht alle Definitionen die Ebenen der Daten-, Anwendungs- und Infrastrukturarchitektur explizit aufführen – z. B. fassen Tamm et al. (2011) und Bradley et al. (2012) diese Ebenen als „IT-Systeme“ bzw. „IT-Infrastruktur“ zusammen – haben die Autoren ein vergleichbares Verständnis.

Erwähnt der Großteil der Autoren zudem diverse Ausprägungen von Beziehungen zwischen den Elementen der Architekturebenen, so führen Schmidt & Buxmann (2011) diese Beziehungen zwischen den Architekturebenen explizit an:

“[...] information processing tasks, the applications used to perform these tasks as well as all the technical components necessary to run the applications [...]” (Schmidt & Buxmann 2011, S. 169)

Die vorliegende Arbeit folgt einem holistischen Verständnis einer Unternehmensarchitektur und bezieht die Ebenen der Geschäftsprozess-, Daten-, Anwendungs- und Infrastrukturarchitektur sowie deren Beziehungen zueinander ein (vgl. Jonkers et al. 2006; Ross et al. 2006; Schmidt & Buxmann 2011).

2.2.4 IT-Architektur

Der Begriff Unternehmensarchitektur (oder Enterprise Architecture) wird in der Praxis häufig nur einer Teilmenge der o. a. vier Architekturebenen zugeordnet (Ross et al. 2006, S. 48). Ein Beispiel hierfür ist die bereits im vorherigen Abschnitt 2.2.3 angeführte Definition von Richardson et al. (1990), welche die Teilmenge der Daten-, Anwendungs- und Infrastrukturarchitektur einbezieht, die Geschäftsprozessarchitektur jedoch außen vor lässt. Da dieser von Richardson et al. (1990) gewählte IT-Fokus in der Literatur und Praxis häufig Verwendung findet und auch für die weitere Arbeit eine wichtige Rolle spielt, wird der Begriff der IT-Architektur im Folgenden eingeführt und vom Begriff der Unternehmensarchitektur abgegrenzt.

Jeanne W. Ross führt in ihrer Arbeit „Creating a Strategic IT Architecture Competency: Learning in Stages“ die folgende Definition für eine IT-Architektur an:

“Thus, at the enterprise level, an IT architecture is the organizing logic for applications, data and infrastructure technologies, as captured in a set of policies and technical choices, intended to enable the firm’s business strategy. Accordingly, the enterprise architecture implies certain IT capabilities. These capabilities are the objectives of the IT architecture, specifying what the architecture enables the business to do.” (Ross 2003, S. 31)

Eine IT-Architektur umfasst nach dieser Definition mit der Daten-, Anwendungs- und Infrastrukturarchitektur die gleichen Gestaltungsobjekte wie bereits von Richardson et al. (1990) angeführt. Gleichzeitig stellt Ross (2003) den Zusammenhang der IT-Architektur zur Geschäfts-Perspektive her: Die IT-Architektur muss zum einen die IT-Anforderungen – bedingt durch die (geplante) Ausgestaltung der Geschäftsprozessarchitektur – erfüllen, bildet zum anderen jedoch auch IT-Potenziale als Basis für die weitere Entwicklung der Geschäftsberei-

che heraus. Abbildung 6 fasst die Abgrenzung der IT-Architektur von der Unternehmensarchitektur sowie deren Beziehung zueinander zusammen.

Wie bereits in Abschnitt 2.2.2 angeführt, verwendet diese Arbeit – in Anlehnung an den ISO/IEC/IEEE (2011) – eine systemtheoretische Perspektive auf eine Architektur im Kontext von Informationssystemen. Diese Perspektive erlaubt eine Unterteilung eines hierarchischen Systems (Simon 1962) in eine Menge miteinander verbundener Subsysteme (vgl. hierzu auch das in Abschnitt 2.1.1 eingeführte hierarchische Konzept nach Ropohl (2009)). Für die IT-Architektur entspricht dies den Subsystemen „Datenarchitektur“, „Anwendungsarchitektur“ und „Infrastrukturarchitektur“ sowie zusätzlich der „Geschäftsprozessarchitektur“ im Falle der Unternehmensarchitektur (vgl. Abbildung 6).

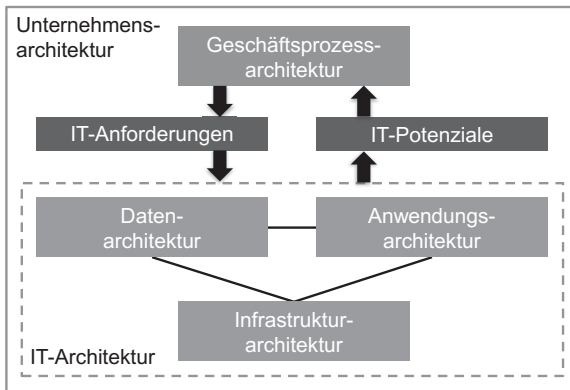


Abbildung 6: Abgrenzung Unternehmens-/IT-Architektur nach Schütz et al. (2013a)

Da die Komponenten und Beziehungen dieser Subsysteme unterschiedliche Eigenschaften aufweisen (vgl. Tabelle 2), erlaubt eine solche Unterteilung in die genannten Subsysteme eine separate Betrachtung, während die wechselseitigen Beziehungen der Subsysteme untereinander weiterhin berücksichtigt werden können.

2.2.5 Architekturmanagement

Das Architekturmanagement (engl. „Enterprise Architecture Management“ (EAM)) ist eine Management-Disziplin (Ahlemann et al. 2012), welche die Prozesse, Methoden, Werkzeuge, Verantwortlichkeiten und Standards zur Planung, Umsetzung, Kontrolle und Steuerung der Entwicklung der Unternehmensarchitektur umfasst (Niemann 2005). In der Literatur finden sich auch zu dem Begriff des Architekturmanagements unterschiedliche Perspektiven, die sich auf eine Veränderung des Fokus des Architekturmanagements im Laufe der Jahre zurückführen lassen. Diese Entwicklung wird in dem folgenden Unterabschnitt 2.2.5.1 aufgezeigt. Nach einer Darstellung der Ziele des Architekturmanagements in Unterabschnitt 2.2.5.2 wird in Unterabschnitt 2.2.5.3 die Standardisierung als eine etablierte Maßnahme des Architekturmanagements vorgestellt. Darüber hinaus wird diskutiert, warum dieser Ansatz nur eine Teilmenge der Ziele des Architekturmanagements berücksichtigt.

2.2.5.1 Die Entwicklung des Architekturmanagements

In den letzten 30 Jahren hat sich das Architekturmanagement von einem eher technischen – mit dem Fokus auf der Modellierung – zu einem die Geschäftsstrategie der Unternehmen berücksichtigenden Managementansatz weiterentwickelt (Ahlemann et al. 2012).

Die Wurzeln des Architekturmanagements sind dabei in Zachmans Framework für die Architektur von Informationssystemen zu finden (Zachman 1987). Dieses beinhaltet Empfehlungen dazu, welche architektonischen Aspekte (im Zachman Framework lauten diese Daten, Funktionen und Netzwerk) aus verschiedenen Perspektiven (u. a. Planer, Designer oder Eigentümer) bei der Ausgestaltung einer IT-Architektur herangezogen und modelliert werden sollten. In der Arbeit von Sowa & Zachman (1992) wurde das initiale Zachman Framework um organisationale Aspekte ergänzt (Menschen, Zeit und Motivation). Der auf der Modellierung der Architekturen liegende Fokus blieb jedoch erhalten. Im Laufe der 90er Jahre zeigte sich, dass ein rein modellbasierter, primär strukturbeschreibender Ansatz (vgl. hierzu auch Krcmar 1990; Scheer 1991) nicht ausreicht, um den Entwicklungen in der IT Herr zu werden (Ahlemann et al. 2012). Die immer weiter steigende IT-Durchdringung in den Unternehmen und die in diesem Kontext häufig durch die einzelnen Geschäftsbereiche dezentral getroffenen IT-Entscheidungen führten zur Entstehung sog. „Business Silos“. Diese zeichneten sich durch IT-Lösungen aus, die für den jeweiligen fachlichen oder geographischen Kontext zugeschnitten waren. Hierdurch entstehende positive Effekte (wie z. B. die Freiheitsgrade der Entwickler, sich stets aus den innovativsten und für den Anwendungsfall optimal passenden Technologien bedienen zu können) wurden in kurzer Zeit durch die gleichzeitig entstehenden Schwierigkeiten bei der Integration neuer Anwendungen in die bestehende Anwendungsarchitektur aufgewogen (Ross 2003). Aus dieser Problemstellung verschob sich die Rolle des EAM von der Modellierungssicht hin zu einer Management-Disziplin mit IT-Managementprozessen für die Planung, Umsetzung, Kontrolle und Steuerung der IT-Architektur. Anstelle einer Konzentration auf einzelne Anwendungen etablierte sich eine ganzheitliche Perspektive auf die Organisation der IT-Architektur des Unternehmens. Diese Entwicklung spiegelt sich auch in den Architekturrahmenwerken wieder, die zwar eine – häufig auf den Erkenntnissen von Zachman (1987) basierende – Form der Modellierung vorsehen, gleichzeitig aber auch Vorgaben für die Planung, Umsetzung, Kontrolle und Steuerung der IT-Architektur enthalten. Beispiele hierfür sind die auf dem Rahmenwerk von Zachman basierenden EAP (Spewak & Hill 1993), FEAF (The Chief Information Officers Council 1999) oder IAF (Integrated Architecture Framework; Wout et al. 2010). In der näheren Vergangenheit nach 2000 finden sich in der IS-Literatur zudem vermehrt Arbeiten, die sich Fragestellungen des Managements von Unternehmensarchitekturen und der Verbindung dieser mit der Unternehmensstrategie widmen. In diesen Arbeiten zeigt sich, dass ein erfolgreiches Architekturmanagement einer engen Abstimmung mit der Strategie des Unternehmens bedarf (z.B. Pulkkinen & Hirvonen 2005; Ross et al. 2006; Tamm et al. 2011). Dies wird nicht zuletzt auch aus der Integration des Literaturstranges des Business-IT Alignment deutlich (Bradley et al. 2012; Tamm et al. 2011).

2.2.5.2 Ziele des Architekturmanagements

Das in Abschnitt 2.2.2 eingeführte Verständnis einer Architektur im Kontext der Informationssysteme lässt vermuten, dass Architekturen relativ statische Konstrukte sind, die sich zwar weiterentwickeln, dies jedoch nach vorgegebenen Prinzipien. Da sich Organisationen stetig an ein sich verändertes (Geschäfts-)Umfeld anpassen, besteht auch an die Unternehmens- und IT-Architektur die Anforderung, sich diesen Änderungen anpassen zu können. Wie bereits angeführt, sind Planung, Umsetzung, Kontrolle und Steuerung der Entwicklung einer Architektur der Kern aktueller Architekturmanagement-Ansätze. Die durch das Architekturmanagement dabei verfolgten Ziele sind im Wesentlichen in der Unterstützung der Geschäftsziele durch die Bereitstellung und Aufrechterhaltung einer *(kosten-)effizienten und flexiblen* Unternehmens- bzw. IT-Architektur zu sehen (vgl. Ahlemann et al. 2012; Richardson et al. 1990; Ross et al. 2006; Schmidt & Buxmann 2011; Tamm et al. 2011). Für die weitere Arbeit werden diese Zielgrößen im Folgenden definiert.

2.2.5.2.1 IT-Flexibilität

Für den Begriff *Flexibilität* existiert keine allgemein akzeptierte Definition. Shewchuk & Moodie (1998) geben an, mehr als 70 verschiedene Definitionen in der Literatur identifiziert zu haben. In der Managementliteratur wird Flexibilität definiert als:

“[...] the degree to which an organization possesses a variety of actual and potential procedures, and the rapidity with which it can implement these procedures to increase the control capability of the management and improve the controllability of the organization over its environment.” (Byrd & Turner 2000, S. 170)

Aufbauend auf dieser Definition nehmen Tallon & Kraemer (2003) eine Differenzierung von IT-Flexibilität und strategischer Flexibilität vor:

“[...] IT flexibility reflects the ability to easily and quickly introduce new sources of IT support for the business strategy, while strategic flexibility reflects the ability to easily and quickly change key aspects of the business strategy in areas such as pricing, market reach, etc.” (Tallon & Kraemer 2003, S. 2)

Die vorliegende Arbeit orientiert sich an dem Begriffsverständnis von „IT-Flexibilität“ nach Tallon & Kraemer (2003) und definiert diese wie folgt:

IT-Flexibilität ist die Eigenschaft einer IT-Architektur – mit angemessenem Aufwand – an neue (Geschäfts-)Anforderungen angepasst werden zu können.

2.2.5.2.2 IT-Effizienz

Den Begriff der *Effizienz* beschreibt Drucker (1986) aus einer organisationalen Perspektive als:

“[...] doing better what is already being done.” (Drucker 1986, S. 36)

Eine vergleichbare Tendenz in Richtung der Optimierung, jedoch mit explizitem Fokus auf dem Ressourceneinsatz, bzw. den Kosten zeigt die Definition von Butler (1999):

„Efficiency refers to the minimization of resources used to discover and implement solutions.“ (Butler 1999, S. 221)

In Anlehnung an die angeführte Ressourcen- bzw. Kostenperspektive wird die folgende Definition des Begriffs „IT-Effizienz“ verwendet:

IT-Effizienz ist die Eigenschaft einer IT-Architektur, bestehende Geschäftsanforderungen mit möglichst geringem Ressourceneinsatz erfüllen zu können.

2.2.5.3 Standardisierung – eine Maßnahme des Architekturmanagements

Eine häufig in der Forschung diskutierte und in der Praxis verwendete Maßnahme für das Architekturmanagement ist die der Standardisierung. In der vorliegenden Arbeit wird der Begriff der Standardisierung im Sinne der Definition der „Standardisierung in betrieblichen Informationssystemen“ nach Buxmann (1996) verwendet:

„Unter Standardisierung in betrieblichen Informationssystemen wird eine Entscheidung über den Einsatz von Standards zur Vereinheitlichung von Systemelementen verstanden [...]“ (Buxmann 1996, S. 10)

Ein Standard ist dabei – der Definition von De Vries (2006) folgend – zu verstehen als:

“[...] approved specification of a limited set of solutions to actual or potential matching problems, prepared for the benefits of the party or parties involved, balancing their needs, and intended and expected to be used repeatedly or continuously, during a certain period, by a substantial number of the parties for whom they are meant.“ (De Vries 2006, S. 4)

In der Praxis finden häufig Standardkataloge (sogenannte „Books of Standards“) Anwendung, in denen Produkte und Technologien dokumentiert sind, die im Zuge einer Neuentwicklung oder Änderung für den Einsatz im Unternehmen freigegeben sind (vgl. Schmidt 2009; The Open Group 2011). In Tabelle 4 ist eine Auswahl wissenschaftlicher Veröffentlichungen im Feld der Unternehmensarchitekturen angeführt, in denen (IT-) Standardisierung eine wesentliche Rolle spielt.

Die bestehende Forschung zur IT-Standardisierung verwendet zur Lösung von Standardisierungsproblemen meist mathematische (Optimierungs-)Modelle, in deren Zielfunktionen die Reduktion von Kosten und somit die Erhöhung der IT-Effizienz eine bedeutende Rolle spielt (Buxmann et al. 1999; Li & Chen 2012; Weitzel et al. 2006; Widjaja 2010). Eng hiermit verbunden ist der Gedanke, dass eine Reduktion der Heterogenität der IT-Architektur i. d. R. wünschenswert ist. Dies ist damit zu begründen, dass ein Abweichen von innerhalb eines Unternehmens vorgegebenen IT-Standards – wie beispielsweise den in den o. a. Standardkatalogen dokumentierten Technologien – mögliche Synergieeffekte nachteilig beeinflusst (Boh & Yellin 2006; Guillemette & Paré 2012; Richardson et al. 1990). Hieraus wird zudem deutlich, dass sich IT-Standardisierung stark an dem in der IS-Forschung etablierten ressourcenbasier-

ten Ansatz orientiert. Sie berücksichtigt hingegen nicht die sich stetig ändernden Anforderungen durch die Geschäftsseite (Allen & Varga 2006).

IT-Standardisierung hat demzufolge einen starken Fokus auf der Erhöhung der IT-Effizienz durch einen kostenoptimierten Ressourceneinsatz. Eng mit der IT-Standardisierung verbunden ist eine Reduktion der Heterogenität der in einer IT-Landschaft vorherrschenden Technologien.

Tabelle 4: IS-Forschungsbeiträge im EA-Kontext mit Standardisierungs-Linse

Quelle	Forschungsansatz	Relevante Ergebnisse für die vorliegende Arbeit
Richardson et al. (1990)	Fallstudie und Erarbeitung von Designprinzipien zur Ausgestaltung der IT-Architektur	<ul style="list-style-type: none"> ▪ Einführung des Gedankens einer prinzipienbasierten Unternehmens- bzw. IT-Architektur ▪ Eine homogene [standardisierte] Architektur verbessert Informationsflüsse sowie die Support-Kosten des Systems
Ross et al. (2006)	Fallstudien bei 69 Unternehmen (aus den USA und Europa) zu Themen des Unternehmensarchitekturmanagements	<ul style="list-style-type: none"> ▪ Etablierung einer neuen, strategischen Perspektive auf das Management von Unternehmensarchitekturen ▪ Hervorhebung von 1) IT-Standardisierungs- und Integrationsentscheidungen bei der Auflösung des Wildwuchses historisch gewachsener IT-Architekturen und 2) der Ausgestaltung wiederverwendbarer Plattformen
Boh & Yellin (2006)	Empirische Untersuchung 1) des Einflusses von Governance- Mechanismen auf EA-Standards sowie 2) der Wirkung des Einsatzes von EA-Standards in den Architekturebenen	<ul style="list-style-type: none"> ▪ Unterstützende Wirkung der Nutzung von IT-Standards bei dem Management der IT-Infrastruktur-, Anwendungs- sowie Datenressourcen ▪ Herstellung der Verbindung zwischen IT-Standardisierung und der Reduktion von Heterogenität und Redundanz sowie der zunehmenden Integration der IT-Architektur
Schmidt & Buxmann (2011)	Konzeptentwicklung sowie empirische Untersuchung der Wirkung eines Managements von Unternehmensarchitekturen	<ul style="list-style-type: none"> ▪ Vorschlag einer Konzeptualisierung des Managements von Unternehmensarchitekturen ▪ Nachweis der Wirkung eines Managements von Unternehmensarchitekturen auf IT-Effizienz und IT-Flexibilität ▪ Herstellung der Verbindung des Managements von Unternehmensarchitekturen mit der Reduktion von IT-Komplexität und einer intensiveren Nutzung von Standardisierung¹⁷
Li & Chen (2012)	Konzeptentwicklung und mathematische Formulierung von IT-Standardisierungsproblemen	<ul style="list-style-type: none"> ▪ Betonung von IT-Standardisierung als wesentliche Aufgabe des IT-Managements ▪ IT-Standardisierung führt zu Kosteneinsparungen im administrativen Bereich der IT und steigert die Kompatibilität zwischen den eingesetzten Produkten ▪ IT-Standardisierung geht zu Lasten des Wunsches der Nutzer nach Vielseitigkeit

Die zweite der in Abschnitt 2.2.5.2 eingeführten Zielgrößen des Architekturmanagements, die IT-Flexibilität, steht bei der IT-Standardisierung jedoch nicht im Vordergrund. Dies wird

¹⁷ Standardisierung findet sich in drei der sieben von Schmidt & Buxmann (2011) eingeführten Indikatoren des Konzeptes EAM wieder.

auch durch die gezielte Reduktion von Heterogenität deutlich: Während Heterogenität von Technologien für die Erreichung eines Kostenoptimums – z. B. durch eine Erhöhung der Informationskosten aufgrund mangelnder Kompatibilität der Schnittstellen – tendenziell eher schädlich ist, kann diese aus einer IT-Flexibilitäts-Perspektive durchaus erwünscht sein. So kann eine größere Heterogenität die Einführung einer neuen Lösung auch begünstigen, da im Zuge der Umsetzung auf eine größere Auswahl verfügbarer Technologien zurückgegriffen werden kann. Heterogenität kann also nicht per se als etwas schädliches angesehen werden. Vielmehr ist ein gewisser Grad an Heterogenität erforderlich, um sowohl gegebene als auch neu hinzukommende Anforderungen umsetzen zu können. Eine Aufgabe des Architekturmanagements besteht demnach darin, einen angemessenen Grad an Heterogenität zu identifizieren.

2.2.6 Die Rolle des Architekten

Die Rolle in der Organisation, die sich in erster Linie dem Management der Unternehmens- bzw. IT-Architektur widmet, ist die des Architekten¹⁸. Der IEEE Standard 1471–2000 definiert den Architekten (engl. „architect“) als:

„The person, team, or organization responsible for systems architecture.“ (IEEE Architecture Working Group 2000, S. 3)

Ahlemann et al. (2012) spezifizieren das Aufgabenspektrum des Architekten durch die Verwendung der Analogie zur Städteplanung¹⁹. Ihnen zufolge muss ein Architekt u. a. zukünftige Anforderungen antizipieren, Pläne erstellen, verschiedene Interessengruppen zusammenbringen und eine ganzheitliche – verschiedene Zielgrößen berücksichtigende – Perspektive einnehmen (Ahlemann et al. 2012, S. 8).

Guillemette & Paré (2012) führen das Management der Komplexität der IT-Architektur als eine der fundamentalen Aufgaben der IT-Abteilungen in Unternehmen an. Für das hierfür verantwortliche Profil verwenden Guillemette & Paré (2012) den Begriff „Architecture Builder“. Aufgabe des „Architecture Builders“ ist es dabei:

“[...] to build and manage an IT infrastructure that supports business processes and reduces architectural complexity.” (Guillemette & Paré 2012, S. 532).

Die im Zusammenhang mit diesen Aufgaben verbundenen Aktivitäten sollen neben der *Reduktion der Komplexität der IT-Architektur* (z. B. durch Standardisierung) auf die Entwicklung einer integrierten und *flexiblen IT-Architektur* abzielen (vgl. Unterabschnitt 2.2.5.3).

¹⁸ In der Praxis finden sich unterschiedliche Rollenbezeichnungen wie z. B. „Enterprise Architect“, „IT Enterprise Architect“ oder „IT Architect“. Da Architektur-Initiativen häufig durch die IT-Bereiche getrieben werden, befassen sich die Mitarbeiter in den angeführten Rollen grundsätzlich mit Themen im IT-nahen Umfeld der IT-Architektur und den aus dieser Perspektive heraus bestehenden Schnittstellen zur Geschäftsprozessebene.

¹⁹ Die Metapher der Städteplanung wird in einer Reihe von Arbeiten verwendet und diskutiert (Namba & Iijima 2004; Schmidt & Buxmann 2011). Eine kritische Betrachtung der Analogie zwischen der Städteplanung und dem Enterprise Architecture Management führt Ross (2003) an. Ihr zufolge wird hierdurch lediglich eine eher technologische Perspektive auf die Beziehung zwischen der IT und den Geschäftsprozessen abgebildet. Das strategische Potenzial einer Unternehmensarchitektur bleibt Ross zufolge außen vor.

2.3 Theoretischer Bezugsrahmen der Arbeit

In diesem Abschnitt werden die in 2.1 und 2.2 eingeführten Konzepte zu einem theoretischen Bezugsrahmen für die vorliegende Arbeit zusammengeführt (vgl. Abbildung 7).

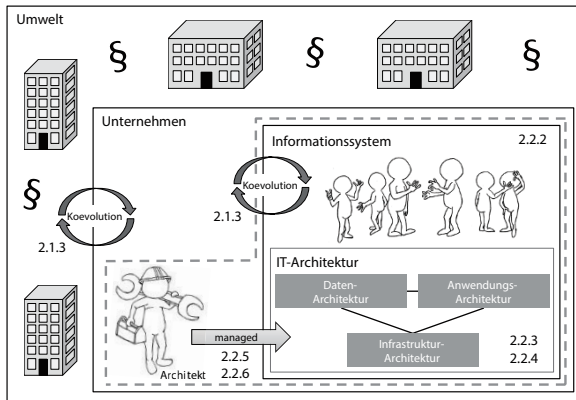


Abbildung 7: Theoretischer Bezugsrahmen der Arbeit

Wie in Abschnitt 2.2.2 dargestellt, wird im Kontext von Informationssystemen ein an die Systemtheorie (vgl. Abschnitt 2.1.1) angelehnter Architekturbegriff verwendet. Eine Architektur wird definiert als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht. Informationssysteme werden in der wissenschaftlichen Literatur häufig als CAS betrachtet. Somit müssten Informationssysteme den in Tabelle 1 angeführten System-Charakteristiken, wie u. a. Selbstorganisation, Emergenz und Koevolution folgen. Wie in Abschnitt 2.1.3 angeführt, zeigt die IT-Architektur (vgl. Abschnitte 2.2.3 und 2.2.4) jedoch keine agentenhaften Charakteristiken, wie sie die Theorie der CAS unterstellt. Speziell die Charakteristik der Koevolution ist hier nicht zutreffend. Um eine solche Koevolution der IT-Architektur sicherzustellen, hat ein Agent im Umfeld des Informationssystems – der IT-Architekt (Abschnitt 2.2.6) – die Aufgabe, die IT-Architektur in der Form aufzubauen und zu managen, dass sie den sich ändernden Anforderungen seiner Umwelt genügt und dabei seine Anpassbarkeit (IT-Flexibilität) nicht verliert. Abschnitt 2.2.5 zeigt, dass bestehende Strategien (speziell die der IT-Standardisierung) primär auf die Erreichung von IT-Effizienz abzielen, das Ziel einer IT-Flexibilität jedoch außer Acht lassen.

Die übergeordnete Fragestellung der Arbeit ist demnach, wie die IT-Architektur hinsichtlich einer sich im ständigen Wandel befindlichen Umwelt gemanagt werden kann. Hierzu soll im Folgenden mit der IT-Komplexität – bzw. genauer der strukturellen Komplexität von IT-Architekturen – eine neue, die bestehenden Perspektiven komplementierende Perspektive untersucht werden. Der Fokus der vorliegenden Arbeit liegt dabei auf dem in Abbildung 7 durch die gestrichelte Linie eingerahmten Teilbereich. Im folgenden Kapitel 3 wird der Begriff der Komplexität in der IS-Literatur untersucht und auf Grundlage einer strukturierten Literaturrecherche die weitere Forschungsagenda der Arbeit abgeleitet.

3 Strukturelle Komplexität in Unternehmens-/IT-Architekturen²⁰

Basierend auf einer strukturierten Literaturrecherche wird in diesem Kapitel (1) ein Überblick über den Stand der IS-Forschung im Kontext von IT-Komplexität gegeben, (2) eine Forschungsagenda für die weitere Arbeit abgeleitet und (3) eine systemtheoretische Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen eingeführt.

In Unterkapitel 3.1 werden zunächst die durchgeführte Literaturrecherche beschrieben und ausgewählte Arbeiten vorgestellt. Auf Grundlage aller als relevant identifizierten Quellen werden anschließend die in der Arbeit behandelten Forschungsfragen abgeleitet. Unterkapitel 3.2 widmet sich anschließend einer holistischen Konzeptualisierung von struktureller Komplexität von Unternehmens-/IT-Architekturen und damit der Beantwortung der ersten Forschungsfrage. Dabei wird in folgenden Schritten vorgegangen: Die im Zuge der Literaturrecherche identifizierte Literatur wird zunächst im Hinblick auf die Verwendung des Konzeptes „Komplexität“ analysiert. Auf Grundlage der hierbei gewonnenen Erkenntnisse wird anschließend eine systemtheoretische Konzeptualisierung von Komplexität in Unternehmens- bzw. IT-Architekturen vorgeschlagen. Das Kapitel schließt in Unterkapitel 3.3 mit einer Zusammenfassung der Ergebnisse der Literaturrecherche sowie einer Diskussion der theoretischen und praktischen Implikationen.

3.1 Strukturierte Literaturrecherche zu Komplexität in der IS-Forschung

Dieses Unterkapitel ist in drei Abschnitte untergliedert. Zunächst wird im folgenden Abschnitt das Vorgehen bei der strukturierten Literatursuche näher erläutert. Anschließend werden in Abschnitt 3.1.2 die identifizierten Arbeiten und Konzepte dargestellt. Darauf folgend werden in Abschnitt 3.1.3 die aus der Literaturrecherche resultierenden Forschungsfragen eingeführt.

3.1.1 Vorgehen bei der strukturierten Literaturrecherche

Die im Folgenden beschriebene strukturierte Literaturrecherche folgt dem von vom Brocke et al. (2009) vorgeschlagenen Vorgehen zur Erstellung strukturierter Literaturübersichten (vgl. hierzu Abbildung 8).

²⁰ Vgl. nachfolgend Schütz et al. (2013b).

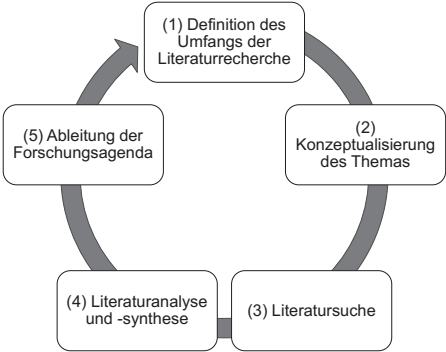


Abbildung 8: Strukturierte Literaturübersichten nach vom Brocke et al. (2009)

In einem ersten Schritt wird zunächst der Umfang der Literaturrecherche definiert. Hierzu wird die von Cooper (1988) eingeführte Taxonomie für Literaturrecherchen herangezogen (vgl. Abbildung 9), die bezüglich des Umfangs, der Form der Darstellung sowie der Zielsetzung sechs Charakteristiken vorsieht.

Charakteristik	Kategorien			
(1) Fokus	Forschungs- ergebnisse	Forschungs- methoden	Theorien	Anwendung
(2) Ziel	Integration		Kritik	Identifikation zentraler Themen
(3) Organisation	Historisch		Konzeptionell	Methodisch
(4) Perspektive	Neutrale Darstellung			Einnahme einer Position
(5) Zielgruppe	Spezialisierte Wissenschaftler	Wissenschaftler	Praktiker	Allgemeinheit
(6) Abdeckung	Vollständig	Vollständig mit selektiver Zitation	Repräsentativ	Zentral

Abbildung 9: Taxonomie einer Literaturübersicht nach Cooper (1988) – Darstellung in Anlehnung an vom Brocke et al. (2009)

Der Fokus der Literaturrecherche (Charakteristik 1) liegt nicht lediglich auf einer der angeführten vier Kategorien, sondern bezieht mit Forschungsergebnissen und den in der Literatur angeführten Anwendungsfeldern zwei Kategorien mit ein.²¹ Die beiden Kategorien der Forschungsmethoden und der Theorien stehen zwar nicht im Zentrum der Betrachtung, werden jedoch ebenfalls untersucht, um Forschungsansätze sowie die etwaige Einordnung des Themas der Komplexität in bestehende theoretische Konzepte nachvollziehen zu können. Die Zielsetzung (Charakteristik 2) der Literaturrecherche besteht sowohl in der Untersuchung des Standes der IS-Forschung im Kontext der IT-Komplexität als auch in der Erarbeitung einer Forschungsagenda für die weitere Arbeit. In Anlehnung an Webster & Watson (2002) erfolgt die Darstellung der Ergebnisse (Charakteristik 3) nach den im Rahmen der Recherche identi-

²¹ Gem. Cooper (1988) beziehen die meisten Literaturrecherchen mehr als einen Fokus mit ein, wobei diese durchaus auch eine unterschiedliche Gewichtung bei der Betrachtung erfahren können.

fizierten Themenbereichen und Konzepten. Zwar wird die Literaturrecherche mit einer systemtheoretischen Perspektive (Charakteristik 4) auf die strukturelle Komplexität von Unternehmens- bzw. IT-Architekturen und deren Subsystemen (vgl. hierzu Abschnitte 2.2.3 und 2.2.4) durchgeführt, eine spezifische Position in Sinne einer Haltung, die einen Einfluss auf die Darstellung der Ergebnisse hat, wird jedoch nicht eingenommen. Die Literaturrecherche richtet sich primär an spezialisierte Wissenschaftler aus den Feldern des Unternehmens-/IT-Architekturmanagements, des Managements von Informationssystemen sowie der Komplexitätsforschung. Aufgrund der interdisziplinären Ausrichtung der systemtheoretischen Perspektive (vgl. hierzu Abschnitte 2.1.2 und 2.1.3), sollen die Ergebnisse auch für Wissenschaftler anderer Forschungsfelder nachvollziehbar sein. Die durchgeführte Literaturrecherche konzentriert sich dabei auf die wesentlichen Publikationsorgane der IS-Forschung (Charakteristik 6). Näheres zur Auswahl der Quellen wird nachfolgend noch näher erläutert.

Nach der Definition des Umfanges der Recherche beginnt mit der Konzeptualisierung des Themas die eigentliche Durchführung der Literaturarbeit (vgl. Abbildung 8, Schritt 2). Hierzu wurde zunächst die grundlegende Literatur zu Unternehmens- und IT-Architekturen (vgl. Tabelle 2 und Tabelle 3) hinsichtlich relevanter Suchbegriffe untersucht und eine initiale Menge an Begriffen gebildet. In einem iterativen Prozess wurde diese Menge um synonyme Begriffe erweitert, die Begriffe in verschiedenen Kombinationen getestet und anschließend durch Hinzufügung logischer Ausdrücke zu einer Suchanfrage verknüpft. Um unterschiedliche Schreibweisen und Wortendungen bei der Suche zu berücksichtigen, wurde dem Begriff „complex“ zudem noch eine sogenannte „wildcard“ („*“) hinzugefügt. Die folgende Suchanfrage für die Literatur- und Volltextdatenbanken in EBSCOhost zeigt die für die Literaturrecherche verwendete Begriffskombination:

„AB complex* AND AB (IT OR system OR landscape OR architecture OR infrastructure OR hardware OR platform OR application OR software OR data)“

Um sicherzustellen, dass ausschließlich qualitativ hochwertige und wissenschaftlichen Ansprüchen genügende Quellen in die Literaturrecherche einfließen, wurde diese auf die Zeitschriften des AIS „Basket of Eight“, ergänzt um die Top 25 Zeitschriften des MIS (Management Information Systems) Journal Rankings (AIS 2012) beschränkt. Darüber hinaus wurden die Tagungsbände der IS-Konferenzen ECIS (European Conference on Information Systems) und ICIS (International Conference on Information Systems) in die Suche miteinbezogen. Für die Suche wurden die Literaturdatenbanken ACM Digital Library, AIS Electronic Library, Business Source Premier (über EBSCOhost), EconLit (über EBSCOhost), IEEE Xplore Digital Library, SciVerse Science Direct sowie Web of Science (Web of Knowledge) herangezogen.

Der Such- und Selektionsprozess (vgl. Abbildung 8, Schritt 3) wurde im Zeitraum von Juli bis Oktober 2012 durchgeführt. Unter Verwendung der angeführten Suchbegriffe wurden die Datenbanken durchsucht und insgesamt 3963 Quellen identifiziert, die sich aus 3664 Zeitschriften- sowie 299 Konferenzbeiträgen zusammensetzten. Im Anschluss wurden die Quellen näher untersucht, um diese auf die für die weiteren Betrachtungen relevanten Beiträge zu beschränken. Hierzu wurden jeweils der Titel sowie die Zusammenfassung durchgesehen und

dabei diejenigen Quellen entfernt, bei denen Komplexität keinen wesentlichen Fokus der Betrachtungen darstellt bzw. der Begriff „complexity“ nur als Füllwort Verwendung findet. Im nächsten Schritt wurden die Volltexte der verbleibenden Beiträge durchgesehen und hinsichtlich der folgenden Exklusionskriterien untersucht und bei Zutreffen entfernt:

- Komplexitätstheorie als Teilgebiet der theoretischen Informatik (z. B. Zuordnung von Problemen zu Komplexitätsklassen, die typischerweise mit Schrankenfunktionen in der O-Notation angegeben werden)
- Dynamische Aspekte der Software Komplexität (z. B. Erweiterungen von McCabe's „cyclomatic number“ (McCabe 1976))

Nach diesem Schritt verblieben 37 relevante Zeitschriftenartikel sowie 8 Konferenzbeiträge. Abbildung 10 zeigt Verteilung der identifizierten Beiträge auf die unterschiedlichen Publikationsorgane.

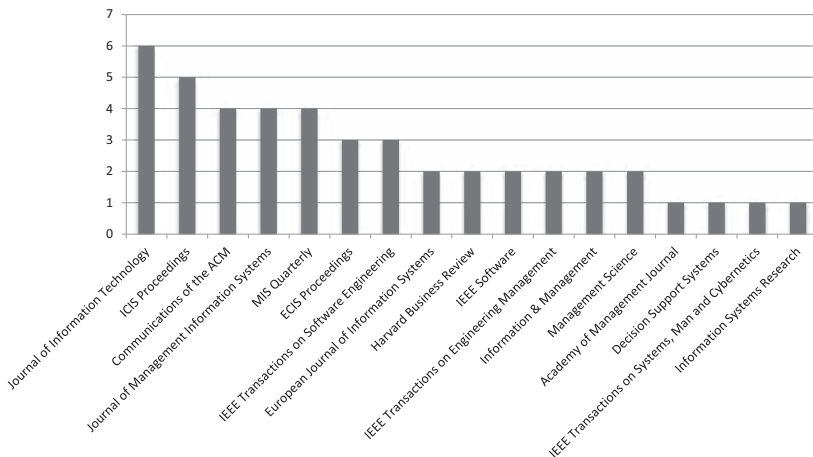


Abbildung 10: Verteilung der Publikationsorgane nach Häufigkeit

3.1.2 Literaturanalyse

Die im Zuge der zuvor dargestellten Literatursuche ermittelten Quellen wurden anschließend analysiert (vgl. Abbildung 8, Schritt 4), um die wesentlichen Schwerpunkte und Konzepte zu identifizieren. Hieraus ergaben sich die folgenden sechs Themenkomplexe, auf die in den nächsten Unterabschnitten anhand ausgewählter Quellen näher eingegangen wird:

- Schnittstelle zwischen Geschäfts- und IT-Architektur
- Komplexität der Datenarchitektur
- Komplexität der Anwendungsarchitektur
- Management komplexer Architekturen
- Informationssysteme als komplexe adaptive Systeme
- Wirkung von Komplexität auf die Eigenschaften von Systemen

3.1.2.1 Schnittstelle zwischen Geschäfts- und IT-Architektur

Der erste hier dargestellte Themenkomplex führt Arbeiten zusammen, die sich im weitesten Sinne mit dem Zusammenspiel von Geschäftsarchitektur und der darunterliegenden IT-Architektur beschäftigen. An dieser Stelle wurde eine weitere Untergliederung in Arbeiten, die sich mit Abhängigkeiten zwischen komplexen Geschäftsprozessen und der IT-Architektur, mit der Komplexität von IS-Projekten oder mit dem Architekten an der Schnittstelle zwischen Geschäfts- und IT-Architektur beschäftigen, vorgenommen.

3.1.2.1.1 Schnittstelle zwischen der Komplexität der Geschäftsprozesse und der IT-Architektur

In einer longitudinalen Fallstudie in einem norwegischen Krankenhaus untersuchen Hanseth et al. (2006) Schwierigkeiten des Einsatzes von Standardisierung bei dem Management von Informationssystemen. Die Autoren führen an, dass diese im Wesentlichen von der sozio-technischen Komplexität – in Form einer zunehmenden Verflechtung technologischer Standards mit lokalen, heterogenen und verteilten praktischen Vorgehensweisen – herrühren. Die diskutierten Schwierigkeiten gehen so weit, dass Standardisierungsmaßnahmen sogenannte reflexive Prozesse (eine Kombination selbstverstärkender und selbstzerstörerischer Prozesse) auslösen, die – verstärkt durch entstehende Seiteneffekte – die ursprüngliche Zielsetzung negativ beeinflussen können („concept of reflexivity“). Anstelle einer zunehmenden Integration und Kontrolle eines fragmentierten komplexen Systems, besteht demnach die Gefahr, dass sich dessen Komplexität weiter erhöht. Die Autoren unterstreichen damit, dass der klassische Fokus auf Standardisierung für ein Management der Komplexität nicht ausreicht.

Den Zusammenhang der IT-Vermögensportfolios mit den organisationalen Zielgrößen der Effizienz und der Innovation untersuchen Xue et al. (2012). Das in der Studie verwendete Model berücksichtigt außerdem den moderierenden Effekt des spezifischen industriellen Umfelds auf diesen Zusammenhang. Miteinbezogene Ausprägungen des industriellen Umfelds sind dabei die Dynamik sowie die Komplexität des Umfelds. Hierbei verwenden Xue et al. (2012) als Indikator für die Komplexität des Umfelds u. a. die Konzentration des Marktes. Je höher die Konzentration – also je stärker die Tendenz hin zu einem Monopol – desto geringer ist die Komplexität des industriellen Umfelds. Für die Messung der Konzentration ziehen die Autoren den Herfindahl-Index heran (vgl. Herfindahl 1950). Die Ergebnisse der Studie stützen die Hypothesen, dass eine geringere Dynamik und Komplexität des industriellen Umfelds den Zusammenhang zwischen getätigten Investitionen in IT-Vermögenswerte und der Effizienz des Unternehmens verstärken. Des Weiteren weisen die Ergebnisse darauf hin, dass eine höhere Komplexität des industriellen Umfelds den Zusammenhang zwischen getätigten Investitionen in IT-Vermögenswerte und Innovationen positiv beeinflusst. Die Ergebnisse unterstreichen, dass Komplexität nicht per se etwas Negatives ist, das stets reduziert werden muss. Vielmehr sollte die Erreichung eines möglichst optimalen Grades an Komplexität angestrebt werden.

3.1.2.1.2 Komplexität von IS-Projekten

Neu zu entwickelnde Lösungen müssen im Rahmen von IS-Projekten in die bestehenden IT-Landschaften der Unternehmen integriert werden. Dies bedeutet für IS-Projekte, dass sowohl neue Technologien als auch die aktuell vorhandene IT-Landschaft zu berücksichtigen sind. Hinzu kommt, dass sich ändernde Geschäftsstrategien und Geschäftsprozesse die Definition von Anforderungen für die IS-Projekte und demzufolge auch die Aufgabe, eine adäquate technische Umsetzung bereitzustellen, erschweren. IS-Projekte können insofern als komplexe Aufgaben bezeichnet werden, als dass sie sowohl (neue) Technologien als auch Geschäftsprozesse betreffen (Lee & Xia 2002; Xia & Lee 2005).

Lee und Xia entwickeln in ihren Arbeiten ein Konstrukt zur Quantifizierung der Komplexität von IS-Projekten (vgl. Lee & Xia 2002; Xia & Lee 2005). Hierfür schlagen sie ein zweidimensionales konzeptuelles Rahmenwerk vor, das die Beschaffenheit sowie den Fokus der Projekt-Komplexität umfasst. Die Beschaffenheit zeigt dabei mit „strukturell“ und „dynamisch“ zwei Ausprägungen auf. Strukturelle Komplexität verstehen Lee und Xia dabei als „[...] eine Funktion der Anzahl der Projekt-Komponenten sowie die Ausgestaltung und Stärke der Beziehungen zwischen den Komponenten.“ (Lee & Xia 2002, S. 81) In Xia & Lee (2005) wird zudem die Varietät der Projektelemente hinzugefügt. Hingegen verstehen Lee und Xia unter der dynamischen Komplexität die durch Änderungen im Geschäfts- sowie dem technologischen Umfeld (Lee & Xia 2002, S. 81) induzierte Komplexität und die hierdurch entstehende Unsicherheit (Xia & Lee 2005). Gleichermaßen unterteilen Lee und Xia den Fokus in zwei Ausprägungen: „IT“ und „Organisation“. Zu dem organisationalen Fokus zählen Lee und Xia u. a. die Organisationsstruktur und die Geschäftsprozesse des Unternehmens, aber auch die Managementunterstützung und die personelle Ausstattung der Projekte. Der IT-Fokus umfasst technologische Komponenten (Hard-/Software) sowie das technologische Know-how. Aus der Kombination dieser ergeben sich für das konzeptionelle Rahmenwerk vier unterschiedliche Dimensionen der Komplexität von Softwareentwicklungsprojekten: die strukturelle organisationale Komplexität, die strukturelle IT-Komplexität, die dynamische organisationale Komplexität sowie die dynamische IT-Komplexität.

3.1.2.1.3 Unterstützung der Architekten an der Schnittstelle zwischen Geschäfts- und IT-Architektur

Adomavicius et al. (2008) beschäftigen sich in ihrer Arbeit mit dem Problem, wie sich zukünftige technologische Entwicklungen auf den Wert bestehender Technologien auswirken. Die große Anzahl verfügbarer Technologien und der zwischen diesen bestehenden Beziehungen erschwert eine derartige Analyse. Adomavicius et al. (2008) gehen diese Problemstellung an, indem sie in ihrer Arbeit ein sogenanntes „IT ecosystem model“ vorschlagen. Zielsetzung der hierbei entwickelten Artefakte ist, eine Repräsentation für die Analyse von IT-Trends anzubieten sowie die Komplexität hinsichtlich der für eine IT-Investitionsentscheidung in Frage kommenden Technologien zu reduzieren (Adomavicius et al. 2008, S. 779). Diese Reduktion erfolgt durch Bildung von Subsystemen (den sogenannten IT Ecosystemen), denen Technologien zugeordnet werden, die miteinander in Verbindung stehen. Die Verwendung

der „process theory perspective“ erlaubt Adomavicius et al. (2008), Erklärungsansätze für die Evolution komplexer Systeme über die Zeit hinweg anzuführen und somit auch Prognosen für die künftige Entwicklung aktueller IT-Trends und deren Wirkung auf bestehende Technologien abgeben zu können.

Die Arbeit von Adomavicius et al. (2008) setzt den Fokus auf die für eine IT-Investition in Frage kommenden (neuen) Lösungen. Die von Adomavicius et al. (2008) vorgestellten Artefakte unterstützen die verantwortlichen Entscheider dabei, Prognosen über die künftige Entwicklung einer aktuellen Technologie treffen zu können. Hingegen geht die Arbeit nicht explizit auf die aktuelle IT-Architektur des Unternehmens ein. Neben der Überlegung, wie sich ein IT-Trend entwickeln wird, stellt sich auch die Frage, wie eine neue Technologie in die bestehende IT-Infrastruktur zu integrieren ist und welche Folgen eine solche Integration auf die zukünftige Beherrschbarkeit der IT-Architektur hat. Wie ein Artefakt ausgestaltet sein sollte, das den Entscheider an dieser Stelle unterstützt, wird durch die aktuelle wissenschaftliche Literatur nicht untersucht.

3.1.2.2 Komplexität der Datenarchitektur

Cappiello et al. (2003) beschäftigen sich mit Datenqualitätsproblemen im Zusammenhang mit einer verteilten Datenhaltung in Unternehmen. Unternehmen stellen ihre Services häufig über verschiedene Kanäle zur Verfügung. Die Umsetzung in der Datenarchitektur der Unternehmen bewegt sich dabei zwischen (a) vollständig integrierten Anwendungsarchitekturen mit einer zentralen Datenhaltung und (b) einer komplett dedizierten Datenhaltung je Funktionalität und Kanal. Eine ausbleibende Integration der Datenhaltung entlang der Kanäle bzw. der Funktionen können zu Datenqualitätsproblemen führen, welche die Qualität der angebotenen Services negativ beeinflussen. Cappiello et al. (2003) führen ein Modell zur Analyse von Architekturentscheidungen – kanal- vs. funktionsorientierte Integration – hinsichtlich deren Wirkung auf die Aktualität, Korrektheit und Vollständigkeit der Daten ein. Anhand einer Simulation auf Basis empirischer Daten zeigen die Autoren, dass keine der beiden Integrationsstrategien die andere stets dominiert.

3.1.2.3 Komplexität der Anwendungsarchitektur

Ein umfangreicher Strang der IS-Literatur widmet sich der Komplexität der Anwendungsarchitektur. Eine Vielzahl von Autoren beschäftigt sich dabei mit der Ermittlung von Metriken zur Quantifizierung der Anwendungscomplexität. Das hierbei betrachtete Grundproblem besteht darin, entscheiden zu können, wann ein (Anwendungs-) System komplexer ist als ein anderes. Weitere Artikel untersuchen die Auswirkungen einer zunehmenden Anwendungscomplexität.

3.1.2.3.1 Quantifizierung von Anwendungskomplexität

Ein in Theorie und Praxis weit verbreitetes Maß zur Quantifizierung von Anwendungskomplexität ist McCabes „cyclomatic number“. McCabe (1976) beschäftigt sich in seiner Arbeit mit der Frage, wie eine (modulare) Software ausgestaltet werden sollte, damit diese sowohl testbar als auch wartbar bleibt. Hierfür stellt McCabe mit der „cyclomatic complexity“ eine Software-Metrik vor, die auf graphentheoretischen Überlegungen beruht. Mit der sogenannten „cyclomatic number“ wird – durch die Darstellung des Programmablaufes mithilfe eines Kontrollflussgraphen – die Anzahl linear unabhängiger Pfade durch ein Programm ermittelt. Die „cyclomatic number“ stellt somit die obere Schranke für die zu betrachtenden Testfälle bzw. der im Zuge der Wartung eines Programms zu berücksichtigenden Programmabläufe dar.

Davis & LeBlanc (1988) untersuchen die Eignung von syntaxbasierten Code-Komplexitäts-Maßen, um Vorhersagen hinsichtlich der Zielgrößen der Erstellungszeit, der Anzahl an Fehlern oder dem Zeitaufwand zur Fehlerbehebung treffen zu können. Neben den klassischen Maßen aus dem Bereich der Softwareentwicklung, d. h. der Halstead-Metrik (vgl. Halstead 1977), McCabes „cyclomatic number“ sowie der Anzahl der Codezeilen, beziehen Davis & LeBlanc (1988) noch eine Reihe weiterer Maßzahlen mit ein, welche die im Programmcode erkennbaren logischen Einheiten (engl. „chunks“) berücksichtigen. Hier findet u. a. die Shannon-Entropie (vgl. Shannon 1948) zur Quantifizierung der Verteilung der logischen Einheiten Anwendung. Zwar kommen Davis & LeBlanc (1988) zu dem Ergebnis, dass keines der untersuchten Maße die anderen in jeder der o. a. zu prognostizierenden Zielgrößen dominiert, jedoch weist speziell die Shannon-Entropie – durch ihre Eigenschaft, die Konzentration einer zu definierenden Charakteristik zu messen – einen sehr vielseitigen Charakter auf.

Tegarden et al. (1995) stellen ein Model für die Messung der Anwendungskomplexität objektorientierter Systeme vor. Diese Form der Komplexität definieren Tegarden et al. (1995) auf Ebene von Variablen, Methoden, Objekten und auf Systemebene. Die vorgeschlagenen Metriken beschränken sich dabei auf diverse anzahlorientierte Maße, wie (auf Ebene der Variablen) die Anzahl der Objekte, in denen ein bestimmter Variablenname definiert ist, oder die Anzahl von Methoden, die auf eine Variable zugreifen.

Als Metrik für die Quantifizierung der Anwendungskomplexität hält Booch (2008) am klassischen Konzept der Anzahl der Codezeilen fest, da eine größere Anzahl an Codezeilen gemäß Booch häufig mit „trägen“, schwer änderbaren Anwendungen, einem relativ großen Team für die Pflege der Anwendung sowie einer größeren Anzahl an beteiligten Interessengruppen zusammenhängt. Zum anderen orientiert sich Booch (2008) bei den von ihm vorgeschlagenen Kennzahlen an der „4 + 1 model view“ nach Kruchten (1995): Für jede der Sichten (logische Sicht, prozessuale Sicht, Entwicklungssicht, physische Sicht und Anwendungsfälle) betrachtet Booch (2008) die Anzahl der jeweils relevanten Ausprägungen für Komponenten und der Beziehungen zwischen diesen.

3.1.2.3.2 Auswirkungen von Anwendungskomplexität

Banker & Slaughter (2000) untersuchen die Wirkung der Frequenz von Änderungen an den Funktionen einer Software sowie der „total data complexity“²² – als Indikator für die Komplexität des Softwaredesigns – auf die Kosten der Weiterentwicklung und die Anzahl der durch die Software verursachten Fehler. Banker & Slaughter (2000) zeigen u. a., dass die „total data complexity“ mit den durch Software verursachten Fehlern sowie den Kosten der Weiterentwicklung von Software in einem positiven Zusammenhang stehen. Hierzu weisen Banker & Slaughter (2000) zudem nach, dass der Grad der Strukturiertheit einer Anwendung (im Sinne eines modularen Aufbaus) den positiven Effekt der „total data complexity“ auf die durch die Software verursachten Fehlern abschwächt.

Banker et al. (1998) untersuchen anhand eines Samples von 23 Anwendungen eines Handelsunternehmens die Wirkung von Anwendungsentwicklungsmethoden (Code-Generatoren und Verwendung von Standardlösungen) auf die Anwendungskomplexität sowie deren Wirkung auf die Wartung und Weiterentwicklung der Anwendungen. Die Anwendungskomplexität wird in drei Dimensionen gemessen: (a) anhand der Menge an Daten relativ zu den im Code implementierten Operationen („component complexity“), (b) anhand der Anzahl der unterschiedlichen Kontrollflüsse in der Anwendung („coordinative complexity“) und (c) anhand der Anzahl an Statements, die den Kontrollfluss während der Laufzeit verändern können („dynamic complexity“). Banker et al. (1998) zeigen, dass die Softwareentwicklungsmethoden eine unterschiedliche Wirkung auf die betrachteten drei Dimensionen der Anwendungskomplexität haben. So kommen sie u. a. zu dem Ergebnis, dass Code-Generatoren in einer signifikanten inversen Beziehung zur „component complexity“ sowie zur „coordinative complexity“ stehen, hingegen aber ein positiver Zusammenhang zur „dynamic complexity“ besteht. Bei dem Einsatz von Standardlösungen wurden genau die entgegengesetzten Zusammenhänge beobachtet, wobei für den inversen Zusammenhang zur „dynamic complexity“ keine Signifikanz nachgewiesen werden konnte. Bezüglich der Anwendungskomplexität konnte ein signifikanter Zusammenhang mit dem Aufwand der Wartung und Weiterentwicklung der Anwendungen nachgewiesen werden.

Den Zusammenhang der strukturellen Komplexität des Programmcodes von Open Source Software mit der Wartbarkeit der Software untersucht Midha (2008) anhand von 450 Open Source Projekten. Eine der Arbeit zugrunde liegende Annahme ist, dass ein relativ simpel strukturiertes Programm einfacher vom Entwickler kognitiv erfasst und somit Wartungsarbeiten, wie z. B. die Behebung von Fehlern, schneller durchgeführt werden können. Nimmt die strukturelle Komplexität der Software durch fortwährende Änderungen und funktionale Erweiterungen hingegen zu, so muss der Entwickler mehr kognitiven Aufwand und Zeit investieren, um den bestehenden Code zu verstehen und einen Fehler zu beheben oder Neuerungen zu integrieren. Midha (2008) zeigt, dass eine höhere strukturelle Komplexität der Software – gemessen unter Verwendung von McCabe's „cyclomatic complexity“ sowie Halstead's „effort“ – (a) mit einer größeren Anzahl an Fehlern in der Software, (b) mit einer geringeren Anzahl

²² Banker & Slaughter (2000) messen die „total data complexity“ anhand der Anzahl von Datenelementen pro Anwendungsfunktionalität.

von Beiträgen am Code durch neue Entwickler und (c) mit einer längeren durchschnittlichen Dauer der Fehlerbehebung zusammenhängt. Hier kommt er zu dem Ergebnis, dass Projektadministratoren ein besonderes Augenmerk auf das Management der strukturellen Komplexität der Open Source Software legen sollten.

In einer Studie ähnlich zu der von Midha (2008) untersuchen Midha & Slaughter (2011) den Zusammenhang der strukturellen Komplexität von Open Source Software auf die Performanz von Wartungsarbeiten an der Software. In dieser Studie betrachten Midha & Slaughter (2011) als abhängige Variable jedoch lediglich die Zeit zur Behebung von Fehlern im Programmcode. Hingegen ergänzen sie das Modell um zwei Konstrukte, nämlich die „Verantwortlichkeit“ (accountability) für die Aufgabe und die „finanziellen Anreize“ (monetary incentives). Für diese wird nachgewiesen, dass sie die Wirkung der strukturellen Komplexität moderieren, indem sie den positiven Zusammenhang abschwächen, gleichzeitig aber auch eine direkte negative Wirkung auf die Zeit zur Behebung von Fehlern im Programmcode haben.

Für die vorliegende Arbeit sind in diesem Themenbereich insbesondere die Ergebnisse von Midha (2008) interessant, da sie nahelegen, dass ein solcher Zusammenhang struktureller Komplexität mit Zielgrößen wie Wartbarkeit nicht ausschließlich innerhalb einer Software, sondern auch in einem Netz aus verschiedenen Softwarelösungen – als wesentliche Elemente einer IT-Landschaft – besteht. Dieser Punkt wird in Abschnitt 3.1.3 wieder aufgegriffen.

3.1.2.4 Management komplexer Architekturen

Das Management der Komplexität von IT-Architekturen wird in der Literatur als eine der wesentlichen Aufgaben des IT-Managements angesehen, wobei auch hier häufig die Standardisierung thematisiert wird (Klompé & Pothuizen 1997).

Einen spezifischen Fokus auf die Standardisierung hinsichtlich der Hersteller der Anwendungen setzen Light et al. (2000). Sie diskutieren in ihrer Arbeit die „best of breed“ und die „single vendor“ Strategie. Anhand eines Fallbeispiels zeigen die Autoren die Vorzüge und Nachteile der beiden Strategien auf. Durch Anwendung des „best of breed“ Ansatzes, der auf der Integration von Standardsoftware einer Vielzahl von Anbietern sowie Eigenentwicklungen basiert, können die individuellen Anforderungen der einzelnen Geschäftsbereiche möglichst optimal unterstützt werden. Um dies zu realisieren, müssen jedoch zur Integration der Anwendungen Schnittstellen für den Datenaustausch implementiert und gepflegt werden. Über eine „single vendor“ Strategie wird hingegen eine, sowohl hinsichtlich der Software als auch der Hardware, einheitliche IT-Landschaft erreicht. Durch die Verwendung bestehender, standardisierter Schnittstellen sind aufwändige Implementierungsarbeiten zur Realisierung des Datenaustausches zwischen den Anwendungen nicht erforderlich. Auf der anderen Seite entsprechen die von einem Anbieter angebotenen Lösungen nicht zwingend den Anforderungen der einzelnen Geschäftsbereiche, was – sofern die Anpassbarkeit (Customizing) der Anwendungen es nicht zulässt – dazu führt, dass die Geschäftsprozesse an die Gegebenheiten der Lösung angepasst werden müssen. Demzufolge erlaubt die „best of breed“ Strategie eine größere Flexibilität hinsichtlich der Ausgestaltung der Geschäftsprozesse, allerdings eine geringe

gere Flexibilität im Hinblick auf Änderungen der bestehenden Umsetzung, da im Zweifel eine Vielzahl an speziell implementierten Schnittstellen anzupassen ist.

Schmidt & Buxmann (2011) untersuchen, inwiefern das Etablieren einer EAM-Funktion in Unternehmen zu einer höheren IT-Flexibilität und IT-Effizienz führt. Die im Zuge einer empirischen Untersuchung ermittelten Ergebnisse stützen die Hypothesen, dass Unternehmen, die ein umfassendes EAM etablieren, sowohl (a) einen höheren Grad an IT-Flexibilität als auch (b) einen höheren Grad an IT-Effizienz aufweisen. Die Definition des in der Arbeit verwendeten Konstrukts „EA Programming“ weist darauf hin, dass ein Architekturmanagement eng mit Standardisierungsstrategien verbunden ist.²³ Weiterhin wird in der Arbeit von einer Reduktion der IT-Komplexität gesprochen und diese mit einer zunehmenden Standardisierung in Verbindung gebracht.

Schneberger & McLean (2003) führen das Konzept der „computing environment complexity“ ein. Dieses stellt eine holistische Perspektive auf die System-Komplexität dar. Schneberger & McLean (2003) verstehen „computing environment complexity“ als die Anzahl (number) und Heterogenität (variety) der Komponenten (computing components) und der Beziehungen zwischen den Komponenten (component interactions) sowie der Änderungsrate des Systems. Zum anderen stellen die Autoren das „complexity cross“ vor, welches eine Abwägung zwischen der Komplexität der Komponenten eines Systems und der Komplexität des Systems an sich darstellt. Diese Abwägung betrachten Schneberger & McLean (2003) im Kontinuum zwischen Zentralität und Dezentralität des Systems. In einem zentralem System (z. B. ein zentraler Mainframe) ist die Komplexität der Komponenten hoch, die Komplexität des Systems jedoch eher niedrig. Die Einführung von PCs (die für sich genommen eine geringere technologische Komplexität aufzeigen als ein Mainframe) führt zu einer geringeren Komplexität der Komponenten, jedoch zu einer höheren Komplexität des Gesamtsystems. Schneberger & McLean (2003) führen an, dass an dieser Stelle ein Trade-Off zwischen eben genau diesen beiden Formen der Komplexität stattfindet. Als Strategien, die für diesen Trade-Off anzuwenden sind, nennen die Autoren u. a. die Reduktion der Komplexität der Komponenten durch den Einsatz einfacherer Technologien, die Reduktion der Heterogenität der System-Komponenten sowie die Reduktion der Abhängigkeiten zwischen den Komponenten.

3.1.2.5 Informationssysteme als komplexe adaptive Systeme

Merali (2006) argumentiert in ihrer Arbeit, dass die Komplexität und die Dynamik von Systemen durch die traditionellen Forschungsansätze nicht ausreichend berücksichtigt werden, da häufig zu stark vereinfachende Annahmen getroffen werden. Daher schlägt sie einen Paradigmenwechsel weg vom klassischen IS-Paradigma hin zu dem der Komplexitätsforschung und somit weg von einem mechanistischen hin zu einem organischen Ansatz vor, der sich durch Konzepte wie Emergenz und Selbstorganisation auszeichnet.

²³ Schmidt & Buxmann (2011) verwenden als Indikatoren für das Konstrukt „EA Programming“ Architekturprinzipien, Standard-Kataloge, Referenzmodelle für die Sicherstellung eines standardisierten Designs bei der Entwicklung individueller Anwendungen sowie unternehmensweit definierte Datenobjekte.

Allen & Varga (2006) diskutieren in ihrer Arbeit aus einer CAS-Perspektive das Problem der Kluft zwischen IT-Systemen und den Anforderungen, welche die Nutzer an IT-Systeme stellen. Die Autoren verstehen Unternehmen als CAS, die mit ihrer Umwelt und deren Mitarbeiter (die Agenten) untereinander interagieren. Der Komplexitätstheorie folgend ist ein Unternehmen das Ergebnis eines historischen Entwicklungspfades lokaler „Experimente“ der Agenten. Dabei findet eine Koevolution der Agenten statt, indem diese lernen und ihr Verhalten an ein sich änderndes Umfeld anpassen. Allen & Varga (2006) geben an, dass das Verständnis des Umfelds (die Ontologie) – also der Realität der Agenten – wesentlich durch die genutzten Informationssysteme geprägt ist. Gleichzeitig setzen die Agenten Informationssysteme ein, um den Entwicklungspfad durch Experimente fortzusetzen. Dies bedeutet, dass die Informationssysteme in der Lage sein müssen, die Agenten mit Informationen, die für deren Experimente notwendig sind, zu versorgen. Dies ist erforderlich, um die Koevolution der Agenten und des Unternehmens zu gewährleisten. Das IT-System als Teil des Informationssystems darf dieser Evolution nicht im Wege stehen, sondern muss sich ebenfalls an die sich ändernden Anforderungen der Agenten anpassen. Da sich IT-Systeme jedoch nicht agentenhaft mit den Anforderungen entwickeln, sondern an die sich ändernde Realität angepasst werden müssen, entsprechen die IT-Systeme zu keinem Zeitpunkt den aktuellen Anforderungen ihrer Umwelt. Allen und Varga schreiben hierzu:

„Because the IT system does not evolve, except by dint of upgrades, new releases, etc., that is, new variables being imagined by agents that are codified and well-defined, then we know IT systems do not represent reality. Reality is always ahead of IT systems.“
(Allen & Varga 2006, S. 234)

Die Kluft zwischen den durch die IT-Systeme zur Verfügung gestellten Informationen und dem tatsächlichen Informationsbedarf der Agenten stellt das betrachtete Problem dar. Die hier diskutierte Problemstellung unterstreicht die Relevanz der Aufgabe des Architekten, die IT-Systeme – d. h. die IT-Architektur – und deren Weiterentwicklung zu managen.

3.1.2.6 Wirkung von Komplexität auf die Eigenschaften von Systemen

Wie im vorangegangenen Unterabschnitt angeführt, gibt Merali (2006) als Zielgrößen des Managements der Komplexität der Technologie u. a. die Erreichung einer angemessenen Reaktionsfähigkeit, Robustheit und Flexibilität an. Den Begriff der Robustheit von Systemen betrachten Alderson & Doyle (2010) sehr differenziert. Alderson & Doyle (2010) diskutieren in ihrer Arbeit die Eigenschaften von komplexen Systemen. Grundidee ist dabei, dass Komplexität sich in biologischen oder technischen Systemen herausbildet, um dem System eine gewisse Robustheit gegenüber einer Menge an Störungen zu verleihen. Unter Robustheit kann beispielsweise die Zuverlässigkeit des Systems hinsichtlich des Ausfalles einzelner Komponenten, die Effizienz hinsichtlich der eingesetzten Ressourcen oder die Weiterentwickelbarkeit über die Zeit hinweg verstanden werden. Gleichzeitig kann die dabei entstehende Komplexität wiederum zu neuer Fragilität des Systems führen. Die Systeme weisen demzufolge ein Systemdesign auf, das robust und fragil zugleich ist. So kann ein System bei-

spielsweise effizient im Hinblick auf den Ressourceneinsatz sein, jedoch gleichzeitig eine geringe Zuverlässigkeit oder Flexibilität aufweisen.

Die Arbeit von Alderson & Doyle (2010) eröffnet eine interessante Perspektive auf Zielkonflikte technischer Systeme. Hierbei stellt sich die Frage, welche Ausprägungen von Robustheit und Fragilität eine komplexe IT-Architektur aufweist, bzw. welche Formen von Robustheit und Fragilität mit welcher Form der Komplexität der Architektur zusammenhängen.

3.1.3 Forschungsagenda

Auf Basis der durchgeführten Literaturanalyse wird im Folgenden die weitere Forschungsagenda der vorliegenden Arbeit abgeleitet (vgl. Abbildung 8, Schritt 5).

Obwohl die Beherrschbarkeit und das Management der Komplexität als eine wesentliche Aufgabe des Architekturmanagements erkannt wurden (vgl. Unterabschnitt 3.1.2.4 sowie Mattern et al. 2003; Ross et al. 2006; Schmidt & Buxmann 2011), existiert in der Literatur keine einheitliche Definition des Begriffs Komplexität. Die Literaturrecherche zeigt auf, dass Komplexität in allen Ebenen einer Unternehmensarchitektur eine Rolle spielt. So werden in der IS-Literatur die Komplexität des Marktumfelds (Xue et al. 2012), der Geschäftsprozesse (Hanseth et al. 2006), der IS-Projekte zur Umsetzung neuer Anforderungen (Lee & Xia 2002; Xia & Lee 2005), der Datenarchitektur (Cappiello et al. 2003), der Anwendungen (Banker et al. 1998; Banker & Slaughter 2000; Booch 2008; Davis & LeBlanc 1988; McCabe 1976; Midha 2008) aber auch das übergeordnete Themenfeld des Managements komplexer Architekturen (Light et al. 2000; Möller et al. 2011; Schmidt & Buxmann 2011) diskutiert. Es existiert folglich Wissen zu Komplexität in den unterschiedlichen Ebenen einer Unternehmensarchitektur, dieses basiert jedoch stets auf einem spezifischen Verständnis des Konzeptes „Komplexität“. Aus einer theoretischen Perspektive ist interessant, wie das Konzept „Komplexität“ in der bestehenden IS-Forschung verwendet wird und inwiefern das Verständnis von Komplexität über die einzelnen Beiträge hinweg verallgemeinert und zu einer holistischen Konzeptualisierung zusammengefasst werden kann. Das Potenzial einer solchen holistischen Konzeptualisierung liegt neben einem einheitlichen Verständnis dabei u. a. darin, dass eine Überprüfung der Übertragbarkeit von Forschungsergebnissen zwischen den einzelnen Architekturebenen erleichtert werden kann. Die erste Forschungsfrage lautet demnach:

Forschungsfrage 1: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen konzeptualisiert werden?

Sowohl die praktisch orientierte Literatur (Mattern et al. 2003; Ross et al. 2006) als auch wissenschaftliche Arbeiten (Möller et al. 2011; Richardson et al. 1990) bringen eine hohe Komplexität der IT-Architektur mit hohen Betriebskosten (geringer Effizienz), geringer Flexibilität sowie hohen IT-Risiken in Verbindung. Eine wissenschaftliche Untersuchung eines solchen Zusammenhanges erfolgte bislang nur vereinzelt und lediglich innerhalb der Architekturebenen. So zeigt u. a. die Arbeit von Midha (2008), dass ein Zusammenhang zwischen der strukturellen Komplexität von Anwendungen und z. B. der Anzahl an Fehlern oder der Wartbarkeit der Anwendung besteht. Diese Ergebnisse legen zugleich nahe, dass ein derartiger Zusam-

menhang struktureller Komplexität mit Zielgrößen wie Wart- oder Änderbarkeit nicht ausschließlich innerhalb einer Software existiert, sondern auch in einem Netz aus verschiedenen Softwarelösungen sowie der zugrunde liegenden Infrastruktur. So ist zu erwarten, dass Änderungen an der Struktur der IT-Landschaft, z. B. durch die Einführung einer Software von einem neuen Anbieter, auch einen Effekt auf Eigenschaften wie die Wart- und Änderbarkeit der IT-Landschaft haben. Hier führen zusätzliche Schnittstellen in der bestehenden Architektur zu neuen Kontrollflüssen (vgl. McCabe 1976) und auch zu einer Reihe zusätzlicher Abhängigkeiten zwischen den Komponenten der IT-Landschaft, die bei weiteren Änderungen verstanden und berücksichtigt werden müssen. Im Rahmen der Literatursuche wurde jedoch keine wissenschaftliche Arbeit identifiziert, die explizit die Auswirkungen struktureller Komplexität der IT-Architektur untersucht. Auch liefert die wissenschaftliche Literatur bislang keine Erkenntnisse, was die Entstehung von Komplexität verursacht und welche Mechanismen das Ausmaß einer entstehenden Komplexität beeinflussen. Hieraus ergeben sich die nächsten beiden Forschungsfragen:

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

Die Messbarkeit der strukturellen Komplexität ist Voraussetzung dafür, einen bestimmten Grad an struktureller Komplexität der IT-Architektur beurteilen und Handlungsbedarf ableiten zu können. In der IS-Literatur wurde bislang noch kein holistischer – auf alle Architekturebenen anwendbarer – Ansatz zur Quantifizierung struktureller Komplexität vorgeschlagen. Allerdings existieren diverse Ansätze in anderen (z. B. der Biologie oder der Volkswirtschaftslehre) oder angrenzenden (wie dem Softwareengineering) Forschungsfeldern, deren Übertragbarkeit auf den Kontext der Unternehmens- und IT-Architekturen untersucht werden kann. Die vierte Forschungsfrage lautet demzufolge:

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen quantifiziert werden?

Die zunehmende Durchdringung der Unternehmen und ihrer Geschäftsprozesse mit Informationssystemen führt zu einer stetig anwachsenden Abhängigkeit der Unternehmen von einer funktionsfähigen IT. Neu aufkommende Geschäftsideen führen häufig zu neuen Anforderungen und somit zu Änderungen, die in die bestehende IT-Landschaft integriert werden müssen. Die Arbeiten von Xia und Lee betonen, dass bei einer Weiterentwicklung der IT-Landschaft stets sowohl die neuen Technologien auf dem Markt als auch die aktuell im Unternehmen vorhandene IT-Landschaft berücksichtigt werden müssen (Lee & Xia 2002; Xia & Lee 2005). Demnach stellen sich für Unternehmen die folgenden beiden Fragen: (1) Welche existierenden Technologien sind für eine IT-Investition sinnvoll und (2) wie sollte eine neue Anforderung angesichts der aktuell vorliegenden Ausgestaltung der IT-Landschaft umgesetzt werden?

Mit der ersten Fragestellung beschäftigt sich die Arbeit von Adomavicius et al. (2008), die den Fokus auf die für eine IT-Investition in Frage kommenden (neuen) Lösungen setzt. In ihrer Arbeit werden Artefakte abgeleitet, welche die verantwortlichen Entscheider (IT-

Manager oder IT-Architekten) dabei unterstützen, Prognosen über die künftige Entwicklung einer Technologie zu treffen. Zu der zweiten Fragestellung, wie eine Anforderung angesichts der aktuell vorliegenden Ausgestaltung der IT-Landschaft integriert werden sollte, liefert die aktuelle Literatur hingegen keine Lösungsansätze. Dabei spielt die Form der Umsetzung eine besonders große Rolle, da diese wiederum die Basis für weitere an der IT-Landschaft vorgenommene Änderungen darstellt und somit einen unmittelbaren Einfluss auf die langfristige Beherrschbarkeit der IT-Architektur hat.

Die Arbeit von Allen & Varga (2006) unterstreicht dieses Problem aus der Perspektive komplexer adaptiver Systeme wie folgt: IT-Systeme stellen ein wesentliches Arbeitsmittel für die Mitarbeiter in einem Unternehmen dar, da es diese mit den für deren Arbeit nötigen Informationen versorgt. Der Informationsbedarf der Mitarbeiter ist jedoch nicht statisch, sondern ändert sich ständig, wodurch neue Anforderungen an die IT-Systeme entstehen. Die IT-Systeme entwickeln sich jedoch nicht agentenhaft (vgl. Abschnitt 2.1.3) mit den Anforderungen weiter, sondern sie werden durch menschliche Agenten (wie dem IT-Architekten) an die sich ändernde Realität der Mitarbeiter angepasst. Die von Allen & Varga (2006) diskutierte Problemstellung unterstreicht die Relevanz der Aufgabe des Architekten, die IT-Systeme – d. h. die IT-Architektur – und deren Weiterentwicklung zu managen, um eine Änder- und Beherrschbarkeit der IT-Systeme sicherzustellen. Die aktuelle wissenschaftliche Literatur bietet keine Antwort darauf, wie ein Artefakt ausgestaltet sein sollte, das den Entscheider an dieser Stelle unterstützt. Hieraus folgt die fünfte Forschungsfrage:

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

3.2 Systemtheoretische Konzeptualisierung von Komplexität in Unternehmensarchitekturen²⁴

Die in Unterkapitel 3.1 vorgestellte Literaturrecherche hat gezeigt, dass Komplexität in den unterschiedlichen Ebenen einer Unternehmens-/IT-Architektur eine Rolle spielt. Die identifizierte Forschung basiert jedoch nicht auf einem einheitlichen Verständnis des Konzeptes „Komplexität“. Obwohl die betrachteten Problemstellungen aus einer systemtheoretischen Perspektive grundsätzlich einen vergleichbaren Charakter aufweisen, nehmen die Arbeiten meist eine spezifische Perspektive ein. In diesem Unterkapitel wird die systemtheoretische Perspektive genutzt, um von diesen spezifischen Perspektiven zu abstrahieren und eine holistische Konzeptualisierung struktureller Komplexität einzuführen. Damit wird die erste Forschungsfrage der vorliegenden Dissertation beantwortet:

Forschungsfrage 1: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen konzeptualisiert werden?

Im nächsten Abschnitt wird hierzu zunächst die systemtheoretische Perspektive auf eine Unternehmens-/IT-Architektur sowie eine systemtheoretische Definition von Komplexität einge-

²⁴ Vgl. nachfolgend auch (Schütz et al. 2013b).

führt. In Anschluss daran werden in Abschnitt 3.2.2 die im Zuge der strukturierten Literaturrecherche zu Komplexität in der IS-Forschung (Unterkapitel 3.1) identifizierten Quellen hinsichtlich der Nutzung des Konzeptes „Komplexität“ untersucht. In Abschnitt 3.2.3 wird schließlich eine Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen vorgeschlagen.

3.2.1 Unternehmens-/IT-Architektur als System

Die vorliegende Dissertation folgt einem holistischen Verständnis einer Unternehmensarchitektur²⁵ und berücksichtigt alle vier in Abschnitt 2.2.3 diskutierten Architekturebenen.²⁶ In Anlehnung an den ISO/IEC/IEEE (2011) wird eine Architektur aus einer systemtheoretischen Perspektive als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht, verstanden (vgl. hierzu auch Abschnitt 2.2.2). Formal kann ein System wie folgt ausgedrückt werden (Klir 2001):

$$(1) \quad S = (T, R)$$

S repräsentiert hierbei das betrachtete System, T die Menge der Komponenten („things“) aus denen das System besteht und R die Menge an Beziehungen („relations“) zwischen den Komponenten. Als Beispiel kann hier eine Menge von Anwendungen (T) einer Anwendungslandschaft sowie die Menge an Schnittstellen (R) zwischen den Anwendungen angeführt werden.

Die Betrachtung der Unternehmensarchitektur als ein System erlaubt es, diese in vier Subsysteme zu unterteilen: Geschäftsprozess-, Daten-, Anwendungs- sowie Infrastrukturarchitektur. Somit kann die Komplexität jedes der Subsysteme bzw. jeder Architekturebene separat betrachtet werden. Um hingegen die Komplexität der gesamten Unternehmensarchitektur zu untersuchen, müssen die Beziehungen zwischen den Subsystemen hinzugezogen werden (vgl. Abbildung 6). Die Beziehung zwischen der Geschäftsprozess- und der Anwendungsarchitektur resultiert aus der Unterstützung der Geschäftsprozesse durch fachliche Anwendungen. Funktionale Anforderungen führen häufig zu spezifischen Datenmodellen und somit zu einer Beziehung zwischen Geschäftsprozess- und Datenarchitektur. Beispiele für Beziehungen zwischen der Anwendungs- und Infrastrukturarchitektur sind u. a. aus den Anwendungen resultierende Spezifikationen für die Infrastruktur, wie z. B. ein spezifisches Betriebssystem.

Bei der Betrachtung struktureller Komplexität stellt die Literatur zur Systemforschung eine Reihe sich ergänzender Charakteristiken heraus. In Summe wird strukturelle Komplexität als ein Systemzustand verstanden, der aus der Anzahl der Komponenten (Flood & Carson 1993; Klir 2001) und Beziehungen (Flood & Carson 1993) sowie der Heterogenität der Komponenten und Beziehungen (Simon 1962) resultiert.

²⁵ Da die IT-Architektur eine Teilmenge der Unternehmensarchitektur darstellt (vgl. Unterkapitel 2.2.4, Abbildung 6) und letztere somit alle Ebenen einer IT-Architektur inkludiert, wird im weiteren Verlauf von Unterkapitel 3.2 auf die explizite Erwähnung der IT-Architektur verzichtet. Es soll aber betont werden, dass die hier dargestellten Ergebnisse auf eine – die Geschäftsprozessarchitektur ausblende – Perspektive auf die IT-Architektur vollständig übertragbar sind.

²⁶ Geschäftsprozess-, Daten-, Anwendungs- und Infrastrukturarchitektur

Auf dieser Basis wird strukturelle Komplexität (C) definiert als Tupel aus der Anzahl (N) und Heterogenität (H) der Komponenten und Beziehungen einer Unternehmensarchitektur:

$$(2) \quad C = (N, H)$$

Diese Definition ist aufgrund der systemtheoretischen Perspektive auf die Unternehmensarchitektur auf alle Subsysteme bzw. Architekturebenen und auch auf die Beziehungen zwischen diesen anwendbar. Im nächsten Abschnitt wird untersucht, inwiefern diese Definition durch die IS-Literatur bestätigt wird.

3.2.2 Verwendung des Konzeptes „Komplexität“ in der IS-Literatur

Die im Rahmen der Literatursuche in Abschnitt 3.1.1 identifizierten Quellen wurden inhaltlich nochmals mit einem spezifischen Fokus auf die in den Arbeiten angeführten Definitionen bzw. die Verwendung des Begriffs Komplexität analysiert. Den Empfehlungen von Webster & Watson (2002) folgend wurde eine Konzeptmatrix herangezogen, um die Arbeiten zu kategorisieren. Als Konzepte wurden die oben angeführten Elemente der systemtheoretischen Definition struktureller Komplexität verwendet. Während der Auswertung der Quellen zeigte sich, dass einige Arbeiten bei der Verwendung des Begriffs Komplexität eine zeitliche Komponente miteinbeziehen. Daraufhin wurde die Konzeptmatrix um die Spalte „Änderung über die Zeit“ erweitert. Das Ergebnis der Analyse ist der Konzeptmatrix in Tabelle 5 zu entnehmen.

Die Analyse der Quellen zeigte, dass nur eine geringe Anzahl von Arbeiten die vier Elemente der systemtheoretischen Definition struktureller Komplexität **explizit** anführen. So definieren beispielsweise Banker & Slaughter (2000) für die Ebene der Datenarchitektur die Komplexität von Daten als “[...] *the number of data elements per unit of functionality*” (Banker & Slaughter 2000, S. 221). Die Komplexität der Anwendungsarchitektur verbinden Davis & LeBlanc (1988) mit “[...] *number of its components or elements, kind or type of elements [and] structure of the relationship between elements*” (Davis & LeBlanc 1988, S. 1366).

Hervorzuhebende Ausnahmen hiervon stellen hingegen die Arbeiten von Schneberger & McLean (2003) und Henningsson & Hanseth (2011) dar, welche die vier Elemente der Definition struktureller Komplexität explizit anführen:

“Complexity can be defined here as the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions” (Henningsson & Hanseth 2011, S. 2).

Den Begriff der „Computing complexity“ definieren Schneberger & McLean (2003) als die Anzahl (number) und die Heterogenität (variety) der Komponenten (computing components) und der Beziehungen zwischen den Komponenten (component interactions) sowie die Änderungsrate des Systems.

Tabelle 5: Konzeptmatrix – Konzeptualisierung von Komplexität

Quelle	Anzahl der Komponenten	Anzahl der Beziehungen	Heterog. der Komponenten	Heterog. der Beziehungen	Änderung über die Zeit
Adomavicius et al. (2008)	X	X	X	X	
Alderson & Doyle (2010)	X	X	X	X	
Allen & Varga (2006)	X	X	X	X	X
Amaral & Uzzi (2007)	X	X	X	X	
Banker & Slaughter (2000)	X				
Banker et al. (1998)	X	X			X
Belardo & Pazer (1985)	X		X		
Boisot (2006)	X				
Booch (2008)	X	X			X
Cappiello et al. (2003)	X		X		
Capra & Merlo (2009)	X				
Davis & LeBlanc (1988)	X	X	X		
Francalanci & Piuri (1999)	X				
Hanseth et al. (2006)	X	X	X	X	X
Henningsson & Hanseth (2011)	X	X	X	X	
Hsieh & Wang (2007)	X		X		
Jiang & Benbasat (2007)	X		X		
Kearney et al. (1986)					
Khoshgoftaar et al. (1992)	X	X	X		X
Kim & Kaplan (2006)	X	X	X	X	
Klompé & Pothuizen (1997)	X		X	X	
Lee & Xia (2002)	X	X	X		X
Light et al. (2000)	X		X		
Long & Denning (1995)	X	X			X
Marshall & Byrd (1998)	X		X		
McCabe (1976)		X			
Merali & McKelvey (2006)	X	X	X	X	
Merali (2006)	X	X	X	X	
Midha & Slaughter (2011)	X	X	X	X	
Midha (2008)	X	X	X	X	
Möller et al. (2011)	X		X		
Müller et al. (2012)	X	X	X	X	X
Roberts et al. (2004)	X	X	X		
Sangwan et al. (2008)	X	X	X		
Sargut & McGrath (2011)	X	X	X	X	
Schmidt & Buxmann (2011)	X	X	X	X	
Schneberger & McLean (2003)	X	X	X	X	X
Shafiei-Monfared & Jenab (2012)	X	X			
Sherif & Xing (2006)	X	X	X	X	
Singh (1997)	X	X			
Sommerville et al. (2012)	X	X		X	
Sullivan (2011)	X	X	X	X	
Tegarden et al. (1995)	X	X			X
Xia & Lee (2005)	X	X	X	X	
Xue et al. (2012)	X		X		
45	43	31	32	20	10

Eine Reihe von Arbeiten führen die vier Elemente zwar nicht explizit an, berücksichtigen diese jedoch **implizit**. Dazu zählen u. a. Arbeiten, welche die Theorie komplexer adaptiver Systeme verwenden. Komplexität ist hier keine Eigenschaft des Systems, sondern vielmehr ein grundlegender Gegenstand der Theorie. CAS betrachtet heterogene, interagierende Agenten. Die Anzahl der interagierenden Agenten spielt demnach auch eine Rolle für die Komplexität des Systems. Weiterhin stellen Beziehungen einen grundlegenden Aspekt von CAS dar, wobei sowohl die Anzahl der Beziehungen zwischen den Agenten als auch die Heterogenität dieser Beziehungen Berücksichtigung finden (Allen & Varga 2006).

Der Arbeit von Kearney et al. (1986) konnte keines der Konzepte zugeordnet werden. Kearney et al. (1986) verwenden den Begriff Komplexität in ihrer Arbeit als die Zeitdauer oder den Speicherbedarf bei der Ausführung eines Programmes bzw. als die Schwierigkeit für einen Softwareentwickler, eine Anwendung zu modifizieren, zu testen oder Fehler zu beseitigen. Letztere sind nach dem Verständnis der vorliegenden Arbeit eher als eine potenzielle Folge der Komplexität einer Anwendung zu verstehen.

Wie in Tabelle 5 ersichtlich, werden die Elemente der systemtheoretischen Definition struktureller Komplexität (die Anzahl der Komponenten, die Anzahl der Beziehungen, die Heterogenität der Komponenten und die Heterogenität der Beziehungen) weitestgehend durch die identifizierten Arbeiten abgedeckt. Da eine „Änderung über die Zeit“ in weniger als 25% der Arbeiten Erwähnung fand, geht dieser Aspekt nicht in den folgenden Vorschlag einer Konzeptualisierung von Komplexität ein.

3.2.3 Vorschlag einer Konzeptualisierung von Komplexität in Unternehmensarchitekturen

Basierend auf den vorausgegangenen Betrachtungen wird in diesem Abschnitt eine systemtheoretische Konzeptualisierung struktureller Komplexität in Unternehmensarchitekturen vorgeschlagen. Hierzu wird das Konzept miteinander in Verbindung stehender Subsysteme einer Unternehmensarchitektur (vgl. Abschnitt 2.2.3) mit der in Abschnitt 3.2.1 angeführten Definition struktureller Komplexität kombiniert. Aus einer systemtheoretischen Perspektive kann eine Unternehmensarchitektur ausgedrückt werden als:

$$(3) \quad S_{EA} = (T_{EA}, R_{EA})$$

T_{EA} stellt die Komponenten des Systems S_{EA} dar und R_{EA} die Beziehungen zwischen den Komponenten T_{EA} . Analog können die Subsysteme Geschäftsprozessarchitektur (GPA), Datenarchitektur (DA), Anwendungsarchitektur (AA) und Infrastrukturarchitektur (IA) wie folgt formuliert werden:

$$(4) \quad S_X = (T_X, R_X), \text{ mit } X \in \{AA, DA, GPA, IA\}$$

Wie in Abschnitt 3.2.1 eingeführt, wird strukturelle Komplexität (C) einer Unternehmensarchitektur definiert als Tupel aus der Anzahl (N) und Heterogenität (H) der Komponenten und Beziehungen einer Unternehmensarchitektur. Die Konzeptualisierung struktureller Komplexität eines Systems muss demnach sowohl die Anzahl N und die Heterogenität H der Komponenten T und der Beziehungen R berücksichtigen. Dies kann wie folgt dargestellt werden:

$$(5) \quad C_Y = (N_Y, H_Y), \text{ mit } Y \in \{T, R\}$$

In Abbildung 11 werden die einzelnen Aspekte der Konzeptualisierung nochmals zusammengefasst.

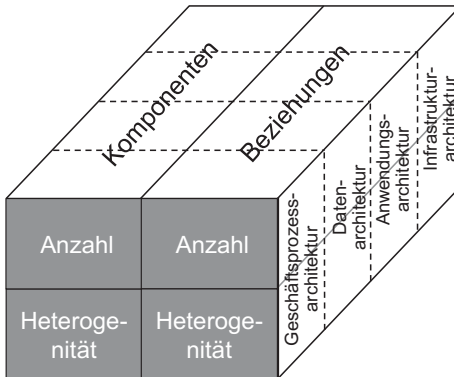


Abbildung 11: Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen ²⁷

Die hier vorgestellte Konzeptualisierung kann auf verschiedene Perspektiven angewendet werden. Sie ermöglicht es, den Fokus sowohl auf die Unternehmensarchitektur als Ganzes, auf die IT-Architektur als Teilmenge der Unternehmensarchitektur oder auf einzelne Subsysteme, wie z. B. die Anwendungsarchitektur, zu legen.

3.3 Zusammenfassung und Diskussion der Ergebnisse

Auf Grundlage einer strukturierten Literaturrecherche wurde in diesem Kapitel zunächst der Stand der Forschung zu IT-Komplexität in der IS-Literatur dargestellt. Basierend auf der diskutierten Literatur wurde anschließend eine Forschungsagenda für die weitere Arbeit abgeleitet. Die fünf Forschungsfragen lauten:

Forschungsfrage 1: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen konzeptualisiert werden?

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen quantifiziert werden?

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

²⁷ Vgl. hierzu Schütz et al. (2013a) und Schütz et al. (2013b).

Die erste Forschungsfrage wurde durch eine Untersuchung der zuvor in Unterkapitel 3.1 identifizierten Quellen bearbeitet. Durch eine Auswertung der IS-Literatur hinsichtlich der Verwendung des Begriffs Komplexität wurde in Unterkapitel 3.2 eine systemtheoretische Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen vorgeschlagen. In den folgenden beiden Abschnitten werden die theoretischen und praktischen Implikationen der in diesem Kapitel durchgeführten Aufbereitung der wissenschaftlichen Literatur dargestellt.

3.3.1 Theoretische Implikationen

Es existiert in der Literatur bislang keine holistische, auf alle Ebenen einer Unternehmens-/IT-Architektur anwendbare, Konzeptualisierung struktureller Komplexität. Dies führt dazu, dass der Begriff sehr uneinheitlich Verwendung findet, obwohl die Art des betrachteten Problems – die strukturelle Komplexität – in den vier Ebenen einen sehr ähnlichen Charakter aufweist. Dies bestätigte auch die durchgeführte strukturierte Literaturrecherche, auf deren Grundlage eine systemtheoretische Konzeptualisierung struktureller Komplexität vorgeschlagen wurde. Die Verwendung einer einheitlichen Konzeptualisierung in weiteren wissenschaftlichen Arbeiten kann das Verständnis der Problemklasse des Managements struktureller Komplexität erleichtern. Zudem eröffnet die Verwendung einer, auf alle vier Architekturebenen anwendbaren Konzeptualisierung die Möglichkeit, die Ergebnisse auf verschiedene Architekturebenen zu übertragen.

3.3.2 Praktische Implikationen

Eine ganzheitliche Perspektive, die alle Ebenen der Architektur beinhaltet, ist aus praktischer Sicht insofern wichtig, als dass Entscheidungen auf einer Ebene nur schwer losgelöst von der restlichen Architektur getroffen werden können. So sind z. B. die vorhandenen Infrastruktur-Komponenten in der Regel das Ergebnis einer Entwicklung des Geschäftsfeldes über die Zeit hinweg. Die Anpassung von existierenden oder die Einführung neuer Geschäftsprozesse kann die Einführung neuer Anwendungen bedingen, die wiederum bestimmte Infrastruktur-Komponenten (Betriebssysteme, Datenbanken, Hardware-Komponenten) benötigen. So können neue Geschäftsideen oder zusätzliche regulatorische Anforderungen zu Änderungen an der gesamten IT-Landschaft führen, und das über alle Architekturebenen hinweg. Aus praktischer Perspektive ist es daher wichtig, ein klares Verständnis bzgl. des Begriffs der Komplexität zu etablieren, um auch über die Grenzen einzelner organisatorischer Bereiche hinweg eindeutig kommunizieren zu können. Die hier eingeführte Konzeptualisierung struktureller Komplexität der Architektur kann in Unternehmen als einheitliches Begriffsverständnis dienen.

4 Qualitative Untersuchung der Ursachen und Auswirkungen von Komplexität in IT-Architekturen

In diesem Kapitel werden im Zuge einer qualitativen Studie die Ursachen und die Auswirkungen struktureller Komplexität in IT-Architekturen näher untersucht. Das Kapitel ist wie folgt aufgebaut: Im folgenden Unterkapitel 4.1 wird zunächst eine Motivation für die in diesem Kapitel behandelten Forschungsfragen gegeben. Anschließend werden in Unterkapitel 4.2 die für die Studie benötigten theoretischen Grundlagen sowie die zugrunde liegenden Annahmen eingeführt. In Unterkapitel 4.3 wird das methodische Vorgehen erläutert, bevor in Unterkapitel 4.4 die Ergebnisse der qualitativen Untersuchung vorgestellt werden. Unterkapitel 4.5 beschließt dieses Kapitel mit einer Zusammenfassung und Diskussion der Ergebnisse.

4.1 Motivation

Gerade aus Perspektive des IT-Managements ist es besonders wichtig, das Phänomen der Komplexität besser zu verstehen. So ist es Aufgabe der unternehmerischen Praxis, IT-Strategien zu definieren, um eine langfristige, die Unternehmensziele unterstützende Ausrichtung der Unternehmens-IT festzulegen (Ross et al. 2006). Ein fehlendes Verständnis der Ursachen und Folgen einer zunehmenden Komplexität verhindert, den Aspekt der Komplexität sowie der Beherrschbarkeit der IT in einer IT-Strategie adäquat berücksichtigen zu können. Jedoch nicht nur bei der Entwicklung einer IT-Strategie kann eine höhere Transparenz der Ursachen struktureller Komplexität und der Mechanismen, die zur Entstehung dieser beitragen, das IT-Management unterstützen. Ein besseres Verständnis des Phänomens der Komplexität kann auch zur Lösung von Aufgabenstellungen, wie der Ausgestaltung der Architekturmanagement-Prozesse oder der Identifikation und der Einbeziehung relevanter Schnittstellen zu anderen organisationalen Prozessen (z. B. dem Anforderungs- oder dem Änderungsmanagement), einen Beitrag leisten.

Wie bereits in Abschnitt 3.1.3 angeführt, konnte durch die Literatursuche keine wissenschaftliche Arbeit identifiziert werden, die sich explizit mit den Ursachen struktureller Komplexität von IT-Architekturen beschäftigt. Auch eine Untersuchung der Mechanismen, wie komplexe IT-Architekturen entstehen und welche Folgen eine Veränderung der Komplexität der IT-Architektur auf Eigenschaften eines Systems (Alderson & Doyle 2010) (wie z. B. IT-Effizienz und IT-Flexibilität) hat, blieb nach dem Kenntnisstand des Autors in der IS-Literatur bislang aus. Die in diesem Kapitel dargestellte Untersuchung soll die bestehende Literatur dahingehend komplementieren und zielt auf die Beantwortung der zweiten und dritten Forschungsfrage der vorliegenden Dissertation ab:

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

4.2 Theoretische Grundlagen und zugrunde liegende Annahmen

In diesem Unterkapitel werden zunächst in den Abschnitten 4.2.1 und 4.2.2 die für diese Untersuchung relevanten theoretischen Grundlagen sowie die zugrunde liegenden Annahmen eingeführt. Darauf aufbauend werden in Abschnitt 4.2.3 die in diesem Kapitel bearbeiteten Forschungsfragen durch eine Reihe von Untersuchungsfragen präzisiert.

4.2.1 IT-Architekturen als künstliche Systeme

In Anlehnung an den ISO/IEC/IEEE (2011) und Simon (1996) sowie der in Abschnitt 2.2.2 eingeführten Definition folgend, werden IT-Architekturen in dieser Arbeit als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht, verstanden. Wie in Abschnitt 3.2.3 dargestellt, wird die Komplexität der IT-Architektur als die Anzahl und Heterogenität der Komponenten und Beziehungen der IT-Architektur definiert. Bei der Komplexität handelt sich folglich um eine Eigenschaft der Systemstruktur zu einem bestimmten Zeitpunkt. Gleichzeitig zeichnet sich die IT-Architektur durch weitere zeitpunktbezogene Eigenschaften aus, z. B. wie schnell und einfach sie an sich ändernde Anforderungen angepasst werden kann (IT-Flexibilität, vgl. 2.2.5.2.1) oder inwiefern bestehende Anforderungen mit möglichst geringem Ressourceneinsatz erfüllt werden (IT-Effizienz, vgl. 2.2.5.2.2).

Es wird angenommen, dass es ausschließlich durch Eingriffe von außen zu einer Veränderung der IT-Architektur und damit auch ihrer Eigenschaften kommt (vgl. Abbildung 12 für eine schematische Darstellung der Veränderung einer IT-Architektur).

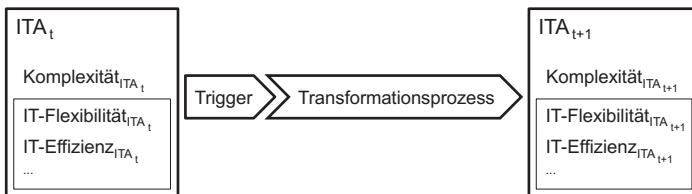


Abbildung 12: Schema der Veränderung der IT-Architektur

Eine Änderung wird demnach durch bestimmte Trigger ausgelöst (vgl. Abbildung 12). Im darauf folgenden Transformationsprozess findet eine aktive und zielgerichtete Veränderung der IT-Architektur statt.²⁸ Die aus diesem Transformationsprozess resultierende IT-Archi-

²⁸ Vgl. hierzu auch das Konzept der Transformation einer As-Is-Architektur (Ausgangsarchitektur) hin zu einer To-Be-Architektur (Zielarchitektur) (Saat et al. 2009).

tektur weist wiederum zeitpunktbezogene Ausprägungen der Komplexität sowie der weiteren Eigenschaften der IT-Architektur auf (wie IT-Flexibilität und IT-Effizienz).

4.2.2 Zusammenhang zwischen Geschäftsprozess- und IT-Architektur

Wie in Abschnitt 2.2.4 erläutert, ist die IT-Architektur ein Teil der Unternehmensarchitektur. Die Geschäftsprozessarchitektur gibt dabei ein Mindestmaß zu implementierender fachlicher Logik vor, das in der IT-Architektur abzubilden ist. Demzufolge gibt die Komplexität der Geschäftsprozessarchitektur (K_{GPA}) auch ein Mindestmaß der Komplexität der IT-Architektur vor. Mithin kann Komplexität nicht per se als etwas Negatives verstanden werden, was es zu vermeiden gilt. Vielmehr ist ein gewisser Grad an Komplexität erforderlich, um eine gegebene Menge an Anforderungen abbilden zu können.

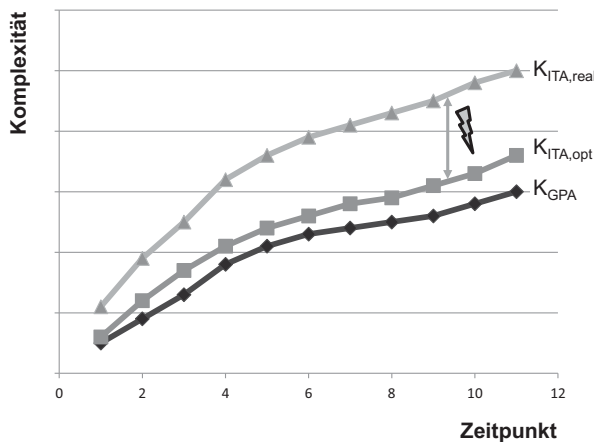


Abbildung 13: Komplexität der Geschäftsprozess- und IT-Architektur

In Abbildung 13 sind drei beispielhafte Kurven skizziert, welche die Veränderung

- der Komplexität der Geschäftsprozessarchitektur (K_{GPA}),
- der Komplexität einer die gegebene Menge an Anforderungen optimal erfüllenden IT-Architektur ($K_{ITA,opt}$) sowie
- der Komplexität der tatsächlich etablierten IT-Architektur ($K_{ITA,real}$) angeben.²⁹

Es wird angenommen, dass sich durch die Änderung der tatsächlichen IT-Architektur eine Komplexität ($K_{ITA,real}$) ergibt, die grundsätzlich über der Komplexität der IT-Architektur liegt,

²⁹ Hinter dem Verlauf der drei Kurven in Abbildung 13 steht die Annahme, dass sich der Abstand zwischen $K_{ITA,real}$ und $K_{ITA,opt}$ bei ausbleibenden Maßnahmen zum Management der strukturellen Komplexität der IT-Architektur tendenziell erhöht. Es soll an dieser Stelle nochmals betont werden, dass es sich um eine beispielhafte Darstellung handelt, um die konzeptionellen Unterschiede aufzuzeigen. Hingegen soll die Darstellung nicht suggerieren, dass diese drei Konzepte stets ermittel- oder messbar sind. Auch die Position der Kurven für K_{GPA} und $K_{ITA,opt}$ ist beispielhaft und bedeutet nicht, dass K_{GPA} stets unterhalb von $K_{ITA,opt}$ liegen muss.

durch welche die Anforderungen – aus dem Blickwinkel der Komplexität – optimal erfüllt werden. Diese Annahme wird durch zwei Überlegungen wie folgt begründet:

Im Sinne der Path Dependence Theory (Liebowitz & Margolis 1995) führen in der Vergangenheit getroffene Entscheidungen zu Systemzuständen, die wiederum mögliche Alternativen für die Weiterentwicklung des Systems in der Zukunft definieren. In diesem Sinne stellt auch eine reale IT-Architektur ein über die Zeit weiterentwickeltes, pfadabhängiges System dar, dessen Änderungen stets auf Basis der bereits bestehenden Gestalt der IT-Architektur durchgeführt werden. $K_{ITA,real}$ basiert demzufolge auf einer stetig fortgeschriebenen IT-Architektur. Im Gegensatz dazu würde ein optimaler Grad der Komplexität ($K_{ITA,opt}$) zu einem bestimmten Zeitpunkt und bei gegebenen Anforderungen den Aspekt der Pfadabhängigkeit bei der Entwicklung des Systems außer Acht lassen und die Erstellung einer IT-Architektur „auf der grünen Wiese“ erlauben. Durch eine sich ändernde Menge an Anforderungen und die hiermit verbundene neue Entscheidungsgrundlage ist es durchaus denkbar, dass sich zum Zeitpunkt t getroffene IT-Investitionsentscheidungen nicht mehr in der optimalen Architektur zum Zeitpunkt $t+1$ wiederfinden. Da ein regelmäßiges Revidieren getroffener IT-Investitionsentscheidungen sowie der damit verbundene Umbau der IT-Landschaft aus praktischer Sicht jedoch kaum durchführbar sind, entsteht hierdurch eine Abweichung zwischen $K_{ITA,real}$ und $K_{ITA,opt}$.

Die zweite Überlegung stellt zugleich den Fokus der hier durchgeführten Studie dar. Sie basiert auf dem grundlegenden Gedanken, dass in dem in Abbildung 12 skizzierten Transformationsprozess neben den erforderlichen Veränderungen auch Veränderungen an der IT-Architektur durchgeführt werden, welche die Komplexität der IT-Architektur unnötig erhöhen können. Dem in Abbildung 14 aufgezeigten beispielhaften Verlauf der Kurven K_{GPA} sowie $K_{ITA,real}$ ist eine Steigerung der Komplexität der Geschäftsprozessarchitektur (z. B. aufgrund der Einführung eines neuen Produktes) um ΔK_{GPA} zu entnehmen. Durch diese Steigerung auf Ebene der Geschäftsprozesse entstehen neue Anforderungen an die IT-Architektur.

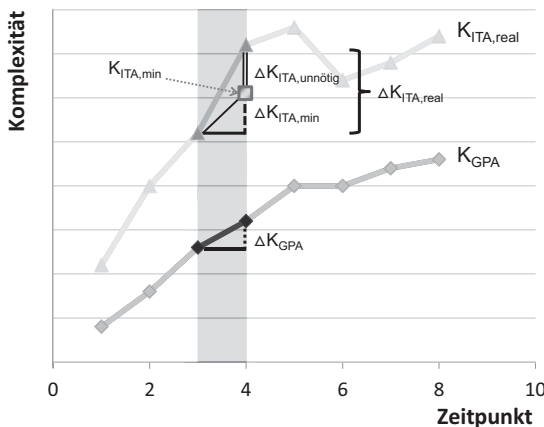


Abbildung 14: Güte einer Transformation in einer spezifischen Entscheidungssituation

Im Zuge der Umsetzung dieser Anforderungen verändert sich die Komplexität der IT-Architektur um $\Delta K_{ITA,real}$. Diese Veränderung der Komplexität der IT-Architektur setzt sich dabei aus den beiden Komponenten $\Delta K_{ITA,min}$ und $\Delta K_{ITA,unnötig}$ zusammen. $\Delta K_{ITA,min}$ stellt hierbei die Veränderung der Komplexität der realen IT-Architektur dar, die für die Umsetzung der neuen Anforderungen erforderlich ist und durch welche die IT-Architektur eine Komplexität von $K_{ITA,min}$ erreichen würde. Ein Abweichen von einer solchen Umsetzung führt zu zusätzlicher Komplexität, die durch $\Delta K_{ITA,unnötig}$ gekennzeichnet ist. Eine derartige Betrachtungsweise einer Entscheidungssituation zu einem Zeitpunkt t und einer Veränderung der realen IT-Architektur von t zu $t+1$ erlaubt demnach eine Beurteilung der Güte dieser spezifischen Architektur-Transformation hinsichtlich der Steigerung der Komplexität. Je geringer $\Delta K_{ITA,unnötig}$, desto höher ist die Güte der Umsetzung der Anforderungen hinsichtlich der Komplexitätssteigerung.

In der folgenden Abbildung 15 werden anhand der eingeführten Nomenklatur und beispielhafter Kurvenverläufe von K_{GPA} und $K_{ITA,real}$ potenzielle Effekte von Eingriffen in die Architektur veranschaulicht.

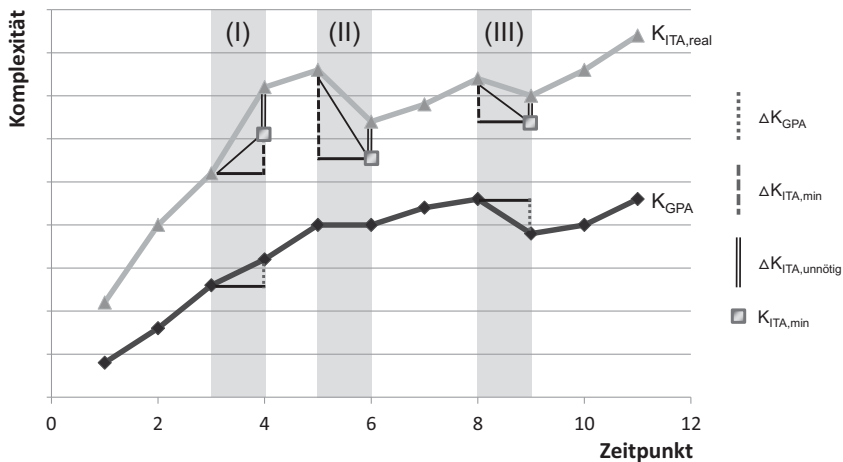


Abbildung 15: Beispiele der Veränderung der strukturellen Komplexität

Der mit (I) markierte Bereich in Abbildung 15 weist eine relativ starke Erhöhung der Komplexität der IT-Architektur auf, wobei ein großer Anteil hiervon einer unnötigen Erhöhung der Komplexität (in Höhe von $\Delta K_{ITA,unnötig}$) zuzuordnen ist. Eine solche Erhöhung unnötiger Komplexität kann beispielsweise dadurch entstehen, dass durch die Umsetzung einer Anforderung eine neue Anwendung eingeführt wird, obwohl diese Anforderung auch durch bereits vorhandene Anwendungen hätte abgedeckt werden können. Im Sinne der in Unterkapitel 3.2 eingeführten Konzeptualisierung von Komplexität in IT-Architekturen würde sich die Einführung einer neuen Anwendung eines neuen Herstellers sowohl in der Anzahl und Heterogenität der System-Komponenten als auch in der Anzahl und Heterogenität der Beziehungen zwischen den Komponenten niederschlagen.

In dem mit (II) markierten Bereich ist hingegen keine Veränderung von K_{GPA} zu erkennen, lediglich $K_{ITA,real}$ weist eine Verringerung auf. Dieser Kurvenverlauf könnte beispielsweise im Zuge einer Migration bestehender, fachlich redundanter Systeme auf eine einheitliche Lösung auftreten. Hierbei würden sich die Heterogenität sowie die Anzahl der eingesetzten Systeme und der Schnittstellen zwischen den Systemen verringern. Im Falle einer solchen Harmonisierung kann ein gewisser Grad unnötiger Komplexität zurückbleiben. So ist es beispielsweise möglich, dass nicht alle Instanzen der fachlich redundanten Systeme auf die einheitliche Lösung übertragen wurden, wodurch trotz der partiellen Harmonisierung noch eine weitere Lösung betrieben und gepflegt werden muss. Dies ist in Abbildung 15 (II) daran erkennbar, dass der Punkt $K_{ITA,min}$ unter dem der tatsächlichen Komplexität $K_{ITA,real}$ liegt. Auch bei dieser Transformation liegt folglich unnötige Komplexität in Höhe von $\Delta K_{ITA,unnötig}$ vor.

Der mit (III) markierte Bereich zeigt im Vergleich zu (I) die genau entgegengesetzte Bewegung der Kurven. Ursachen hierfür könnten jegliche Einflüsse darstellen, welche die Anzahl der durch die IT-Architektur abzudeckenden Anforderungen reduzieren. Beispiele wären die Einstellung einer Produktparte oder die Abspaltung eines Teilbereiches des Unternehmens. Auch hier zeigt Abbildung 15, dass (analog zu (II)) das Potenzial einer weiteren Reduktion der Komplexität durch die Transformation nicht ausgeschöpft wurde.

4.2.3 Einführung der behandelten Untersuchungsfragen

Zur Bearbeitung der zweiten und dritten Forschungsfrage werden diese im Folgenden durch eine Reihe von Untersuchungsfragen³⁰ präzisiert. Wie bereits angeführt, wird angenommen, dass eine Änderung der IT-Architektur nicht durch das System selbst initiiert wird, sondern ausschließlich durch ein Zutun von außerhalb des Systems angestoßen werden kann.

Für die Betrachtung der zweiten Forschungsfrage³¹ sollen in der Studie zunächst Trigger identifiziert werden, die eine Veränderung der IT-Architektur bewirken. Hieraus ergibt sich die erste Untersuchungsfrage (vgl. (1) in Abbildung 16):

Untersuchungsfrage 2.1: Welche Trigger verursachen eine Änderung der IT-Architektur?

Eine Veränderung der IT-Architektur hat grundsätzlich auch eine Veränderung der strukturellen Komplexität der IT-Architektur zur Folge. Wie im vorangegangenen Abschnitt 4.2.2 eingeführt, setzt sich die Veränderung der Komplexität der IT-Architektur ($\Delta K_{ITA,real}$) aus einer für die Umsetzung der Anforderungen erforderlichen Veränderung der Komplexität der IT-Architektur ($\Delta K_{ITA,min}$) und der sogenannten „unnötigen Komplexität“ ($\Delta K_{ITA,unnötig}$) zusammen. Untersuchungsfrage 2.2 beleuchtet, welchen Einfluss die in Untersuchungsfrage 2.1 identifizierten Trigger auf die Höhe der erforderlichen Veränderung der Komplexität der realen IT-Architektur ($\Delta K_{ITA,min}$) haben (vgl. (2) in Abbildung 16):

Untersuchungsfrage 2.2: Welchen Einfluss haben Anforderungen auf die minimale Komplexität der IT-Architektur?

³⁰ Bei der Nummerierung der Untersuchungsfragen wird im Folgenden stets die jeweilige Nummer der Forschungsfrage vorangestellt.

³¹ Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

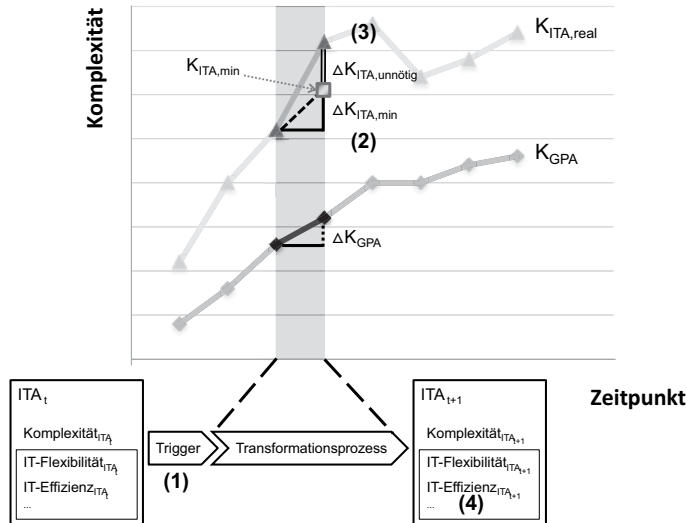


Abbildung 16: Verortung der Untersuchungsfragen

Wie in Abschnitt 4.2.2 diskutiert, wird angenommen, dass es im Rahmen des Transformationsprozesses (vgl. (3) in Abbildung 16) zu Abweichungen von $K_{ITA,min}$ in Form unnötiger Komplexität ($\Delta K_{ITA,unnötig}$) kommen kann. Wie und warum diese Abweichungen entstehen, soll durch die dritte Untersuchungsfrage beantwortet werden:

Untersuchungsfrage 2.3: Welche Mechanismen führen bei der Umsetzung der Anforderungen zu Abweichungen von der minimalen Komplexität der IT-Architektur?

Die Komplexität wurde als eine Eigenschaft der IT-Architektur eingeführt. Wie bereits zu Beginn des Unterkapitels diskutiert, weisen Systeme noch weitere Eigenschaften wie Flexibilität und Effizienz auf (vgl. (4) in Abbildung 16). Die dritte Forschungsfrage³² der vorliegenden Dissertation betrachtet, wie sich die Veränderung der Komplexität einer IT-Architektur auf Eigenschaften des Systems (vgl. Alderson & Doyle 2010) auswirkt. Zur Beantwortung dieser Fragestellung wird diese ebenfalls durch zwei Untersuchungsfragen präzisiert. Dabei sollen zunächst die für das Management von IT-Architekturen relevanten Eigenschaften des Systems identifiziert und anschließend der Einfluss der Komplexität der IT-Architektur auf diese Eigenschaften untersucht werden:

Untersuchungsfrage 3.1: Welche Eigenschaften eines Systems sind relevant für das Management von IT-Architekturen?

Untersuchungsfrage 3.2: Wie werden diese Eigenschaften eines Systems durch die Komplexität der IT-Architektur beeinflusst?

³² Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

4.3 Methodisches Vorgehen

In diesem Unterkapitel wird das methodische Vorgehen der durchgeführten Studie vorgestellt. Hierzu wird in Abschnitt 4.3.1 zunächst das Design der Studie beschrieben. Dabei wird sowohl die Wahl der angewendeten Forschungsmethode begründet als auch die Auswahl des Unternehmens diskutiert. Danach wird in Abschnitt 4.3.2 das Vorgehen bei der Durchführung der Datenerhebung vorgestellt. Neben der Auswahl der Experten für die Expertengespräche werden zudem auch der Inhalt und die Struktur der Interviews vorgestellt. In Abschnitt 4.3.3 wird schließlich das Vorgehen bei der Auswertung der Daten erläutert.

4.3.1 Design der Studie

Aufgrund des explorativen Charakters der betrachteten Forschungsfragen, durch die u. a. Ursachen der Entstehung struktureller Komplexität von IT-Architekturen und die dabei ablaufenden Mechanismen identifiziert werden sollen, beruht die durchgeführte Untersuchung auf einem qualitativen Forschungsansatz, basierend auf einer Fallstudienmethodik (Yin 2014). Diese stellt eine in der IS-Forschung anerkannte (Benbasat et al. 1987) und weit verbreitete Forschungsmethode dar (Dubé & Paré 2003). Sie erlaubt es, Phänomene innerhalb ihres realen Kontexts zu untersuchen, insbesondere, wenn eine Abgrenzung des Phänomens und des jeweiligen Kontextes nur schwer möglich ist (Yin 2014). Dem Kenntnisstand des Autors zufolge hat sich bislang noch keine Studie mit der Entstehung struktureller Komplexität von IT-Architekturen und den mit dieser verbundenen Folgen beschäftigt. Um ein tieferes Verständnis für dieses Phänomen zu entwickeln, bietet sich daher eine qualitative Untersuchung an.

Bei Komplexität handelt es sich um ein vielschichtiges Phänomen, das unterschiedliche Architekturbereiche und die mit diesen verbundenen organisationalen Rollen betrifft. Dyer & Wilkins (1991) argumentieren in ihrer Arbeit, dass Single-Case-Studien besonders geeignet sind, tief greifende Einblicke in den Untersuchungsgegenstand in einem realen Kontext zu erlangen. Daher wurde für die Durchführung der Fallstudie ein Single-Case-Ansatz gewählt. Dies soll ermöglichen, sich sehr intensiv mit dem Fall auseinanderzusetzen zu können und dabei zu vielfältigen Erkenntnissen zu gelangen. Das zu untersuchende Unternehmen sollte dabei eine Reihe von Anforderungen erfüllen, um möglichst vielseitige Erkenntnisse gewinnen zu können. Es sollte...

- (a) ...bereits mehrere Jahre bestehen, damit verschiedene Ausprägungen von Änderungen der IT-Architektur und damit der IT-Komplexität beobachtet werden können. Gemäß der in Unterkapitel 4.2 eingeführten Annahmen ist davon auszugehen, dass der Abstand zwischen $K_{ITA,opt}$ und $K_{ITA,real}$ bei relativ jungen Unternehmen noch gering ist und somit kaum Abweichungen untersucht werden können.
- (b) ...eine ausreichende Größe aufweisen, damit möglichst viele Typen von Anforderungen von vielen unterschiedlichen Interessengruppen beobachtet werden können.
- (c) ...möglichst über eine Vielzahl von Standorten verfügen, damit die Effekte unterschiedlicher Standorte und der mit diesen verbundenen organisatorischen und räumlichen Trennung untersucht werden können.

- (d) ...möglichst in einer Branche tätig sein, die regulatorischen Anforderungen unterliegt, um aus einer systemtheoretischen Perspektive Einflüsse des Supersystems auf das System „Unternehmen“ miteinzubeziehen (vgl. Abschnitt 2.1.1).
- (e) ...möglichst noch keine spezifische Strategie für das Management seiner IT-Architekturen etabliert haben, um keine der zu identifizierenden Einflussfaktoren und Mechanismen, aufgrund vorab getroffener strategischer Entscheidungen auszublenden.³³

Für die Durchführung der Studie konnte eine in der chemischen Industrie tätige Unternehmensgruppe (im Folgenden auch „UG“) gewonnen werden, welche die oben angeführten Anforderungen erfüllt. Die UG besteht seit mehreren Jahrzehnten und beschäftigt mehr als 5000 Mitarbeiter an einer Vielzahl von Standorten weltweit. Die heutige Struktur der UG ist dabei das Ergebnis eines stetigen organischen Wachstums, der Errichtung weiterer Produktionsstätten im Ausland und einer Reihe von Unternehmenszukäufen. Da die akquirierten Unternehmen einen gewissen Grad an Eigenständigkeit beibehalten sollten, hat sich eine kombinierte IT-Governance-Struktur herausgebildet. Neben dem zentral organisierten IT-Bereich, der sich im Verantwortungsbereich der Dachgesellschaft befindet, hat sich zudem eine feudale Struktur (Weill & Ross 2004) etabliert, die den angeschlossenen Gesellschaften grundsätzlich das (Fort-)Führen eigener IT-Abteilungen erlaubt, um u. a. lokalen Anforderungen besser gerecht werden zu können. Gleichzeitig existierten zum Zeitpunkt der Durchführung der Studie noch keine formale IT-Strategie oder anderweitige formale Vorgaben durch die Geschäftsleitung, die regeln, welche IT-Leistungen durch die zentrale IT der Dachorganisation und welche dezentral durch die lokalen IT-Funktionen der angeschlossenen Unternehmen zu erbringen sind.³⁴

Aufgrund der über die Jahre hinweg gewachsenen IT-Landschaft der UG und der zuvor dargestellten Gegebenheiten wurde diese als besonders geeignet für die Untersuchung beurteilt. Bedingt durch häufige Änderungen an der IT-Architektur ist davon auszugehen, dass die interviewten Experten eine Vielzahl an Erfahrungen hierzu einbringen können. Darüber hinaus konnte ein besonderer Zugang zu der UG gewonnen werden. Zwei Forschungsassistenten, die bereits mehrere Jahre im zentralen IT-Bereich der Dachgesellschaft tätig waren, unterstützten in den Phasen der Datenerhebung und –auswertung. Aufgrund der mehrjährigen Erfahrung der Forschungsassistenten wurde damit ein besseres Verständnis der unternehmensspezifischen Projekte und der dort üblichen Begrifflichkeiten sichergestellt.

Um potenziell verschiedene Sichtweisen auf bzw. Ausprägungen der entstandenen Komplexität der IT-Architektur untersuchen zu können, wurde ein integrierter Ansatz gewählt (Yin 2014). Hierbei wurden sowohl die Anwendungsarchitektur als auch die Infrastruktur-

³³ So könnte z. B. eine „Single-Vendor-Strategie“ dazu führen, dass grundsätzlich keine von den Standards abweichenden Anwendungen oder Hardware-Komponenten eingeführt werden dürfen.

³⁴ Dieser Aspekt grenzt die vorgefundene IT-Governance-Struktur von einem föderalistischen Konzept ab, bei dem die Zuständigkeiten für bestimmte Teilbereiche klar vorgegeben sind. Weill & Ross (2004) definieren „the federal model as coordinated decision making involving both the center and the business units“ (Weill & Ross 2004, S. 61).

architektur der UG untersucht. Dadurch konnten die wesentlichen Teile der IT-Architektur abgedeckt werden.³⁵

4.3.2 Datenerhebung

Die Erhebung der qualitativen Daten erfolgte im Zeitraum von Anfang Juni bis Ende September 2014 in Form von leitfadengestützten, semistrukturierten Experteninterviews (Myers & Newman 2007) mit Schlüsselpersonen aus den Bereichen der Anwendungs- bzw. der Infrastrukturarchitektur. Dabei wurden die Interviewpartner so ausgewählt, dass möglichst viele verschiedene Perspektiven auf die Anwendungslandschaft und die IT-Infrastruktur in die Untersuchung miteinbezogen werden konnten. Im Bereich der Anwendungsarchitektur waren dies in erster Linie Abteilungsleiter der IT-Abteilungen der UG, die in der Organisation primäre Anlaufstelle für Fragestellungen bzgl. der Anwendungslandschaft sind. Ergänzt wurde dieser Teil der Untersuchung um einen Abteilungsleiter einer fachlichen Abteilung. Im Bereich der Infrastrukturarchitektur wurden Experten der zentralen IT der Dachgesellschaft aus der Netzwerk- und Systeminfrastruktur sowie der IT-Sicherheit und zwei Leiter der IT-Bereiche dezentraler Standorte befragt. Insgesamt wurden neun Interviews mit elf Experten und einer Gesamtdauer von mehr als zehn Stunden geführt. An dem Interview mit dem Abteilungsleiter „Logistische Prozesse SAP“ nahmen zudem noch zwei Mitarbeiter mit spezifischem Know-how teil. In Tabelle 6 ist eine Übersicht der im Rahmen der qualitativen Untersuchung befragten Experten aufgeführt.

Die Experteninterviews wurden nach gründlicher Vorbereitung und in Abstimmung mit dem Autor von den mit den Interviewpartnern und Projekten vertrauten Forschungsassistenten durchgeführt. Während der Interviews waren stets beide Forschungsassistenten anwesend, wobei jeweils einer der Forschungsassistenten dem Bereich Anwendungs- bzw. Infrastrukturarchitektur zugeordnet war und die Gesprächsführung übernahm. Im Vorfeld der Experteninterviews wurde gemeinsam mit den Forschungsassistenten ein Leitfaden³⁶ entwickelt, der auf das in Abbildung 12 dargestellte Schema der Veränderung von IT-Architekturen abgestimmt war und primär der Strukturierung und Unterstützung der Gesprächsführung diente. Da die qualitative Untersuchung nicht das Ziel hatte, auf einem Modell basierende Hypothesen zu untersuchen, sondern ein besseres Verständnis der Entstehung von Komplexität in IT-Architekturen zu entwickeln, wurde ein explorativer Ansatz gewählt und die Fragestellungen des Leitfadens somit gezielt offen gestaltet. Nach einem einleitenden Teil, in dem die Experten in das Themengebiet des Interviews eingeführt und ihre relevanten biographischen Daten abgefragt wurden, folgten die fachspezifischen Fragestellungen. Letztere wurden den Experten in gekürzter Form vor dem Interview als Agenda und für die Vorbereitung des Gesprächs wie folgt zur Verfügung gestellt:

- a) Die letzten architekturrelevanten IT-Projekte und deren Verlauf
- b) Die Komplexität der Anwendungs- bzw. Infrastrukturarchitektur

³⁵ Wie in der Praxis häufig der Fall, stellt die Datenarchitektur einen integralen Bestandteil des Aufgabenbereiches der für Anwendungsarchitektur zuständigen Funktion dar.

³⁶ Der im Rahmen der Studie verwendete Interviewleitfaden ist im Anhang aufgeführt.

- c) Eigenschaften der Anwendungs- bzw. Infrastrukturarchitektur
- d) Bestehende Mittel zur Steuerung von Komplexität

Tabelle 6: Übersicht der befragten Experten der qualitativen Untersuchung

Architekturbereich	Funktion / Fachgebiet des Experten	Dauer des Interviews	Erfahrung im aktuellen Tätigkeitsfeld in Jahren
Anwendung	Demand Manager und Abteilungsleiter kaufmännische Prozesse SAP	60 Minuten	> 5 Jahre (> 17 Jahre im SAP-Umfeld)
Anwendung	Demand Manager und Abteilungsleiter IT-Vertriebsprozesse SAP	45 Minuten	> 5 Jahre (> 9 Jahre im SAP-Umfeld)
Anwendung	Demand Manager und Abteilungsleiter logistische Prozesse SAP	96 Minuten	> 5 Jahre
Anwendung	Supply Manager logistische Prozesse SAP	(96 Minuten)	> 5 Jahre (> 19 Jahre im Unternehmen)
Anwendung	Administrator SAP Basis	(96 Minuten)	> 13 Jahre
Anwendung	Projektmanager SAP CRM	52 Minuten	> 1,5 Jahre (> 3,5 Jahre im SAP-Umfeld)
Anwendung	Leitung Marketing-Lösungen	80 Minuten	k.A.
Infrastruktur	Leitung IT-Sicherheit (zentral)	97 Minuten	> 5 Jahre (> 20 Jahre in der IT der Gesellschaft)
Infrastruktur	Leitung Netze und Systeme in der IT-Infrastruktur (zentral)	55 Minuten	> 5 Jahre (> 10 Jahre in der IT der Gesellschaft)
Infrastruktur	IT-Leitung eines angeschlossenen Unternehmens	66 Minuten	> 6 Jahre (> 10 Jahre in der IT der Gesellschaft)
Infrastruktur	IT-Leitung eines angeschlossenen Unternehmens	106 Minuten	> 4 Jahre (> 11 Jahre in der IT der Gesellschaft)

Durch den ersten Teil (a) hatten die Experten Gelegenheit, anhand von Projekten über durchgeführte Änderungen an der IT-Architektur zu berichten. Projekte boten sich als Gesprächsgegenstand besonders an, da die Auslöser der Projekte, der Zustand der IT-Architektur (bzgl. Struktur und Eigenschaften) vor und nach der Durchführung der Projekte, aber auch die jeweiligen Transformationsprozesse diskutiert werden konnten. Im darauf folgenden Teil (b) wurde den Experten zunächst die systemtheoretische Konzeptualisierung struktureller Komplexität vorgestellt (vgl. Unterkapitel 3.2), um ein einheitliches Verständnis für die weitere Diskussion sicherzustellen. Auf dieser Grundlage sollten die Experten die bestehende Komplexität der IT-Architektur in ihrem Bereich beurteilen und gegebenenfalls auch einen Bezug zu den bereits diskutierten Projekten herstellen. Darüber hinaus wurden die Experten nach weiteren Ursachen der bestehenden strukturellen Komplexität befragt. Im Anschluss wurden im dritten Teil (c) die Auswirkungen der bestehenden Form struktureller Komplexität der Anwendungs- bzw. Infrastrukturarchitektur diskutiert. Von besonderem Interesse war hierbei, inwieweit sich die aktuelle Struktur auf die IT-Prozesse der Bereiche (z. B. hinsichtlich Änderbarkeit oder Wartbarkeit der IT) oder die mit der IT assoziierten Kosten positiv oder negativ auswirken. Der letzte Teil (d) des Experteninterviews zielte schließlich darauf ab, inwie-

fern bereits Maßnahmen etabliert sind, die bei dem Umgang mit der bestehenden Komplexität des Bereiches unterstützen.

Bis auf eine Ausnahme wurden alle Experteninterviews aufgezeichnet und standen für die weitere Auswertung zur Verfügung. Im Zuge des nicht aufgezeichneten Interviews wurde eine Mitschrift erstellt, die ebenfalls für die weitere Analyse hinzugezogen wurde. Zusätzlich wurden während der Interviews ergänzende Unterlagen zur Veranschaulichung der Projekte und Zusammenhänge verwendet, die von den Forschungsassistenten eingesehen werden konnten, jedoch aus Gründen der Vertraulichkeit im Unternehmen verbleiben mussten.

4.3.3 Auswertung der Daten

Zur Vorbereitung der Auswertung der geführten Experteninterviews wurden diese zunächst mithilfe der Software f4 transkribiert. Die resultierenden Transkripte der zehnstündigen Aufzeichnungen umfassen in Summe mehr als 79.000 Worte.

Im Anschluss wurde eine zweistufige Codierung der Daten³⁷ vorgenommen (vgl. Abbildung 17).³⁸ In der ersten Stufe wurde eine initiale Codierung mittels Provisional Coding (Dey 1993; Miles & Huberman 1994) durchgeführt. Als Coding-Unit wurden dabei inhaltlich zusammengehörige Passagen herangezogen, die aus einzelnen Sätzen bis hin zu ganzen Absätzen bestehen konnten.

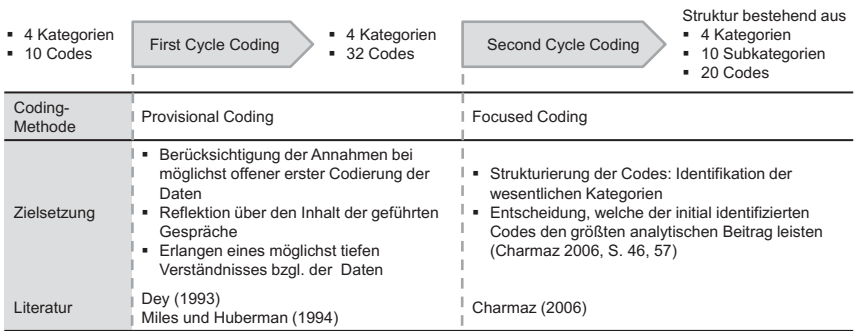


Abbildung 17: Übersicht über den Coding-Prozess

Die Codierung anhand der Provisional Coding-Methode begann mit vier Kategorien und zehn initialen Codes, basierend auf den in Unterkapitel 4.2 dargestellten Annahmen sowie der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung von Komplexität. Ausgehend von diesen initialen Codes wurden die Daten zunächst möglichst offen bzgl. der Ursachen, der Entstehung sowie der Folgen struktureller Komplexität hin analysiert, um weitere, die o. a. theoretischen Überlegungen komplementierende oder auch hiervon abweichende, Erklärungsansätze zuzulassen. Wie die Provisional Coding-Methode vorsieht, wurden die

³⁷ Die in Abbildung 17 verwendeten Begriffe des First- und Second Cycle Codings wurden aus Saldaña (2013) entnommen.
³⁸ Für die Codierung wurde die Software MAXQDA11 verwendet.

initialen Codes dabei angepasst, teilweise entfernt und neue Codes hinzugefügt (Saldaña 2013).

In der zweiten Stufe wurde mittels Focused Coding (Charmaz 2014) eine Ausdifferenzierung und Strukturierung der codierten Daten vorgenommen. Focused Coding ist eine Methode, mit der die häufigsten bzw. wesentlichen Codes identifiziert werden, um hieraus eine Kategorisierung der Daten durchführen zu können. Im Rahmen dieser Kategorisierung wurden Trigger identifiziert, die eine Änderung der IT-Architektur verursachen (vgl. Untersuchungsfrage 2.1). Aus den Codes zur Entstehung struktureller Komplexität wurden die Kategorien „Einfluss von Anforderungen auf die minimale IT-Komplexität“ (vgl. Untersuchungsfrage 2.2) sowie „Mechanismen, die ein Abweichen von der minimalen IT-Komplexität erklären“ (vgl. Untersuchungsfrage 2.3) ausgearbeitet. Die durch die Umsetzung der Anforderungen entstehenden Ausprägungen struktureller Komplexität dienen dem besseren Verständnis der untersuchten Zusammenhänge sowie als Bindeglied zu den Folgen struktureller Komplexität. Letztere wurden im Zuge des Focused Codings in der Kategorie „Durch strukturelle Komplexität beeinflusste Zieleigenschaften des Systems“ zusammengefasst (vgl. Untersuchungsfragen 3.1 und 3.2).

Die Codierung wurde ausschließlich durch den Autor der vorliegenden Dissertation vorgenommen. Dabei wurden die in Saldaña (2013) vorgeschlagenen Maßnahmen zur Validierung des Arbeitsfortschrittes während des Solo-Codings berücksichtigt. So fand ein regelmäßiger Austausch mit anderen Doktoranden, den Forschungsassistenten sowie den beteiligten Praktikern statt: Während der ersten Stufe der initialen Codierung sicherten Abstimmungsgespräche mit den Forschungsassistenten ein korrektes Verständnis der in den Interviews besprochenen Projekte und Zusammenhänge. Dies unterstützte die Entwicklung der Codes, ausgehend von der initialen Menge zu Beginn des Provisional Codings hin zu der Menge der vorläufigen Codes vor der zweiten Stufe – dem Focused Coding. Im Rahmen des Focused Codings erwiesen sich die Diskussionen mit Kollegen bei der Kategorisierung der Codes im Hinblick auf die in Abbildung 16 dargestellte Systematisierung des Transformationsprozesses als äußerst hilfreich. Ein vorläufiger Stand der Ergebnisse wurde zudem drei Interviewpartnern im Rahmen eines Workshops vorgestellt und mit diesen diskutiert. Das hierbei erhaltene Feedback floss ebenfalls in die im folgenden Unterkapitel vorgestellten finalen Ergebnisse ein.

4.4 Ergebnisse der qualitativen Untersuchung

Anhand des in Unterkapitel 4.2 eingeführten Schemas der Veränderung der IT-Architektur werden im Folgenden die Ergebnisse der qualitativen Untersuchung vorgestellt (vgl. Abbildung 18).

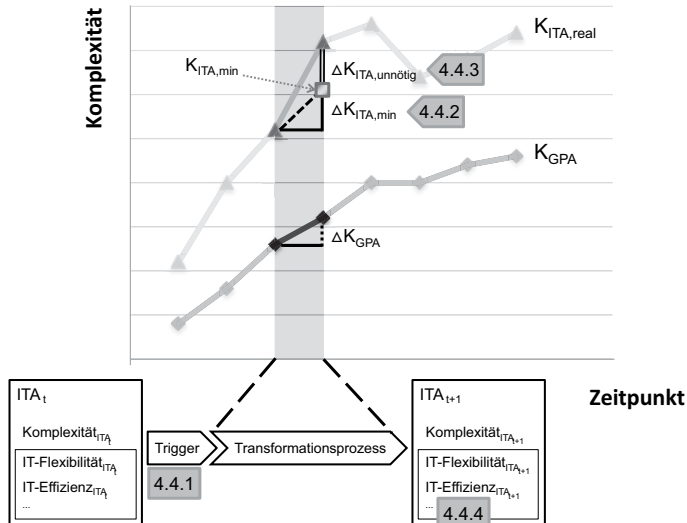


Abbildung 18: Struktur der Ergebnisse der qualitativen Untersuchung

Die Strukturierung der Abschnitte erfolgt in Anlehnung an die in Abschnitt 4.2.3 eingeführten Untersuchungsfragen. In Abschnitt 4.4.1 werden zunächst sowohl externe als auch interne Anforderungen als Trigger von Veränderungen der IT-Architektur diskutiert. Darauf folgend werden in Abschnitt 4.4.2 die in den Interviews erörterten Einflüsse von Anforderungen auf die Veränderung der minimal erforderlichen IT-Komplexität vorgestellt. In Abschnitt 4.4.3 werden Mechanismen untersucht, die ein Abweichen von dieser minimalen Veränderung der IT-Komplexität verursachen bzw. begünstigen. Weiterhin wird an dieser Stelle dargestellt, welche Architekturmaßnahmen in den Interviews identifiziert wurden und wie diese mit den o. a. Mechanismen in Beziehung stehen. Die durch die strukturelle Komplexität von IT-Architekturen beeinflussten Eigenschaften eines Systems werden in Abschnitt 4.4.4 diskutiert. Abbildung 19 zeigt eine Vorschau auf die im Zuge der Interviews identifizierten Kategorien. Die Transkription der Interviews hatte das Ziel, die Inhalte der Expertenäußerungen zu erfassen. Zur besseren Lesbarkeit wurden die Zitate in der vorliegenden Arbeit geglättet, indem diese von Wortwiederholungen, Füllwörtern und leichten Satzbaufehlern befreit wurden. Der Inhalt der Aussagen wurde dabei nicht verändert.

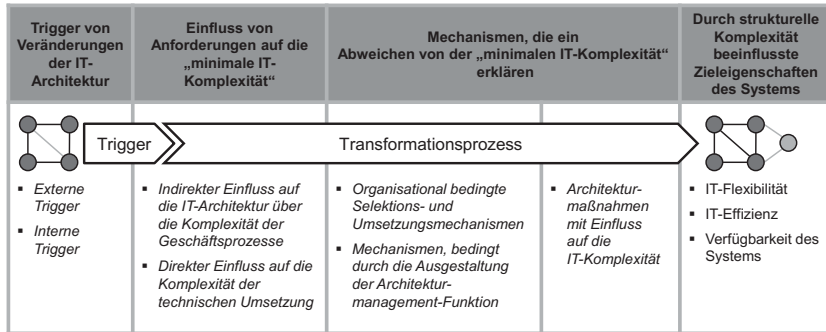


Abbildung 19: Vorschau auf die identifizierten Kategorien

4.4.1 Trigger von Veränderungen der IT-Architektur

Als Trigger werden im Folgenden die Auslöser einer Änderung der IT-Architektur verstanden. Wie die Auswertung der Interviews ergeben hat, können diese ihren Ursprung außerhalb sowie innerhalb des Unternehmens haben. Hieraus resultiert die Unterteilung in sogenannte „externe“ und „interne“ Trigger, die in den folgenden beiden Unterabschnitten beschrieben werden. Dieser Abschnitt widmet sich damit Untersuchungsfrage 2.1 („Welche Trigger verursachen eine Änderung der IT-Architektur?“).

4.4.1.1 Externe Trigger

Externe Trigger zeichnen sich dadurch aus, dass sie nicht aus dem Unternehmen heraus entstehen, sondern aus dem Umfeld auf das Unternehmen einwirken. Im Kontext der untersuchten UG konnten mit anorganischem Wachstum sowie gesetzlichen Anforderungen zwei Ausprägungen externer Trigger identifiziert werden.

4.4.1.1.1 Anorganisches Wachstum

Unter anorganischem Wachstum wird der Teil des Wachstums verstanden, der nicht aus eigener Kraft des Unternehmens, sondern vielmehr durch Zukauf von Unternehmen entsteht (Sherman 2010). Diese Wachstumsstrategie fand im untersuchten Fall in der Vergangenheit bereits häufig Anwendung, wodurch eine Reihe von zuvor unabhängigen Unternehmen in die Struktur der UG integriert werden mussten. Experten aus dem Bereich der Anwendungsarchitektur diskutieren Unternehmenszukäufe explizit als einen Auslöser für eine Änderung der IT-Architektur und nachgelagert auch für die Entstehung zusätzlicher struktureller Komplexität. In der Infrastrukturarchitektur liegt der Fokus hingegen eher auf den mit den Zukäufen verbundenen Folgen für das Management der Infrastruktur.

„Im Produktionsumfeld würde ich tatsächlich auch noch das Thema anorganisches Wachstum nennen. [...] Also eine [Gesellschaft A] hat man zugekauft. Da geht aber keiner hin und sagt, wenn ich die [Gesellschaft A] zukaufe, dann werfe ich die ganze EDV

dort weg und etabliere eine [Dachgesellschaft]-EDV. Mittlerweile haben wir [Gesellschaft A], [Gesellschaft B] und [Gesellschaft C] hinzu bekommen und diese noch mit verschiedenen Standorten. Dadurch habe ich in der Produktion sehr viele Standorte, die sich sehr heterogen aufgestellt haben.“ (Experte aus der Anwendungsarchitektur)³⁹

Ein Unternehmenszukauf kann dabei auch nachgelagert eine Änderung der IT-Architektur nach sich ziehen, z. B. für den Fall, dass sich der Parallelbetrieb zweier fachlich redundanter Systeme als nachteilig erweist. Ein Experte aus dem Bereich der Anwendungsarchitektur erläutert im Zuge des Interviews die Beweggründe der Migration eines CRM-Systems einer angeschlossenen Gesellschaft auf die zentrale CRM-Lösung (vgl. Sektion 4.4.3.3.2). Die Tatsache, dass Kundeninformationen nicht zentral in einem System abgelegt waren, erschwerte in der Vergangenheit die Abstimmung der einzelnen Gesellschaften bezüglich der Kundenansprache:

„Wenn ich die Daten im gleichen System sehe, habe ich ja die Chance zu erkennen, ob ich Dinge doppelt mache. Sonst kommt es eben vor, dass ein Kunde von drei Marken dreimal besucht wird und andere Kunden dafür gar nicht.“ (Experte aus der Anwendungsarchitektur)

Bleibt im Falle von Unternehmenszukäufen eine Harmonisierung in Form einer Vereinheitlichung der Systeme der Gesellschaften hingegen aus, so berichten die Experten von einem starken Zuwachs der Komplexität der IT-Architektur, bedingt durch die steigende Anzahl redundanter Systeme:⁴⁰

„[Gesellschaft A] hat einen eigenen Internetzugang, der betreut werden muss, eine eigene Firewall, die betreut werden muss und einen eigenen Proxyserver, der betreut werden muss. Wir haben ja alles in [der Dachgesellschaft] redundant, hoch verfügbar und eine performante Anbindung.“ (Experte aus der Infrastrukturarchitektur)

Anorganisches Wachstum ist auch für die Experten der Infrastruktur ein relevanter Auslöser für die Veränderung der IT-Architektur und folglich auch für die Entstehung struktureller Komplexität. Somit besteht bei der Zusammenführung der IT-Landschaften der Unternehmen stets die Gefahr, dass die Anzahl und Heterogenität der Anwendungen und Infrastruktur-Komponenten weiter ansteigt. Ohne eine Standardisierung der Systeme bzw. eine Migration redundanter Systeme (vgl. Unterabschnitt 4.4.3.3) kann es dazu kommen, dass sich Anzahl und Heterogenität der Schnittstellen durch die Umsetzung zusätzlicher Anforderungen zunehmend erhöhen.

³⁹ Aus Gründen der Anonymisierung wird in den angegebenen Zitaten lediglich der Bereich transparent gemacht, dem der jeweilige Experte zugeordnet ist.

⁴⁰ In diesem Zitat wird von zwei Formen der Redundanz gesprochen. So handelt es sich bei den angeführten redundanten Komponenten (Internetzugang, Firewall und Proxyserver) um fachlich redundante Systeme, die ausschließlich bei dem angeschlossenen Unternehmen genutzt werden und somit um "unnötige" Redundanz an einem dezentralen Standort. Der im zweiten Satz des Zitates verwendete Redundanz-Begriff ist positiv zu verstehen. Bei diesen Redundanzen handelt es sich um solche, die eine Verfügbarkeit der Systeme im Falle eines Vorfalls sicherstellen sollen.

4.4.1.1.2 Gesetzliche Anforderungen

Die gesetzlichen Anforderungen sind für beide Architekturebenen ein relevanter Auslöser von Änderungen an der IT-Architektur. Als gesetzliche Anforderungen werden alle Anforderungen verstanden, die aus einer sich verändernden oder neuen Gesetzgebung heraus entstehen. Hierbei kann es sich um unterschiedlichste Vorgaben aus diversen Bereichen handeln. In den Interviews werden beispielsweise Vorgaben bzgl. der Rechnungslegung, dem Datenschutz oder zu Zoll- und Lieferbeschränkungen genannt. Durch Produktionsstätten und Gesellschaften im Ausland kommen zudem noch landesspezifische Ausprägungen der gesetzlichen Anforderungen hinzu:

„Es ist immer eine Herausforderung, dass landesspezifische [...] Aspekte wie Zollbeschränkungen, Lieferbeschränkungen oder Rechnungsstellung berücksichtigt werden müssen. Da haben natürlich alle eine andere Gesetzgebung.“ (Experte aus der Infrastrukturarchitektur)

Im Falle neuer oder sich ändernder gesetzlicher Bestimmungen sind diese durch die IT umzusetzen. Die Menge und Diversität an gesetzlichen Vorgaben schlägt sich entsprechend in der Ausgestaltung der Prozesse und demnach auch in der IT-technischen Umsetzung nieder, was zu einer höheren Heterogenität auf der Anwendungsebene führt:

„Das liegt aber oft auch an den gesetzlichen Bestimmungen. Wenn ich an das Ausland denke, dann müssen dort gewisse Dinge auf den Formularen aufgedruckt werden. Es müssen z. B. das Ursprungsland und der Chargensplit mit angegeben werden, sonst sagt der russische Zoll „nein“. Die lassen den LKW dann wirklich zurückfahren. Tatsächlich. Da müssen wir uns beugen, das tut weh. Hierfür haben wir Ausnahmen programmiert.“ (Experte aus der Anwendungsarchitektur)

Mit Blick auf die Infrastrukturarchitektur bedeuten gesetzliche Vorgaben beispielsweise, dass auch bereits abgelöste IT-Systeme, deren Daten jedoch nicht vollständig migriert wurden, aufgrund von Anforderungen an die Aufbewahrungsfristen von Geschäftsunterlagen teilweise noch weiter betrieben und gegebenenfalls neue Schnittstellen zu diesen Systemen implementiert werden müssen.

„Man muss es aus Revisionsgründen, weil man die Daten zehn Jahre aufheben muss, im Hintergrund laufen lassen.“ (Experte aus der Infrastrukturarchitektur)

4.4.1.2 Interne Trigger

Im Gegensatz zu den externen Triggern entstehen die internen Trigger innerhalb des Unternehmens. Dazu zählen im Kontext der durchgeführten Studie sowohl fachliche Anforderungen, die durch die Geschäftstätigkeit der Unternehmung neu aufkommen, als auch technische Anforderungen, die sich aus der bestehenden IT-Architektur ergeben.

4.4.1.2.1 Fachliche Anforderungen

Unter fachlichen Anforderungen werden Anforderungen an eine IT-Lösung verstanden, die grundsätzlich von Mitarbeitern des Geschäftsbereiches an die IT-Funktion bzw. einen verantwortlichen Demand Manager – als eine Funktion, die Anforderungen eines bestimmten Bereiches aufnimmt und konsolidiert – herangetragen werden. Bei diesen Anforderungen kann es sich um Änderungswünsche hinsichtlich einer bestehenden IT-Lösung, aber auch um die Einführung einer neuen IT-Lösung zur Unterstützung eines neuen oder bestehenden Geschäftsprozesses handeln. Neue fachliche Anforderungen stellten sich in den Gesprächen über die durchgeführten Projekte als der wesentliche Auslöser von Änderungen der IT-Architektur heraus.

Im Bereich der Anwendungsarchitektur wird u. a. von Anforderungen an die Anpassungen von Geschäftsprozessen in SAP berichtet. Weiter erwähnte Anforderungen betreffen die Vereinheitlichung der Content Management Systeme der diversen Web-Auftritte, um einen einheitlichen Online-Markenauftritt der angeschlossenen Gesellschaften sicherzustellen oder die Einführung einer weiteren – den Anforderungen einer bestimmten Ländergesellschaft entsprechenden – E-Learning-Plattform.

Die Experten aus der Infrastrukturarchitektur beschäftigen sich – entsprechend ihrer Funktion – tendenziell eher mit der technischen Umsetzung einer geplanten Lösung, weshalb in den Gesprächen die strukturelle Komplexität der IT-Architektur hervorgehoben wird, welche aus den diversen fachlichen Anforderungen der dezentralen Einheiten resultiert. Hierzu wird angeführt, dass es aufgrund der unterschiedlichen fachlichen Anforderungen und der Eigenständigkeit einzelner Gesellschaften häufig zur Entstehung von Insellösungen kommt. Dabei handelt es sich um Lösungen, die auf spezifische Problemstellungen ausgerichtet sind und nicht oder nur sehr schwer auf ähnliche Problemstellungen anderer angeschlossener Gesellschaften angewendet werden können.

Ein aktuelles Themenfeld, das sowohl von den Experten der Anwendungs- als auch der Infrastrukturarchitektur diskutiert wird, ist „Mobility“. So kommt es u. a. seitens des Außendienstes vermehrt zu Anforderungen, zur Arbeitserleichterung auch über mobile Endgeräte auf relevante Daten aus den unterschiedlichsten Datenquellen zugreifen zu können. Diese Anforderungen betreffen somit eine Vielzahl von datenführenden Systemen und verschiedenste mobile Endgeräte. Gleichzeitig führt die zunehmende Durchdringung des Alltages mit modernen IT-Services im privaten Umfeld (wie z. B. Smartphone-Apps) zu einer Erwartungshaltung der Nutzer gegenüber der Unternehmens-IT, die durch diese nur schwer erfüllt werden kann:

„Ein Problem, das ich sehe, ist, dass die Anforderungen der Benutzer immer mehr werden – gerade was das Thema Mobilität betrifft. Ich möchte überall alles verfügbar haben, ob das jetzt Daten oder Anwendungen sind. Den Benutzer interessiert dabei kaum, wie das funktioniert. Er will seinen Kram haben und er will es schnell haben. Benutzer wollen neue Sachen relativ schnell haben. Und wir auf der IT-Seite haben das Problem,

dass wir das Ganze ins Netz, in eine Infrastruktur oder in eine Anwendungsstruktur integrieren müssen.“ (Experte aus der Infrastrukturarchitektur)

Von einem ähnlichen Sachverhalt wird zudem im Kontext der Anforderung der Einführung cloud-basierter Lösungen berichtet:

„Wir müssen davon ausgehen, dass die Anforderungen in immer kürzeren Abständen kommen werden und immer schneller umgesetzt werden müssen. Ja, weil der Markt es theoretisch mit Cloud-Lösungen hergibt. Die Fachabteilungen wollen das haben und sagen: ‚Daheim kann ich es, warum kann ich es hier nicht?‘“ (Experte aus der Infrastrukturarchitektur)

Auch in diesem Bereich existiert demzufolge eine gewisse Erwartungshaltung⁴¹ seitens der Fachbereiche, aus der neue Anforderungen entstehen – sowohl im Hinblick auf die Art der Lösung als auch die Geschwindigkeit der Umsetzung. Die für die Lösung der Anforderungen erforderlichen Eingriffe stellen wiederum eine Veränderung der IT-Architektur dar.

4.4.1.2.2 Technische Anforderungen

Unter technischen Anforderungen sind diejenigen Anforderungen zu verstehen, die nicht primär einen fachlichen Hintergrund haben, sondern sich vielmehr auf die technische Umsetzung der fachlichen Anforderung beziehen. So überrascht es nicht, dass diese vornehmlich Gegenstand der Interviews mit den Experten der Infrastrukturarchitektur sind.

In den Interviews mit den Ansprechpartnern der Anwendungsarchitektur werden in diesem Kontext lediglich Release-Wechsel auf aktuelle Versionspakete (die i. d. R. eine Menge von Anpassungen und Aktualisierungen einer bestehenden Softwarelösung darstellen) oder die Ablösung einer implementierten Softwarelösung, z. B. aufgrund der Einstellung von Software-Updates durch den Hersteller, angeführt.

Im Bereich der Infrastrukturarchitektur wird ebenfalls die Ablösung alter Hardware-Komponenten angesprochen, deren Software durch die Hersteller nicht mehr adäquat betreut und weiterentwickelt wird. Ein angeführtes Beispiel im Bereich der Infrastruktur ist die Ablösung der bestehenden Telefonanlage, da diese moderne Funktionen, wie die Anbindung an die Kontakte des E-Mail-Clients oder der CRM-Lösung, nicht unterstützt.

4.4.2 Einfluss von Anforderungen auf die „minimale IT-Komplexität“

Nachdem im vorangegangenen Abschnitt die Trigger einer Änderung der IT-Architektur untersucht wurden, widmet sich dieser Abschnitt der Untersuchungsfrage 2.2 und beleuchtet, wie die Anforderungen auf die „minimale IT-Komplexität“ wirken. Abbildung 20 verortet die in diesem Abschnitt vorgestellten Ergebnisse im Gesamtkontext der Studie.

⁴¹ In der Literatur wird auch eine zunehmende Kundenorientierung bei der strategischen Ausrichtung thematisiert (Priem et al. 2013).

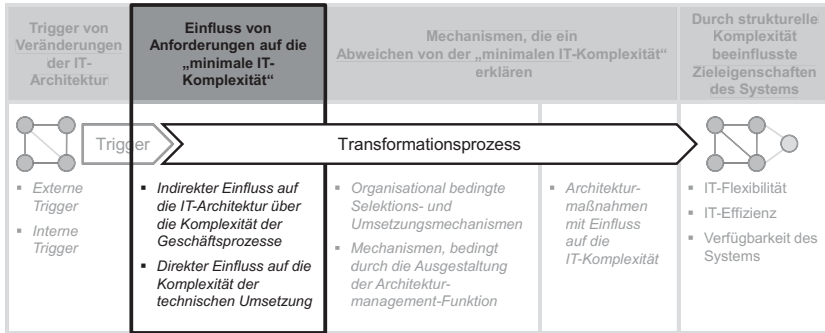


Abbildung 20: Einordnung des Einflusses von Anforderungen auf $K_{ITA,min}$ in den Untersuchungskontext

Anforderungen können hierbei entweder einen direkten Effekt auf die IT-Architektur haben (Unterabschnitt 4.4.2.2) oder über eine Erhöhung der Komplexität der Geschäftsprozesse – als Vorläufer der eigentlichen technischen Umsetzung – auf die IT-Architektur wirken (Unterabschnitt 4.4.2.1; vgl. Abbildung 21). Hierdurch kann die Wirkungsweise der im vorangegangenen Abschnitt eingeführten unterschiedlichen Typen von Triggern weiter ausdifferenziert werden.

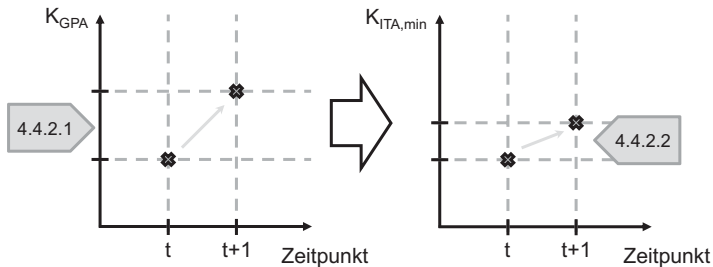


Abbildung 21: Einflüsse auf $K_{ITA,min}$

4.4.2.1 Indirekter Einfluss auf die IT-Architektur über die Komplexität der Geschäftsprozesse

Bei der Analyse der transkribierten Interviews fällt zunächst auf, dass primär Experten aus dem Bereich der Anwendungsarchitektur derartige Einflüsse diskutieren. Dies lässt sich dadurch erklären, dass die Ausgestaltung der Geschäftsprozesse ein Themenfeld ist, das sich inhaltlich näher an dem Aufgabenfeld der Anwendungsarchitektur befindet und es vornehmlich dort Berührungspunkte gibt (vgl. hierzu auch Abschnitt 2.2.4 und Abbildung 6). Die von den Experten angeführten Aspekte umschreiben zum einen eine zunehmende Komplexität der Geschäftsprozesse (Sektion 4.4.2.1.1) aufgrund der eingehenden Anforderungen und zum anderen den Umstand, dass zusätzlich eine standortspezifische Ausgestaltung von Prozessen (Sektion 4.4.2.1.2) den Umfang an IT-seitig umzusetzenden Anforderungen erhöht.

4.4.2.1.1 Komplexität des Objektes „Geschäftsprozess“

Dieser Punkt bezieht sich auf die zunehmende Komplexität einzelner Geschäftsprozesse aufgrund neuer Anforderungen. Hier steht demzufolge die Ausgestaltung der Prozesse an sich im Vordergrund. Das folgende Zitat eines Mitarbeiters aus dem Bereich der Anwendungsarchitektur stellt den Bezug zwischen eingehenden Anforderungen bezüglich der Ausgestaltung der Prozesse und den Folgen für die erforderliche Komplexität der IT-Architektur her:

„Man könnte die Komplexität der Geschäftsprozesse auch als Differenzierung unserer Firma gegenüber der Konkurrenz verstehen. Wir differenzieren uns mit zusätzlichen Funktionen von unseren Konkurrenten. Insofern ist die IT-Komplexität ein Resultat dessen, was beispielsweise vertriebsseitig gefordert wird. Wir als IT können nur bedingt etwas an der hinzukommenden Komplexität ändern.“ (Experte aus der Anwendungsarchitektur)

In diesem Kontext führt der Experte ein Beispiel aus dem angesprochenen Vertriebsumfeld an:

„Der Vertrieb gibt vor, was wir leisten müssen. Wir müssen z. B. bei der Kommissionierung für [Kunde] die Paletten so vorbereiten, dass [Kunde] die Paletten einfach in die einzelnen Filialen reinstellen kann. Die müssen nichts mehr machen. Also landet die Komplexität, für jede Filiale die gewünschte Palette zusammenzustellen, bei uns: Eine Einheit von dem, zwei Einheiten von dem, fünf von dem. Wir sagen nicht, der bekommt einfach eine Palette von [Produkt], sondern wir übernehmen diesen komplexen Vorgang. Hierbei handelt es sich aber um eine Differenzierung von uns gegenüber der Konkurrenz. An der Anforderung an sich können wir an der Stelle also wenig machen. Die Komplexität können wir nur im Bereich der Umsetzung der IT-Prozesse beeinflussen und hier etwas verbessern.“ (Experte aus der Anwendungsarchitektur)

Das folgende Zitat stellt den Zusammenhang zu dem in Sektion 4.4.1.2.1 diskutierten internen Trigger der fachlichen Anforderungen her:

„Komplexe fachliche Anforderungen von unseren Anwendern erfordern auch immer komplexere Applikationen und Systeme.“ (Experte aus der Anwendungsarchitektur)

Neben fachlichen Anforderungen führen auch zusätzliche oder sich ändernde gesetzliche Anforderungen (vgl. Sektion 4.4.1.1.2) zu einer zunehmenden Komplexität der IT-seitig unterstützten Geschäftsprozesse. So berichtet ein Experte von der Anforderung einer landesspezifischen Bedruckung der Produkte, um den jeweiligen gesetzlichen Anforderungen zu genügen:

„Für die Schweiz muss beispielsweise der VOC-Gehalt [volatile organic compounds; Flüchtige organische Verbindungen] ausgewiesen werden. Fehlt diese Angabe, dann darf die Ware nicht eingeführt werden.“ (Experte aus der Anwendungsarchitektur)

Des Weiteren führen die Experten in diesem Kontext Anforderungen aus dem Bereich der Rechnungslegung an, die ebenfalls zu zusätzlicher Komplexität auf Prozessebene führten.

4.4.2.1.2 Heterogenität der Prozesse

In Abgrenzung zur vorherigen Sektion wird in dieser Sektion die Vielfalt bestehender Prozesse behandelt, die sich in den Interviews vorrangig anhand standortspezifischer Gegebenheiten festmachen lässt. Hiermit verändert sich der Fokus von der Ausgestaltung einzelner Prozesse und der durch diese entstehende IT-Komplexität auf eine Menge heterogen ausgestalteter Prozesse, die einem (gleichen) fachlichen Zweck dienen.

Ein in den Interviews angesprochener Bereich sind die Prozesse in der Logistik. Die verschiedenen Standorte der angeschlossenen Gesellschaften haben in diesem Bereich stark voneinander abweichende Vorgehensweisen etabliert, die technisch abgebildet werden müssen:

„Die Standorte haben verschiedene Konzepte, wie sie ihre Logistik abwickeln. In [Standort Dachgesellschaft] haben wir regionale Spediteure. Deutschland ist in Regionen aufgeteilt und anhand der Postleitzahlen ist jeder Region jeweils ein Spediteur zugeordnet. [...] [Gesellschaft A hingegen hat] einen Spediteur für ganz Deutschland. Dieser liefert an die Umschlagdepots und stellt dann zu; und da fängt es schon an. An und für sich ist der Prozess ‚Warenauslieferung an Kunde‘ der gleiche und es wird auch im gleichen SAP System abgebildet. Im Grunde mit den gleichen Logiken dahinter, aber eben dann doch nicht, weil die Prozessabwicklung eine ganz andere ist. Da hat man da schon zwei Varianten. [...] Dann haben wir Standorte wie [Gesellschaft B], die gar nicht zum Kunden ausliefern. Die produzieren und liefern dann woanders hin.“ (Experte aus der Anwendungsarchitektur)

Die unterschiedliche Ausgestaltung von Prozessen kann auch im Rahmen der Zusammenführung von Systemen zu Problemen führen. Im Zuge der Umstellung der ERP-Lösung eines Standortes auf SAP wurden die vorher in der ursprünglichen Lösung dargestellten Prozesse in SAP nachgebildet. Dies war erforderlich, da die Prozesse der Gesellschaft vom Standard abweichen, jedoch nicht ohne Weiteres an diesen anpassbar waren:

„Bei [Gesellschaft A] zum Beispiel wurden einige Prozesse implementiert. Die mussten wir eins zu eins kopieren, [wodurch wir] [...] eine Komplexitätssteigerung im System erfahren haben [...].“ (Experte aus der Anwendungsarchitektur)

Derartige die Komplexität erhöhende Maßnahmen können nachgelagert zu Schwierigkeiten bei der Betreuung, Wartung und Änderung der Systeme führen. Hier berichtet der Experte von den erwarteten Schwierigkeiten bei einem anstehenden Releasewechsel:

„Jetzt haben wir einen Releasewechsel vor uns, [...] auf eine komplett neue Version inklusive einer anderen Datenbankbasis. Da knallt es natürlich an allen Ecken bei der Migration und da muss man im Prinzip noch mal nachprogrammieren.“ (Experte aus der Anwendungsarchitektur)

Der Umfang der unterschiedlichen, in den Systemen abzubildenden Prozesse wird aus dem folgenden Zitat deutlich:

„Wir haben unsere Komplexität, weil wir viele Vertriebsgesellschaften haben. Jeder hat seinen mehr oder weniger eigenen Prozess und das erschwert den Umgang mit dem Sys-

tem und macht es auch relativ intransparent. Anstelle von einem oder fünf Prozessen [...] haben wir 26 in SAP zu betreuen. Ich denke an Preise und Konditionsschemata, die wir eingebaut haben. Da gibt es einen Kollegen, der kennt sich damit aus. Wenn der nicht da ist, dann ist es schon schwer, das Ganze nachzuvollziehen.“ (Experte aus der Anwendungsarchitektur)

Die Umsetzung der unterschiedlichen Prozessvarianten führt dabei aber auch zu weiteren, teilweise unvorhersehbaren Schwierigkeiten:

„Wir hatten Anfang des Jahres eine Anforderung von den schwedischen Kollegen,[...] etwas an der Lagerortsteuerung zu ändern. Das haben wir auch gut hinbekommen und an deren Anforderungen angepasst. Leider hat in der Folge die Lagersteuerung an einem französischen Standort nicht mehr funktioniert, was dann wieder behoben werden musste.“ (Experte aus der Anwendungsarchitektur)

So entsteht durch die Implementierung einer Vielzahl von Prozessvarianten eine Reihe von Abhängigkeiten, die im Laufe der Zeit nur noch sehr schwer nachvollziehbar sind. Diese Tendenzen zeigen sich in diversen Bereichen, wie z. B. bei der Umsetzung individueller Prozesse im Vertrieb (u. a. für die Bearbeitung der Kundenaufträge) oder der Produktion, für die auch die jeweilige Ausstattung der Standorte mit den für die Produktion erforderlichen Maschinen eine Rolle spielt (vgl. Sektion 4.4.2.2.2).

Wie die angeführten Beispiele zeigen, spielt die Heterogenität der Prozesse vor allem im Kontext der standortspezifischen Ausgestaltung der Prozesse eine Rolle. Demnach trägt in erster Linie das in Sektion 4.4.1.1.1 diskutierte anorganische Wachstum des Unternehmens zu einer zunehmenden Prozessvielfalt bei. Darüber hinaus führen auch die landesspezifischen gesetzlichen Gegebenheiten (Sektion 4.4.1.1.2) zu voneinander abweichenden Ausprägungen der Prozesse.

4.4.2.2 Direkter Einfluss auf die Komplexität der IT-Architektur

Unabhängig von der durch eine Anforderung hervorgerufenen Veränderung der Komplexität auf Ebene der Geschäftsprozesse, können sich Anforderungen auch unmittelbar auf die minimale Komplexität der IT-Architektur auswirken. So zeigen die Experteninterviews, dass sowohl die zunehmende Automatisierung von Abläufen als auch standortspezifische Anforderungen an die technische Umsetzung einen Einfluss auf die minimale Komplexität der IT-Architektur haben.

4.4.2.2.1 Automatisierung

Unter Automatisierung wird die IT-Unterstützung von Arbeitsschritten verstanden, die alternativ durch einen Mitarbeiter manuell durchgeführt werden müssten. Eine zunehmende Automatisierung ändert demnach nicht zwangsweise den grundlegenden Ablauf eines Geschäftsprozesses, sondern sie verändert (erhöht) den Grad der IT-Unterstützung. Dies hat zur Folge, dass es – bei grundsätzlich unveränderter Komplexität der Geschäftsprozessarchitektur – zu einem Anstieg der erforderlichen Komplexität der IT-Architektur kommt. Die Experten

beobachten die klare Tendenz, dass Abläufe der Geschäftsprozesse zunehmend automatisiert werden sollen:

„Letztendlich werden die Prozesse mehr und mehr [in den Systemen] abgebildet. Es wird immer mehr automatisiert. Das ist ganz normal. [...] Die Anforderungen, welche die IT erfüllen kann, werden durch die Möglichkeiten immer größer und das bedingt zwangsweise, dass Sachen automatisiert werden, die vorher nicht denkbar gewesen waren, die aber jetzt notwendig sind. Auch um konkurrenzfähig zu bleiben.“ (Experte aus der Anwendungsarchitektur)

Gerade im Kontext der in Unterabschnitt 4.4.2.1 diskutierten Vielfalt an Prozessvarianten kann eine zunehmende Automatisierung zu einem Anstieg der erforderlichen Komplexität der IT-Architektur führen, da für die diversen Prozessvarianten auch unterschiedliche Ausprägungen der Automatisierung oder zumindest die Behandlung prozessspezifischer Ausnahmen berücksichtigt werden müssen.

4.4.2.2.2 *Standortspezifische Anforderungen an die technische Umsetzung*

Im Gegensatz zu den oben beschriebenen fachlichen Aspekten steht hier die technische Perspektive im Mittelpunkt der Betrachtung. Hintergrund ist dabei, dass der individuelle Technologiebedarf⁴² der einzelnen Standorte bei der Planung der Umsetzung von Anforderungen miteinzubeziehen ist. So berichten die Experten, dass die Auswahl der technischen Lösungen häufig nach den Anforderungen der jeweiligen Standorte erfolgt. Beispielsweise wird an einem relativ kleinen Standort mit überschaubarem IT-Umfang nicht die gleiche Switch-Lösung für das Netzwerk angeschafft wie an größeren Standorten. An dieser Stelle entscheidet man sich aufgrund der hohen Anschaffungskosten der Standardlösung für eine günstigere Alternative. Durch das Abweichen von der Standardlösung entsteht damit zusätzliche Heterogenität im Hinblick auf die Technologien und somit Komplexität.

Gleichzeitig beinhaltet dieser direkte Einfluss einen Aspekt, der die Auswirkungen der indirekten Einflüsse auf die entstehende Komplexität zusätzlich verstärkt („History Matters“). So spielt bei neuen Anforderungen die aktuell vorhandene technische Umgebung der jeweiligen Standorte eine wesentliche Rolle für die Wahl möglicher Lösungsalternativen. Gerade im Bereich der Produktion weisen die Standorte – trotz vergleichbarer fachlicher Funktionen – eine sehr große Heterogenität an Maschinen und Herstellern auf. Die Maschinen der jeweiligen Hersteller verfügen meist über proprietäre Systemsoftware. Dies führt dazu, dass die Umsetzung neuer Anforderungen, auch wenn diese grundsätzlich für alle Standorte ähnlich ausgeprägt sind, deutlich voneinander abweichen:

„Es gibt vielleicht nur eine Möglichkeit einen Fertigungsauftrag im SAP zu erfassen. [...] Da haben wir nicht viele Varianten etabliert, aber die Komplexität kommt erst nachgelagert. Die Frage ist nun, wie produziere ich den Fertigungsauftrag? Gebe ich ihn an eine [...] -Anlage, dann bekommt diese den Auftrag in einer bestimmten Struktur und arbeitet

⁴² Dieser direkte Einfluss auf die Komplexität der IT-Architektur ist eng mit dem in Sektion 4.4.1.2.2 eingeführten Trigger der technischen Anforderungen verknüpft.

ihn ab. Übergebe ich den Auftrag an einen Turm in [Standort Dachgesellschaft], dann habe ich eine ganz andere Vorgehensweise [...] und auch andere Rückmeldungen, die ich nachher bekomme. Oder ich gebe ihn nach [Gesellschaft D], dort habe ich ein [System], das die Daten noch mal ganz anders erwartet und zurückmeldet.“ (Experte aus der Anwendungsarchitektur)

Bei der Implementierung zusätzlicher fachlicher Anforderungen in den mit diesen Anlagen verknüpften Systemen müssen die technischen Anforderungen der bestehenden Anlagen entsprechend berücksichtigt werden. Wie aus dem vorangegangenen Zitat ersichtlich, führt dies zu einer Vielzahl unterschiedlicher Schnittstellen und Datenstrukturen.

4.4.3 Mechanismen, die ein Abweichen von der „minimalen IT-Komplexität“ erklären

In Abschnitt 4.2 wurde diskutiert, dass im Zuge des Transformationsprozesses der IT-Architektur Abweichungen von der „minimalen IT-Komplexität“ entstehen können, wodurch sich ein gewisser Grad „unnötiger IT-Komplexität“ einstellt. In diesem Abschnitt werden die hierbei zugrunde liegenden Mechanismen untersucht, aufgrund derer bei der Umsetzung der Anforderungen derartige Abweichungen auftreten (Untersuchungsfrage 2.3). Diese werden in organisational bedingte Selektions- und Umsetzungsmechanismen (Unterschnitt 4.4.3.1) und durch die Ausgestaltung der Architekturmanagement-Funktion bedingte Mechanismen (Unterabschnitt 4.4.3.2) untergliedert. Abbildung 22 stellt dar, wie sich die in den folgenden beiden Unterabschnitten vorgestellten Ergebnisse in den Gesamtkontext der Studie eingliedern.

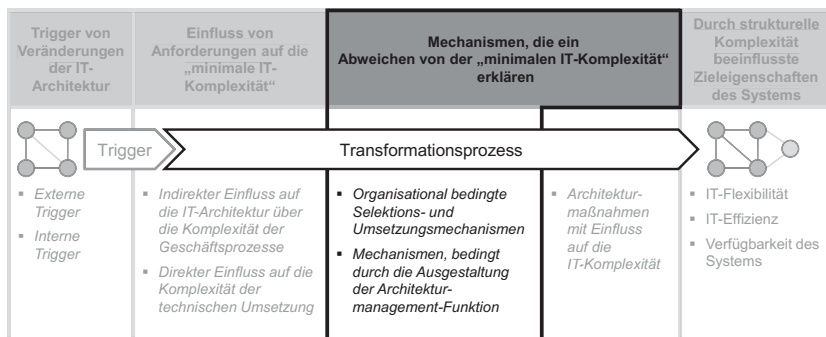


Abbildung 22: Einordnung der Mechanismen – die $\Delta K_{IT,unnötig}$ erklären – in den Untersuchungskontext

In Unterabschnitt 4.4.3.3 werden anschließend die im Rahmen der Auswertung der Interviews identifizierten Architekturmaßnahmen (wie z. B. Standardisierung) dargestellt. Hierbei zeigte sich, dass die Intensität der Anwendung der Architekturmaßnahmen von den im jeweiligen Projektkontext vorkommenden Mechanismen abhängt.

4.4.3.1 Organisational bedingte Selektions- und Umsetzungsmechanismen

Im Bereich der organisational bedingten Selektions- und Umsetzungsmechanismen werden Mechanismen betrachtet, die aufgrund der organisatorischen Einordnung der IT-Funktion

einen Einfluss auf die Auswahl (Selektion) möglicher Lösungsalternativen haben und die Implementierung der Anforderungen in der IT-Architektur (Umsetzung) der UG betreffen. In den Experteninterviews konnten mit Selektions- und Umsetzungsmechanismen aufgrund der teilweise etablierten dezentralen Organisation der IT sowie der Eigenmächtigkeit der Fachabteilungen bei IT-Entscheidungen zwei unterschiedliche Mechanismen identifiziert werden, die ein Abweichen von dem Mindestmaß an IT-Komplexität erklären können.

4.4.3.1.1 Selektions- und Umsetzungsmechanismen durch die dezentrale Organisation der IT

Der Definition von Brown & Magill (1998) folgend handelt es sich um eine zentralisierte IT-Funktion, wenn die Entscheidungskompetenz bzgl. der Ausgestaltung der IT primär im Verantwortungsbereich der zentralen IT-Abteilung des Unternehmens liegt. Hingegen liegt diese Entscheidungskompetenz im Falle einer dezentralen IT-Funktion primär bei dem Management der einzelnen Geschäftsbereiche.

Hinsichtlich der gesamten UG liegt eine kombinierte IT-Governance-Struktur vor, bestehend aus einem zentral organisierten IT-Bereich – aufgehängt in der Dachgesellschaft – sowie einer feudalen Struktur (Weill & Ross 2004), die den angeschlossenen Gesellschaften das Führen eigener IT-Abteilungen grundsätzlich ermöglicht:

„Wir haben quasi eine Mischform was Zentralisierung angeht. Wir haben etliche zentralisierte Systeme [...]. Das ERP- bzw. das gesamte SAP-System ist zentralisiert. Da sind die Standorte relativ eingeschränkt in dem, was sie tun können. Gleichzeitig haben wir an größeren Standorten doch auch relativ große dezentrale Strukturen, da es an den Standorten [...] IT-Teams, Abteilungen oder einzelne Personen gibt, die gewisse Dinge eigenständig tun können, dürfen und zum Teil auch sollen.“ (Experte aus der Anwendungsarchitektur)

Bei den Systemen, die dezentral von den Standorten verantwortet werden, handelt es sich vornehmlich um Systeme für die Lager- oder Produktionssteuerung, für deren Ausgestaltung und Pflege die individuellen Gegebenheiten der Standorte eine große Rolle spielen. Diese Eigenständigkeit führte in der Vergangenheit auch dazu, dass die Standorte nicht nur ein eigenes System betrieben, sondern auch eine eigene Wahl hinsichtlich des Produktes (Hersteller, Version etc.) getroffen haben:

„Hier führen die dezentralen Strukturen dazu, dass man [...] ein dezentrales System [...] nicht nur verwendet, sondern sich auch ein eigenes aussucht.“ (Experte aus der Anwendungsarchitektur)

Diese Selbstständigkeit dezentraler Einheiten hat zur Folge, dass die zentrale IT bei dem Management der Entwicklung der IT-Komplexität teilweise relativ eingeschränkt ist:

„Die organisatorischen Strukturen erhöhen auch die Komplexität. Wir haben hier einen Konzern, der aus verschiedenen Standorten und verschiedenen Landesgesellschaften und teilweise sehr selbstständigen Bereichen besteht, die sehr autonom agieren können. Hierdurch entsteht Komplexität, die schlecht von uns direkt zu verwalten ist.“ (Experte aus der Anwendungsarchitektur)

In diesem Kontext werden von den Experten auch die wesentlichen Reibungspunkte eines Architekturmanagements mit der teilweise feudalen Struktur ausgemacht. Lösungswünsche werden teils umgesetzt, auch wenn sie aus einer zentralen Perspektive sowohl in technischer als auch in fachlicher Hinsicht nicht immer sinnvoll zu sein scheinen. Die hierbei getroffenen Entscheidungen der Einführung einer von bestehenden Lösungen abweichenden Technologie führen dabei häufig zu einer Abweichung von dem Mindestmaß an IT-Komplexität.

Als einen möglichen Ansatz, diesen Punkten entgegenzuwirken, wird u. a. die Verbesserung der Kommunikation zwischen der zentralen und der dezentralen IT angesehen:⁴³

„Man muss sich miteinander austauschen und Projekte und Neueinführungen müssen vorher abgestimmt werden. Hier muss mehr Kommunikation stattfinden. Und wir benötigen eine Instanz, die eine Eskalation verhindert und verbindliche Entscheidungen treffen kann.“ (Experte aus der Anwendungsarchitektur)

Dabei betont der Experte, dass es nicht das Ziel der zentralen IT ist, alles zu vereinheitlichen, sondern dass die jeweiligen Anforderungen der dezentralen Standorte weiterhin Berücksichtigung finden müssen:

„Eine diktatorische Lösung – „Hurra, SAP für alle! Wir schmeißen Axapta, Navision und alles andere raus!“ – funktioniert auch nicht. Denn dann vergessen wir die landesspezifischen oder vertriebsgesellschaftstypischen Anforderungen.“ (Experte aus der Anwendungsarchitektur)

Es existiert demnach ein Spannungsfeld aus einer erforderlichen Individualität der einzelnen Standorte und der Erreichung von Synergien durch eine weniger heterogene IT-Architektur. Für die dezentralen Standorte stellen die bestehenden Freiheiten bzgl. der Auswahl und Umsetzung einer Lösung zudem eine Form der Flexibilität dar, da diese ohne umfangreiche Abstimmungsprozesse eingeführt werden kann:

„Man hat natürlich eine gewisse Flexibilität, wenn ein Standort [sich] dezentral für eine Anwendung entscheiden kann und dann auch in der Lage ist, diese einzuführen. Das hat den Vorteil, dass man im Zweifel schneller ist, als wenn zunächst überprüft werden muss, wer das alles noch gebrauchen könnte oder ob es vielleicht bereits eine Lösung gibt. Das kostet natürlich alles Zeit. Die kann man sparen, das geht schneller. Das ist aus Sicht des Anfordernden die individuellere Lösung, die auch besser auf die eigene Problemstellung passt.“ (Experte aus der Anwendungsarchitektur)

Ein in diesem Kontext durch einen Experten angeführtes Beispiel beschreibt die Entwicklung einer mobilen Applikation durch eine der angeschlossenen Gesellschaften. Diese Applikation ermöglicht es den Außendienstmitarbeitern der Gesellschaft, unterwegs auf relevante Kundeninformationen zugreifen zu können. Zentral hätte dies nach Aussage des Experten zu lange gedauert, also erfolgte eine dezentrale Umsetzung, auch wenn eine solche Lösung ebenfalls für die Außendienstmitarbeiter anderer Bereiche der UG hätte interessant sein können.

⁴³ Eine Architekturmanagement-Funktion stellt eine mögliche Instanz dar, die eine solche Kommunikation sicherstellen kann (vgl. Unterabschnitt 4.4.3.2).

Wie ein Experte anführt, entwickeln und verschärfen sich die mit der Einführung von Insellösungen verbundenen Schwierigkeiten erst im Laufe der Zeit. Durch die Integration der Lösung in die IT-Architektur und die Implementierung der dafür erforderlichen Schnittstellen wächst die Komplexität der IT-Architektur zunehmend an:

„Inseln bleiben selten Inseln. Irgendjemand baut in der Regel Brücken. Ich kenne jetzt kein System, bei dem man mal gesagt hat, [...] das steht einfach da, das kann man benutzen. Das fängt bei den Daten an, seien es Kundendaten oder Materialdaten. Irgendwelche Zahlen oder Daten benötigt man eigentlich immer und die kommen meist aus einem anderen System. [...] [Hierbei kann sich herausstellen, dass] im Zweifelsfall die Entscheidung für ein System vielleicht auch anders getroffen worden wäre, hätte man sich frühzeitig Gedanken über das System gemacht. [...] Deshalb glaube ich auch, dass es nur gefühlte Vorteile sind, und diese ganz ganz schnell – auch nicht innerhalb von Jahren, sondern eher innerhalb von Monaten – von der Realität eingeholt werden [...].“ (Experte aus der Anwendungsarchitektur)

4.4.3.1.2 Selektions- und Umsetzungsmechanismen durch die Eigenmächtigkeit der Fachabteilungen bei IT-Entscheidungen

Neben der in der vorangegangenen Sektion beschriebenen Verteilung der Entscheidungskompetenzen auf die zentrale IT der Dachgesellschaft und die dezentral organisierten IT-Abteilungen angeschlossener Gesellschaften ist noch ein weiterer Aspekt zu beobachten. So greifen auch Fachabteilungen in die IT-Architektur ein, indem diese teils eigenmächtig über die Anschaffung einer spezifischen Lösung entscheiden und diese Lösung unter Umständen sogar bereits anschaffen, bevor die IT eingebunden wurde.⁴⁴ In dieser Situation bleibt der IT-Funktion im Zweifel keine andere Möglichkeit, als diese Anschaffung in die Landschaft zu integrieren:

„Da wird manchmal eine proprietäre Software eingeführt. Den Fall haben wir jetzt gerade in [Fachabteilung A]. Die haben Update 7 ausgerollt. Das haben wir denen implementiert, ganz nach Gusto der damaligen Leitung. Der neue Leiter sagt, mit dem System kann er nichts anfangen und hat jetzt eine neue Software gekauft. Nun ruft er mich an, er braucht Schnittstellen zum CRM und zu SAP. Und das bringt natürlich Freude.“ (Experte aus der Anwendungsarchitektur)

Der Auslöser dieser Anschaffung lag dabei darin, dass der Leiter Funktionalitäten bei der bestehenden Lösung vermisste. Ähnliches berichten weitere Experten aus der Anwendungs- und der Infrastrukturarchitektur. Hier wurde zum einen eine Anwendung angeschafft, um ein Be-

⁴⁴ Hierbei soll die beobachtete Eigenmächtigkeit der Fachabteilungen bei IT-Entscheidungen von dem in der Literatur etablierten Begriff der Schatten-IT abgegrenzt werden. So handelt es sich bei einer sogenannten Schatten-IT um IT-Systeme, die durch die Fachabteilungen eigenständig angeschafft/entwickelt und betrieben werden. Eine wesentliche Problematik der Schatten-IT liegt in der fehlenden Integration der Systeme in die zentralen Governance-Strukturen (Fürstenau & Rothe 2014). Im Gegensatz hierzu ist der Betrieb der IT-Systeme durch die Fachabteilungen bei den in dieser Sektion betrachteten Selektions- und Umsetzungsmechanismen keine erforderliche Voraussetzung.

dürfnis nach zusätzlicher Funktionalität zu bedienen und zum anderen neue Infrastruktur-Komponenten in Form von Tablets, um das Thema Mobility voranzutreiben:

„Wir haben durchaus das ein oder andere Mal die Situation, dass die Fachabteilung kommt und sagt, wir haben da eine Anwendung – bitte anbinden!“ (Experte aus der Anwendungsarchitektur)

„Mit Hochdruck wurde das Thema iPads vorangetrieben. [...] Das wurde stark forciert und die IT zunächst gar nicht gefragt. Das heißt, als die IT involviert wurde, waren die ersten iPads schon gekauft. Dann hieß es nur noch: Integriert die bitte!“ (Experte aus der Infrastrukturarchitektur).

Die hier beschriebenen Selektions- und Umsetzungsmechanismen führen durch eine Abweichung von zentralen, standardisierten Systemen zu einer höheren Anzahl und Heterogenität der Komponenten sowie der Schnittstellen. Im letzten Beispiel wurde im Nachgang erkannt, dass die angedachte mobile Lösung nicht in die Microsoft-lastige IT-Architektur passt und daraufhin die Erarbeitung einer Windows-basierten Lösung veranlasst.

4.4.3.2 Mechanismen, bedingt durch die Ausgestaltung der Architekturmanagement-Funktion

Aus dem folgenden Zitat aus dem Gespräch mit einem Experten der Anwendungsarchitektur werden die drei in diesem Abschnitt diskutierten Aspekte des Architekturmanagements deutlich:

„Das Leben mit der Komplexität oder der Umgang mit ihr ist, glaube ich, das Wichtige. Und da ist tatsächlich Dokumentation aus meiner Sicht ein Aspekt, der ausgebaut werden müsste. Und dann eben auch der angesprochene Prozess für das Leben und den Umgang mit der Komplexität. Eine Struktur zu definieren, die berücksichtigt, wie mit Anforderungen umgegangen wird, die die Architektur oder größere Themen betreffen.“ (Experte aus der Anwendungsarchitektur)

Die in diesem Zitat angesprochene Notwendigkeit, Vorgehensweisen für das „Leben und Arbeiten“ mit der Komplexität der IT-Architektur zu etablieren, ist ein deutlicher Hinweis auf das Erfordernis einer Architekturmanagement-Funktion (vgl. hierzu auch Abschnitt 2.2.5 und Niemann (2005)). Weiter werden hier explizit die Dokumentation der IT-Architektur sowie ein strukturiertes Vorgehen bei Anforderungen mit Einfluss auf die Architektur angesprochen. Eben diese drei Aspekte werden im Folgenden aufgegriffen und diskutiert, inwiefern es hierdurch zu einer Abweichung von dem Mindestmaß an IT-Komplexität kommt.

4.4.3.2.1 Architekturmanagement-Funktion mit Management-Support

Grundgedanke des im Folgenden dargestellten Mechanismus ist, dass es im Unternehmen einer dedizierten Funktion bedarf, in deren Aufgabenbereich das zielgerichtete Management der Komplexität der IT-Architektur fällt (vgl. Abschnitt 2.2.6) und die ausreichend Unterstützung durch das Management erfährt. Das Fehlen einer solchen einflussreichen Funktion kann dazu führen, dass es bei architekturrelevanten Entscheidungen zu einer Abweichung von dem

Mindestmaß an IT-Komplexität kommt. Zum Zeitpunkt der Durchführung der Studie war in der UG noch keine Architekturmanagement-Funktion etabliert. Nach Beendigung der Studie wurde eine solche jedoch im Rahmen eines IT-Strategie-Projektes eingeführt.

In der UG existierte grundsätzlich bereits das Bewusstsein, dass die Komplexität der IT-Architektur ein relevantes Thema darstellt, welches bei zukünftigen Entscheidungen Berücksichtigung finden sollte. Um IT-seitig notwendige Maßnahmen durchzuführen, fehlte es bislang an Vorgaben der Geschäftsleitung, in welche Richtung sich die IT-Architektur der UG entwickeln soll. Solche sind jedoch erforderlich, um die IT in die Lage zu versetzen, sich bei Fragen zur Ausgestaltung der IT-Architektur gegen den Fachbereich durchsetzen zu können:

„Natürlichen ist auch der Geschäftsleitung mittlerweile bewusst, dass es nicht sinnvoll sein kann, die Komplexität weiter zu erhöhen und dass wir zumindest maximal auf dem Level bleiben sollten, auf dem wir gerade sind. Wir haben uns auch in der IT wirklich sehr bemüht, die Komplexität nicht weiter zu erhöhen. Wir versuchen, sie abzubauen. Wir haben damit aber [...] Schwierigkeiten, [...] weil sich die Fachseite dadurch potenziell eingeschränkt sieht. Und der klare Auftrag, jetzt zwingend zu standardisieren, zu harmonisieren und die Komplexität damit gezielt abzubauen, liegt in der Form nicht vor.“ (Experte aus der Anwendungsarchitektur)

Hieraus wird deutlich, dass eine Architekturmanagement-Funktion, die keinen unmittelbar für die Fachbereiche sichtbaren Mehrwert schafft, die Rückendeckung der Geschäftsleitung benötigt. Diese Rückendeckung ist ebenfalls im Zuge des Umgangs mit neuen Anforderungen relevant, damit die Wirksamkeit von IT-seitigen Architekturentscheidungen sichergestellt werden kann. Dieser Punkt wird durch einen weiteren Experten der Anwendungsarchitektur betont:

„Wir haben bereits versucht, das auf organisatorischer Ebene durch ein ‚Change Advisory Board‘ zu regeln. Neu eingehende Änderungen haben wir zunächst intern besprochen und wenn wir zu dem Ergebnis gekommen sind, dass wir dafür schon etwas haben oder dass eine [gewünschte] Umsetzung evtl. ‚ungeschickt‘ ist, sind diese [Anforderungen] in das Change Advisory Board eingebracht worden. Dort wurden die Anforderungen diskutiert und entschieden, ‚ja gut, machen wir‘ oder eben ‚nein, das ist nicht vernünftig‘. Oft blieb es danach aber nicht bei dem ‚Nein‘, sondern es wurde dann trotzdem gemacht. Und das ist dieser Rückhalt, den ich angesprochen habe, den wir einfach an dieser Stelle brauchen. Wenn die IT-Ebene ‚Nein‘ sagt, dann muss sie entweder mit vernünftigen Argumenten überzeugt werden oder es muss bei dem ‚Nein‘ bleiben.“ (Experte aus der Anwendungsarchitektur)

Im Zuge eines IT-Strategie-Projektes, bei welchem die IT-Architektur einen wesentlichen Teil darstellt, soll dies stärker berücksichtigt werden:

„Ich bin ganz froh, dass es das Strategieprojekt gibt. [...] Weil ich absolut sicher bin, dass wir über das Thema Architektur nicht nur Geld einsparen, sondern die Ressourcen, die wir haben, besser einsetzen könnten – wenn man das Thema Architektur richtig angehen kann.“ (Experte aus der Infrastrukturarchitektur)

Der Experte aus der Infrastrukturarchitektur betont zudem, dass die IT heute mit einer Vielzahl von Aufgaben konfrontiert ist. Ein Architekturmanagement stellt hierbei eine neue Aufgabe dar, die zwar notwendig ist, jedoch nicht ohne Weiteres von der IT übernommen werden kann; nicht zuletzt, da ein Architekturmanagement eigene Werkzeuge benötigt:

„Eine IT kann meiner Meinung nach – also eine moderne IT – einfach nicht mehr so funktionieren, wie noch vor sechs oder acht Jahren. Wir haben die Anforderungen von draußen, alte [...] [Systeme, die] betrieben werden müssen, organisatorische Probleme mit den [...] [angeschlossenen Gesellschaften] und deren Insellösungen, bis hin zu profanen Sachen wie eine funktionierende Dokumentation. [...] Und das führt dazu, dass wir uns anscheinend mit dem Thema Komplexität der Architektur selbst wirklich wehtun. [...] Das Problem ist, dass uns einfach auch die Methoden, die Mittel, Ideen oder vielleicht auch an der einen oder anderen Stelle die Ressourcen fehlen, um das Thema anzugehen.“ (Experte aus der Infrastrukturarchitektur)

Da zum Zeitpunkt der Durchführung der Studie kein dediziertes Architekturmanagement in der UG existierte, gab es entsprechend keine Funktion, die in der Lage gewesen wäre, Veränderungen, welche die strukturelle Komplexität der Architektur weiter erhöhen, zu unterbinden. So berichten die Experten von diversen Sonderlösungen für einzelne Bereiche, bei deren Umsetzung die Nutzbarkeit der Lösung für andere Bereiche keine wesentliche Rolle spielte.

4.4.3.2.2 Architekturdokumentation

Wie in Unterabschnitt 2.2.5.1 angeführt, stellt die Modellierung – und somit die Dokumentation – der Architektur eine elementare Facette des Architekturmanagements dar, was sich von den Wurzeln des Architekturmanagements (Zachman 1987) bis zu den heute etablierten Architekturrahmenwerken (The Open Group 2011) nicht geändert hat. So ist die Dokumentation der aktuellen Unternehmens-/IT-Architektur der Ausgangspunkt sowie wesentliches Werkzeug für die Transformation hin zu einer Soll-Architektur (Aier et al. 2012). Auch während der Auswertung der Expertengespräche stellte sich die Architekturdokumentation als ein wesentlicher Faktor des Architekturmanagements heraus, wobei eine unzureichende Dokumentation der Architektur zu suboptimalen Lösungen und somit zu einer Abweichung von dem Mindestmaß struktureller Komplexität der IT-Architektur führen kann.

Im Bereich der Anwendungsarchitektur des zentralen IT-Bereiches liegt bereits eine relativ gute Dokumentation vor, die bei anstehenden Änderungen an der IT-Architektur jedoch nicht immer zu Hilfe genommen wird, wodurch die Gefahr einer Abweichung vom Mindestmaß an Komplexität besteht:

„Ich glaube, das Problem ist nicht unbedingt, dass man nicht weiß, was man hat. Allerdings wird bei neuen Anforderungen [...] eine zentrale Prüfung, ob woanders bereits eine Lösung existiert, oder ob eine neue Lösung auch gemeinsam genutzt werden kann, eine solche Prüfung eher vernachlässigt. [...] Wir schauen schon manchmal, was die anderen so gemacht haben. Das geht ja dann auch deutlich schneller, wenn man ein System, das man schon hat, auf eine breitere Nutzerbasis stellt. Das ist häufig deutlich

schneller, als ein neues System auszusuchen und alle Schnittstellen neu implementieren zu müssen.“ (Experte der Anwendungsarchitektur)

In der Infrastrukturarchitektur und über die Grenzen der einzelnen Gesellschaften hinweg, ist die Dokumentation der IT-Architektur hingegen deutlich schwächer ausgeprägt. Als ein grundlegendes Problem führt ein Experte aus der Infrastrukturarchitektur an, dass in diesem Bereich nicht vollständig transparent ist, welche IT-Komponenten an den einzelnen Standorten existieren und welche Projekte mit IT-Bezug an den einzelnen Standorten der angeschlossenen Gesellschaften durchgeführt werden. Dies kann dazu führen, dass Lösungen implementiert werden, die nicht mit der bereits bestehenden Architektur abgestimmt sind und somit fachliche Redundanzen und eine zusätzliche Heterogenität in der IT-Architektur entstehen:

„Und besonders schwierig ist, dass keiner so genau weiß, was die IT [der angeschlossenen Gesellschaften] gerade macht. Das hört spätestens im Ausland auf. Da wissen wir überhaupt nicht, was da steht [...]. Die [Gesellschaft E] betreibt zum Beispiel schon seit Jahren ein Dropbox-System. Da hätte man sich theoretisch dranhängen können. Ja, wir stellen aber in [Standort Dachgesellschaft] ein eigenes hin. Das ist immer auch das Schwierige, dass die IT in [Standort Dachgesellschaft] manchmal nicht weiß, was die anderen da noch so stehen haben.“ (Experte aus der Infrastrukturarchitektur)

Dieser Punkt soll nun vorübergehend dadurch behoben werden, dass ein Mitarbeiter die laufenden Projekte der einzelnen Standorte mit Schnittpunkten zur IT überwacht. Für den SAP-Bereich ist dies bereits umgesetzt und soll nun auf den gesamten IT-Bereich ausgeweitet werden.

Ein weiterer in den Interviews diskutierter Punkt ist die Tatsache, dass nicht alle zwischen den einzelnen Komponenten der IT-Architektur bestehenden Schnittstellen dokumentiert sind. Die fehlende Transparenz über die im Laufe der Zeit stetig angewachsene Menge an Abhängigkeiten der einzelnen Komponenten führt dabei zu Schwierigkeiten bei der Durchführung von Änderungen:

„Sowohl das Integrieren oder auch das Wiederherausnehmen von Infrastrukturkomponenten oder von Anwendungen ist dann schwierig, wenn ich das so komplex gestalte, dass ich nicht mehr weiß, wer mit wem spricht. Und deshalb war ja auch mal das Erstellen einer IT-Landkarte angedacht, um aufzuzeigen, wer denn mit wem kommuniziert und welche Anwendung was benutzt. [...] Und ich behaupte nicht einmal, dass wir jetzt hyper komplex sind, sondern ich glaube einfach, dass das Problem ist, dass das, was wir haben, nicht überschaubar für uns ist und wir es nicht überschaubar machen.“ (Experte aus der Infrastrukturarchitektur)

Dokumentation ist demnach wesentlich, um Abhängigkeiten und Seiteneffekte bei Änderungen erkennen zu können. Ein Experte aus der Anwendungsarchitektur betont in diesem Kontext die Wichtigkeit von Prozessen, die festlegen, wie mit der Dokumentation umzugehen ist. Dies betrifft dabei sowohl die Erstellung als auch die Verwendung der Dokumentation als Grundlage für die Durchführung von Änderungen:

„Wenn ich also an irgendeiner Stelle etwas verändern möchte, aber gar nicht weiß, was dadurch alles noch passieren könnte... Diese Probleme wird man durch Dokumentation natürlich nicht vollständig lösen können, aber sicherlich kann sie unterstützen. Dazu benötigt man aber auch einen gelebten Prozess, durch den vorgegeben wird, wann und wie diese Dokumentation zu verwenden ist. Es bringt nichts, wenn ich alles dokumentiert habe und wenn eine neue Anwendung oder eine neue Anforderung hinzukommen soll, dann doch einfach – wie gehabt – loslege und wenn ich dann fertig bin, die Dokumentation anpasse. Dann habe ich auch nichts gewonnen.“ (Experte aus der Anwendungsarchitektur)

Die mit einer nicht ausreichenden Dokumentation und der hierdurch fehlenden Transparenz über die bestehende IT-Architektur verbundenen Mechanismen wurden hier bereits angedeutet. Stehen keine Dokumentation und keine Modelle der aktuellen IT-Architektur zur Verfügung, können nur schwer zielgerichtete Entscheidungen zur langfristigen Weiterentwicklung der Gesamtarchitektur getroffen und ein Abweichen von dem Mindestmaß an Komplexität verhindert werden. Hier diskutierte Beispiele der Folgen einer fehlenden derartigen Übersicht sind die Entstehung unnötiger Redundanzen oder die Schwierigkeiten des Umgangs mit einem System mit einer Vielzahl unterschiedlicher Schnittstellen und den damit verbundenen Abhängigkeiten der System-Komponenten.

4.4.3.2.3 Einbettung des Architekturmanagements in die Unternehmensprozesse

In Sektion 4.4.3.2.1 wurde das Erfordernis einer Architekturmanagement-Funktion in einem Unternehmen beschrieben, die zur Ausführung ihrer Aufgaben – der zielgerichteten Fortentwicklung der IT-Architektur – zudem auch über ausreichende Entscheidungskompetenzen verfügen muss. In dieser Sektion wird die Form der organisationalen Einbettung der Architekturmanagement-Funktion auf Ebene der IT-Prozesse als ein weiterer relevanter Mechanismus diskutiert. Grundgedanke ist hierbei, dass nur durch eine adäquate Berücksichtigung dieser Funktion in allen Prozessen, die einen Einfluss auf die Veränderung der strukturellen Komplexität der IT-Architektur haben, ein Abweichen von dem Mindestmaß an IT-Komplexität verhindert werden kann.

Wie die Experten in den Interviews berichten, existiert bereits eine Reihe von Regelprozessen für das Vorgehen bei bestimmten Änderungen an der IT-Architektur. Hierbei handelt es sich um wiederkehrende, vom Umfang überschaubare Maßnahmen, wie beispielsweise die Einführung von Druckern oder anderen Peripheriegeräten. Diese Regelprozesse sollen sicherstellen, dass von bestimmten festgelegten Standards nicht abgewichen wird. Auch bei wesentlich umfangreicheren Eingriffen in die IT-Architektur sind vereinzelt standardisierte Vorgehensweisen etabliert. So erfolgt die Einführung von SAP bei den Großhändlern anhand eines koordinierten Verfahrens, in dem die IT und die Fachabteilungen, wie Materialwirtschaft oder Finanzbuchhaltung, eng zusammenarbeiten.

Wie aus dem folgenden Zitat eines Experten der Anwendungsarchitektur deutlich wird, sollte ein derartiges Vorgehen im Falle wiederkehrender Maßnahmen auch auf weitere Änderungen der IT-Architektur übertragen werden. Dies ermöglicht es, auch beim Ersetzen bestehender

Komponenten eine bewusste Entscheidung in Richtung einer geringeren Heterogenität der IT-Architektur treffen zu können:

„Eigentlich gleicht unsere Systemlandschaft, die wir im Bereich Produktion haben, einer Patchworkdecke. [...] Wo wir hinkommen müssten ist, dass wir sagen: Hier akzeptieren wir die Patchworkdecke. Wenn wir davon aber etwas austauschen, dann machen wir da einen grünen Flicker rein. [...] Ob das Grün jetzt hellgrün, mittelgrün oder dunkelgrün ist, dass können wir dann entscheiden. Dann haben wir aber am Ende – wenn wir mal darauf schauen – eine grüne Decke.“ (Experte aus der Anwendungsarchitektur)

Ein weiterer Experte ergänzt hierzu:

„Also benötigen wir eine Art Toolbox, aus der wir uns bedienen können. Wir nehmen immer das, wenn die Anforderung kommt.“ (Experte aus der Anwendungsarchitektur)

Der Fokus sollte dabei nicht ausschließlich auf Vorgaben bezüglich des Vorgehens im Falle einer Änderung der bestehenden Systemkomponenten der Architektur liegen, sondern es sollte zudem geregelt werden, wie im Falle der Umsetzung neuer Anforderungen vorzugehen ist. Hierzu berichtet ein weiterer Experte aus der Anwendungsarchitektur:

„Bei dem Release ECC 6.0 im SAP wurde eine neue Funktion eingeführt, durch welche die Verwendung von Webservices ermöglicht wurde. Da sind auf die Schnelle viele Services etabliert worden. Im Vorfeld wurde hierfür aber nicht gezielt eine Infrastruktur aufgebaut. Da wurde hier mal was gemacht und da mal was, und dabei sind verschiedene Infrastruktursysteme verwendet worden. Wir sind gerade dabei, das zu konsolidieren, dass dies eine geradlinige Sache wird, weil das Thema ja immer weiter vorangetrieben wird. Also da wäre es sinnvoll, wenn etwas Neues zu etablieren ist, dass man erst mal einen Rundumschlag macht und analysiert, inwiefern ein allgemeines Interesse besteht. So könnte man eine solche Anforderung auf ein solides Fundament stellen [...].“ (Experte aus der Anwendungsarchitektur)

Die Experten aus der Anwendungs- und der Infrastrukturarchitektur betonen weiterhin die Wichtigkeit einer frühzeitigen Einbeziehung der IT in Veränderungsprozesse:

„Das hängt ganz stark davon ab, wann wir eingebunden werden. Es gibt Fälle, da werden wir recht früh eingebunden. Es gibt Fälle, da werden wir recht spät eingebunden. Ich habe gerade das Beispiel [Projektname] genannt, da sind wir jetzt in der Ausschreibungsphase schon dabei. Da können wir darauf hinwirken, dass wir eine SAP-interne Lösung nutzen oder eine Lösung, die wir bereits in [Standort der Dachgesellschaft] einsetzen [...]. Wenn wir spät dazugeholt werden, dann bekommen wir irgendetwas vorge-setzt und dann ist es schwer, das abzulehnen und wieder aus dem System herauszunehmen.“ (Experte aus der Anwendungsarchitektur)

Um eine rechtzeitige Einbeziehung der IT bei derartigen Fragestellungen sicherstellen zu können, ist ein Bewusstsein dafür erforderlich, dass eine neue Anforderung oder Änderung Einfluss auf die IT haben kann und eine diesbezügliche Überprüfung erfolgen sollte:

„Manchmal ist es auch einfach so, dass man nicht darüber nachdenkt, was alles dazu gehört. Wenn ich mir eine Software anschau, dann denke ich [im Fachbereich] vielleicht gar nicht so sehr über den Tellerrand... Dass ein Drucker angesteuert und [...] eine Schnittstelle zu SAP gebaut werden muss, das vielleicht schon noch. Aber gibt es sonst noch Schnittstellen? Der Hersteller sagt mir, das brauche ich an Hardware – das ist ok. Aber wie das verknüpft werden soll... Ja, ich sage mal, darüber muss sich auch ein Einkäufer oder jemand in der Produktion gar keine Gedanken machen, weil das nicht sein Aufgabengebiet ist. Aber das Bewusstsein müssen wir schärfen. Heutzutage gibt's eigentlich gar nichts mehr, was nicht irgendwo mit der IT verflochten ist. Das müsste geschärft werden. Und dass dann einfach früher gefragt wird: Schaut euch das bitte mal an. Was sagt ihr denn dazu? Von einer anderen Perspektive, von einer anderen Sichtweise aus.“
(Experte aus der Infrastrukturarchitektur)

Speziell die Frage nach der rechtzeitigen Einbeziehung der IT im Rahmen von Projekten, die einen Einfluss auf die IT-Architektur haben, beschäftigt die Experten beider Architekturbereiche. So führen die Experten hierzu sowohl Positiv- als auch Negativbeispiele an. Ein Experte aus der Anwendungsarchitektur berichtet, dass die IT bei der Auswahl und Einführung des Content Management Systems für den Internet-Auftritt frühzeitig hinzugezogen und in diesem Zusammenhang auch technische Anforderungen an die verwendeten Server mit der IT abgestimmt wurden. Auf der anderen Seite führt ein weiterer Experte aus der Anwendungsarchitektur als Beispiel einer eigenmächtigen Entscheidung der Fachseite die Anschaffung eines bestimmten CRM-Systems an (vgl. hierzu auch Sektion 4.4.3.1.2). Hier berichtet der Experte, dass die zentrale IT erst involviert wurde, nachdem die – vom etablierten Standard abweichende – Lösung bereits angeschafft war.

Zusammenfassend zeigen die Ergebnisse dieser Sektion, dass die Architekturmanagement-Funktion in einer Vielzahl von Prozessen, die einen Einfluss auf die Veränderung der strukturellen Komplexität der IT-Architektur haben, einbezogen werden sollte. Bei wiederkehrenden, relativ gleichförmigen Anforderungen an die IT, die durch die Verwendung von klar definierten Regelprozessen umgesetzt werden können, kann eine Architekturmanagement-Funktion die bei Änderungen der Architektur heranzuziehenden Technologien in Form von Architektur-Standards vorgeben.

Um eine zielgerichtete Entwicklung der IT-Architektur zu gewährleisten und eine möglichst geringe Abweichung vom Mindestmaß an Komplexität sicherzustellen, kann eine Architekturmanagement-Funktion verbindliche Architekturprinzipien (The Open Group 2011) vorgeben, die bei der Durchführung von Änderungen an bestehenden Komponenten einzuhalten sind. Haben die Änderungen hingegen die Einführung einer neuen Technologie zur Folge, für die noch keine Vorgabe oder Empfehlung seitens der Architekturmanagement-Funktion existiert, sollte das Architekturmanagement frühzeitig hinzugezogen werden, um bei der Einführung mögliche Synergien zu heben.

Weiterhin zeigen die Interviews mehrheitlich, dass das Architekturmanagement in Projekten der Fachbereiche frühzeitig involviert werden sollte, um mögliche Auswirkungen auf die IT-Architektur identifizieren und berücksichtigen zu können. Dies kann in der Praxis beispiels-

weise durch die Integration eines verpflichtenden Architektur-Checks realisiert werden. Im Rahmen eines solchen Architektur-Checks hat die Architekturmanagement-Funktion die Möglichkeit, gemeinsam mit dem IT-Bereich die durch ein Projekt verursachten Änderungen der IT bzw. der IT-Architektur zu beurteilen.

Bleibt eine Berücksichtigung der Architekturmanagement-Funktion in einem oder mehreren dieser Bereiche aus, können durch ungünstige Architekturentscheidungen Abweichungen von dem Mindestmaß an IT-Komplexität entstehen.

4.4.3.3 Architekturmaßnahmen mit Einfluss auf die IT-Komplexität

Bei der Auswertung der Interviewtranskripte wurden mit der Standardisierung und der Migration redundanter Systeme zwei Architekturmaßnahmen identifiziert. Je nachdem, wie stark die diskutierten Mechanismen in einem spezifischen Projektkontext greifen, kommen diese Architekturmaßnahmen mehr oder weniger intensiv zur Anwendung. Betrifft ein Projekt beispielsweise ausschließlich den Entscheidungsbereich der zentralen IT – wodurch z. B. die angeführten organisationalen Selektions- und Umsetzungsmechanismen nicht zu einer Abweichung von dem Mindestmaß an IT-Komplexität führen – so kann die Architekturmaßnahme der Standardisierung potenziell stärker genutzt werden. Schafft ein Fachbereich jedoch ohne vorherige Abstimmung mit der zentralen oder dezentralen IT eine Lösung an, welche die Problemstellung des Fachbereiches möglichst gut berücksichtigt, so wird es eher zu einer Abweichung von etablierten Standards und demnach auch zu einer größeren Abweichung vom Mindestmaß an IT-Komplexität kommen (Mechanismus aus Sektion 4.4.3.1.2). Die Architekturmaßnahmen dienen an dieser Stelle als Erklärungsansatz für den Einfluss der identifizierten Mechanismen auf den vorherrschenden Grad an Komplexität der IT-Architektur. Wie in Abbildung 23 dargestellt, werden sie daher den Mechanismen, die ein Abweichen von $K_{ITA,min}$ erklären, zugeordnet:

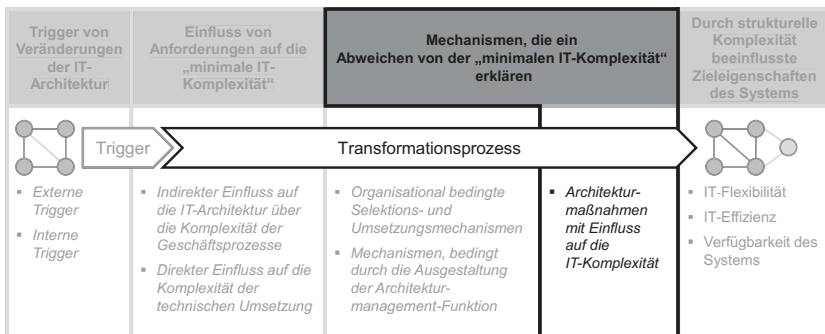


Abbildung 23: Einordnung der Architekturmaßnahmen in den Untersuchungskontext

4.4.3.3.1 Standardisierung

Standardisierung wird sowohl von den Experten aus der Anwendungs- als auch der Infrastrukturarchitektur im Zuge der Interviews als eine der verwendeten Architekturmaßnahmen angeführt. Wie bereits in Sektion 4.4.3.1.1 erwähnt, ist eine Standardisierung in dem untersuchten Unternehmen in erster Linie im Bereich SAP-ERP vorzufinden, wobei sich dies primär auf die in Deutschland ansässigen Gesellschaften bezieht. Über die Landesgrenzen hinweg sind aufgrund landesspezifischer Anforderungen auch bewusst abweichende ERP-Lösungen im Einsatz. Mit der softwareseitigen Standardisierung der ERP-Lösung im Inland wurde in diesem Bereich eine weitgehend standardisierte Infrastruktur etabliert.

„Wenn man die Landschaft im Großhandel vor der Umstellung gesehen hat, die hatten alle unterschiedliche Systeme – unterschiedliche Warenwirtschaftssysteme – und die [Systeme] konnten auch nicht miteinander reden. Wir sind dann hingegangen und haben gesagt, ok, wir machen ein SAP System und eine Infrastruktur.“ (Experte aus der Infrastrukturarchitektur)

Die Großhändler sind heute über ein Citrix-Netzwerk an das zentrale SAP-System der UG angebunden. Die Umstellung auf diese Lösung betraf dabei auch die von den Großhändlern verwendeten Endgeräte. An dieser Stelle waren vor Einführung der Citrix-Lösung PCs im Einsatz, die zwar auch grundsätzlich standardisiert, jedoch zu den unterschiedlichsten Zeitpunkten angeschafft worden waren und somit verschiedene Versionsstände der Betriebssysteme aufwiesen. Da diese Geräte ausschließlich einen SAP-Client und Office-Anwendungen ausführen mussten, war aus der Perspektive der kostenoptimierenden Großhändler nachvollziehbar, dass diese die Systeme nicht regelmäßig durch neue ersetzen wollten. Aus der Perspektive der zentralen IT stellte dies hingegen ein Problem dar:

„Welche Probleme uns dadurch entstehen, das sieht der Großhändler ja nicht. Und man kann es ja auch nachvollziehen. Er will so kostengünstig wie möglich sein. Der sagt, naja, was interessiert mich die IT? Ich verwende das Ding einfach, bis es auseinanderfällt. Aus dem Blickwinkel der Gruppe ist das natürlich fatal, weil dann Geräte mit verschiedenen Software-Update-Programmen auf verschiedensten Versionen von Betriebssystemen gemanagt werden müssen. Das ist natürlich schwieriger, als wenn wir nur ein vorgegebenes Set an Geräten hätten.“ (Experte aus der Infrastrukturarchitektur)

Nach der Einführung der Citrix-Lösung erfolgt der Zugriff auf die zentrale SAP-Instanz nun über vollständig standardisierte physische Citrix-Clients, die wesentlich einfacher durch die zentrale IT zu betreuen sind.

Wie dieses Beispiel zeigt, ist die Standardisierung eine Maßnahme, die grundsätzlich die Heterogenität der eingesetzten Komponenten und Schnittstellen reduziert und demnach einen Effekt auf die strukturelle Komplexität der IT-Architektur hat. Aus den Gesprächen mit den Experten der Infrastrukturarchitektur geht hervor, dass eine starke Tendenz in Richtung einer zunehmenden Standardisierung auf der Infrastrukturebene besteht:

„Ich werde also versuchen – wenn ich es als Schichtenmodell betrachte – mich auf den unteren Ebenen möglichst auf Standards zu einigen; weil die Komplexität in den oberen

Ebenen, der Anwendungsebene und der Prozessebene, so hoch geworden ist, dass ich versuchen muss, das Untere zu standardisieren oder sogar von anderen machen zu lassen.“ (Experte aus der Infrastrukturarchitektur)

Hingegen stellt sich bei den Interviews mit den Experten der Anwendungsarchitektur heraus, dass eine zunehmende Standardisierung auf Ebene der Infrastruktur nicht gleichzeitig zu einer Reduktion der Heterogenität und somit der Komplexität führt. Vielmehr besteht die Gefahr, dass diese lediglich von einer Ebene (der Infrastrukturarchitektur) in die nächste Ebene (die Anwendungsarchitektur) verlagert wird. Hierzu berichtet ein Experte der Anwendungsarchitektur von seinen Erfahrungen aus der Standardisierung der CRM-Lösung:

„Ziel war, deren bestehende Lösung abzulösen. [...] Wenn man dies hinsichtlich der Komplexität der Architektur betrachtet, war das Ziel zu vereinheitlichen und zu harmonisieren. Wenn man sich das Projekt ansieht, dann ist das im ersten Schritt so nicht passiert. Denn im ersten Schritt wurden 1:1 Funktionalitäten integriert, die das alte System hatte. Diese wurden mühsam in das bestehende System dazuprogrammiert. Auf technischer Seite hat man also nun ein System weniger, auf fachlicher Seite hat man im Grunde immer noch zwei Systeme.“ (Experte aus der Anwendungsarchitektur)

Auf der einen Seite konnte durch die Standardisierung der CRM-Lösung die Heterogenität der CRM-Anwendungen und der Schnittstellen reduziert werden, auf der anderen Seite ist die den etablierten Prozessen zugrunde liegende Komplexität nun in den CRM-Lösungen implementiert. Für die Experten der Infrastrukturarchitektur bedeutet dies einen Wegfall der Infrastruktur-Komponenten und -Schnittstellen der abgelösten CRM-Anwendung und dementsprechend eine Reduktion der Komplexität der Infrastrukturarchitektur. Für die Anwendungsarchitektur hingegen ist die Komplexität der fachlichen Prozesse, die zuvor von mehreren Anwendungen abgedeckt wurden, in eine Anwendung verlagert worden. Dies hätte verhindert werden können, wenn zunächst eine Vereinheitlichung der Prozesse stattgefunden hätte. Ein weiterer Experte aus der Anwendungsarchitektur greift diesen Punkt auf und hält fest, dass bereits erste Anstrengungen unternommen werden, Prozesse möglichst abzustimmen und zu vereinheitlichen:

„Es gibt organisatorische Versuche zu harmonisieren, die nennen sich dann auch wirklich Harmonisierungsarbeitskreise – gebildet aus verschiedenen Standorten – in denen die Prozesse diskutiert werden [...] wenn neue dazu kommen oder auch um bestehende Prozesse zu vereinheitlichen. Also auch organisatorisch versucht man bereits bezüglich der Prozesse eine gewisse Vereinheitlichung hinzubekommen.“ (Experte aus der Anwendungsarchitektur)

4.4.3.3.2 Migration

Wie die Standardisierung stellt auch die Migration eine wesentliche Maßnahme zur Veränderung der IT-Architektur dar. Eine Migration in der IT stellt einen Umzug von Daten von einer bestehenden zu einer alternativen Anwendung oder auch ganzer Anwendungen von einer bestehenden auf eine alternative technische Plattform dar. Zielsetzung ist hierbei häufig die Ab-

lösung einzelner Komponenten (beispielsweise aufgrund einer Einstellung der herstellerseitigen Unterstützung), die Umsetzung zusätzlicher fachlicher oder technischer Anforderungen (die nicht mehr durch die bestehende Lösung bedient werden können) oder die Durchführung einer gezielten Konsolidierung der IT-Landschaft (Bisbal et al. 1997).

Nach dem in Sektion 4.4.3.3.1 herangezogenen Verständnis von „Standardisierung“ besteht diese aus mindestens einer Migration: Durch eine Standardisierungsmaßnahme wird in der Regel mindestens ein bestehendes System durch ein – als Standard definiertes – anderes System ersetzt. Im Falle der Standardisierung einer Anwendung erfolgt dabei häufig auch eine Migration der Anwendungsdaten sowie der dort implementierten Prozesse auf das Zielsystem. Eine Migration erfordert aber nicht zwingend eine Standardisierung. Handelt es sich bei dem (beispielsweise die Daten und Prozesse) aufnehmenden System nicht um ein als Standard bestimmtes System, so findet während der Migration entsprechend keine Standardisierung statt.

In den Experteninterviews wird das Thema Migration in zwei Kontexten diskutiert. Zum einen spielt Migration im Rahmen der Zusammenführung fachlich und technisch identischer Lösungen auf eine gemeinsame technische Plattform eine wesentliche Rolle. Hierzu berichten die Experten aus beiden Architekturbereichen von einer angeschlossenen Gesellschaft, die eine Infrastruktur für u. a. einen eigenen Exchange Server vorhalten. Da eine Migration der E-Mail-Konten auf einen gemeinsam genutzten Exchange Server bislang ausblieb, entsteht sowohl Mehraufwand bezüglich des Betriebes der Server als auch bei Änderungen an den Systemen, die Daten mit den Exchange Servern austauschen.

Zum anderen spielt die Migration – wie bereits beschrieben – im Zuge der Standardisierung eine wichtige Rolle. Hier berichtet ein Experte aus der Anwendungsarchitektur von einer erfolgreichen Migration eines vormals eigenständigen Produktionssystems in die SAP-Landschaft:

„Als ich hier in der Firma angefangen habe, gab es in der Produktion [...] ein separates Produktionssystem. Das haben wir im Laufe des letzten Jahrzehntes in SAP überführt. Und da hat man diese zwei Landschaften in eine SAP-Welt zusammengeführt. Dabei ist die Komplexität der Systemlandschaft vermindert worden.“ (Experte aus der Anwendungsarchitektur)

Wie dieses Zitat zeigt, kann es durch eine Migration zu einer Verringerung der strukturellen Komplexität der IT-Architektur kommen. Handelt es sich bei dem Zielsystem einer Migration um eine im Unternehmen vorherrschende Lösung oder entfällt durch eine Migration sogar eine Lösung (eines bestimmten Herstellers) vollständig aus der IT-Architektur, so reduziert sich die Heterogenität (im Hinblick auf die Hersteller) und folglich auch die strukturelle Komplexität der IT-Architektur.

Ein weiteres von einem Experten der Anwendungsarchitektur erläutertes Beispiel greift die bereits in Sektion 4.4.3.3.1 angeführte Standardisierung der CRM-Lösung auf. Hierbei handelt es sich sowohl um eine Standardisierung auf Ebene der verwendeten Lösung als auch um eine Migration der vormals in der abgelösten CRM-Lösung implementierten Prozesse und

Daten. Begründet wurde diese Migrationsentscheidung mit einer unzureichenden Weiterentwicklung des abgelösten Produktes durch den Hersteller sowie die, durch eine verteilte Kunden-Datenbasis entstehenden, Schwierigkeiten für den Vertrieb bei der Kundenansprache.

„Die Notwendigkeit, [...] [die bestehende Lösung abzulösen, bestand darin,] dass das System relativ alt war und von Seiten des Herstellers keine wirklichen Neuerungen mehr kamen. Man musste also auf kurz oder lang weg davon. Dann war aber auch ein sehr wichtiges Argument von der Vertriebsseite [...], dass sie eigentlich alle unsere Marken in einem System haben wollen, um auf lange Sicht zusammenarbeiten zu können. Wenn ich die Daten im gleichen System sehe, habe ich ja die Chance zu erkennen, ob ich Dinge doppelt mache. Sonst kommt es eben vor, dass ein Kunde von drei Marken dreimal besucht wird und andere Kunden dafür gar nicht. [...] [Die Beweggründe der Ablösung] waren also tatsächlich sowohl der fachliche Aspekt, als auch der technische, ein bestehendes System, das nicht mehr zukunftsfähig war, abzulösen.“ (Experte der Anwendungsarchitektur)

Die Migration eines Systems auf eine als Standard festgelegte Lösung kann auf verschiedenen Ebenen zu einer Reduktion der strukturellen Komplexität der IT-Architektur führen. In dem hier geschilderten Fall entfällt durch die Migration eine Anwendung, was zu einer Verringerung der Anzahl und Heterogenität der Komponenten auf Ebene der Anwendungsarchitektur führt. Auch auf Ebene der Infrastruktur entfallen die Hardware-Komponenten der abgelösten Software. Durch eine Vereinheitlichung der Kundendaten kann zudem eine geringere Heterogenität auf Ebene der Datenarchitektur hinsichtlich der syntaktischen Verschiedenheit der Kundendaten erreicht werden. Wie jedoch in Sektion 4.4.3.3.1 bereits erläutert wurde, erfolgte bei der Migration keine Harmonisierung der Prozesse, was zur Folge hatte, dass die Vorgehensweisen der abgelösten Lösung in der neuen Lösung nachgebildet werden mussten, wodurch die Komplexität dieser Lösung zunahm. Eine differenzierte Betrachtung der Maßnahmen ist demnach stets erforderlich.

4.4.4 Durch strukturelle Komplexität beeinflusste Zieleigenschaften des Systems

Ein Architekturmanagement verfolgt bei der Ausgestaltung der IT-Architektur bestimmte Ziele (vgl. Unterabschnitt 2.2.5.2). In der Literatur werden hierbei in erster Linie die Bereitstellung und Aufrechterhaltung einer effizienten und flexiblen IT-Architektur diskutiert, um damit eine Erreichung der Geschäftsziele der Unternehmung zu unterstützen (vgl. Ahlemann et al. 2012; Richardson et al. 1990; Ross et al. 2006; Schmidt & Buxmann 2011; Tamm et al. 2011). Anhand der Daten aus den Experteninterviews wurde untersucht, welche Eigenschaften eines Systems für das Management von IT-Architekturen relevant sind (Untersuchungsfrage 3.1) und wie diese Eigenschaften durch die strukturelle Komplexität der IT-Architektur beeinflusst werden (Untersuchungsfrage 3.2). In Abbildung 24 werden die in den folgenden Unterabschnitten vorgestellten Ergebnisse in den Gesamtkontext der Studie eingeordnet.

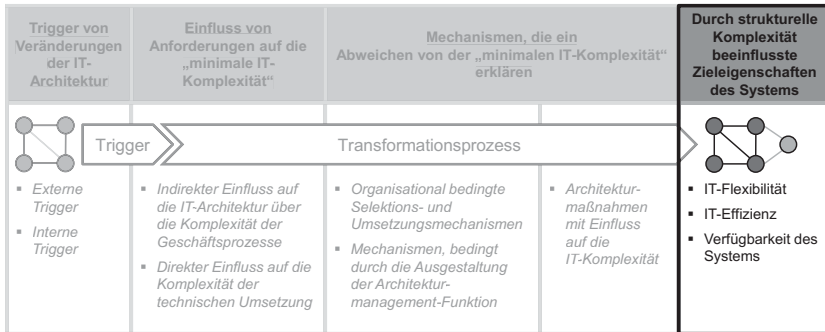


Abbildung 24: Einordnung der Zieleigenschaften der IT-Architektur in den Untersuchungskontext

Die Interviews mit den Experten ergeben, dass neben den bereits angeführten Zieleigenschaften der IT-Flexibilität (Unterabschnitt 4.4.4.1) und der IT-Effizienz (Unterabschnitt 4.4.4.2) auch die Verfügbarkeit des Systems (Unterabschnitt 4.4.4.3) eine wesentliche Rolle spielt.

4.4.4.1 IT-Flexibilität

Wie bereits in Sektion 2.2.5.2.1 eingeführt, stellt IT-Flexibilität die Eigenschaft einer IT-Architektur dar, mit angemessenem Aufwand an neue (Geschäfts-)Anforderungen angepasst werden zu können. Aus den Experteninterviews werden unterschiedliche Facetten der Zieleigenschaft der IT-Flexibilität deutlich, die in den folgenden drei Sektionen näher erläutert werden.

4.4.4.1.1 Werkzeugkasten-Facette

Eine Facette der IT-Flexibilität – im Sinne der oben angeführten Definition – stellt die Verfügbarkeit von Technologien dar, welche für die Integration neuer Anforderungen erforderlich sind. Hier wird die Menge in einer IT-Architektur vorhandener Technologien als eine Art Werkzeugkasten verstanden, der die Werkzeuge für die Umsetzung neuer Anforderungen bereitstellt. Ein Experte aus der Anwendungsarchitektur äußert sich hierzu wie folgt:

„Also eine Auswirkung [struktureller Komplexität] ist, dass wenn eine neue Anforderung eingeht, diese schon zu 95 % mit den bestehenden Möglichkeiten abgedeckt werden kann. Das ist ein Vorteil der Komplexität.“ (Experte aus der Anwendungsarchitektur)

Unter den „bestehenden Möglichkeiten“ ist an dieser Stelle in erster Linie die Menge unterschiedlicher vorhandener Technologien – also die Heterogenität von Komponenten und Schnittstellen – zu verstehen. Im Hinblick auf die IT-Flexibilität kann sich demzufolge ein gewisser Grad an Komplexität (vgl. Abbildung 13) bei der Umsetzung künftiger Anforderungen als vorteilhaft herausstellen. Gleichzeitig deutet dies darauf hin, dass Architekturmaßnahmen wie die Standardisierung (vgl. Sektion 4.4.3.3.1) maßvoll eingesetzt werden sollten. Eine zu hohe Rigidität bei Architekturentscheidungen könnte an dieser Stelle zu einer geringeren IT-Flexibilität führen. Werden im Rahmen von Architekturentscheidungen hinge-

gen ein zu geringes Augenmerk auf die Standardisierung der eingesetzten Technologien gelegt und „zu viele“ unterschiedliche Technologien eingeführt, so kann sich dies auch negativ auf die IT-Flexibilität auswirken. In einem solchen Fall müssen bei der Durchführung von Änderungen die unterschiedlichen Schnittstellentechnologien der bestehenden Komponenten berücksichtigt werden. Die Werkzeugkasten-Facette deutet demzufolge auf einen u-förmigen Zusammenhang zwischen der Heterogenität der Komponenten bzw. Beziehungen und der IT-Flexibilität hin (vgl. Abbildung 25).

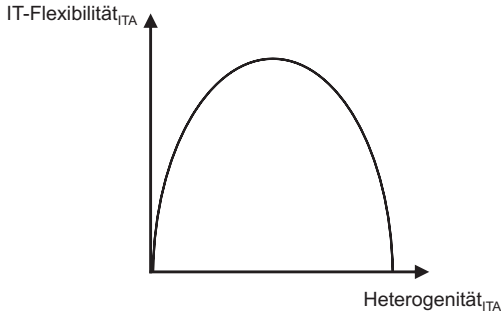


Abbildung 25: Vermuteter Zusammenhang der Heterogenität der IT-Architektur und der IT-Flexibilität

Zur Erreichung einer möglichst hohen IT-Flexibilität ist es demnach erforderlich, die adäquaten Werkzeuge für die Menge bestehender (und ggf. künftiger) Anforderungen zu identifizieren. Ein Experte aus dem Bereich der Infrastrukturarchitektur führt ein Beispiel im SAP-Kontext an, in dem durch Standardisierung der Komponenten – also durch Reduktion der Heterogenität – die IT-Flexibilität erhöht werden konnte. Dabei wurden sowohl ein Citrix-Netzwerk eingeführt, über welches die Anbindung der Großhändler an das zentrale SAP-System erfolgt, als auch die Endgeräte bei den angeschlossenen Großhändlern vereinheitlicht. Weitere Großhändler können aufgrund der einheitlichen IT-Architektur nun schnell und mit relativ wenig Aufwand integriert werden:

„Wir sind hier relativ flexibel und ein neuer Großhändler stellt uns nicht vor große Herausforderungen. [...] Von unserer Abteilung her – von der IT – sind da eigentlich keine größeren Anforderungen mehr zu stemmen. Und das können wir nur, weil wir einen standardisierten Prozess haben, den wir [...] [in solchen Fällen heranziehen], der an sich nicht komplex ist und durch den auch keine neue Komplexität entsteht. Deswegen muss jeder Großhändler, der bei uns auf das System möchte, auch dieser Infrastruktur beitreten. Sonst geht das nicht.“ (Experte aus der Infrastrukturarchitektur)

Das Besondere an dem hier dargestellten Ausschnitt der IT-Architektur der UG ist, dass die Aufgabengebiete klar abgesteckt sind und die Anforderungen – also die Anbindung eines neuen Großhändlers – einen wiederkehrenden und relativ gleichmäßigen Charakter aufweisen. Es ist davon auszugehen, dass die Identifikation der erforderlichen Werkzeuge in einem dynamischeren Kontext eine weitaus schwierigere Aufgabe darstellt.

4.4.4.1.2 Abhängigkeits-Facette

Eine weitere in den Interviews sehr intensiv diskutierte Facette der IT-Flexibilität bezieht sich auf die Anpassbarkeit des bestehenden Systems. Die Schwierigkeiten bei der Anpassbarkeit resultieren dabei aus der großen Menge von Schnittstellen und Abhängigkeiten zwischen den einzelnen Komponenten der IT-Architektur. Ein Experte aus der Infrastrukturarchitektur berichtet hierzu von schwer vorhersehbaren Folgen einer Änderung aufgrund der bestehenden Abhängigkeiten. Die vorhandene IT-Architektur ist dabei bereits so umfangreich und stark in die Produktionstechnik integriert, dass die Bereitstellung einer mit der Produktivumgebung vollständig identischen Testumgebung kaum noch möglich ist. Dies erschwert die Vermeidung von unerwarteten Seiteneffekten bei Änderungen der IT-Architektur:

„Ja, es kann immer vorkommen, [...] dass an Einzelstellen eine Änderung durchgeführt wird, deren Auswirkungen an anderer Stelle nicht absehbar waren. Selbst bei der besten Risikoanalyse. Wie oft steht irgendwas, weil wieder etwas transportiert wurde... Es wurden Schnittstellen angepasst und es wurde [in der Testumgebung] getestet. Dann sieht die Produktivumgebung vielleicht doch ein bisschen anders aus, und irgendwo hängt dann ein Zahnradchen, weil man eben eine [Maschine] nicht nochmal [in der Testumgebung] stehen hat. Ich sehe nicht, ob der Roboter richtig fährt, wenn ich es nicht am realen Roboter mache.“ (Experte aus der Infrastrukturarchitektur)

Neben der technischen Perspektive auf die Verschiedenheit und Menge der Abhängigkeiten, spielt auch die Vielseitigkeit der durch die IT-Architektur unterstützten Prozesse eine wesentliche Rolle. So existiert an den verschiedenen Standorten eine Vielzahl unterschiedlich implementierter Prozessvarianten, die alle im Zuge einer Änderung eines zentralen Systems berücksichtigt werden müssen. Ein Experte aus der Anwendungsarchitektur berichtet hierzu:

„Wir hatten Anfang des Jahres eine Anforderung von den schwedischen Kollegen, [...] etwas an der Lagerortsteuerung zu ändern. Das haben wir auch gut hinbekommen und an deren Anforderungen angepasst. Leider hat in der Folge die Lagerortsteuerung an einem französischen Standort nicht mehr funktioniert, was dann [wieder behoben werden musste]. [...] Hier zeigt sich die Komplexität durch die unterschiedlichen Anforderungen und die hohe Anzahl der verschiedenen Möglichkeiten, die es gibt. Dadurch ist auch die Testdurchführung schwierig und langwierig. [...] Man erfasst gar nicht alle Varianten eines Prozesses, wie jetzt in dem Fall. Für Schweden hat man es natürlich ausprobiert und auch für die deutschen Standorte [...], aber eben für diesen speziellen französischen Fall hat man den Test in der Testphase nicht durchgeführt.“ (Experte aus der Anwendungsarchitektur)

Darüber hinaus berichten die Experten von Schwierigkeiten im Kontext größerer Änderungen, wie z. B. Releasewechsel, aufgrund der Komplexität auf Ebene der Geschäftsprozessarchitektur:

„Es gibt viele Varianten. Wenn man dann etwas umstellt, müssen die anderen, bisher neun Prozesse, die jetzt durch einen zehnten ergänzt werden, alle durchgespielt werden. Man ist sich aber oft nicht ganz im Klaren, was alles dazu gehört. Das merkt man, wenn

wir einen Releasewechsel machen. Dann müssen wir mit den Anwendern zusammen die ganzen Szenarien durchspielen. Ein Releasewechsel ist ein hochkomplexer, aufwendiger Vorgang, der dazu führt, dass wir, sobald wir dann umstellen, immer noch etwas entdecken, was eben nicht durch die Tests abgedeckt wurde.“ (Experte aus der Anwendungsarchitektur)

4.4.4.1.3 Planungs-Facette

Eine weitere Facette der IT-Flexibilität betrifft den vor der Durchführung einer Anpassung zu berücksichtigenden Planungs- und Abstimmungsaufwand. Dieser nimmt im Falle einer zunehmenden strukturellen Komplexität der Architektur tendenziell zu, da beispielsweise bei der Änderung eines zentralen Systems, welches über eine Vielzahl von Schnittstellen zu Anwendungen verschiedener Fachbereiche verfügt, eine entsprechend große Menge an Ansprechpartnern bei der Lösungsfindung zu involvieren ist. In Abgrenzung zu der in Sektion 4.4.4.1.2 diskutierten Abhängigkeits-Facette existiert demnach auch eine Wirkung auf organisationaler Ebene, die hier als eine eigene Facette der IT-Flexibilität verstanden wird. Wie das folgende Zitat eines Experten aus der Anwendungsarchitektur zeigt, hat diese Facette bereits von Beginn eines Projektes an einen Einfluss:

„Nachteil der Komplexität ist, dass ich einen extrem hohen Abstimmungs- und Planungsaufwand habe. Ich muss sehr lange genau überlegen, wie bilde ich das jetzt ab, mit wem bilde ich das ab, wen brauche ich alles, wen muss ich alles einbinden [...]“ (Experte aus der Anwendungsarchitektur)

Aus dem folgenden Zitat eines Experten aus der Infrastrukturarchitektur zum Planungsprozess einer Änderung wird deutlich, dass der einzubeziehende Personenkreis neben den Fachbereichen auch weitere Interessengruppen umfassen kann:

„Wer sind die Stakeholder des Changes? Von Compliance, über Fachbereich, Technik, Know-how... Ja, und über die Kosten muss auch gesprochen werden. Welches sind die beteiligten Kompetenzen? Wen kann ich fragen? Wer ist irgendwo betroffen? [...] Welche Auswirkungen wird das haben? [...] Mit Komplexität steigt die Planungsphase nicht linear, sondern wirklich exponentiell an.“ (Experte aus der Infrastrukturarchitektur)

Der dargestellte Abstimmungsaufwand und die damit verbundene benötigte Zeit der Umsetzung einer Anforderung kann Unzufriedenheit bei den Fachbereichen bezüglich der IT-Flexibilität hervorrufen:

„Und daraus entsteht dann Unzufriedenheit in der Fachabteilung. Die sagen dann: ‚Ich wollte doch nur das grüne Lämpchen haben‘. So, aber das muss über sehr viele Gremien mit sehr vielen verschiedenen Leuten diskutiert werden, weil es auch einfach sehr verteilt ist.“ (Experte aus der Anwendungsarchitektur)

Diese Unzufriedenheit der Fachabteilung kann in der Folge auch dazu führen, dass vorhandene eigene Möglichkeiten und Freiheiten bezüglich der Ausgestaltung von IT-Lösungen – unabhängig von der zentralen IT – vermehrt genutzt werden (vgl. Unterabschnitt 4.4.3.1). Zunehmende dezentral getroffene Entscheidungen können wiederum einen Anstieg struktureller

Komplexität der IT-Architektur begünstigen (vgl. Unterabschnitt 4.4.3.1), was zu einer weiteren Abnahme der IT-Flexibilität führen kann.

4.4.4.2 IT-Effizienz

IT-Effizienz wird definiert als die Eigenschaft einer IT-Architektur, bestehende Geschäftsanforderungen mit möglichst geringem Ressourceneinsatz erfüllen zu können (vgl. Sektion 2.2.5.2.2). Im Rahmen der Interviews betonten die Experten die Effekte der Heterogenität sowie der Redundanz, die nicht der Ausfallsicherheit der Systeme dient, auf die IT-Effizienz. Eine Verringerung der Komplexität der IT-Architektur durch Reduktion der Heterogenität sowie ein Abbau fachlich redundanter Lösungen kann demnach eine Erhöhung der IT-Effizienz zur Folge haben.

Wie bereits in Sektion 4.4.3.1.1 erläutert, führen unterschiedliche fachliche Anforderungen sowie die Eigenständigkeit einzelner Gesellschaften häufig zur Entstehung von Insellösungen. Diese sind grundsätzlich auf eine spezifische Problemstellung einer angeschlossenen Gesellschaft ausgerichtet und können nur sehr schwer auf ähnliche Problemstellungen anderer angeschlossener Gesellschaften angewendet werden. Von einer Reihe solcher individueller Lösungen berichtet ein Experte im Kontext der Content Management Systeme für die Web-Auftritte der angeschlossenen Gesellschaften. In diesem Bereich konnte eine Vereinheitlichung durchgesetzt und Kosteneinsparungen ermöglicht werden:

„Man spart hier deutlich Kosten. [...] Wir haben eine Umlage ermittelt, die muss jeder zahlen. Das sind [...] € im Jahr. Dafür stehen unsere Server und unser Typo3 zur Verfügung. Außerdem bekommen die Gesellschaften einen Grundsupport durch die Agenturen.“ (Experte aus der Anwendungsarchitektur)

Auch die bereits in Sektion 4.4.3.3.2 diskutierte Migration eines vormals eigenständigen Produktionssystems in die SAP-Landschaft führte dazu, dass nicht nur ein System eines anderen Herstellers abgeschaltet werden konnte, sondern dass das benötigte Know-how sich nun auf ein System beschränkt. Auch die Standardisierung auf die SAP-Lösung im Großhandel (vgl. Sektion 4.4.3.3.1) führte nicht nur zu einer besseren Integrierbarkeit neuer Großhändler (vgl. Unterabschnitt 4.4.4.1), sondern auch zu einer Bündelung von Kompetenzen und Know-how, wodurch der Support weitestgehend zentral abgewickelt werden kann:

„Wir haben im Großhandel einheitliche Prozesse eingeführt. Wir können den Großhändlern von hier aus auch einen gewissen Support anbieten und brauchen nicht unterschiedliche Leute für den Support. Klar gibt es Spezialthemen. Es gibt immer Sachen, in denen sich Leute spezialisiert haben. Aber den ‚normalen‘ Support bekommen wir hier gut abgewickelt.“ (Experte aus der Infrastrukturarchitektur)

Neben den Kosteneinsparungen für die Einführung und den Support der Großhändler im SAP-Umfeld konnten auch Wartungskosten für die verwendeten Endgeräte eingespart werden. Wie bei der Diskussion zur Architekturmaßnahme der Standardisierung in Sektion 4.4.3.3.1 bereits angeführt, kam es im Zuge der Einführung der Citrix-Lösung zu einer Standardisierung der Endgeräte, von denen aus der Großhandel auf SAP zugreift. Vor dieser

Maßnahme mussten diverse Endgeräte mit unterschiedlichen Versionsständen von Betriebssystemen gewartet und upgedatet werden.

Auch im Infrastrukturbereich werden Anstrengungen unternommen, möglichst nicht von den intern als Standard vorgegebenen Komponenten abzuweichen. Hier besteht das Bewusstsein, dass eine zusätzliche Heterogenität durch die Einführung anderer Hersteller von Datenbanken oder Betriebssystemen auch zusätzliche Kosten für die Bereitstellung von aktuell nicht vorhandenem Know-how verursacht:

„Also einen gewissen Einfluss haben wir. Wenn jetzt jemand sagt: ‚Ich hätte gerne eine Anwendung. Die läuft auf Unix – oder Linux heutzutage – und auf Windows.‘ Dann sagen wir: ‚Wir nehmen die Windows-Applikation‘. Einfach, weil das andere ein [zusätzlicher] Aufwand ist. Im Linux-Bereich sind wir wenig unterwegs und das ist auch ganz bewusst so gehalten, weil das Personal ja ausgebildet werden müsste. Und ich bräuchte nicht nur einen, der das kann, sondern ich bräuchte mindestens zwei oder drei Personen. Und je weniger verschiedene Softwareprodukte ich habe, desto einfacher ist es, das Personal auch auf dem neuesten Stand zu halten.“ (Experte aus der Infrastrukturarchitektur)

Eine entsprechende Argumentation besteht auch bei der Auswahl einer Datenbank-Lösung:

„Wir können SQL, wir können Oracle. [...] Wenn es also heißt: ‚Da ist eine Datenbank hintendran‘, dann werden wir immer bestrebt sein, in erster Linie SQL zu verwenden. Das ist geclustert, da haben wir eine sehr hohe Verfügbarkeit. Wenn SQL gar nicht geht, werden wir Oracle nehmen. Weil auch da haben wir das Know-how in der Firma. Und bei allen anderen Datenbanken ist das Know-how nicht da. Da müssten wir uns jemanden einkaufen, der das Know-how hat. Und das kostet Geld [...].“ (Experte aus der Infrastrukturarchitektur)

Ein weiterer von den Experten ausgiebig diskutierter Zusammenhang besteht zwischen dem Betrieb von nicht lediglich fachlich redundanten, sondern vollkommen identischen Lösungen und den damit verbundenen Kosten. Diese Redundanz steht dabei nicht in Verbindung mit der Gewährleistung einer erhöhten Ausfallsicherheit der Systeme, sondern erzeugt in erster Linie lediglich unnötige Mehrkosten für Anwendungen sowie Infrastruktur. So berichtet ein Experte aus der Infrastrukturarchitektur wie folgt:

„Ich könnte die Exchange Server-Lizenzen sparen, ich könnte die Windows Server-Lizenzen sparen, ich könnte mir die Firewall inklusive der Wartung sparen, ich könnte mir die Administration von dem ganzen System sparen, ich könnte mir das Drumherum was an Tickets im Servicedesk aufschlägt sparen. [...] Man könnte sogar den Internetzugang dort kündigen. Ja, dort läuft ein Linux Proxy, den könnte man komplett abschalten und und und... [...] Wir haben ja alles in [Standort Dachgesellschaft] redundant, hoch verfügbar, performante Anbindung.“ (Experte aus der Infrastrukturarchitektur)

Diese Redundanzen der Exchange Server führen zudem zu zusätzlichem Aufwand bei der Aktualisierung und Wartung von angeschlossenen Systemen:

„Wir haben eine Software im Einsatz von Update [Hersteller von CRM-Software – Anm. d. Verf.]. Die nennt sich Groupware und kommuniziert mit Outlook und schaufelt die Daten hin und her. Und jetzt hat [Gesellschaft A] auch ‚Update.seven‘ und [Gesellschaft A] stellt nun den eigenen Exchange Server um. Also müssen jetzt die Kollegen hier [...] bei uns Anpassungen durchführen, weil [Gesellschaft A] einen eigenen Exchange Server betreibt. [...] Hätten wir einen gemeinsamen Exchange Server, dann hätten wir letztes Jahr gemeinsam das Upgrade durchgeführt und Ende. Jetzt kommt wieder eins. Das ist etwas, was im Grunde hinderlich ist.“ (Experte aus der Anwendungsarchitektur)

4.4.4.3 Verfügbarkeit des Systems

Zusätzlich zu den bereits vorgestellten Zieleigenschaften der IT-Flexibilität und IT-Effizienz diskutieren die Experten auch häufig das Thema der Verfügbarkeit des Systems. *Verfügbarkeit wird dabei definiert als die Eigenschaft einer IT-Architektur, den legitimierten Nutzern des Systems den Zugriff auf die durch das System implementierten Funktionen und Daten jederzeit zu gewährleisten* (vgl. Avizienis et al. 2004; Landwehr 2001; Olovsson 1992).

In Abgrenzung zu der Abhängigkeits-Facette der Zieleigenschaft der IT-Flexibilität, bei welcher in erster Linie die schwere Vorhersehbarkeit von Seiteneffekten bei Änderung der IT-Architektur eine Rolle spielt, stehen bei der Verfügbarkeit vornehmlich die Ausfallrisiken eines komplexen Systems im Vordergrund. Ein angesprochener Punkt betrifft beispielsweise die Befähigung der Mitarbeiter der Rufbereitschaft, im Falle eines Vorfalls entsprechende Maßnahmen einleiten zu können. Je komplexer ein System – also je mehr heterogene Komponenten mit unterschiedlich ausgeprägten Schnittstellen und Protokollen – desto schwieriger ist es für den Mitarbeiter, die Ursachen eines Vorfalls zu erkennen und zu beheben:

„Wir haben das Problem mit der Rufbereitschaft. Es sind einfach zu viele Systeme, als dass jeder, der an der Rufbereitschaft teilnimmt, alle Systeme [perfekt] kennen könnte. Also muss er sich darauf verlassen, dass wenn etwas ausfällt, eine Dokumentation vorhanden ist, mit der er arbeiten kann. Also wer gut dokumentiert, der verringert zwar nicht die Komplexität der Systeme, denn dafür ist die Dokumentation natürlich nicht bestimmt, aber der ermöglicht eine Korrektur, wenn es Probleme gibt.“ (Experte aus der Infrastrukturarchitektur)

Weiterhin erklärt der Experte, dass bereits einiges an Dokumentation für solche Fälle vorliegt, jedoch auch immer wieder neue Fehler auftreten, bei denen die Erfahrung und das Geschick der Mitarbeiter gefragt sind:

„Wenn ein ganz neuer Fehler auftritt, dann ist es auch für diejenigen in der Rufbereitschaft schwierig zu entscheiden, was zu tun ist. Aber ich sage mal, das sind alles Leute, die in dem Metier schon lange unterwegs sind. Die versuchen etwas und wenn es funktioniert hat, ist es gut, und wenn nicht, dann wird etwas anderes probiert. Also das ist schon so ein „trial and error“. Dort, wo es schon dokumentiert ist, keine Frage, dort kann man relativ schnell etwas machen. Aber dort, wo ein ganz neuer Fehler auftritt, lebt man von der Erfahrung.“ (Experte aus der Infrastrukturarchitektur)

Die von dem Experten angeführte Erfahrung ist in einer IT-Architektur mit einer großen Anzahl heterogener Systeme jedoch nur schwer vorzuhalten und ein einzelner Mitarbeiter kommt dabei an seine Grenzen:

„Wenn man sich um dreißig, vierzig, fünfzig Produkte kümmert, dann kann man immer nur der Generalist sein, nicht mehr der Spezialist.“ (Experte aus der Infrastrukturarchitektur)

Versäumt es ein Unternehmen, sich von alten Systemen zu trennen, für die kein ausreichender Support des Herstellers oder geschultes internes Personal vorhanden ist, kann dies empfindliche Auswirkungen im Falle eines Vorfalles haben. Ein Experte aus der Infrastrukturarchitektur berichtet von einem derartigen Altsystem:

„Das heißt, der Support war kritisch und jedes Mal, wenn es zu Ausfällen kam, ist natürlich alles ausgefallen. Das heißt, da ich an die Ware nicht ran kam, stand im Prinzip die gesamte Kommissionierung, die aus diesem Lager kommt, still.“ (Experte aus der Infrastrukturarchitektur)

Aus den Interviews ergibt sich mit der „Verfügbarkeit“ an dieser Stelle nur eines der im Literaturstrang der IT-Sicherheit diskutierten Attribute. Es ist durchaus naheliegend, dass noch weitere sicherheitsorientierte Aspekte (vgl. Avizienis et al. 2004; Landwehr 2001; Olovsson 1992) eine wesentliche Rolle spielen. Beispielsweise erhöht eine zunehmende Heterogenität der Hersteller von IT-Lösungen auch die Menge potenzieller IT-Sicherheitsschwachstellen und damit die Empfindlichkeit gegenüber Angriffen von außen.

4.4.5 Darstellung der Ergebnisse anhand einer Auswahl von Projekten

Die in den vorangegangenen Abschnitten vorgestellten Ergebnisse werden im Folgenden verwendet, um die Effekte der jeweils durchgeführten Änderung der IT-Architektur auf die strukturelle Komplexität anhand einer Auswahl der in den Interviews diskutierten Projekte darzustellen.

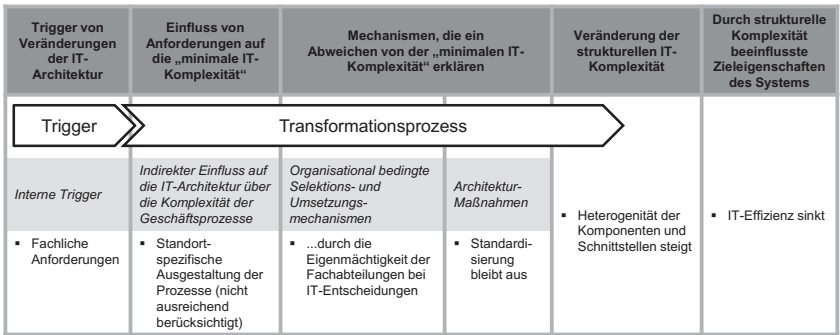


Abbildung 26: Ausprägungen für die Projekte A und B

Bei den ersten beiden Projekten A und B (vgl. Abbildung 26) handelte es sich jeweils um eine fachliche Anforderung. Im Zuge der Projekte wurde eine neue E-Learning-Plattform (Projekt

A) bzw. eine CRM-Software (Projekt B) in die bestehende IT-Landschaft integriert. In beiden Fällen entsprach die etablierte Lösung nicht den standortspezifischen Anforderungen, was die Fachbereiche dazu veranlasste, eigenmächtig eine alternative Lösung zu wählen und anzuschaffen. Da die IT-Funktion bei beiden Anschaffungen nicht hinzugezogen wurde, konnten keine Bemühungen in Richtung der Standardlösung unternommen werden. Durch die Integration dieser von den bestehenden Lösungen abweichenden Softwareprodukte stieg die strukturelle Komplexität der IT-Architektur sowohl in Form der Heterogenität der Komponenten als auch der Schnittstellen an. Aufgrund des nun erforderlichen Betriebes neuer Lösungen stiegen die Kosten der IT, was zu einer geringeren IT-Effizienz führte.

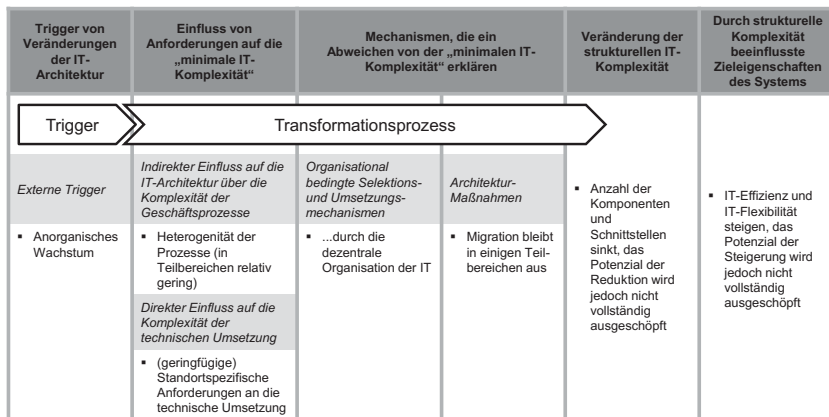


Abbildung 27: Ausprägungen für Projekt C

Auslöser von Projekt C (vgl. Abbildung 27) war der Zukauf eines Unternehmens, das in die bestehende IT-Landschaft integriert werden sollte. Bei gemeinsamer Betrachtung der beiden zuvor unabhängigen IT-Bereiche der Unternehmen bestand durch den Zukauf die Möglichkeit, einen Teil der bestehenden Systeme abzubauen und hierdurch Synergien zu heben. Demzufolge hätte (bei gemeinsamer Betrachtung) durch den Zusammenschluss IT-Komplexität abgebaut werden können, da $K_{ITA,min}$ (aus aggregierter Perspektive) aufgrund der teilweise ähnlichen Prozesse und der vereinzelt vergleichbaren standortspezifischen Anforderungen an die technische Umsetzung sank. Die tatsächliche IT-Komplexität ($K_{ITA,real}$) nahm zwar ebenfalls ab, lag jedoch noch deutlich über $K_{ITA,min}$. Dies ist damit zu begründen, dass von der dezentralen IT der zugekauften Gesellschaft entschieden wurde, eine Reihe von Servern nicht in die bestehende IT-Landschaft der Dachgesellschaft zu integrieren, obwohl dies sowohl technisch als auch im Hinblick auf die vorhandenen Kapazitäten möglich gewesen wäre. Hierdurch blieb eine weitergehende Migration der Systeme und Daten aus, wodurch vorhandene Potenziale zur Reduktion struktureller Komplexität nicht ausgeschöpft wurden. Durch den Parallelbetrieb der beiden, an zwei verschiedenen Standorten vorgehaltenen IT-Systeme entstanden vermeidbare Kosten – u. a. für Personal und Lizenzen. Zudem mussten durch andere Systeme bedingte Anpassungen weiterhin an beiden Standorten durchgeführt und getes-

tet werden, was die Flexibilität der Durchführung von Änderungen unnötig belastete und die Anzahl potenzieller Fehlerquellen erhöhte.

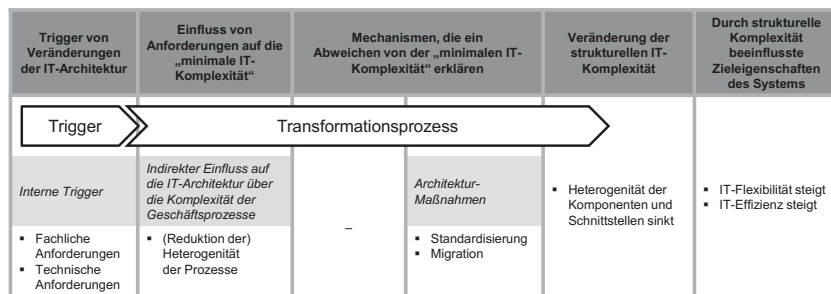


Abbildung 28: Ausprägungen für Projekt D

In Projekt D (vgl. Abbildung 28) wurde im Zuge der Einführung einer einheitlichen Betriebsdatenerfassung im Produktionsbereich ein bestehendes proprietäres System abgelöst und die Daten nach SAP – als etablierte Standardlösung – migriert. Gleichzeitig konnten durch die Ablösung des Altsystems proprietäre Schnittstellenformate durch SAP-Standardschnittstellen ersetzt werden. Die Vereinheitlichung des Vorgehens zur Erfassung der Betriebsdaten hat demnach einen indirekten Einfluss auf die IT-Architektur über die Verringerung der Komplexität der Geschäftsprozesse, welche eine Reduktion von $K_{ITA,min}$ zur Folge hatte. Da in diesem Projekt keine ungünstig beeinflussenden Mechanismen wirkten, die weiterhin unnötige Komplexität aufrechterhielten, konnte die strukturelle Komplexität der IT-Architektur – in Form einer geringeren Heterogenität der Komponenten und Schnittstellen – reduziert werden. Hierdurch wurden sowohl die IT-Effizienz als auch die IT-Flexibilität positiv beeinflusst.

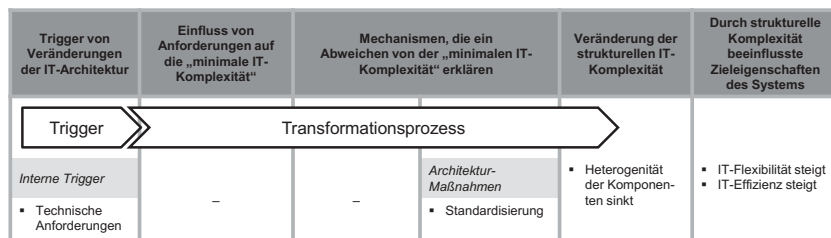


Abbildung 29: Ausprägungen für Projekt E

Ein weiteres Projekt E (vgl. Abbildung 29), das auch bereits an einigen Stellen in Form von diversen Zitaten Erwähnung fand, hatte die Umstellung der IT der Großhändler auf das Citrix-Netzwerk zum Gegenstand. Diese Änderung hatte dabei weniger fachliche, als vielmehr technische Beweggründe. Aufgrund der sehr heterogenen Ausstattung mit Endgeräten auf der Seite der Großhändler (u. a. diverse Betriebssystemversionen), war die Versorgung dieser – mit beispielsweise aktuellen Sicherheitsupdates – sehr aufwändig. Nach dem Citrix-Rollout verfügen alle Großhändler über vollständig standardisierte Endgeräte, die ohne großen Aufwand zentral betreut werden können. Da standortspezifische Anforderungen bei der Umset-

zung keine Rolle spielten und die Beschaffung und Implementierung über die zentrale IT durchgeführt wurden, konnte die strukturelle Komplexität der IT-Architektur durch die Verringerung der Heterogenität der Komponenten deutlich reduziert werden. Neben den bereits angeführten Vorteilen im Hinblick auf die IT-Effizienz konnte auch die IT-Flexibilität gesteigert werden. Änderungen erfolgen am zentral betreuten System und mögliche Seiteneffekte sind durch die gleichförmige Ausgestaltung der angeschlossenen Infrastruktur gut zu managen. Auch die Anbindung neuer Großhändler kann nun zeitnah und nach einem klar definierten Vorgehen erfolgen.

4.5 Zusammenfassung und Diskussion der Ergebnisse

In diesem Kapitel wurden im Rahmen einer qualitativ-empirischen Fallstudie die Ursachen und die Auswirkungen struktureller Komplexität in IT-Architekturen untersucht und damit die folgenden beiden Forschungsfragen bearbeitet:

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

Die Ergebnisse der qualitativ-empirischen Fallstudie sind in Abbildung 30 zusammengefasst.

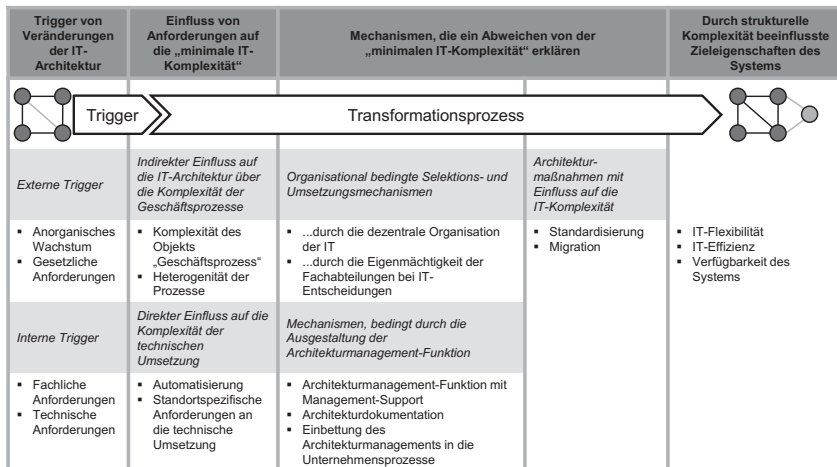


Abbildung 30: Überblick über die Ergebnisse der qualitativen Untersuchung

Mit dem Fokus auf der Veränderung der strukturellen Komplexität der IT-Architektur wurden unterschiedliche Phasen des Transformationsprozesses der IT-Architektur beleuchtet. Annahmen dieser Betrachtungsweise waren dabei, dass eine IT-Architektur als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht, verstanden wird und dass dieses System ausschließlich durch Eingriffe von außen verändert

werden kann. In der Studie wurden sowohl unternehmensinterne als auch -externe Auslöser (Trigger) von Änderungen identifiziert. Bei allen Auslösern handelt es sich um Anforderungen, die im Zuge einer Transformation der IT-Architektur umzusetzen sind.

In der Studie wurde die Veränderung struktureller Komplexität der IT-Architektur in zwei Komponenten unterschieden: die minimal erforderliche ($\Delta K_{ITA,min}$) und die „unnötige“ – über dieses Mindestmaß hinausgehende – Veränderung der Komplexität der IT-Architektur ($\Delta K_{ITA,unnötig}$). Diese Unterscheidung erlaubt eine gesonderte Betrachtung zweier Klassen von Effekten. Die erste Klasse bezieht sich auf Einflussfaktoren, die eine Veränderung des Mindestmaßes an Komplexität bewirken. Hingegen erlaubt die zweite Klasse von Effekten die Untersuchung von Mechanismen, durch die es zu einer Abweichung von diesem Mindestmaß $K_{ITA,min}$ und somit zur Entstehung „unnötiger“ Komplexität kommt.

Aus den Interviews ergab sich, dass Abweichungen von $K_{ITA,min}$ in erster Linie durch die verteilten Entscheidungskompetenzen zwischen zentraler und dezentraler IT sowie die Eigenmächtigkeit der Fachbereiche bei IT-Entscheidungen verursacht wurden. Komplementiert wurden diese Mechanismen durch die fehlende Architekturmanagement-Funktion und die ebenfalls fehlende Einbettung dieser in die Unternehmensprozesse.

Gleichzeitig erlaubte die Studie die Identifikation von Architekturmaßnahmen, die durch das Unternehmen eingesetzt werden und die sich auf die Komplexität der IT-Architektur auswirken. Die Studie zeigte, dass die Architekturmaßnahmen der Standardisierung und der Migration bislang vornehmlich im Kontext zentraler IT-Entscheidungen zu beobachten sind, was nicht zuletzt durch die fehlende Architekturmanagement-Funktion begründet werden kann.

Weiterhin wurde untersucht, welche Eigenschaften eines Systems relevant für das Management von IT-Architekturen sind und wie diese Eigenschaften durch die Komplexität der IT-Architektur beeinflusst werden. Mit IT-Flexibilität, IT-Effizienz und Verfügbarkeit des Systems wurden drei relevante Zieleigenschaften festgestellt. Der Einfluss der Komplexität auf die IT-Flexibilität erwies sich dabei als nicht eindeutig. Dies kann anhand der unterschiedlichen identifizierten Facetten der IT-Flexibilität begründet werden. Im Hinblick auf die sogenannte Werkzeugkasten-Facetten führt ein gewisser Grad an Komplexität – hier die Heterogenität – zunächst zu mehr Flexibilität, da die Architekten eine Mindestmenge an Werkzeugen benötigen, um Änderungen mit angemessenem Aufwand durchführen zu können. Durch die Abhängigkeits-Facetten erschwert eine Zunahme der Komplexität die Vorhersagbarkeit der Seiteneffekte bei der Durchführung von Änderungen. Die Planungs-Facetten der IT-Flexibilität macht sich durch eine Zunahme des Abstimmungsaufwandes im Falle einer Erhöhung der strukturellen Komplexität bemerkbar.

Eine zusätzlich identifizierte Zieleigenschaft der IT-Architektur ist die Verfügbarkeit des Systems. Zunehmende strukturelle Komplexität der IT-Architektur birgt die Gefahr einer Erhöhung der Ausfallzeiten des Systems, da bei einer Vielzahl unterschiedlicher Komponenten die Fehleridentifikation im Falle von Vorfällen zunehmend erschwert wird.

In den folgenden beiden Abschnitten werden die theoretischen sowie die praktischen Implikationen der vorgestellten Studie diskutiert.

4.5.1 Theoretische Implikationen

In der Studie wurde eine neue Perspektive – basierend auf der strukturellen Komplexität der IT-Architektur – auf die Entwicklung und das Management einer Unternehmens-/IT-Architektur eingeführt. Durch die Unterscheidung von erforderlicher Komplexität ($K_{ITA,min}$) und unnötiger Komplexität ($K_{ITA,unnötig}$) wurde deutlich, dass strukturelle Komplexität der IT-Architektur nicht per se etwas Negatives ist, was es zu reduzieren gilt. Vielmehr ist ein Teil der strukturellen Komplexität der IT-Architektur notwendig, um gegebene und künftige Anforderungen erfüllen zu können.

Wie die Untersuchung gezeigt hat, ermöglicht die Unterscheidung dieser beiden Formen der Komplexität eine differenzierte Betrachtung der Entstehung struktureller Komplexität der IT-Architektur. Zum einen wird nachvollziehbar, welchen Effekt Eingriffe in die Unternehmens-/IT-Architektur auf $K_{ITA,min}$ haben. Zum anderen unterstützt die Unterscheidung bei dem Verständnis, welche Mechanismen während der Transformation der IT-Architektur zu einer Entstehung von $K_{ITA,unnötig}$ führen. Des Weiteren erlaubt sie, die Auswirkungen struktureller Komplexität auf die Zieleigenschaften der IT-Architektur besser zu verstehen.

Die Perspektive auf die Veränderung der strukturellen Komplexität der IT-Architektur komplementiert bzw. integriert bestehende Perspektiven und Ansätze der Forschung zu Unternehmensarchitekturen. So lässt sich beispielsweise das Konzept der Transformation einer bestehenden Architektur (As-Is) hin zu einer Zielarchitektur (To-Be) (Aier et al. 2012; Niemann 2005; Saat et al. 2009) über die Anforderungen, welche an die IT-Architektur gestellt werden, integrieren. Eine Zielarchitektur kann demzufolge einen Teil der Anforderungen bedingen, die im Rahmen eines Transformationsprozesses zu berücksichtigen sind. Mit hin enthält $K_{ITA,min}$ auch die durch die Zielarchitektur induzierte Komplexität der IT-Architektur. Diese wird als Teil des Veränderungspfad es dabei explizit in Kauf genommen. Darüber hinaus kann über die Mechanismen während des Transformationsprozesses (beispielsweise durch Architekturprinzipien) beeinflusst werden, wie auf eine Zielarchitektur zugesteuert werden soll. Auch können die Mechanismen beobachtet werden, die gegebenenfalls ein Abweichen von der Zielarchitektur verursachen. Auf der anderen Seite ist es für die hier verwendete Komplexitäts-Perspektive nicht zwingend erforderlich, eine Zielarchitektur – in Form einer modellhaften Abbildung der IT-Architektur – zu definieren und fortzuschreiben. Vielmehr können, beispielsweise über die zielgerichtete Ausgestaltung der Architekturmanagement-Funktion und der dort angewendeten Architekturprinzipien, Grundsatzentscheidungen hinsichtlich bestimmter Technologien und Verfahren getroffen werden, welche die zukünftige Gestalt der IT-Architektur leiten. Die hieraus entstehenden Architekturmaßnahmen beinhalten u. a. die der IT-Standardisierung, als einen weiteren in der Forschung etablierten Ansatz des Architekturmanagements (vgl. Unterabschnitt 2.2.5.3).

4.5.2 Praktische Implikationen

Die Studie hat gezeigt, dass in den verschiedenen Phasen des Transformationsprozesses unterschiedliche Aspekte zur Entstehung struktureller Komplexität der IT-Architektur führen. Über die Einflüsse der Anforderungen, die einen Anstieg des Mindestmaßes struktureller

Komplexität der IT-Architektur bedingen sowie über die Mechanismen im Zuge der Umsetzung der Anforderungen, durch die eine Abweichung von diesem Mindestmaß entsteht, kann ein Unternehmen gezielt gegenlenken. In dem betrachteten Fall kann beispielsweise eine klare Regelung der Kompetenzen zwischen zentraler IT und den dezentralen IT-Abteilungen der angeschlossenen Gesellschaften eine zielgerichtete Entwicklung der IT-Architektur unterstützen. Darüber hinaus stellt eine Architekturmanagement-Funktion, die mit ausreichenden Kompetenzen ausgestattet und in alle architekturrelevanten Unternehmensprozesse integriert ist, eine essenzielle Funktion für das Management der strukturellen Komplexität der IT-Architektur dar. Ausgerichtet an den für das Unternehmen relevanten Zieleigenschaften der IT-Architektur kann die Architekturmanagement-Funktion wesentlich zu dem Erreichen der Geschäftsziele des Unternehmens beitragen (Ahlemann et al. 2012).

Des Weiteren kann die Einnahme der hier vorgestellten Perspektive auf die Entwicklung der strukturellen Komplexität bei der Ausgestaltung der IT-Strategie eines Unternehmens unterstützen. So können beispielsweise durch eine Abstimmung der Unternehmensziele mit den Zieleigenschaften der IT-Architektur grundsätzliche Entscheidungen über den erwünschten Grad an Standardisierung in verschiedenen Bereichen des Unternehmens getroffen werden.

Die Pfadabhängigkeit von Architekturentscheidungen stellte eine der Studie zugrunde liegende Annahme dar. So sind sowohl unnötig komplexe Strukturen in der IT-Architektur als auch evtl. nicht vollständig zurückgebaute Komponenten nicht mehr genutzter Geschäftsprozesse die Basis für die Umsetzung neuer Anforderungen. An dieser Stelle sollten Unternehmen regelmäßig überprüfen, inwiefern die Vielfalt der bestehenden Prozesse und IT-Systeme tatsächlich noch erforderlich ist. Beispielsweise könnten im Rahmen der Umsetzung einer neuen Anforderung die involvierten Prozesse und Systeme dahingehend überprüft werden. Identifizierte unnötige Komplexität könnte dabei, je nach Umfang, entweder im Zuge der Umsetzung der Anforderung oder im Rahmen eines eigenständigen Vorhabens abgebaut werden.

4.5.3 Limitationen und weiterer Forschungsbedarf

Diese Studie unterliegt einigen Limitationen. Hierbei ist primär die begrenzte Verallgemeinerbarkeit von Fallstudienresultaten anzuführen. Aufgrund der Verwendung eines Single-Case-Ansatzes können die Ergebnisse grundsätzlich mit vorliegenden unternehmensspezifischen Gegebenheiten zusammenhängen. Allerdings werden die Ergebnisse aufgrund des Abstraktionsniveaus und der durchaus für ein Großunternehmen typischen Ausgestaltung der IT als grundsätzlich auch auf andere Unternehmen übertragbar angesehen. Der Zugang zu dem Unternehmen ermöglichte das Erlangen eines tiefen Verständnisses für die identifizierten Zusammenhänge. Es ist jedoch davon auszugehen, dass die Mengen der identifizierten Ausprägungen für Trigger (vgl. Abschnitt 4.4.1), Einflüssen von Anforderungen auf die minimale IT-Komplexität (vgl. Abschnitt 4.4.2), Mechanismen (vgl. Abschnitt 4.4.3) sowie Zieleigenschaften (vgl. Abschnitt 4.4.4) sicherlich nicht erschöpft sind. Weitere Untersuchungen könnten diese erweitern. Hierzu bietet sich die Verwendung der für diese Studie erarbeiteten Herangehensweise im Kontext anderer Unternehmen an.

Darüber hinaus könnten weitere Studien mit der Unternehmensgruppe die Veränderung der zu beobachtenden Mechanismen nach der Einführung eines Architekturmanagements untersuchen. An dieser Stelle wäre interessant zu beobachten, wie sich der Umfang der Kompetenzen des Architekturmanagements auf die entstehende unnötige Komplexität auswirkt.

Außerdem besteht weiterer Forschungsbedarf im Hinblick auf das Zusammenspiel der in Abschnitt 4.2.2 diskutierten Path Dependence Theory mit dem in dieser Studie betrachteten Phänomen unnötiger struktureller Komplexität von IT-Architekturen. Diese stellen gemeinsam einen Erklärungsansatz für die Abweichung der tatsächlichen von der optimalen strukturellen Komplexität der IT-Architektur dar. In diesem Zusammenhang könnte eine weitere Studie untersuchen, inwieweit die Pfadabhängigkeit bei der Weiterentwicklung der IT-Architektur die Entstehung sowie die Auswirkungen unnötiger Komplexität (beispielsweise auf die Zieleigenschaften) verstärkt.

5 Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen⁴⁵

In diesem Kapitel wird eine Maßzahl zur Quantifizierung struktureller Komplexität in Unternehmens-/IT-Architekturen erarbeitet. Nach einer Motivation im folgenden Unterkapitel 5.1 werden in Unterkapitel 5.2 verwandte Arbeiten in angrenzenden Forschungsgebieten sowie der praxisorientierten Literatur diskutiert. Anschließend wird in Unterkapitel 5.3 die GQM-Methode zur Ableitung von Kennzahlen eingeführt, die zur Erarbeitung der vorgeschlagenen Maßzahl Verwendung findet. Wie aus der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung deutlich wird, stellt die Heterogenität eine wesentliche Dimension der Komplexität dar. Die an ein Maß zur Quantifizierung von Heterogenität zu stellenden Anforderungen werden in Unterkapitel 5.4 dargestellt. Im Anschluss daran werden in Unterkapitel 5.5 eine Reihe häufig verwendeter Konzentrationsmaße als Kandidaten für die Quantifizierung von Heterogenität im Hinblick auf die eingeführten Anforderungen analysiert. Ergänzend werden in Unterkapitel 5.6 mit Richness und Evenness zwei Konzepte aus der Biologie eingeführt und auf den betrachteten Kontext übertragen. Auf Grundlage der Ergebnisse der Unterkapitel 5.5 und 5.6 wird in Unterkapitel 5.7 eine auf einem Maß aus der Informationstheorie (der Shannon-Entropie) basierende Form der Quantifizierung struktureller Komplexität in IT-Architekturen vorgeschlagen. In Unterkapitel 5.8 wird dieses Maß in drei Fallbeispielen mit Unternehmen aus der Finanz- und Versicherungsbranche sowie aus dem Transportsektor angewendet. Unterkapitel 5.9 beschließt das Kapitel mit einer Zusammenfassung und Diskussion der Ergebnisse.

5.1 Motivation

Wie in der vorangegangenen qualitativen Untersuchung gezeigt wurde, hat die Veränderung der strukturellen Komplexität von IT-Architekturen u. a. einen Einfluss auf die IT-Effizienz, die IT-Flexibilität sowie die Verfügbarkeit der IT-Systeme. Durch die vielgestaltigen Wirkungen ist die strukturelle Komplexität der IT-Architektur als wesentliche Steuerungsgröße des IT-Managements anzusehen (Schmidt et al. 2013). Da sich strukturelle Komplexität durch eine Veränderung der IT-Architektur einstellt, fällt die Aufgabe eines Managements der Entwicklung der Komplexität in erster Linie dem Architekturmanagement zu. Voraussetzung für die Beurteilung eines bestimmten Grades oder der Veränderung struktureller Komplexität ist deren Messbarkeit. Nur durch die Messung ist es möglich, geplante Veränderungen gezielt in Richtung eines angemessenen Grades an Komplexität zu beeinflussen sowie einen Handlungsbedarf abzuleiten. Wie bereits in Abschnitt 3.1.3 angeführt, wurde in der Literatur im Bereich des Information Systems Research bislang noch kein holistischer, auf alle Architek-

⁴⁵ Vgl. nachfolgend Schütz et al. (2013b).

turebenen (vgl. Abschnitte 2.2.3 sowie 2.2.4) anwendbarer Ansatz zur Quantifizierung struktureller Komplexität vorgeschlagen. Seitens der Praxis finden primär spezifische Indikatoren Anwendung, die lediglich auf einzelne Facetten der Komplexität abzielen. Da diese Indikatoren zudem häufig auf unterschiedlichen Berechnungsmethoden basieren, sind diese nur schwer vergleich- und kommunizierbar (Schmidt et al. 2013). Basierend auf der in Unterkapitel 3.2 hergeleiteten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen, wird in diesem Kapitel eine Möglichkeit zur Quantifizierung von Komplexität eingeführt. In diesem Kapitel wird demnach die vierte Forschungsfrage der vorliegenden Dissertation bearbeitet:

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/ IT-Architekturen quantifiziert werden?

Hierzu werden bestehende Ansätze anderer (z. B. der Biologie oder der Volkswirtschaftslehre) oder angrenzender (z. B. dem Softwareengineering) Forschungsfelder herangezogen und deren Übertragbarkeit auf den Kontext der Unternehmens- und IT-Architekturen untersucht.

5.2 Verwandte Literatur

Im Feld der Informatik existiert eine Reihe an Konzepten zur Quantifizierung unterschiedlicher Formen von Komplexität. In der theoretischen Informatik wird der Begriff „computational complexity“ verwendet, um darzustellen, wie effizient ein Problem gelöst werden kann (Ladner 1977; Rudie & Willems 1995). Eine vergleichbare Betrachtung erfolgt im Bereich der Algorithmik. Die „Kolmogorov Komplexität“ eines binären Strings ist hierbei gleichbedeutend mit der minimalen Länge eines Programms, welches diesen String erzeugt (Muehnik & Vereshchagin 2001). Mit der sogenannten „cyclomatic number“ führt McCabe (1976) ein Maß zur Bestimmung der Komplexität des Ablaufes einer Anwendung ein. Hierbei werden die Anwendungen als gerichtete Ablaufgraphen strukturiert und die Anzahl der linear unabhängigen Pfade bestimmt. Ein weiteres klassisches Maß zur Bestimmung der Komplexität einer Anwendung ist die Ermittlung der Anzahl der Zeilen des Quellcodes (Booch 2008).

Alle hier angeführten Maße zur Quantifizierung von Komplexität aus dem Bereich der Informatik sind spezifisch ausgerichtet auf die Bewertung der Komplexität eines zu lösenden Problems bzw. der einer Anwendung, die ein Problem löst. Für eine Verwendung zur Quantifizierung der Komplexität auf anderen Ebenen der IT-Architektur erscheinen sie daher ungeeignet.

Ein weiterer Ansatz aus der Literatur, der nicht unmittelbar mit der Quantifizierung von Komplexität in Verbindung gebracht wird, ist der von Lagerström et al. (2014). Hierbei wird die sogenannte Design Structure Matrix verwendet, um aus dem Grad der Vernetzung der Systemkomponenten Rückschlüsse auf die Struktur der Anwendungslandschaft ziehen zu können und hierdurch Bereiche zu identifizieren, die einen höheren Grad an Komplexität aufweisen. Auf Grundlage einer typologischen Betrachtung der Anwendungslandschaft können verschiedenen Typen von Architektur-Strukturen abgeleitet werden. Dieser Ansatz ist zwar grundsätzlich auch auf andere Architekturebenen übertragbar, jedoch bleibt unklar, wel-

che Schlüsse aus den Ergebnissen im Hinblick auf die Komplexität der IT-Architektur gezogen werden können. Durch den Fokus auf die Struktur der Vernetzung bleiben die spezifischen Attribute der Systemkomponenten (wie z. B. der Hersteller oder der Typ der Schnittstelle zwischen den Systemkomponenten) außen vor, wodurch eine Beurteilung der Heterogenität der IT-Architektur nicht möglich ist.

In der Literatur zu Unternehmensarchitekturen schlägt Dern (2011) in seiner Arbeit ein Kennzahlensystem zur Ermittlung der Gesamtkomplexität einer IT-Landschaft vor. Dieses setzt sich aus Kennzahlen für die wahrgenommene geschäftliche Komplexität und zur Bewertung der Anwendungslandschaft sowie der IT-Infrastruktur zusammen. Auch in der Arbeit von Engels et al. (2008) werden eine Reihe quantitativer Bewertungskriterien für Anwendungslandschaften vorgestellt. Diese Indikatoren zielen auf die Bereiche „funktionale Redundanz“, „fachliche Strukturierung“, „Domänenreinheit“, „Kategorienreinheit“ und „nichtzyklische Abhängigkeiten“ ab.

Die Ansätze von Dern (2011) und Engels et al. (2008) bieten zwar für verschiedene Ebenen der IT-Architektur Möglichkeiten zur Quantifizierung der Komplexität, jedoch folgen diese keinem einheitlichen Konzept von Komplexität. Die unterschiedlichen Berechnungsvorschriften der Kennzahlen für die einzelnen Teilbereiche weisen zudem keine klare theoretische Basis auf.

Insgesamt konnte in den angrenzenden Forschungsgebieten und der praktisch orientierten Literatur kein holistischer Ansatz zur Quantifizierung struktureller Komplexität von IT-Architekturen identifiziert werden.

5.3 Methodisches Vorgehen

In der Literatur existiert eine Reihe unterschiedlicher Ansätze zur Definition von Maßzahlen (Fenton & Pfleeger 1998; Van Solingen & Berghout 1999). Diese Arbeit verwendet die systematische Vorgehensweise des Goal Question Metric Paradigmas (Basili & Rombach 1988), um Maßzahlen für die Quantifizierung struktureller Komplexität von Unternehmens-/IT-Architekturen zu entwickeln. Goal Question Metric (GQM) ist ein von Basili & Rombach (1988) vorgeschlagener systematischer und „zielorientierter“ Ansatz für die Herleitung von Maßzahlen, welcher vorzugsweise im Feld des Softwareengineerings Anwendung findet. GQM sieht dabei eine dreistufige Vorgehensweise vor:

- (1) Definition eines Zieles, basierend auf eine Menge an Kriterien
- (2) Ableitung von Fragestellungen auf Grundlage des formulierten Zieles
- (3) Definition von Maßzahlen, die geeignet sind, die Fragestellungen in Form einer Quantifizierung zu beantworten

Diese Vorgehensweise wird in diesem Kapitel als Methode zur Definition einer abstrakten Maßzahl zur Quantifizierung struktureller Komplexität in Unternehmens-/IT-Architekturen verwendet. Die Zielsetzung beschreibt hierbei, welches Wissen durch die Messung hinzu-erwonnen werden soll. Zur systematischen Formulierung und Dokumentation des Zieles der Messung kann das durch Basili & Rombach (1988) eingeführte Template herangezogen wer-

den. Hierdurch wird sichergestellt, dass die durch ein GQM-Ziel abzudeckenden Aspekte⁴⁶ berücksichtigt werden. Durch Anwendung des Templates zur Formulierung der Zielsetzung wird das folgende **Ziel** definiert:

*„Analysiere die Unternehmensarchitektur als System **zum Zwecke** diese zu charakterisieren **in Bezug auf** die strukturelle Komplexität **vom Blickwinkel** der Unternehmens-/IT-Architekten“.*

Bei dieser Zieldefinition ist die Bezugnahme auf „die strukturelle Komplexität“ bewusst generisch formuliert. Der Architekt hat hier die Freiheit zu wählen, nach welchen Kriterien die strukturelle Komplexität der Architektur beurteilt werden soll. Beispielsweise kann eine Betrachtung der IT-Architektur anhand der eingesetzten Typen von Datenbankmanagementsystemen erfolgen. Gleichermaßen ist der Gegenstand der Analyse mit der „Unternehmensarchitektur als System“ weit gefasst. Dies kann genau in dieser Form verwendet, jedoch auch auf einzelne Teilbereiche der Architektur, also die Subsysteme der Gesamtarchitektur, angewendet werden. Hierbei hat der Architekt die Möglichkeit, den Fokus der Analyse gezielt auf eine bestimmte Architekturebene oder auch Ausschnitte für einzelne Bereiche der Organisation zu setzen.

Basierend auf der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen werden die folgenden **Fragestellungen** aus der angeführten Zielsetzung abgeleitet:

(Q1) Wie hoch ist die Anzahl der Komponenten und Beziehungen des Systems?

(Q2) Wie hoch ist die Heterogenität der Komponenten und Beziehungen des Systems?

Zur Sicherstellung der Konsistenz mit der in Abschnitt 3.2.1 eingeführten systemtheoretischen Definition von Komplexität in Unternehmens-/IT-Architekturen müssen die Maßzahlen, welche die angeführten Fragestellungen Q1 und Q2 beantworten, die folgenden Anforderungen erfüllen:

- Je größer die Anzahl N – bei konstanter Heterogenität H – der Komponenten und Beziehungen einer Unternehmens-/IT-Architektur, desto höher die strukturelle Komplexität der Unternehmens-/IT-Architektur (vgl. Q1).
- Je größer die Heterogenität H – bei konstanter Anzahl N – der Komponenten und Beziehungen einer Unternehmens-/IT-Architektur, desto höher die strukturelle Komplexität der Unternehmens-/IT-Architektur (vgl. Q2).

Während die Anzahl N relativ einfach zu bestimmen ist (durch Abzählen der Komponenten und Beziehungen des Systems), sind für die Bestimmung der Heterogenität H nähere Betrachtungen erforderlich. Im folgenden Unterkapitel werden Anforderungen an ein Heterogenitätsmaß aus der Literatur zusammengetragen.

⁴⁶ Diese Aspekte umfassen das Objekt und den Zweck der Messung, den Qualitätsfokus sowie den im Zuge der Messung eingenommenen Blickwinkel.

5.4 Anforderungen an ein Maß zur Quantifizierung von Heterogenität

Der Begriff Heterogenität findet in der IT in einer Vielzahl von Kontexten Anwendung. Die vorliegende Arbeit folgt der generischen Definition von Heterogenität nach Widjaja et al. (2012):

“Heterogeneity in IT landscapes is a statistical property and refers to the diversity of attributes of elements in the IT landscape.” (Widjaja et al. 2012, S. 4)

Dabei können die in der Definition angeführten Elemente (*elements*) sowohl als Komponenten als auch Beziehungen einer IT-Architektur verstanden werden. Die Attribute (*attributes*) stellen Eigenschaften dar, welche die Komponenten oder Beziehungen näher beschreiben. Im Falle der Datenbankmanagementsysteme (als Komponenten) können beispielsweise der Hersteller, der Typ, die Version sowie das Release der Datenbank die Attribute der Elemente darstellen. Die tatsächlich in der IT-Landschaft vorkommenden Ausprägungen von Attributkombinationen werden als Charakteristiken bezeichnet.

Als Beispiel hierfür ist in Abbildung 31 die Heterogenität der Datenbankmanagementsysteme (DB) eines Unternehmens angeführt. Diese Heterogenität kann als eine Häufigkeitsverteilung verstanden und anhand eines Histogramms visualisiert werden. Jede der in Abbildung 31 dargestellten Säulen repräsentiert eine Charakteristik. In diesem Fall setzen sich die Charakteristiken, nach der die Datenbankinstanzen unterschieden werden, aus dem Hersteller sowie der Typenbezeichnung der Datenbank zusammen.

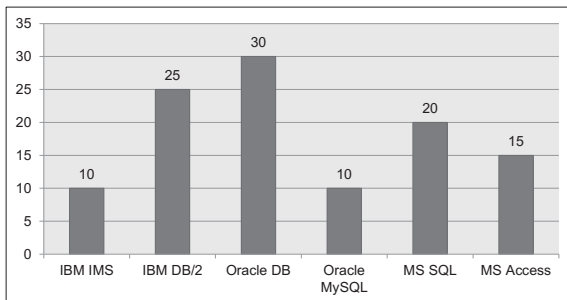


Abbildung 31: Heterogenität eingesetzter DBs anhand der absoluten Häufigkeiten

Eine bekannte Möglichkeit, Heterogenität zu messen, liegt in der Verwendung von Konzentrationsmaßen. Im Kontext kartellrechtlicher Überprüfungen wird von den amerikanischen Behörden der Herfindahl-Hirschmann-Index eingesetzt, um die Marktkonzentration vor bzw. nach einem möglichen Unternehmenszusammenschluss zu ermitteln (U.S. Department of Justice and the Federal Trade Commission 2010). Neben der Berechnung der Marktmacht von Firmen (Kwoka Jr 1985) werden Konzentrationsmaße in verschiedenen ökonomischen Themenfeldern, wie beispielsweise zur Messung der geographischen Konzentration von Industrien (Garrison & Paulson 1973), herangezogen.

Um in der weiteren Untersuchung eine Eignung unterschiedlicher Maße zur Quantifizierung von Heterogenität beurteilen zu können, wird zunächst eine Menge von Eigenschaften eingeführt, welche durch die infrage kommenden Maße erfüllt werden sollten. In der wissenschaftlichen Literatur existiert eine Reihe von Vorarbeiten zu Konzentrationsmaßen und den Anforderungen, die an diese gestellt werden. In den folgenden Abschnitten werden die wesentlichen Konzepte aufgeführt und auf den Kontext der Unternehmens-/IT-Architekturen übertragen.⁴⁷

5.4.1 Zusätzliche Charakteristiken

Eine Charakteristik ist ein Tupel von Attributen einer Komponente oder Beziehung eines Systems. In dem in Abbildung 31 dargestellten Beispiel besteht die Charakteristik, nach der eine Komponente (hier ein Datenbankmanagementsystem) unterschieden wird, aus dem Hersteller und der Bezeichnung des Produktes. Würden weitere Komponenten eingeführt werden, deren Attribute von bereits bestehenden Komponenten abweichende Ausprägungen aufweisen (beispielsweise eine Adabas-Datenbank der Firma Software AG), so würde sich die Heterogenität der IT-Landschaft erhöhen (vgl. die Veränderung der Heterogenität H_A von Architektur A hin zu H_B von Architektur B in Abbildung 32). Diese hinzugefügten Komponenten würden sich dabei in einer weiteren Charakteristik (Software AG, Adabas) zeigen.

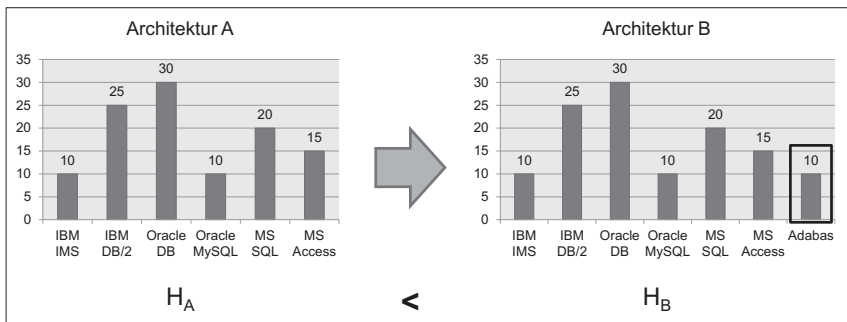


Abbildung 32: Zusätzliche Charakteristiken

Wird demgegenüber eine der bestehenden Charakteristiken vollständig eliminiert, beispielsweise durch eine Migration der Daten der Oracle MySQL Datenbanken auf die MS SQL Lösung, so führt dies zu einer Verringerung der Heterogenität (vgl. Abbildung 33).

⁴⁷ Die Arbeit von Kohlweyer (2011) ist in die folgenden Ausführungen eingeflossen.

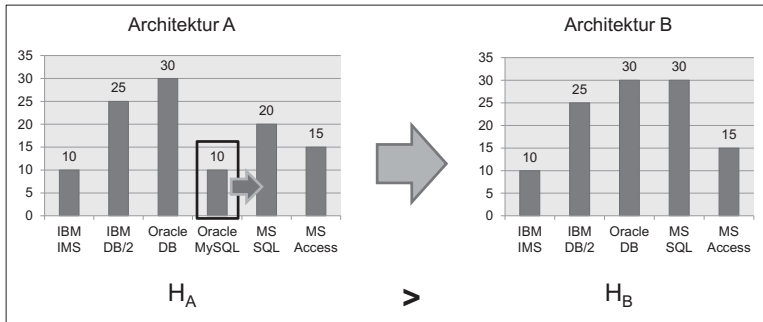


Abbildung 33: Eliminieren einer Charakteristik

Bei der Beurteilung der Konzentration von Märkten entspricht diese Betrachtungsweise dem Markteintritt eines neuen bzw. dem Marktaustritt eines bestehenden Wettbewerbers. Dabei wird angenommen, dass eine entgegengesetzte Reaktion von Konzentrationsmaßen auf eine derartige Veränderung erfolgt. Ein Markteintritt geht dabei mit einer geringeren und ein Marktaustritt entsprechend mit einer höheren Marktkonzentration einher (Häni 1987; Marfels 1972b). Hierbei ist zu beachten, dass dies jeweils nur dann gilt, wenn der zugehende bzw. der abgehende Anteil eine bestimmte Größe am Gesamtmarkt nicht überschreitet. Ab einer gewissen Größe des zugeordneten relativen Marktanteiles einer neuen Charakteristik, ist mit einer Zunahme der Konzentration zu rechnen. Gleiches gilt für den Marktaustritt eines Wettbewerbers mit einem bislang sehr großen Marktanteil.

Aus diesen Betrachtungen ergibt sich die erste Anforderung an ein Maß zur Quantifizierung von Komplexität:

Anforderung 1: Die Einführung zusätzlicher Charakteristiken (mit einem ausreichend geringen Anteil an der Menge der Elemente) führt zu einer Erhöhung und das Eliminieren von Charakteristiken entsprechend zu einer Verringerung der Heterogenität.

5.4.2 Effekt kleiner Gruppen von Elementen

Eine kleine Gruppe von Elementen ist als eine Software oder Technologie zu verstehen, die einen vergleichsweise geringen Teil der IT-Architektur ausmacht. Ein Beispiel hierfür könnte eine Software eines Herstellers sein, von dem nur wenige oder keine weiteren Produkte in der IT-Architektur integriert sind. Auch solche kleinen Gruppen von Elementen stellen aus der Perspektive der IT-Architektur eine hoch relevante Größe dar, da diese einen spürbaren Einfluss auf das Management der IT-Architektur haben können (vgl. hierzu Abbildung 34).

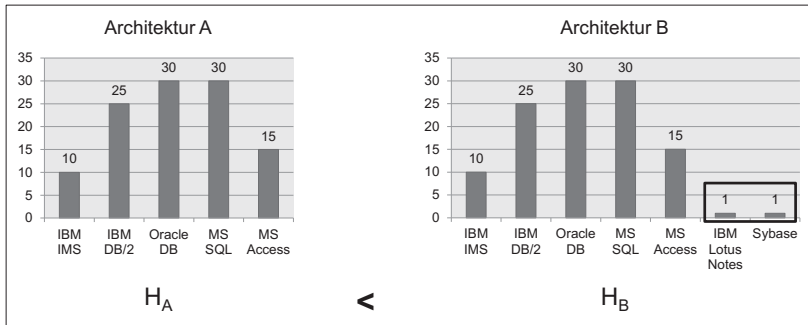


Abbildung 34: Effekt kleiner Gruppen von Elementen

Beispielsweise können solche Komponenten über spezifische, proprietäre Schnittstellen verfügen. Hierdurch entstehende Kompatibilitätsprobleme zwischen den Systemkomponenten können dabei zu einem deutlichen Mehraufwand im Zuge der Umsetzung von Änderungen führen. Neben dem technisch bedingten Aufwand spielen auch organisatorische Aspekte eine Rolle. So muss auch für solche Systeme Personal mit spezifischem Know-how vorgehalten sowie eine Dokumentation erstellt und nachgehalten werden.

An dieser Stelle weicht die Betrachtungsweise von der Literatur zur Messung der Konzentration von Märkten ab. Hier wird angeführt, dass ein Konzentrationsmaß nur geringfügig auf Unternehmen mit geringen Marktanteilen reagierte. Der Grundgedanke liegt dabei darin, dass solche Unternehmen einen überschaubaren Einfluss auf den Markt haben (Häni 1987). Zudem lassen sich die potenziell sehr große Menge an kleinen Unternehmen sowie deren Marktanteile nur mit sehr großem Aufwand bestimmen (Bruckmann 1969).

Um diesen Effekt kleiner Gruppen in einem Maß zur Quantifizierung von Heterogenität zu berücksichtigen, ist die folgende Anforderung zu erfüllen:

Anforderung 2: Charakteristiken mit einer geringen Anzahl zugeordneter Elemente werden merklich durch das Heterogenitätsmaß berücksichtigt.

5.4.3 Verschiebung in der Klassifikation

Unter einer Verschiebung in der Klassifikation ist eine Veränderung der Zuordnung einer oder mehrerer Komponenten zu den Charakteristiken zu verstehen. Als Beispiel kann erneut die Heterogenität der DBs betrachtet werden. Im Falle einer Migration einiger Datenbankinstanzen von einem vergleichsweise selten vorkommenden DBs (z. B. MS Access) hin zu einer relativ dominanten Technologie (z. B. MS SQL) verringert sich die Heterogenität der IT-Landschaft (vgl. Abbildung 35).

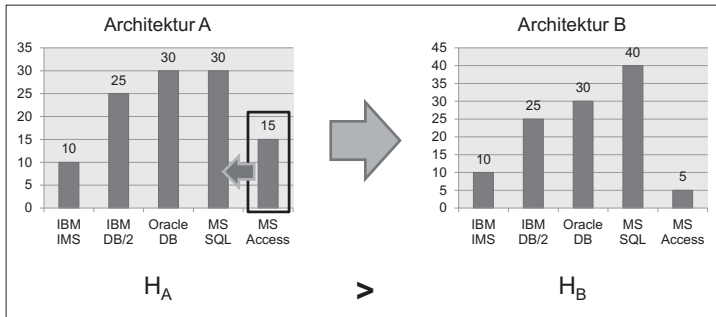


Abbildung 35: Verschiebung in der Klassifikation

Im entgegengesetzten Fall einer Verlagerung von Komponenten einer dominanten hin zu einer deutlich weniger dominanten Technologie, muss es jedoch nicht zwingend zu einer Erhöhung der Heterogenität kommen. Ist die Anzahl der verlagerten Komponenten so groß, dass der zuvor weniger dominanten Technologie nach der Verlagerung eine größere Anzahl an Komponenten zuzuordnen ist als der dominanten Technologie vor der Verlagerung, so kommt es ebenfalls zu einer Verringerung der Heterogenität. Sind der vor der Verlagerung weniger dominanten Technologie auch nach der Verlagerung weiterhin weniger Komponenten zuzuordnen als der dominanten Technologie vor der Verlagerung, so kommt es zu einer Erhöhung der Heterogenität.

Im Kontext der Marktkonzentration ist die angeführte Verschiebung in der Klassifikation gleichbedeutend mit einer Verschiebung der Marktanteile der Unternehmen. Ringt ein bereits relativ dominantes Unternehmen einem Wettbewerber mit geringerem Marktanteil weitere Marktanteile ab, so entspricht dies einer Erhöhung der Marktkonzentration (Bruckmann 1969; Marfels 1972b). Im Fall einer Verschiebung der Marktanteile von einem relativ dominierenden Wettbewerber hin zu einem Unternehmen mit geringerem Marktanteil kommt es grundsätzlich zu einer Verringerung der Marktkonzentration (Häni 1987). „Grundsätzlich“ daher, da hier analog die oben angeführte Einschränkung bzgl. des Umfangs der von der Verschiebung betroffenen Marktanteile für den Effekt auf die Konzentration eine Rolle spielt.

In Anlehnung an Gollop & Monahan (1991) lautet die dritte Anforderung an ein Maß zur Quantifizierung von Heterogenität, welche die angeführten Sonderfälle bei einer Verschiebung der Klassifikation berücksichtigt, wie folgt:

Anforderung 3: Eine Erhöhung der Ungleichheit in der Verteilung der Elemente auf die Charakteristiken führt zu einer Verringerung der Heterogenität (und umgekehrt).

5.4.4 Effekt proportionaler Änderungen

Die vierte Anforderung an ein Heterogenitätsmaß zielt darauf ab, dass das Maß nicht durch proportionale Veränderungen der den Charakteristiken zugeordneten Elemente beeinflusst wird. In Abbildung 36 sind die Verteilungen der Datenbankinstanzen zweier IT-Architekturen dargestellt. Diese unterscheiden sich dabei weder in der Anzahl der Charakteristiken, noch in

den relativen Häufigkeiten der Ausprägungen der jeweiligen Charakteristiken. Der einzige Unterschied zwischen diesen beiden Architekturen liegt darin, dass in Architektur B jeder der Charakteristiken exakt die doppelte Anzahl von Komponenten zugeordnet ist. Ein Heterogenitätsmaß sollte diese beiden Architekturen als gleich heterogen bewerten.

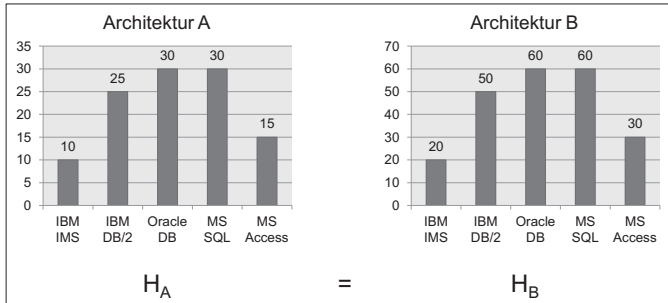


Abbildung 36: Effekt proportionaler Änderungen

Reine Größeneffekte einer IT-Architektur sollen demzufolge durch das Heterogenitätsmaß keine Berücksichtigung finden. Vielmehr soll sich das Maß auf eine Quantifizierung der Verteilung der Elemente auf eine Menge von Charakteristiken beschränken (Hall & Tideman 1967). Hierdurch wird u. a. ermöglicht, dass die Heterogenität unterschiedlicher Unternehmen, Bereiche oder Abteilungen miteinander verglichen werden kann. Besagte Größeneffekte werden hingegen explizit durch Frage Q1 („Wie hoch ist die Anzahl der Komponenten und Beziehungen des Systems?“) und der dieser zugeordneten Maßzahl berücksichtigt.

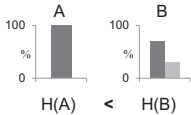
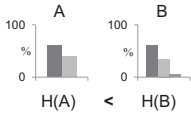
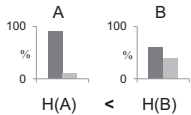
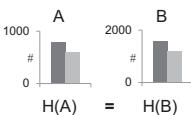
Analog gilt auch bei der Bewertung der Marktkonzentration, dass die absolute Größe des Marktes bei der Beurteilung der Machtverhältnisse der Wettbewerber nicht weiter relevant ist (Häni 1987).

Aus diesen Überlegungen ergibt sich die vierte Anforderung an ein Maß zur Quantifizierung der Heterogenität:

Anforderung 4: Proportionale Veränderungen der den Charakteristiken zugeordneten Elemente haben keinen Einfluss auf die Heterogenität.

In Tabelle 7 werden die in den vorangegangenen Abschnitten eingeführten Anforderungen zusammengefasst. Diese werden in dem folgenden Unterkapitel 5.5 für die Bewertung einer Reihe ausgewählter Maßzahlen herangezogen.

Tabelle 7: Anforderungen an ein Maß zur Quantifizierung von Heterogenität⁴⁸

Anforderung	Beschreibung	Beispiel	Oracle DB/2 IMS DB	Literatur
Zusätzliche Charakteristiken	Die Einführung zusätzlicher Charakteristiken (mit einem ausreichend geringen Anteil an der Menge der Elemente) führt zu einer Erhöhung der Heterogenität und das Eliminieren von Charakteristiken entsprechend zu einer Verringerung	 <p>H(A) < H(B)</p>		Häni (1987), Marfels (1972b)
Effekt kleiner Gruppen von Elementen	Charakteristiken mit einer geringen Anzahl zugeordneter Elemente werden merklich durch das Heterogenitätsmaß berücksichtigt	 <p>H(A) < H(B)</p>		Häni (1987), Bruckmann (1969)
Verschiebung in der Klassifikation	Eine Erhöhung der Ungleichheit in der Verteilung der Elemente auf die Charakteristiken führt zu einer Verringerung der Heterogenität (und umgekehrt)	 <p>H(A) < H(B)</p>		Häni (1987), Bruckmann (1969), Gollop & Monahan (1991), Marfels (1972b)
Effekt proportionaler Änderungen	Proportionale Veränderungen der den Charakteristiken zugeordneten Elemente haben keinen Einfluss auf die Heterogenität	 <p>H(A) = H(B)</p>		Hall & Tideman (1967), Häni (1987)

5.5 Analyse häufig verwendeter Konzentrationsmaße

In diesem Unterkapitel werden die im Rahmen der Arbeit betrachteten Kennzahlen eingeführt und anschließend deren Eignung für eine Verwendung als Maß zur Quantifizierung von Heterogenität dargestellt.⁴⁹ Bei den Kennzahlen handelt es sich um den Herfindahl-Hirschman-Index (Abschnitt 5.5.1), den Horvath-Index (Abschnitt 5.5.2) sowie die Shannon-Entropie (Abschnitt 5.5.3). Bei den in den folgenden Abschnitten dargestellten Berechnungsvorschriften der Kennzahlen stellt p_i die relative Häufigkeit einer Charakteristik i in einer IT-Architektur dar. x_i bezeichnet die absolute Anzahl von Elementen, die einer Charakteristik i zugeordnet sind und n die Anzahl der verschiedenen in einer IT-Architektur identifizierbaren Charakteristiken:

$$(6) \quad p_i = \frac{x_i}{\sum_{j=1}^n x_j}$$

⁴⁸ Vgl. hierzu auch Schütz et al. (2013b).

⁴⁹ Vgl. hierzu auch Kohlweyer (2011).

5.5.1 Herfindahl-Hirschman-Index

Der Herfindahl-Hirschman-Index (HHI) ist eine der am weitesten verbreiteten Kennzahlen im Bereich der Konzentrationsmessung. Benannt ist der HHI nach den Ökonomen Orris Clemens Herfindahl und Albert Otto Hirschman, die den Index unabhängig voneinander entwickelten (Herfindahl 1950; Hirschman 1945; Hirschman 1964). Verwendung findet der HHI u. a. durch amerikanische Institutionen, wie das Department of Justice und die Federal Reserve, zur Bestimmung der Wirkung von Unternehmenszusammenschlüssen auf die Wettbewerbssituation in Märkten (U.S. Department of Justice and the Federal Trade Commission 2010).

Wie die meisten Konzentrations- und Diversifikationsmaße hat der HHI die folgende Struktur (Jacquemin & Berry 1979):

$$(7) \quad \text{Index} = \sum_{i=1}^n p_i w_i$$

Hierbei steht w_i für die Gewichtung der jeweiligen Charakteristik i . Im Falle des HHI werden die relativen Häufigkeiten der Charakteristiken (oder die relativen Marktanteile der Unternehmen) mit sich selbst gewichtet. Hieraus ergibt sich die folgende Berechnungsvorschrift für den HHI:

$$(8) \quad HHI = \sum_{i=1}^n p_i^2$$

Der Wertebereich des HHI lautet wie folgt (Piesch 1975):

$$(9) \quad \frac{1}{n} \leq HHI \leq 1$$

Es sei an dieser Stelle erwähnt, dass sich der HHI als Konzentrationsmaß exakt entgegengesetzt verhält, wie es intuitiv von einem Heterogenitätsmaß zu erwarten ist. Eine hohe Konzentration einer Charakteristik ist demnach gleichbedeutend mit einer geringen Heterogenität. Um den HHI in ein Heterogenitätsmaß zu überführen muss lediglich die folgende Transformation vorgenommen werden:

$$(10) \quad H_{HHI} = 1 - HHI = 1 - \sum_{i=1}^n p_i^2$$

Die Beurteilung des HHI hinsichtlich der in Unterkapitel 5.4 angeführten Anforderungen erfolgt im Weiteren in der ursprünglichen Form.

Bezüglich der in Abschnitt 5.4.1 dargestellten Anforderung in Bezug auf den Zugang zusätzlicher bzw. das Eliminieren bestehender Charakteristiken reagiert der HHI erwartungsgemäß. Entgegen der Feststellung von Hart (1971), dass der HHI mit einer zunehmenden Anzahl an Charakteristiken stets abnehme, weist Piesch (1975) nach, dass die Reaktion des HHI von der Größe des Anteiles der zugehenden (oder abgehenden) Charakteristik abhängt. Dies soll am folgenden Beispiel einer Verteilung von 158 Elementen auf sieben Charakteristiken verdeutlicht werden:

$$(11) \quad x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (80, 40, 20, 10, 5, 2, 1)$$

Der Zugang einer zusätzlichen Charakteristik, der x_{neu} Elemente zuzuordnen sind, verursacht eine Veränderung des HHI um ΔHHI , eine Funktion von x_{neu} :

$$(12) \quad \text{delta_HHI} = \left(\sum_{i=1}^7 \left(\frac{x_i}{\sum_{j=1}^7 x_j + x_{neu}} \right)^2 \right) + \left(\frac{x_{neu}}{\sum_{j=1}^7 x_j + x_{neu}} \right)^2 - \sum_{i=1}^7 \left(\frac{x_i}{\sum_{j=1}^7 x_j} \right)^2$$

Abbildung 37 ist ein Auszug des Verlaufes der Funktion *delta_HHI* zu entnehmen. Im Bereich $0 < x \leq 164,0185$ nimmt die Funktion negative Werte an. Dies bedeutet, dass sich der HHI für diese Verteilung bei Zugang einer Charakteristik mit bis zu 164 Elementen verringert (die Heterogenität also steigt). Ab 165 Elementen ist die neue Charakteristik hingegen so dominant, dass sich der HHI erhöhen würde (gleichbedeutend mit einer abnehmenden Heterogenität).

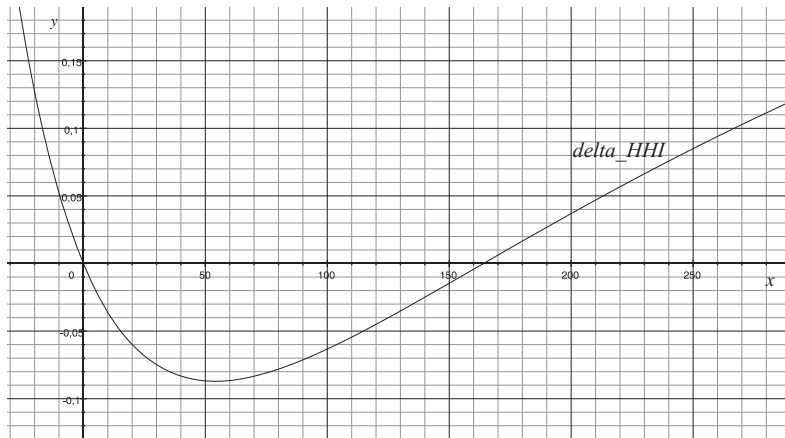


Abbildung 37: Zusätzliche Charakteristiken – HHI

Hinsichtlich der in Abschnitt 5.4.2 diskutierten Anforderung bezüglich des Effektes kleiner Gruppen von Elementen, spielt die Gewichtung der Charakteristiken eine wesentliche Rolle. Die Gewichtung der Anteile erfolgt bei dem HHI mit dem jeweiligen Anteil selbst, was sich nachteilig auf die zweite Anforderung an ein Heterogenitätsmaß auswirkt. Charakteristiken mit großen Anteilen werden hierdurch relativ stark gewichtet, wohingegen Charakteristiken mit relativ geringen Anteilen eine entsprechend schwache Gewichtung erfahren. Wie bereits bei der Einführung dieser Anforderung angeführt, ist diese Eigenschaft für ein Maß zur Berechnung der Konzentration von Märkten durchaus erwünscht, da Unternehmen mit sehr kleinem Marktanteil dadurch einen verschwindend kleinen Einfluss auf die Höhe des Maßes haben. Für das gesuchte Maß zur Quantifizierung der Heterogenität von IT-Architekturen ist diese Eigenschaft des HHI hingegen von Nachteil. In Anlehnung an die Darstellung in Marfels (1972a) sind in Tabelle 8 die Werte für den HHI bei einer dominanten Charakteristik sowie einer Gleichverteilung des verbleibenden Anteiles auf $n-1$ Charakteristiken dargestellt. Bereits bei einem Anteil der dominanten Charakteristik von 0,25 ist bei dem Zugang der elften gleichverteilten Charakteristik lediglich eine Veränderung an der dritten Nachkommastelle zu beobachten. Bei einem Anteil der dominanten Charakteristik von 0,75 ist keine wesentliche Reaktion des HHI mehr zu erkennen. Der Effekt kleiner Gruppen von Elementen ist folglich als sehr gering einzustufen.

Tabelle 8: Effekt kleiner Gruppen von Elementen – HHI⁵⁰

Anteil der dominanten Charakteristik	n-1 gleichverteilte Charakteristiken								
	1	10	11	50	51	100	101	500	501
0,1	-	0,09100	0,08364	0,02620	0,02588	0,01810	0,01802	0,01162	0,01162
0,25	-	0,11875	0,11364	0,07375	0,07353	0,06813	0,06807	0,06363	0,06362
0,5	0,50000	0,27500	0,27273	0,25500	0,25490	0,25250	0,25248	0,25050	0,25050
0,75	0,62500	0,56875	0,56818	0,56375	0,56373	0,56313	0,56312	0,56263	0,56262
0,9	0,82000	0,81100	0,81091	0,81020	0,81020	0,81010	0,81010	0,81002	0,81002

Wie Marfels (1972b) zeigt, wird die in Abschnitt 5.4.3 eingeführte Anforderung, wie ein Maß zur Quantifizierung von Heterogenität auf die Verschiebung in der Klassifikation reagieren sollte, durch den HHI erfüllt (Marfels 1972b, S. 473 f.).

Die zudem geforderte Eigenschaft der Maßzahl, keine Reaktion bei proportionalen Veränderungen der Anzahl der Elemente (vgl. Abschnitt 5.4.4) zu zeigen, wird ebenfalls durch den HHI erfüllt. Da die Berechnung des HHI auf Grundlage der relativen Häufigkeiten der Charakteristiken erfolgt, spielen proportionale Veränderungen um einen Faktor a keine Rolle:

$$(13) \quad HHI' = \sum_{i=1}^n \left(\frac{ax_i}{\sum_{j=1}^n ax_j} \right)^2 = \sum_{i=1}^n \left(\frac{ax_i}{a \sum_{j=1}^n x_j} \right)^2 = \sum_{i=1}^n \left(\frac{x_i}{\sum_{j=1}^n x_j} \right)^2 = HHI$$

5.5.2 Horvath-Index

Der auf Janos Horvath zurückgehende Horvath-Index (HOR) wurde als Maß zur Quantifizierung der Industriekonzentration eingeführt und wird wie folgt berechnet (Horvath 1970):

$$(14) \quad HOR = p_1 + \sum_{i=2}^n p_i^2 (2 - p_i)$$

Hier fällt auf, dass der relative Anteil der ersten Charakteristik (p_1) eine besondere Behandlung erfährt, auf die an späterer Stelle noch eingegangen wird. p_i ist für die Berechnung des HOR dabei so zu wählen dass folgendes gilt:

$$(15) \quad p_1 \geq p_i \quad \forall i \neq 1$$

Es handelt sich bei p_1 demnach um den größten relativen Anteil. Der Wertebereich des HOR kann wie folgt angegeben werden (Marfels 1972a):

$$(16) \quad \frac{3n^2 - 3n + 1}{n^3} \leq HOR \leq 1$$

Anhand des bereits im vorangegangenen Abschnitt verwendeten Beispiels der Verteilung aus Formel 11 kann auch das Verhalten des HOR bei dem Zugang zusätzlicher Charakteristiken nachvollzogen werden. Δ_{HOR} stellt dabei die von x_{neu} abhängige Funktion der Veränderung des HOR bei Erweiterung der Charakteristiken um eine weitere Charakteristik bestehend aus x_{neu} vielen Elementen dar. Wie Formel 19 zu entnehmen ist, muss bei der Funktion der Veränderung des HOR eine Fallunterscheidung vorgenommen werden. Dabei berücksichtigt

⁵⁰ Eine erweiterte Darstellung in Anlehnung an Marfels (1972a), S. 204.

die Berechnungsvorschrift, ob x_{neu} die Charakteristik mit dem größten relativen Anteil darstellt (vgl. Formel 15), was einen Einfluss auf die Gewichtung des relativen Anteiles von x_{neu} hätte:

$$(17) \quad HOR_{alt} = (p_1 + \sum_{i=2}^7 p_i^2 (2 - p_i))$$

$$(18) \quad X = \sum_{i=1}^7 x_i + x_{neu}$$

$$(19) \quad \text{delta_HOR} = \begin{cases} \left(\frac{x_1}{X} + \sum_{i=2}^7 \left(\frac{x_i}{X} \right)^2 \left(2 - \frac{x_i}{X} \right) \right) + \left(\frac{x_{neu}}{X} \right)^2 \left(2 - \frac{x_{neu}}{X} \right) - HOR_{alt} & x_{neu} \leq x_1 \\ \left(\frac{x_{neu}}{X} + \sum_{i=1}^7 \left(\frac{x_i}{X} \right)^2 \left(2 - \frac{x_i}{X} \right) \right) - HOR_{alt} & x_{neu} > x_1 \end{cases}$$

Wie Abbildung 38 zu entnehmen ist, zeigt auch der HOR den erwarteten Verlauf. Analog zum HHI nimmt der HOR mit dem Eingang der neuen Charakteristik ab, solange x_{neu} eine gewisse Anzahl zugeordneter Elemente nicht überschreitet. In dem betrachteten Beispiel ist der Anteil der neuen Charakteristik erst ab einem Wert von ca. 168 Elementen so groß, dass der HOR mit dem Zugang der neuen Charakteristik einen größeren Wert als zuvor aufweist. Der HOR erfüllt demzufolge die Anforderung bzgl. des Zugangs zusätzlicher Charakteristiken.

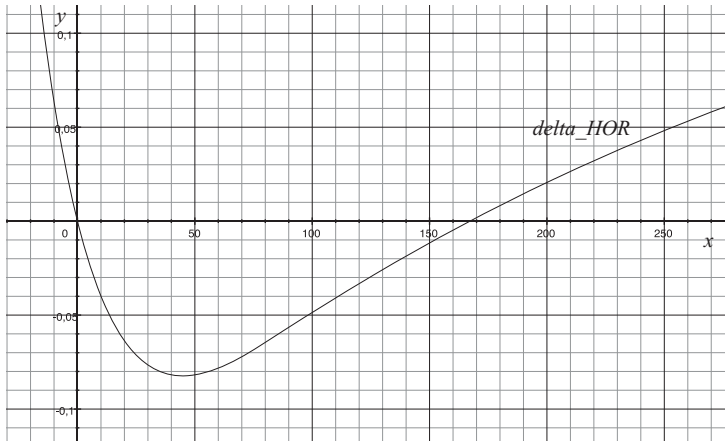


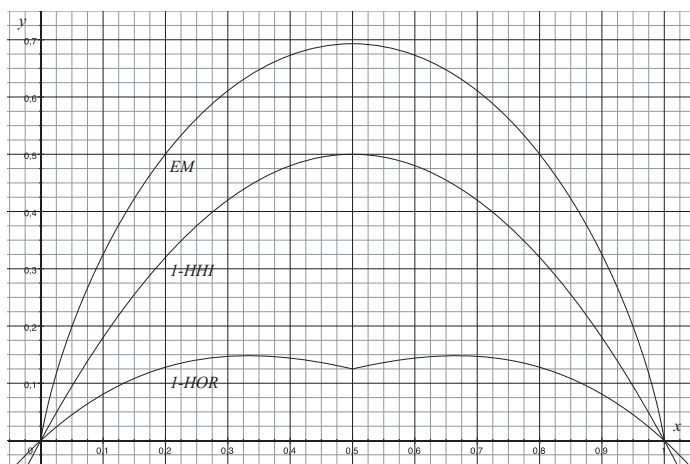
Abbildung 38: Zusätzliche Charakteristiken – HOR

Wie bereits erwähnt, erfährt die Charakteristik mit dem größten relativen Anteil eine besondere Behandlung. Diese wird in der Berechnungsvorschrift des HOR mit 1 gewichtet. Alle weiteren relativen Anteile erhalten eine Gewichtung von $p_i (2 - p_i)$. Damit ist die Gewichtung aller Charakteristiken mit $i > 1$ um den Faktor $(2 - p_i)$ größer als die des HHI. Dennoch ist analog zum HHI ein lediglich geringer Effekt kleiner Gruppen von Elementen zu erkennen (vgl. Tabelle 9).

Tabelle 9: Effekt kleiner Gruppen von Elementen – HOR⁵¹

Anteil der dominanten Charakteristik	n-1 gleichverteilte Charakteristiken								
	1	10	11	50	51	100	101	500	501
0,1	-	0,25471	0,24125	0,13211	0,13148	0,11613	0,11597	0,10324	0,10323
0,25	-	0,35828	0,34879	0,27233	0,27190	0,26121	0,26110	0,25225	0,25224
0,5	0,87500	0,54875	0,54442	0,50995	0,50976	0,50499	0,50494	0,50100	0,50100
0,75	0,85938	0,76234	0,76123	0,75249	0,75244	0,75125	0,75124	0,75025	0,75025
0,9	0,91900	0,90199	0,90181	0,90040	0,90039	0,90020	0,90020	0,90004	0,90004

Entgegen des HHI erfüllt der HOR die Anforderung bezüglich der Verschiebung in der Klassifikation (vgl. Abschnitt 5.4.3) nicht (Marfels 1972b, S. 475). Dies wird bereits bei Variation der Anteile zwischen zwei Charakteristiken deutlich (vgl. hierzu Marfels 1972b). In Abbildung 39 ist der Verlauf der Funktionen 1-HOR, 1-HHI sowie EM (Entropie; vgl. Abschnitt 5.5.3) für alle möglichen Verteilungen der Elemente auf zwei Charakteristiken dargestellt. Während 1-HHI sowie die Entropie bei einer Verschiebung der Elemente von der dominanten Charakteristik hin zu der zweiten Charakteristik bis zu einer Gleichverteilung der Elemente steigen, fällt 1-HOR bereits vorher ab und erreicht ein lokales Minimum bei Gleichverteilung der Elemente. Das Verhalten des HOR ist dabei der Vorgehensweise bei der Gewichtung der Anteile geschuldet. Ein formaler Nachweis, dass der HOR auch für mehr als zwei beteiligte Charakteristiken die Anforderung bezüglich der Verschiebung in der Klassifikation nicht erfüllt, ist Marfels (1972b) zu entnehmen.

**Abbildung 39: Verlauf der Funktionen 1-HOR, 1-HHI sowie EM für alle möglichen Verteilungen der Elemente zweier Charakteristiken**

Die Anforderung hinsichtlich der Invarianz bei proportionalen Veränderungen der Anzahl der Elemente (vgl. Abschnitt 5.4.4) wird durch den HOR hingegen erfüllt:

⁵¹ Eine erweiterte Darstellung in Anlehnung an Marfels (1972a), S. 205.

$$\begin{aligned}
 (20) \quad HOR' &= \frac{ax_1}{\sum_{j=1}^n ax_j} + \sum_{i=2}^n \left(\frac{ax_i}{\sum_{j=1}^n ax_j} \right)^2 \left(2 - \left(\frac{ax_i}{\sum_{j=1}^n ax_j} \right) \right) \\
 &= \frac{ax_1}{a \sum_{j=1}^n x_j} + \sum_{i=2}^n \left(\frac{ax_i}{a \sum_{j=1}^n x_j} \right)^2 \left(2 - \left(\frac{ax_i}{a \sum_{j=1}^n x_j} \right) \right) \\
 &= \frac{x_1}{\sum_{j=1}^n x_j} + \sum_{i=2}^n \left(\frac{x_i}{\sum_{j=1}^n x_j} \right)^2 \left(2 - \left(\frac{x_i}{\sum_{j=1}^n x_j} \right) \right) = HOR
 \end{aligned}$$

5.5.3 Shannon-Entropie

Die Entropie ist ein Maß, das in unterschiedlichen wissenschaftlichen Disziplinen Anwendung findet. Im Bereich der Physik stellt die Entropie eine thermodynamische Zustandsgröße eines Systems dar (Clausius 1865). Boltzmann (1877) stellt in seiner Arbeit die Verbindung zwischen dem Entropie-Begriff aus der Thermodynamik mit der Wahrscheinlichkeitsrechnung heraus.⁵²

Auch im Bereich der Informationstheorie spielt die Entropie eine wesentliche Rolle.⁵³ Das auch als Shannon-Entropie (Shannon 1948) bekannte Maß, benannt nach Claude Elwood Shannon, zur Quantifizierung des mittleren Informationsgehaltes einer Nachricht hat dabei die folgende Form:

$$(21) \quad EM_{\log_2} = - \sum_{i=1}^n p_i \log_2(p_i)$$

Die Wahl der Basis des Logarithmus spielt dabei keine wesentliche Rolle, da diese lediglich einen proportionalen Effekt hat:

$$(22) \quad \log_a x = \log_b x \times \log_a b$$

Für die weitere Arbeit wird der natürliche Logarithmus verwendet:

$$(23) \quad EM = - \sum_{i=1}^n p_i \ln(p_i)$$

Wie bei den zuvor betrachteten Maßen, hängt auch der Wertebereich des Entropie-Maßes von der Anzahl der betrachteten Charakteristiken (n) ab (Theil 1967):

$$(24) \quad 0 \leq EM \leq \ln n$$

Analog zu dem HHI und dem HOR wird anhand des in Abschnitt 5.5.1 verwendeten Beispiels der Verteilung aus Formel 11 das Verhalten des EM bei dem Zugang zusätzlicher Charakteristiken überprüft. ΔEM ist eine von x_{neu} abhängige Funktion der Veränderung des EM bei Zugang einer weiteren Charakteristik, bestehend aus x_{neu} vielen Elementen:

$$(25) \quad EM_{alt} = - \sum_{i=1}^7 p_i \ln(p_i)$$

⁵² Die Übergänge hin zu einem thermodynamischen Gleichgewicht – und die hiermit verbundene Erhöhung der Entropie des Systems – sind dabei vergleichbar mit einer Transition von einem weniger wahrscheinlichen zu einem wahrscheinlicheren Zustand.

⁵³ Das Entropie-Konzept aus der Informationstheorie wurde bereits häufig in den Wirtschaftswissenschaften verwendet, um beispielsweise die industrielle Konzentration zu bestimmen. Die Eignung hierfür wurde von Theil (1967) nachgewiesen. Hexter & Snow (1970) verwenden die Entropie in ihrer Arbeit zur Ermittlung der gesamtwirtschaftlichen Konzentration.

$$(26) \quad X = \sum_{i=1}^7 x_i + x_{neu}$$

$$(27) \quad \text{delta_EM} = - \left(\left(\sum_{i=1}^n \frac{x_i}{X} \ln \left(\frac{x_i}{X} \right) \right) + \frac{x_{neu}}{X} \ln \left(\frac{x_{neu}}{X} \right) \right) - EM_{alt}$$

Bei Betrachtung von Abbildung 40 fällt zunächst auf, dass delta_EM bei Variation von x_{neu} entgegen delta_HHI (vgl. Abbildung 37) und delta_HOR HHI (vgl. Abbildung 38) zunächst positive Werte annimmt und ab ca. 173 Elementen negative Funktionswerte aufweist. Da es sich sowohl bei dem HHI als auch dem HOR um Konzentrationsmaße handelt, müssen diese, wie bereits in Abschnitt 5.5.1 angeführt (vgl. Formel 10), noch jeweils in ein Heterogenitätsmaß transformiert werden. Dies ist bei dem EM nicht erforderlich, da es bereits die erwartete Richtung bei der Veränderung der Funktionswerte aufweist: steigender (fallender) Funktionswert bei höherer (niedriger) Heterogenität. Abbildung 40 zeigt den bezüglich der Anforderung aus Abschnitt 5.4.1 erwarteten Verlauf der Funktion delta_EM . Durch den Zugang einer weiteren Charakteristik steigt das EM an (die Heterogenität steigt), solange die neue Charakteristik nicht zu dominant ist. Ab ca. 173 Elementen hat die neue Charakteristik in dem betrachteten Beispiel einen so großen Anteil an der Summe der Elemente, dass das EM abnimmt (die Heterogenität sinkt). Das EM erfüllt demnach die Anforderung bzgl. des Zugangs zusätzlicher Charakteristiken.

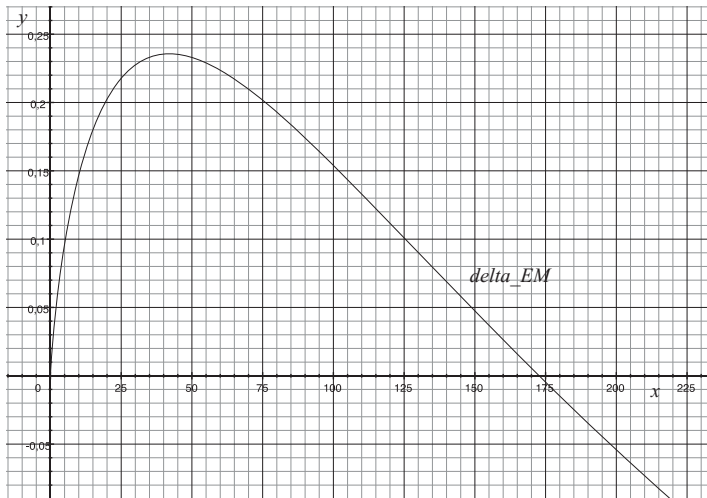


Abbildung 40: Zusätzliche Charakteristiken – EM

Charakteristiken mit einer relativ geringen Anzahl zugeordneter Elemente werden durch das EM deutlich stärker berücksichtigt als es bei dem HHI und dem HOR der Fall ist. Wie Tabelle 10 zu entnehmen ist, führt im Falle eines dominanten Anteiles von 0,5 auch der Zugang der 51. gleichverteilten Charakteristik zu einer Änderung des EM an der zweiten Nachkommastelle. Selbst in dem Szenario einer dominanten Charakteristik mit einem Anteil von 0,9 hat die 101. gleichverteilte Charakteristik einen spürbaren Effekt auf das EM. Sowohl der HHI (Tabelle 8) als auch der HOR (Tabelle 9) reagieren in diesem Fall nicht wahrnehmbar.

Tabelle 10: Effekt kleiner Gruppen von Elementen – EM

Anteil der dominanten Charakteristik	n-1 gleichverteilte Charakteristiken								
	1	10	11	50	51	100	101	500	501
0,1	-	2,39741	2,48319	3,84590	3,86373	4,46974	4,47869	5,91823	5,92003
0,25	-	2,28927	2,36076	3,49635	3,51120	4,01621	4,02368	5,22329	5,22479
0,5	0,69315	1,84444	1,89209	2,64916	2,65906	2,99573	3,00071	3,80045	3,80145
0,75	0,56234	1,13798	1,16181	1,54034	1,54529	1,71363	1,71612	2,11599	2,11649
0,9	0,32508	0,55534	0,56487	0,71629	0,71827	0,78560	0,78660	0,94654	0,94674

Um dies weiter zu verdeutlichen zeigt Abbildung 41, wie die Anteile der Charakteristiken in Abhängigkeit ihrer Größe durch die drei betrachteten Maße gewichtet werden. Für den HOR wurde für die Darstellung angenommen, dass der größte Anteil 0,5 beträgt. Der Sprung in der Gewichts-Funktion des HOR erfolgt stets an dem Wert für x , der dem größten Anteil entspricht (hier $x = 0,5$). An dem grundsätzlichen Verlauf der Funktion ändert sich jedoch nichts.

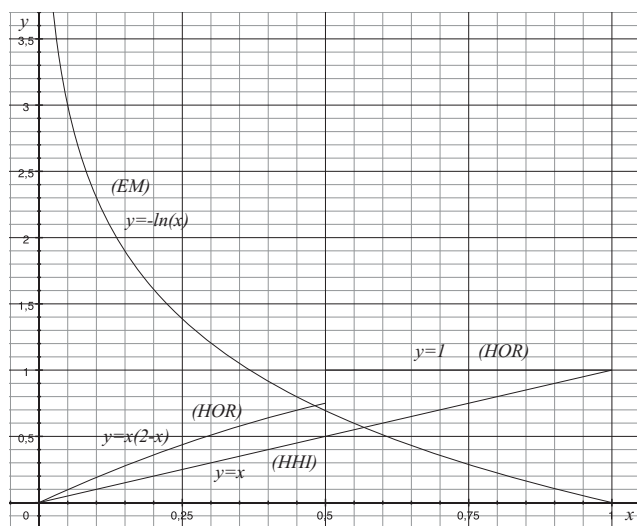


Abbildung 41: Gewichtungen der Anteile

Durch Abbildung 41 wird ersichtlich, dass das EM Charakteristiken mit einer kleinen Anzahl von Elementen deutlich höher gewichtet als eine sehr dominante Charakteristik. Hingegen gewichten sowohl der HHI als auch der HOR Charakteristiken, denen nur ein kleiner Anteil von Elementen zugeordnet ist, wesentlich geringer als Charakteristiken mit größeren Anteilen. Die Anforderung bezüglich des Effektes kleiner Gruppen von Elementen wird demnach durch das EM erfüllt.

Die Anforderung bezüglich der Verschiebung in der Klassifikation (vgl. Abschnitt 5.4.3) wird durch die Entropie ebenfalls erfüllt. Der formale Nachweis hierfür ist Theil (1967) zu entnehmen.

Auch die Anforderung, dass proportionale Änderungen der Anzahl der den Charakteristiken zugeordneten Elemente keinen Effekt auf das EM aufweisen sollen, wird erfüllt:

(28)

$$\begin{aligned} EM' &= -\sum_{i=1}^n \frac{ax_i}{\sum_{j=1}^n ax_j} \ln\left(\frac{ax_i}{\sum_{j=1}^n ax_j}\right) \\ &= -\sum_{i=1}^n \frac{ax_i}{a\sum_{j=1}^n x_j} \ln\left(\frac{ax_i}{a\sum_{j=1}^n x_j}\right) \\ &= -\sum_{i=1}^n \frac{x_i}{\sum_{j=1}^n x_j} \ln\left(\frac{x_i}{\sum_{j=1}^n x_j}\right) = EM \end{aligned}$$

5.5.4 Zusammenfassung der Analyse der vorgestellten Maße

In Tabelle 11 werden die Ergebnisse der Analyse der betrachteten Maße zusammengefasst.

Tabelle 11: Ergebnisse der Analyse der betrachteten Maße

Maß	Zusätzliche Charakteristiken	Effekt kleiner Gruppen von Elementen	Verschiebung in der Klassifikation	Effekt proportionaler Änderungen	Abschnitt
HHI	✓	✗	✓	✓	5.5.1
HOR	✓	✗	✗	✓	5.5.2
EM	✓	✓	✓	✓	5.5.3
Abschnitt	5.4.1	5.4.2	5.4.3	5.4.4	

Wie Tabelle 11 zeigt, erfüllt lediglich die Entropie alle vier formulierten Anforderungen an ein Maß zur Quantifizierung von Heterogenität. In den folgenden Unterkapiteln wird daher eine auf der Entropie basierende Kennzahl vorgeschlagen.

5.6 Richness und Evenness – Zwei Maße aus der Biologie

Nachdem in dem vorangegangenen Unterkapitel bekannte Konzentrationsmaße hinsichtlich deren Eignung zur Quantifizierung von Heterogenität beurteilt wurden, wird in diesem Unterkapitel eine Diskussion aus dem Bereich der Biologie aufgegriffen. In der Biologie spielt die Biodiversität eine wesentliche Rolle. Diese wird nach den begrifflichen Konventionen der Vereinten Nationen zur biologischen Diversität definiert als:

„[...] the variability among living organisms from all sources including, inter alia, terrestrial, marine and other aquatic ecosystems and the ecological complexes of which they are part: this includes diversity within species, between species and of ecosystems.“
(United Nations 1992, S. 146)

Die zu Anfang angesprochene Diskussion in dem Bereich der Messung der Biodiversität bezieht sich auf die Arbeit von Peet (1974), der das Fehlen eines klaren Verständnisses des Begriffs der Diversität und die damit einhergehende Vielfalt unterschiedlicher Konzepte, Defini-

tionen, Modelle und Maßzahlen beklagt. Grundsätzlich betrachtet die Biodiversität mit der Anzahl der Spezies und deren relativen Vielfalt untereinander zwei unterschiedliche Facetten. Um diese Diversität in der Biologie zu messen, existieren u. a. vergleichbare Ansätze, wie sie bereits in dem vorangegangenen Unterkapitel vorgestellt wurden. So finden in der Biologie sowohl die Shannon-Entropie als auch der Herfindahl-Hirschman-Index Verwendung. Letzterer wird in der Biologie auch als Simpson's Index bezeichnet (Simpson 1949). Wie im vorangegangenen Unterkapitel 5.5 gezeigt, reagieren diese Maße sowohl auf eine sich verändernde Anzahl von Charakteristiken (vgl. Anforderung in Abschnitt 5.4.1; was im Kontext der Diversität in der Biologie als die Anzahl der Spezies verstanden werden kann) als auch auf eine Veränderung in der Verteilung der Charakteristiken (vgl. Anforderung in Abschnitt 5.4.3). Peet (1974) bezeichnet diese Maße deshalb auch als „dual-concept diversity“ (Peet 1974, S. 287). Die Tatsache, dass diese Maße Veränderungen beider Facetten, also der Anzahl der Spezies sowie deren relativen Vielfalt, berücksichtigen, kann dabei sowohl als Stärke als auch als Schwäche angesehen werden. Die Stärke liegt darin, dass diese Maßzahlen eine aggregierte Aussage über den Grad der Heterogenität treffen. Dies hat den Vorteil, dass die Veränderung der Heterogenität eines Ökosystems bzw. eines Teilbereiches der IT-Architektur unter Verwendung einer Kennzahl abgebildet werden kann. Die Schwäche besteht zum einen in der erschwerten Vergleichbarkeit unterschiedlicher Ökosysteme bzw. unterschiedlicher Bereiche der IT-Architektur. Diese ist den in Unterkapitel 5.5 angeführten Wertebereichen der Maßzahlen geschuldet, die von der Anzahl der vorhandenen Spezies bzw. Charakteristiken abhängen (vgl. Formeln 9 und 24). Zum anderen besteht eine Schwäche der Reaktion auf beide Facetten darin, dass nur anhand der Veränderung der Kennzahlen nicht zu erkennen ist, was die Veränderung verursacht hat. Darüber hinaus obliegt es den Berechnungsvorschriften der Kennzahlen, zu beurteilen, welche dieser Veränderungen (die Anzahl der Charakteristiken oder die Form der Verteilung der Elemente auf die Charakteristiken) einen größeren Einfluss auf die Höhe der Heterogenität hat.

Lloyd & Ghelardi (1964) stellen heraus, dass diese beiden durch ein Heterogenitätsmaß berücksichtigten Facetten als „species number“ sowie „evenness“ bezeichnet werden können (Lloyd & Ghelardi 1964, S. 271). Die Bezeichnung „species number“ wurde jedoch häufig kritisiert, da diese impliziert, dass die Anzahl der Spezies exakt bestimmbar wäre (Peet 1974). McIntosh (1967) verwendet in seiner Arbeit den alternativen Begriff der „Richness“, der auch in der vorliegenden Dissertation zur Bezeichnung der Anzahl der betrachteten Charakteristiken einer IT-Architektur weiterhin Verwendung finden wird.

Dass das Konzept der Richness für die Biologie nicht ausreichend ist, um die Biodiversität zu beurteilen, ist leicht ersichtlich (McIntosh 1967): Betrachtet werden sollen ein Ökosystem A, in dem eine Spezies 95 % der Lebewesen ausmacht und fünf weitere Spezies jeweils 1 % sowie ein Ökosystem B, in dem die sechs Spezies gleichverteilt sind. Beide Ökosysteme verfügen demnach über sechs Spezies und würden nach dem Konzept der Richness die gleiche Biodiversität aufweisen. Ganz offensichtlich sind diese beiden Ökosysteme aber nicht als äquivalent anzusehen. Richness genügt demzufolge nicht, da diese keine Aussage hinsichtlich der Dominanz einer Spezies oder gar der Verteilung der Lebewesen auf die Spezies zulässt. Dieser Unterschied zwischen den beiden Ökosystemen wird durch die bereits diskutierten

Heterogenitätsmaße berücksichtigt, kann jedoch auch isoliert betrachtet werden. Durch das erstmals von Lloyd & Ghelardi (1964) formal diskutierte Konzept der „Evenness“ wird auf die reine Verteilung der relativen Häufigkeiten der Spezies eines Ökosystems abgezielt.

In den folgenden Abschnitten werden Möglichkeiten zur Quantifizierung der Konzepte Richness und Evenness vorgestellt, die sich auf die folgenden Unterfragen, die Frage Q2 (vgl. Unterkapitel 5.3) weiter spezifizieren, beziehen:

(SQ 2.1) Wie groß ist die Anzahl der unterschiedlichen Charakteristiken?

(SQ 2.2) Wie sind die vorhandenen Elemente auf die Charakteristiken verteilt?

5.6.1 Richness

In der Biologie ist die Richness, also die Artenvielfalt, nur schwer zu ermitteln, da diese in der Regel nur auf Grundlage von Stichproben bestimmt werden kann. Unterschiedliche Stichproben können hierbei zu unterschiedlichen Ergebnissen bezüglich der Richness führen. Weiterhin wird häufig ein Zusammenhang zwischen der Größe der Stichprobe und der Anzahl der identifizierten Spezies beobachtet. Um dies zu lösen, wurde in der Biologie eine Reihe von Ansätzen zur Schätzung der Richness entwickelt (Peet 1974).

Für den Kontext der IT-Architekturen stellt sich die Ermittlung dieser Kennzahl als wesentlich einfacher dar. Wie bereits in Abschnitt 4.2.1 erläutert, werden IT-Architekturen in der vorliegenden Dissertation als künstliche, durch Menschen erstellte, gepflegte und weiterentwickelte Systeme verstanden (ISO/IEC/IEEE 2011; Simon 1996). Dies bedeutet, dass alle bestehenden Komponenten und Beziehungen in der IT-Architektur grundsätzlich bereits bekannt oder im Rahmen einer Aufnahme der IT-Landschaft vollständig ermittelbar sind.

Bei Richness handelt es sich im Sinne der in Unterkapitel 5.5 eingeführten Maßzahlen jeweils um die *Anzahl der Charakteristiken* n , welche für die Berechnung der diskutierten Heterogenitätsmaße stets ohnehin zu ermitteln sind. Mithin ist diese Kennzahl aus praktischer Perspektive ohne weiteren Aufwand berechenbar und beantwortet die Unterfrage SQ 2.1.

5.6.2 Evenness

Ein Maß für Evenness, sprich die Quantifizierung der Verteilung der Elemente, kann aus den in Unterkapitel 5.5 untersuchten Kandidaten für ein Heterogenitätsmaß H wie folgt ermittelt werden (Hill 1973; Peet 1974):

$$(29) \quad \text{Evenness} = (H - H_{\min}) / (H_{\max} - H_{\min})$$

Wie die Analyse der Konzentrationsmaße in Unterkapitel 5.5 gezeigt hat, eignet sich die Entropie für ein Maß zur Quantifizierung der Heterogenität besonders, da das EM auch Charakteristiken mit einem geringen relativen Anteil der Elemente vergleichsweise stark berücksichtigt. Daher wird im Folgenden ein auf der Entropie basierendes Maß für Evenness vorgeschlagen.

In Abschnitt 5.5.3 wurde angeführt, dass das EM sein Minimum von 0 im Falle einer Zuordnung aller Elemente auf eine Charakteristik annimmt. Das Maximum des EM von $\ln(n)$ stellt sich hingegen bei einer Gleichverteilung der Elemente auf n Charakteristiken ein. Daher gilt:

(30) $H_{min} = EM_{min} = 0$

(31) $H_{max} = EM_{max} = \ln(n)$

Hieraus ergibt sich Evenness wie folgt:⁵⁴

(32)
$$\begin{aligned} Evenness &= (H - H_{min}) / (H_{max} - H_{min}) \\ &= (EM - 0) / (\ln(n) - 0) \\ &= EM / \ln(n) \end{aligned}$$

Für den Fall, dass lediglich eine Charakteristik vorliegt (und n demnach den Wert 1 annimmt), wird Evenness auf einen Wert 0 gesetzt:

(33)
$$Evenness(n) = \begin{cases} EM / \ln(n) & n > 1 \\ 0 & n = 1 \end{cases}$$

Der Wertebereich von Evenness lautet demnach wie folgt:

(34) $0 \leq Evenness(n) \leq 1$

Für die Interpretation von Evenness ist zu beachten, dass dieser Wertebereich nicht gleichmäßig ausgeschöpft wird. Die Werte bewegen sich tendenziell näher bei dem Maximum von 1. In Tabelle 12 sind die Werte für Evenness im Falle einer dominanten Charakteristik sowie einer Gleichverteilung des verbleibenden Anteiles auf $n-1$ Charakteristiken dargestellt.

Tabelle 12: Evenness bei einer dominanten sowie $n-1$ gleichverteilten Charakteristiken

Anteil der dominanten Charakteristik	n-1 gleichverteilte Charakteristiken						
	1	10	11	50	51	100	101
0,1	-	0,9998	0,9993	0,9781	0,9779	0,9685	0,9684
0,3	-	0,9269	0,9213	0,8518	0,8512	0,8309	0,8306
0,5	1,0000	0,7692	0,7614	0,6738	0,6730	0,6491	0,6488
0,6	0,9710	0,6648	0,6568	0,5692	0,5684	0,5450	0,5447
0,7	0,8813	0,5428	0,5353	0,4539	0,4531	0,4317	0,4314
0,8	0,7219	0,4007	0,3944	0,3263	0,3257	0,3080	0,3078
0,9	0,4690	0,2316	0,2273	0,1822	0,1818	0,1702	0,1701
0,95	0,2864	0,1308	0,1281	0,1002	0,1000	0,0929	0,0928

Bei Betrachtung von nur zwei Charakteristiken fällt auf, dass bei einem Anteil der dominanten Charakteristik von 95 % der Elemente, der Wert für Evenness bereits bei ca. 0,29 liegt. Eine Verschiebung von 5 % der Elemente zugunsten der Charakteristiken mit geringerem Anteil, führt zu einem relativ starken Anstieg der Evenness auf fast 0,47. Auch in den restlichen in Tabelle 12 dargestellten Beispielen ist eine starke Tendenz hin zum Maximum zu erkennen.

⁵⁴ Vgl. hierzu auch Pielous J' (Hill 1973, S. 429; Pielou 1966, S. 143).

Wie bereits die zuvor für Richness ermittelte Kennzahl n , lässt sich auch Evenness aus den für die Entropie erforderlichen Daten berechnen. Evenness beantwortet zudem die Unterfrage SQ 2.2.

An dieser Stelle drängt sich die Frage auf, ob Evenness in Form der auf den Wertebereich von $[0,1]$ normierten Entropie das EM nicht vollständig ersetzen kann. Dies ist nicht möglich, da durch die Normierung die erste Anforderung aus Abschnitt 5.4.1 verletzt wird. Diese verlangt, dass ein Heterogenitätsmaß auf den Zugang zusätzlicher Charakteristiken reagiert. Durch die Normierung in Form der Division durch $EM_{max} = \ln(n)$ geht diese Eigenschaft der Entropie verloren. Dies kann wie folgt veranschaulicht werden: Betrachtet man eine beliebige Anzahl von m gleichverteilten Charakteristiken, so nimmt die Entropie hierfür einen Wert von $EM_m = \ln(m)$ an. Kommt zu dieser Menge gleichverteilter Charakteristiken eine weitere Charakteristik, mit exakt der gleichen Anzahl zugeordneter Elemente, hinzu, so nimmt die Entropie den Wert $EM_{m+1} = \ln(m+1)$ an. Das Maß Evenness weist hingegen aufgrund der Gleichverteilung der Elemente auf m bzw. $m+1$ Charakteristiken jeweils einen Wert von 1 auf:

$$(35) \quad \text{Evenness} = \frac{EM_m}{\ln(m)} = \frac{EM_{m+1}}{\ln(m+1)} = 1$$

5.7 Vorschlag für ein Maß zur Quantifizierung von Komplexität

Auf Grundlage der in den vorangegangenen Unterkapiteln durchgeführten Bewertung der Konzentrationsmaße (vgl. Unterkapitel 5.4 und 0) sowie der Überlegungen zu den Konzepten Richness und Evenness (vgl. Unterkapitel 5.6) wird in diesem Unterkapitel ein Maß zur Quantifizierung der Komplexität von IT-Architekturen vorgeschlagen. Hierzu wird in Abschnitt 5.7.1 zunächst einleitend eine Zuordnung der identifizierten Maßzahlen zu den Fragestellungen der GQM-Methode vorgenommen. Anschließend werden diese in Abschnitt 5.7.2 formal zu einem Maß zur Quantifizierung von Komplexität zusammengeführt.

5.7.1 Zuordnung der identifizierten Maßzahlen

Wie bereits im Zuge der Einführung in das methodische Vorgehen zur Ermittlung des Komplexitätsmaßes nach GQM angegeben, soll das folgende Ziel erreicht werden (vgl. Unterkapitel 5.3):

„Analysiere die Unternehmensarchitektur als System **zum Zwecke** diese zu charakterisieren **in Bezug auf** die strukturelle Komplexität **vom Blickwinkel** der Unternehmens-/IT-Architekten“.

Für die Ableitung der Fragestellungen wird die in Unterkapitel 3.2 eingeführte systemtheoretische Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen herangezogen:

- (Q1) Wie hoch ist die Anzahl der Komponenten und Beziehungen des Systems?
- (Q2) Wie hoch ist die Heterogenität der Komponenten und Beziehungen des Systems?

Komplementiert wird Q2 durch die folgenden beiden Unterfragen (vgl. Unterkapitel 5.6):

(SQ 2.1) Wie groß ist die Anzahl der unterschiedlichen Charakteristiken?

(SQ 2.2) Wie sind die vorhandenen Elemente auf die Charakteristiken verteilt?

Die Frage **Q1** ist – unter der Voraussetzung einer bestehenden Dokumentation der IT-Architektur – relativ einfach durch Abzählen der Komponenten und Beziehungen des Systems zu beantworten. Hieraus ergibt sich die Maßzahl N , die der Anzahl der Komponenten oder Beziehungen des gewählten Kontextes entspricht.

Zur Beantwortung von **Q2** wurde eine Reihe von Konzentrationsmaßen hinsichtlich ihrer Eignung zur Quantifizierung von Heterogenität überprüft (vgl. Unterkapitel 5.5). Hierbei stellte sich das Entropie-Maß als besonders geeignet heraus (vgl. Abschnitt 5.5.3). Das **Entropie-Maß** wird demzufolge als Maßzahl zur Beantwortung von Q2 herangezogen.

Wie in Unterkapitel 5.6 dargestellt, kann ein Maß zur Quantifizierung von Heterogenität durch zwei weitere Maßzahlen ergänzt werden, um dem Architekten die Möglichkeit einer tieferen Analyse zu geben. Die Anzahl der Charakteristiken n stellt dabei eine Maßzahl zur Quantifizierung des Konzeptes „**Richness**“ (z. B. zur Analyse der Menge bestehender Technologien) dar und dient der Beantwortung von **SQ 2.1**. Das auf den Wertebereich $[0,1]$ normierte Entropie-Maß ($EM/\ln(n)$) ist eine Maßzahl der „**Evenness**“ einer Verteilung von Werten (z. B. zur Analyse der relativen Häufigkeit bestehender Technologien) und beantwortet **SQ 2.2**.

5.7.2 Ein auf der Shannon-Entropie basierendes Maß

Konsistent mit der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen wird die Heterogenität der Komponenten T (H_T) bzw. der Beziehungen R (H_R) wie folgt definiert:⁵⁵

$$(36) \quad H_T := (EM_T, n_T, (EM_T/\ln n_T))$$

$$(37) \quad H_R := (EM_R, n_R, (EM_R/\ln n_R))$$

EM_T und EM_R repräsentieren dabei das Entropie-Maß der Komponenten T , bzw. der Beziehungen R . n_T und n_R stehen für die jeweilige Anzahl der betrachteten Charakteristiken (Richness). $EM_T/\ln n_T$ und $EM_R/\ln n_R$ geben die Verteilung der Charakteristiken der Komponenten T und der Beziehungen R wieder (Evenness).

Die Anzahl der Komponenten (N_T) und der Beziehungen (N_R) ergibt sich durch Abzählen der jeweiligen Elemente der betrachteten Menge an Komponenten und Beziehungen. Im Gegensatz zu n_T und n_R , die mathematisch gesehen die Mächtigkeit endlicher Mengen von Elementen darstellen, werden bei der Ermittlung von N_T und N_R mehrfach vorkommende Ausprägungen von Elementen auch mehrfach gezählt. Als Beispiel kann eine Menge von Instanzen

⁵⁵ Der Ausdruck kann auf die ersten beiden Elemente reduziert werden, da sich das dritte vollständig aus den ersten beiden berechnen lässt. Zum Zwecke der Darstellung aller drei Konzepte (Heterogenität, Richness und Evenness) wird die angeführte Form gewählt.

von Datenbankmanagementsystemen betrachtet werden. Die Komponenten T sind hier die Elemente einer Menge von Datenbank-Instanzen [IMS/DB, IMS/DB, DB2, DB2, IMS/DB, IMS/DB, DB2]. Die Anzahl der Komponenten (N_T) beträgt in diesem Beispiel sieben, die Anzahl der Charakteristiken (n_T) hingegen zwei.

Mit diesem Verständnis bezüglich der Anzahl und Heterogenität der Elemente (vgl. Gleichungen 36 und 37) ergeben sich aus der Konzeptualisierung von Komplexität (vgl. Gleichung 5) die folgenden Maßzahlen zur Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen:

$$(38) \quad C_T = \left(N_T, (EM_T, n_T, (EM_T / \ln n_T)) \right)$$

$$(39) \quad C_R = \left(N_R, (EM_R, n_R, (EM_R / \ln n_R)) \right)$$

An dieser Stelle soll betont werden, dass im Zuge der Ermittlung der Maßzahlen kein spezifischer Kontext bei der Definition des Zieles der GQM-Methode vorgegeben wurde. Daher sind die Maßzahlen nicht auf einen bestimmten Kontext innerhalb der Unternehmensarchitektur beschränkt, sondern können für alle Subsysteme der Architektur angewendet werden. Damit können IT-Architekten diese Maßzahlen für die unterschiedlichsten Einsatzgebiete ausprägen. Hierzu müssen sie lediglich das zu untersuchende (Sub-)System (z. B. die Infrastrukturarchitektur), die Komponente oder Beziehung (z. B. die Datenbankmanagementsysteme) sowie die Charakteristiken, nach denen die Komponenten oder Beziehungen unterschieden werden sollen (z. B. den Hersteller), auswählen. Idealerweise können die IT-Architekten die für die Berechnung der Maßzahlen erforderlichen Daten direkt einem zentralen Daten-Repository oder einem EA Management-Tool entnehmen.

5.8 Anwendung des Maßes

In diesem Unterkapitel wird die Anwendbarkeit des eingeführten Maßes zur Quantifizierung struktureller Komplexität von IT-Architekturen anhand der Architekturdaten von drei Unternehmen aufgezeigt. Hierbei soll insbesondere veranschaulicht werden, für welche spezifischen Fragestellungen des IT-Architekten sich die jeweilige Form der Quantifizierung der Heterogenität – Dual-Konzept in Form der Entropie oder alternativ bzw. ergänzend Richness und Evenness – besonders eignet. In Abschnitt 5.8.1 wird zunächst die Infrastrukturarchitektur eines Unternehmens aus der Finanzbranche hinsichtlich der Komplexität der eingesetzten Datenbanksysteme sowie Betriebssysteme untersucht. In Anschluss wird in Abschnitt 5.8.2 mithilfe des Komplexitätsmaßes der Schnitt der Domänenstruktur eines Unternehmens aus dem Transportsektor beurteilt. Das dritte Unternehmen ist in der Versicherungsbranche tätig. Im Kontext dieses Unternehmens wird in Abschnitt 5.8.3 untersucht, wie sich ein Transformationsprojekt der IT-Architektur auf die strukturelle Komplexität auswirkt.

5.8.1 Fall 1: Unternehmen aus der Finanzbranche⁵⁶

In diesem Abschnitt wird der vorgestellte Ansatz zur Quantifizierung struktureller Komplexität von IT-Architekturen anhand der Architekturdaten einer führenden, international tätigen Universalbank angewendet.

Den IT-Architekten der Bank wurde die Vorgehensweise zur Quantifizierung der strukturellen Komplexität im Rahmen gemeinsamer Workshops vorgestellt. Für eine Anwendung der Maßzahlen wurde eine Analyse des Infrastrukturbereiches als besonders interessant angesehen, da in diesem Bereich eine besonders hohe Komplexität vermutet wurde. Um dies anhand einer Maßzahl zu überprüfen, wurde der Fokus bei der Analyse der Komplexität der IT-Landschaft auf die Infrastrukturarchitektur gelegt. Als zu betrachtende Elemente wurden sowohl die Datenbankmanagementsysteme (DB) als auch die Betriebssysteme (OS) gewählt. Zur Bestimmung der Charakteristiken wurden der Hersteller sowie die jeweilige Version des Produktes als Unterscheidungskriterien verwendet. In Zusammenarbeit mit den IT-Architekten der Bank wurde folgendes Ziel definiert (vgl. die GQM-Methode in Unterkapitel 5.3):

„Analysiere die Infrastrukturarchitektur *zum Zwecke* diese zu charakterisieren *in Bezug auf* die von den Komponenten der Datenbankmanagementsysteme sowie der Betriebssysteme ausgehende strukturelle Komplexität *vom Blickwinkel* der IT-Architekten“.

Als Maßzahlen zur Bearbeitung der Zielsetzung wurden die folgenden Ausprägungen der in Unterkapitel 5.7 vorgeschlagenen Maßzahlen zur Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen herangezogen (vgl. Gleichung 38):

$$(40) \quad C_{OS} = \left(N_{OS}, (EM_{OS}, n_{OS}, (EM_{OS} / \ln n_{OS})) \right)$$

$$(41) \quad C_{DB} = \left(N_{DB}, (EM_{DB}, n_{DB}, (EM_{DB} / \ln n_{DB})) \right)$$

Die gesamten für die Berechnung der Maßzahlen erforderlichen Architekturdaten wurden dem zentralen ARIS-Repository des Unternehmens entnommen. Die IT-Landschaft der Bank ist anhand fachlicher Domänen strukturiert. Diese Domänen bestehen aus Subdomänen, die wiederum in eine Menge funktionaler Einheiten untergliedert sind. Jede Anwendung der IT-Landschaft ist dabei exakt einer dieser funktionalen Einheiten zugeordnet.

Insgesamt wurden 136 funktionale Einheiten von 32 Subdomänen auf die verwendeten Datenbankmanagementsysteme und Betriebssysteme hin analysiert. Hierzu wurden zunächst alle Anwendungen ermittelt, die den funktionalen Einheiten zugeordnet sind. Im Anschluss daran wurden die jeweils durch die Anwendungen verwendeten Datenbanken sowie die erforderlichen Betriebssysteme bestimmt. Basierend auf diesen Informationen wurden daraufhin die oben angeführten Maßzahlen für alle funktionalen Einheiten der Bank berechnet.

⁵⁶ Vgl. nachfolgend Schütz et al. (2013b).

Tabelle 13: Verteilung und Komplexitätsmaße für die Datenbankmanagementsysteme

DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7	DB 8	N_{DB}	EM_{DB}	n_{DB}	$\frac{EM_{DB}}{\ln n_{DB}}$
237	54	70	20	245	23	89	304	1042	1,7416	8	0,8375

Tabelle 13 zeigt die absoluten Häufigkeiten der jeweiligen Datenbankinstanzen (DB 1 bis DB 8), die Gesamtanzahl der vorhandenen produktiven Datenbankinstanzen (N_{DB}), das Entropie-Maß (EM_{DB}), die Anzahl der genutzten, unterschiedlichen Datenbank-Typen (n_{DB}) – Richness – sowie das normierte Entropie-Maß ($EM_{DB}/\ln n_{DB}$) zur Bewertung der Verteilung – Evenness – der Datenbankinstanzen. Eine äquivalente Darstellung für die Betriebssysteme ist Tabelle 14 zu entnehmen. Diese beiden Tabellen stellen dabei eine aggregierte Sicht auf die Datenbankmanagementsysteme sowie die Betriebssysteme der Gesamtbank dar. Aus Tabelle 13 ist zu erkennen, dass in der Bank insgesamt acht (n_{DB}) unterschiedliche Datenbankmanagementsysteme verwendet werden. Die Höhe der berechneten Evenness von 0,8375 ist ein Indiz für eine relativ heterogene Verteilung der acht Datenbankmanagementsysteme, was sich durch Betrachtung der Verteilung der Instanzen auf die Charakteristiken bestätigt. Hinsichtlich der Betriebssysteme zeigt Tabelle 14 eine höhere Anzahl von Charakteristiken (Richness) bei gleichzeitig geringerer Evenness.

Tabelle 14: Verteilung und Komplexitätsmaße für die Betriebssysteme

OS 1	OS 2	OS 3	OS 4	OS 5	OS 6	OS 7	OS 8	OS 9	OS 10	OS 11	OS 12	OS 13	OS 14	OS 15	OS 16	N_{OS}	EM_{OS}	n_{OS}	$\frac{EM_{OS}}{\ln n_{OS}}$
9	55	3	48	544	85	30	283	363	24	417	1	41	14	20	2	1939	1,9405	16	0,6999

Diese Kombination ist ein Indiz für potenzielle Kandidaten für die Durchführung einer Migration: Die Verteilung der Betriebssysteme zeigt mit OS 3, OS 12 und OS 16 gleich drei Betriebssysteme, für die untersucht werden sollte, ob diese auch durch andere, dominantere Betriebssysteme ersetzt werden können. Die sieben Domänen und deren 32 Subdomänen wurden in Zusammenarbeit mit zwei IT-Architekten der Bank und auf Basis der errechneten Maßzahlen analysiert. In diesem Zuge konnte mithilfe der Maßzahlen Handlungsbedarf im Bereich der Infrastrukturarchitektur identifiziert werden, welcher die subjektive Wahrnehmung der IT-Architekten bestätigte. Eine der untersuchten Domänen wird im Folgenden ausführlicher beschrieben.

Die betrachtete Domäne besteht aus vier Subdomänen, die sich wiederum aus insgesamt 13 funktionalen Einheiten zusammensetzen. In Tabelle 15 werden die berechneten Maßzahlen für die in der Domäne verwendeten Datenbankmanagementsysteme und Betriebssysteme zusammengefasst. Im Hinblick auf die Betriebssysteme zeigt sich auf Domänenebene eine vergleichsweise große Richness bei gleichzeitig moderater Evenness. Auch hier ist dies ein Hinweis darauf, dass einige der Betriebssysteme nur von einer relativ geringen Anzahl von

Anwendungen benötigt werden. Bei genauerer Betrachtung auf Ebene der Subdomänen zeigte sich, dass in den Subdomänen 1 und 4 insgesamt vier Betriebssysteme jeweils von lediglich einer Anwendung verwendet wurden. Die IT-Architekten der Bank zogen diese daraufhin für eine Migration auf ein alternatives Betriebssystem in Betracht.

Tabelle 15: Komplexitätsmaßzahlen für eine ausgewählte Domäne der Bank⁵⁷

	Datenbankmanagementsysteme				Betriebssysteme			
	N_{DB}	EM_{DB}	n_{DB}	$\frac{EM_{DB}}{\ln n_{DB}}$	N_{OS}	EM_{OS}	n_{OS}	$\frac{EM_{OS}}{\ln n_{OS}}$
Subdomäne 1	82	1,0703	6	0,5974	178	1,4061	9	0,6399
Subdomäne 2	37	1,4203	6	0,7927	48	1,5179	6	0,8471
Subdomäne 3	76	1,3592	5	0,8445	89	1,5004	6	0,8374
Subdomäne 4	19	0,8785	3	0,7997	24	1,2904	7	0,6632
Domäne	214	1,5448	7	0,7939	339	1,6651	12	0,6701

Auch bezüglich der Datenbankmanagementsysteme zeigte sich ein vergleichbarer Fall. Besonders Subdomäne 1 wies hier durch die geringe Evenness auf Verbesserungspotenziale hin. Tatsächlich ergab ein Blick in die Datenbanknutzung der Anwendungen, dass zwei der sechs durch Subdomäne 1 verwendeten Datenbanken lediglich einer Anwendung zugeordnet waren. Eine weitere Datenbank war drei Anwendungen zugeordnet. Durch eine Migration dieser auf eine alternative, dominantere Datenbanklösung kann die Heterogenität und dadurch die Komplexität reduziert werden. In diesem Fall würde eine solche Migration auch zu einer Verringerung der Komplexität auf Domänenebene führen, da die Datenbank von keiner weiteren Anwendung der Domäne verwendet wird. Auch in diesem Fall bestätigten die IT-Architekten einen Handlungsbedarf.

Basierend auf dem vorgeschlagenen Komplexitätsmaß konnte gemeinsam mit den IT-Architekten Handlungsbedarf in unterschiedlichen Bereichen abgeleitet werden. Weiterhin betonten die IT-Architekten die Vorteile des holistischen Ansatzes, der auf eine Vielzahl weiterer Kontexte anwendbar ist.

In dem betrachteten Anwendungsfall erwies sich vor allem die Kombination aus Richness und Evenness als gutes Instrument, um Hinweise auf einen potenziellen Handlungsbedarf auf Ebene aggregierter Daten zu erhalten, ohne unmittelbar mit großen Datenmengen umgehen zu müssen. Zudem kann hierdurch ein Einstieg in die weitere, tiefergehendere Analyse gefunden werden.

⁵⁷ Vgl. hierzu auch Schütz et al. (2013b).

5.8.2 Fall 2: Unternehmen aus dem Transportsektor

Im zweiten Anwendungsfall wurde die vorgestellte Maßzahl zur Quantifizierung struktureller Komplexität auf die Anwendungsebene der Unternehmensarchitektur eines international tätigen Großunternehmens (>40.000 Mitarbeiter) aus dem Transportsektor angewendet. Die Anwendungsarchitektur ist anhand eines Domänenmodells strukturiert, in dem die Anwendungen nach Fachlichkeit und starker Kohärenz in Domänen und Subdomänen geclustert sind.

Die folgenden Betrachtungen beziehen sich auf die Fachlichkeit der Schnittstellen zwischen den Anwendungen. Dabei wird angenommen, dass eine Schnittstelle zu einer Anwendung, die einer bestimmten Domäne zugeordnet ist, auch zu einem gewissen Grad fachspezifisches Know-how dieser Domäne beinhalten. Weist eine Anwendung Schnittstellen zu Anwendungen einer Vielzahl anderer Domänen auf, wird für die Durchführung von Änderungen an einer solchen Anwendung oder deren Schnittstellen mehr fachspezifisches Know-how benötigt, als bei einer Anwendung, die ausschließlich mit Anwendungen einer Domäne kommuniziert. Eine Unterscheidung hinsichtlich der Art der Implementierung der Schnittstellen (technische Perspektive) war nicht Gegenstand der Untersuchung.

In Kooperation mit dem Unternehmen wurde anhand des Subsystems der Anwendungsarchitektur die Komplexität der Beziehungen (C_R) (Schnittstellen) ermittelt. Hierzu wurde das folgende Ziel formuliert:

„Analysiere die Anwendungsarchitektur **zum Zwecke** diese zu charakterisieren **in Bezug auf** die fachliche Komplexität der Schnittstellen **vom Blickwinkel** der IT-Architekten“.

Auch dieses Fallbeispiel basiert auf den Daten aus Systemen des Unternehmens. Anhand einer Domäne (Domäne 8) wurden 38 Anwendungen hinsichtlich ihrer fachlichen Schnittstellenkomplexität untersucht. Hierbei wurden die Anzahl und die Heterogenität der Schnittstellen anhand der Unterschiedlichkeit der Zieldomänen ermittelt. Tabelle 16 sind die errechneten Maßzahlen der untersuchten Domäne zu entnehmen. Im Zuge der Ausgestaltung der Domänen wurde das Ziel einer hohen Kohärenz der den Domänen zugeordneten Anwendungen verfolgt. Aus diesem Grund ist es nicht überraschend, dass die meisten Schnittstellen zu Anwendungen der eigenen Domäne 8 führen. Neben Domäne 8 weisen auch die Domänen 4 und 10 eine vergleichsweise hohe Anzahl von Schnittstellen mit Anwendungen der betrachteten Domäne 8 auf.

Tabelle 16: Verteilung der Schnittstellen und Maßzahlen der betrachteten Domäne 8

Anzahl der Schnittstellen mit Domäne...											N_R	EM_R	n_R	$\frac{EM_R}{\ln n_R}$
1	2	3	4	5	6	7	8	9	10	11				
2	3	18	50	13	16	2	113	5	44	31	297	1,8331	11	0,7645

Für die weitere Analyse wurden zunächst für alle Anwendungen der Domäne 8 die Maßzahlen der fachlichen Komplexität im Hinblick auf die Verteilung der Schnittstellen auf die einzelnen Domänen berechnet. Der Fokus lag nicht darin, potenzielle Kandidaten für eine Migration zu identifizieren, sondern darin, die Zuordnung der Anwendungen zu der betrachteten

Domäne zu beurteilen. Hierfür war interessant zu untersuchen, inwiefern Anwendungen mit einer geringen Evenness, sprich einer tendenziellen Ungleichverteilung, aus struktureller Perspektive der richtigen Domäne zugeordnet sind. Im Sinne einer hohen Kohärenz der Domänen wäre dies zu erwarten. Bei Anwendungen, deren Schnittstellen relativ gleichmäßig auf unterschiedliche Domänen aufgeteilt sind, kann diese strukturelle Eigenschaft hingegen nicht als Indiz herangezogen werden. Daher wurde die weitere Betrachtung auf Anwendungen beschränkt, deren Evenness einen relativ geringen Wert aufweist. Gleichzeitig handelte es sich bei den ausgewählten Anwendungen um solche, die über eine relativ große Anzahl an Schnittstellen verfügen. Ergänzt wurde die Auswahl um eine weitere Anwendung, deren relativ geringe Anzahl an Schnittstellen sich auf vergleichsweise viele Domänen verteilt. In Tabelle 17 sind die Maßzahlen für die ausgewählten Anwendungen aufgeführt.

Tabelle 17: Errechnete Maßzahlen für die ausgewählten Anwendungen der Domäne 8

Anw. Nr.	Anzahl der Schnittstellen mit Domäne...											N_R	EM_R	n_R	$\frac{EM_R}{\ln n_R}$
	1	2	3	4	5	6	7	8	9	10	11				
3	1	0	1	1	0	6	0	17	2	5	1	34	1,5161	8	0,7291
11	0	1	1	2	3	2	0	13	0	5	1	28	1,6372	8	0,7873
14	0	0	0	29	0	0	0	1	0	0	3	33	0,4375	3	0,3982
24	0	0	1	2	0	1	0	11	0	1	1	17	1,2001	6	0,6698
37	0	0	3	3	0	0	0	3	0	1	1	11	1,4990	5	0,9314

Wie bereits in Tabelle 16 gezeigt, sind Schnittstellen zu den Domänen 4 und 10 von besonderem Interesse. Auf diese wird in den folgenden Betrachtungen regelmäßig Bezug genommen.

Anwendung 3 verfügt über die größte Anzahl an Schnittstellen, die auf Anwendungen aus insgesamt acht Domänen aufgeteilt sind. EM_R ist mit einem Wert von 1,5161 noch relativ weit von seinem theoretischen Maximum von 2,0794 ($\ln(8)$) entfernt. Der Grund dafür liegt in der – anhand der Evenness von 0,7291 erkennbaren – relativ homogenen Verteilung der Schnittstellen auf die Domänen, wie Tabelle 17 belegt. Ein sehr ähnliches Bild zeigen die Anwendungen 11 und 24. Die Schnittstellen sind bei Anwendung 11 etwas heterogener verteilt (Evenness von 0,7873) und bei Anwendung 24 deutlich homogener (Evenness von 0,3982) als bei Anwendung 3. Für alle drei Anwendungen sind der Großteil der Schnittstellen Domäne 8 zugeordnet. Die Anzahl der Schnittstellen, die zu Domäne 10 führen sind teilweise relativ hoch, jedoch noch deutlich unter der Anzahl der Schnittstellen zu Domäne 8. Mithin ist die Zuordnung dieser Anwendungen unauffällig.

Eine hohe Heterogenität zeigt hingegen Anwendung 37. Der Wert des EM_R von 1,499 liegt relativ nahe am theoretischen Maximum von 1,6094 ($\ln(5)$). Wie die Evenness von 0,9314 zeigt, sind die Schnittstellen auf die fünf Domänen relativ gleichmäßig verteilt. Die hohe Heterogenität rührt hierbei aus dem Aufgabenbereich der Anwendung her. Dieser besteht u. a. in der zentralen Verwaltung von Daten und Geschäftsregeln für eine Reihe anderer Anwendungen.

Anwendung 14 weist dagegen trotz der sehr hohen Anzahl an Schnittstellen eine sehr geringe Heterogenität (EM_R von 0,4375 bei theoretischem Maximum von 1,0986 ($\ln(3)$)) auf. Die 33

Schnittstellen von Anwendung 14 sind lediglich drei Domänen zugeordnet. Die Evenness von 0,3982 weist auf eine starke Ungleichverteilung der Schnittstellen hin. Wie die linke Seite von Tabelle 17 zeigt, führen 29 der 33 Schnittstellen zu Anwendungen genau einer Domäne. Jedoch handelt es sich hierbei nicht um die eigene Domäne 8, sondern um die bereits oben erwähnte Domäne 4. Anwendung 14 steht demnach in sehr engem Austausch mit Domäne 4. Die Mehrheit ihrer Schnittstellen dient dazu, Planungsdaten mit Ist-Produktionsdaten zwischen den Domänen auszutauschen. Hier sollte überprüft werden, ob diese Anwendung durch die Verantwortlichen von Domäne 4 nicht einfacher zu verwalten wäre.

In dem hier angeführten Fallbeispiel wurde die Maßzahl zur Quantifizierung struktureller Komplexität verwendet, um die Zuordnung der Anwendungen zu Domänen zu untersuchen. Die Evenness konnte hier als Einstieg in eine detailliertere Betrachtung herangezogen werden.

5.8.3 Fall 3: Unternehmen aus der Versicherungsbranche⁵⁸

Der dritte Anwendungsfall wurde in Zusammenarbeit mit einer Spezialversicherung durchgeführt. Bei dem kooperierenden Unternehmen handelt es sich um eine international tätige Tochtergesellschaft eines namhaften deutschen Versicherungskonzernes. Im Rahmen eines Architektur-Transformationsprojektes wurde eine neue Zielarchitektur für die IT-Landschaft der Spezialversicherung entwickelt. Ein Treiber für die Überarbeitung der Bebauung lag unter anderem in der relativ hohen Redundanz der eingesetzten Systeme an den einzelnen Standorten. Darüber hinaus hat sich über die Jahre hinweg eine Vielzahl Office-basierter Anwendungen, sogenannte IDV-Anwendungen (Individuelle Datenverarbeitung), herausgebildet. Hierbei handelt es sich um Anwendungen, die typischerweise nicht durch die zentrale IT, sondern unmittelbar in den Fachbereichen erstellt und gepflegt werden. Diese entstehen häufig ad-hoc, um einen kurzfristigen Informationsbedarf zu befriedigen. Sie können sich im Laufe der Zeit jedoch auch als Informationsquelle für Regelprozesse in den Fachbereichen etablieren. In den letzten Jahren sind die IDV-Anwendungen – nicht zuletzt durch Reform des Versicherungsaufsichtsrechtes in Europa (Solvency II) – ins Visier der Aufsicht geraten (BaFin 2009), was zu Handlungsbedarf bei einem Großteil an Versicherungsunternehmen führte. Neben den angeführten Schwierigkeiten bzgl. der vorhandenen Redundanzen sowie der Vielzahl an IDV-Anwendungen sollten im Zuge des Transformationsprojektes die IT-Landschaft modernisiert und existierende Individualschnittstellen durch die Einführung einer Integrationsplattform weitestgehend abgebaut werden. Weiterhin sollten die Vorgaben der Konzernmutter im Hinblick auf die technologischen Standards der Gruppe Berücksichtigung finden.

Die vorgestellte Maßzahl wurde im Kontext des Architektur-Transformationsprojektes verwendet, um die Effekte der geplanten Veränderung der IT-Landschaft auf die strukturelle Komplexität zu validieren. Gemeinsam mit den beteiligten IT-Architekten der Versicherung wurde eine Reihe von Zielen definiert (vgl. hierzu Tabelle 18). Diese Ziele weisen dabei grundsätzlich die folgende Struktur auf:

⁵⁸ Vgl. nachfolgend Schmidt et al. (2013).

„Analysiere die [Architekturebene] *zum Zwecke* diese zu charakterisieren *in Bezug auf* die strukturelle Komplexität, verursacht durch [Untersuchungsfokus] *vom Blickwinkel* der IT-Architekten“.

Die jeweils durch die Zielsetzung betrachtete Architekturebene und der Untersuchungsfokus sind in Tabelle 18 dargestellt. Darüber hinaus gibt die Tabelle an, anhand welcher Elemente und Unterscheidungskriterien die Komplexitätskennzahlen ermittelt wurden.

Tabelle 18: Zieldefinition sowie Elemente und Unterscheidungskriterien

Komplexitätsaspekt	Architekturebene	Untersuchungsfokus	Element	Unterscheidungskriterium
C_{T_1}	AA	Anwendungen	Komponenten	Hersteller
C_{R_1}	AA	Schnittstellen	Beziehungen	Schnittstellentechnologie
C_{R_2}	AA	Schnittstellen	Beziehungen	Schnittstellennutzung durch Anwendungen
C_{T_2}	IA	Plattformen	Komponenten	Plattformausprägung
C_{T_3}	GPA, AA	von Geschäftsfeldern verwendete Anwendungen	Komponenten	Produkt
C_{T_4}	GPA, AA	von Anwendungen bereitgestellte Geschäftsfunktionen	Komponenten	Produkt
C_{T_5}	GPA, AA	Anwendungsnutzung durch die Organisationseinheiten	Komponenten	Produkt

Für ein besseres Verständnis der hierfür abgeleiteten Kennzahlen sind in Tabelle 19 Interpretationshilfen sowie Beispiele, welche Veränderungen durch die jeweilige Kennzahl gemessen werden sollen, angeführt. Für die angegebenen sieben Komplexitätsaspekte wurden die entsprechenden Maßzahlen für die aktuelle Architektur sowie die geplante Zielarchitektur berechnet. Die hierfür erforderlichen Daten konnten dem konzernweit eingesetzten Architektur-Repositorium entnommen werden. Eine Zusammenfassung der berechneten Maßzahlen ist Tabelle 20 zu entnehmen. Für diesen Anwendungsfall wurde ausschließlich die Entropie als Maßzahl für die Heterogenität berechnet, da nicht die Ursachen einer veränderten Komplexität näher untersucht werden sollten, sondern die Veränderung der Komplexität im Falle der Transformation der bestehenden Architektur hin zur Zielarchitektur. Für diesen Zweck ist die Entropie als Dual-Konzept Maßzahl besonders geeignet, da diese sowohl Veränderungen der Anzahl der Charakteristiken als auch deren Verteilung in einer Kennzahl darstellt.

Tabelle 19: Interpretation und Beispiele für die Veränderung der Maßzahlen⁵⁹

<i>Komplexitätsaspekt</i>	<i>Anzahl N</i>	<i>Heterogenität H</i>
$C_{T,1}$	Anzahl der produktiven Anwendungen	Heterogenität der Anwendungen nach Herstellern
	Reduktion durch Außerbetriebnahme von Anwendungen	Reduktion u. a. durch die Migration von Lösungen auf diejenige eines in der IT-Landschaft dominanten Anbieters
$C_{R,1}$	Anzahl der produktiven Schnittstellen	Heterogenität der Schnittstellen hinsichtlich der Schnittstellentechnologie
	Reduktion durch Außerbetriebnahme von Schnittstellen	Reduktion u. a. durch die Ersetzung proprietärer Schnittstellentechnologien durch standardisierte Schnittstellentechnologien
$C_{R,2}$	Anzahl der von den Anwendungen genutzten Schnittstellen	Heterogenität der Schnittstellennutzung durch Anwendungen
	Reduktion durch geringere Nutzung der Schnittstellen	Reduktion durch Verschiebung von Schnittstellen mit relativ geringer Nutzung hin zu Schnittstellen mit relativ hoher Nutzung
$C_{T,2}$	Anzahl der Unterstützung von Anwendungen durch Plattformen	Heterogenität der Plattformnutzung durch Anwendungen
	Reduktion durch Verringerung der Anzahl der für eine Anwendung erforderlichen Plattformen	Reduktion durch Fokussierung auf eine geringere Anzahl an Plattformvarianten
$C_{T,3}$	Unterstützung der Geschäftsfelder durch Anwendungen	Heterogenität der Anwendungsnutzung durch die Geschäftsfelder
	Reduktion durch geringeren Bedarf an Anwendungsunterstützung der Geschäftsfelder	Reduktion durch Abschaltung von Anwendungen sowie der Ablösung von spartenspezifischen durch spartenübergreifenden Anwendungen
$C_{T,4}$	Anzahl der von den Anwendungen bereitgestellten Geschäftsfunktionen	Heterogenität der Geschäftsfunktionen nach bereitstellenden Anwendungen
	Reduktion durch Verringerung von funktionaler Redundanz	Reduktion durch Abschaltung von Anwendungen sowie der Verschiebung von Geschäftsfunktionen hin zu Kernsystemen
$C_{T,5}$	Unterstützung von Organisationseinheiten durch Anwendungen	Heterogenität der Anwendungsnutzung durch Organisationseinheiten
	Reduktion durch Verringerung der Anzahl der durch Organisationseinheiten genutzten Anwendungen	Reduktion durch Abschaltung von Anwendungen sowie der Ablösung von „Insellösungen“ durch „global“ genutzte Anwendungen

⁵⁹ Vgl. hierzu auch Schmidt et al. (2013).

Tabelle 20: Berechnete Maßzahlen vor und nach der geplanten Transformation⁶⁰

<i>Komplexitätsaspekt</i>	<i>Anzahl N</i>		<i>Heterogenität H (EM)</i>	
	<i>Ist</i>	<i>Ziel</i>	<i>Ist</i>	<i>Ziel</i>
C_{T_1}	51	23	3,9	3,08
C_{R_1}	40	39	3,36	0,66
C_{R_2}	40	67	3,69	3,45
C_{T_2}	29	31	2,18	1,78
C_{T_3}	91	42	3,9	3,11
C_{T_4}	148	108	3,09	2,5
C_{T_5}	65	44	3,88	3,12

Wie Tabelle 20 zu entnehmen ist, führt die geplante Architektur zu einer deutlichen Reduktion der Komplexität der Anwendungen (C_{T_1}). Sowohl die Anzahl der produktiven Anwendungen verringert sich von 51 auf 23, als auch die Heterogenität der Anwendungen im Hinblick auf die Hersteller geht von einem Wert für EM von 3,9 auf 3,08 zurück. Die Reduktion der Anzahl der Anwendungen ist dabei zu einem Großteil mit der Außerbetriebnahme von IDV-Anwendungen zu begründen. Die nur relativ moderate Verringerung der Heterogenität liegt an der Tatsache, dass auch in der Zielarchitektur Individualentwicklungen weiterhin einen großen Anteil der Anwendungen der IT-Landschaft ausmachen. Bei Komplexitätsaspekt C_{R_1} fällt besonders der starke Rückgang der Heterogenität der Schnittstellentechnologien von einem Wert von 3,36 auf 0,66 auf. Dies ist mit der Einführung einer zentralen Integrationslösung zu begründen, die zu einer umfangreichen Ablösung unterschiedlicher Schnittstellentechnologien sowohl für realtime- als auch batch-orientierte Übertragungen führt. Die Ablösung individueller durch wiederverwendbare Schnittstellen wird begleitet durch die Einführung zusätzlicher Schnittstellen zu beispielsweise neuen Anwendungen für das Reporting. Hierdurch ist für die Zielarchitektur ein deutlicher Anstieg der genutzten Schnittstellen von 40 auf 67 zu beobachten (C_{R_2}).

Auch im Hinblick auf die Komplexität der Plattformen (C_{T_2}) ist auf der einen Seite eine Reduktion der Heterogenität der verwendeten Plattformen zu erkennen (von 2,18 auf 1,78). Auf der anderen Seite nimmt die Intensität der Plattformnutzung jedoch zu (von 29 auf 31). Die niedrigere Heterogenität kann dabei mit der Reduzierung der Anzahl an Plattformvarianten begründet werden. Hingegen kommt es durch die Einführung zusätzlicher Anwendungen, wie u. a. Reporting und Solvency II, zu einem Anstieg der Plattformnutzung.

In Abbildung 42 sind die Veränderungen der strukturellen Komplexität bei Durchführung der geplanten Transformation für die Komplexitätsaspekte C_{T_1} , C_{T_2} , C_{R_1} sowie C_{R_2} visuell veranschaulicht.

⁶⁰ Vgl. hierzu auch Schmidt et al. (2013).

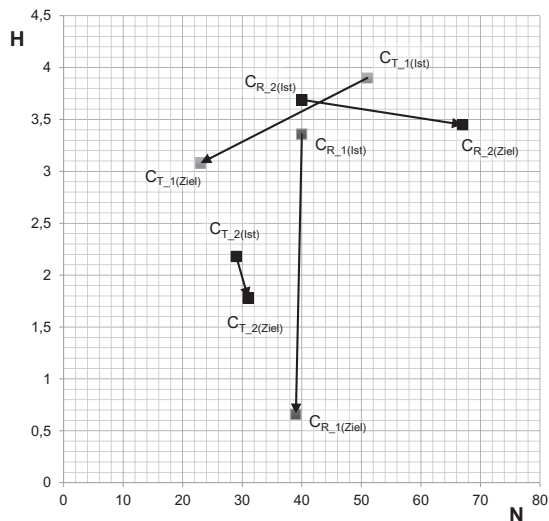


Abbildung 42: Veränderung der Komplexität bei Durchführung der Transformation (1)

Bei den sowohl die Geschäftsprozess- als auch die Anwendungsarchitektur betreffenden Komplexitätsaspekten $C_{T,3}$, $C_{T,4}$ und $C_{T,5}$ weisen die Veränderungen der jeweiligen Anzahl und Heterogenität stets die gleiche Richtung auf (vgl. Tabelle 20 sowie Abbildung 43).

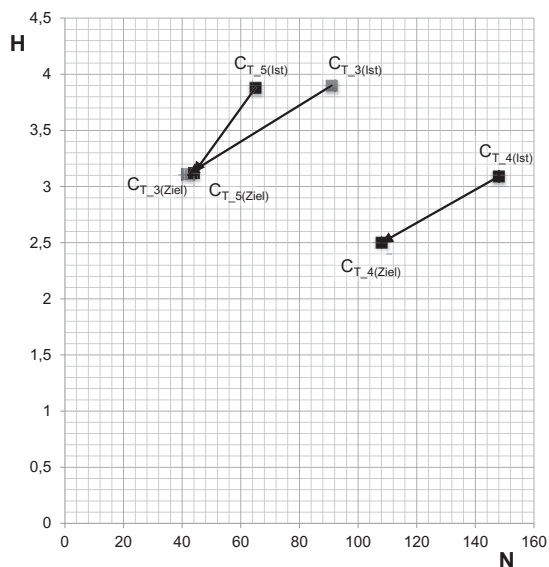


Abbildung 43: Veränderung der Komplexität bei Durchführung der Transformation (2)

Bezüglich der Komplexität der Anwendungsnutzung durch die Geschäftsfelder (C_{T_3}) führt die Einführung geschäftsfeldübergreifender Anwendungen sowie die Konsolidierung der bestehenden Lösungen zu einer deutlichen Reduktion der Komplexität. Auch die Komplexität der durch Anwendungen bereitgestellten Geschäftsfunktionen (C_{T_4}) fällt in der Zielarchitektur wesentlich geringer aus als in der aktuellen Architektur. Dies ist durch die Konzentration der Funktionalität auf eine deutlich geringere Anzahl zentraler Kernsysteme – vornehmlich im Bereich Accounting – sowie der gleichzeitigen Abschaltung geschäftsfeldspezifischer Lösungen zu begründen.

Ein vergleichbarer Effekt ist auf Ebene der Anzahl und Heterogenität der durch die Organisationseinheiten genutzten Anwendungen zu beobachten (C_{T_5}). Hier führt die Ablösung spezifischer, lokaler Lösungen durch zentrale Anwendungen zu einer Reduktion der Komplexität.

Die hier vorgestellten Anwendungsbeispiele der vorgeschlagenen Maßzahl wurden durch die Spezialversicherung zur Beurteilung der Veränderung der aktuellen IT-Architektur hin zur geplanten Zielarchitektur verwendet. Dabei wurde die Veränderung der Komplexität als eines der Kriterien zur Bewertung der angedachten Transformation herangezogen.

5.9 Zusammenfassung und Diskussion der Ergebnisse

In diesem Kapitel wurde die vierte Forschungsfrage der vorliegenden Dissertation bearbeitet:

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/ IT-Architekturen quantifiziert werden?

Auf Basis der in Unterkapitel 3.2 eingeführten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen wurde hierzu die erste holistische Vorgehensweise zur Quantifizierung von Komplexität in Unternehmensarchitekturen entwickelt. Holistisch insofern, als dass das Maß auf alle Ebenen (bzw. Subsysteme) einer Unternehmensarchitektur anwendbar ist.

Die Entropie erwies sich als ein geeignetes Maß zur Quantifizierung von Heterogenität. Vor allem die Eigenschaft der Entropie, Charakteristiken mit einer relativ kleinen Anzahl von Elementen merkbar zu berücksichtigen, ist eine klare Stärke der Entropie gegenüber alternativen Konzentrationsmaßen. Auch das eingeführte Konzept „Evenness“ profitiert von dieser Eigenschaft der Entropie. Charakteristiken mit einer geringen Anzahl zugeordneter Elemente werden für den Architekten sichtbar gemacht und können im Hinblick auf eine mögliche Migration untersucht werden.

Sowohl das Entropie-Maß als Dual-Konzept (vgl. Abschnitt 5.5.3 sowie Unterkapitel 5.6) als auch das auf der Entropie basierende Konzept „Evenness“ (in Kombination mit dem Konzept „Richness“) erwiesen sich im Rahmen der Anwendungsbeispiele als sehr aussagekräftige Maßzahlen.

5.9.1 Theoretische Implikationen

Die entwickelte Maßzahl greift mit dem Konzept der Entropie auf ein anerkanntes und auch in anderen Disziplinen etabliertes Maß zurück. Die vorgeschlagene Maßzahl gründet demnach auf einer sehr gut erforschten Basis (Theil 1967). Ein Beitrag der in diesem Kapitel vorgestellten Forschung besteht demzufolge in der Übertragung bestehender Konzepte aus den Bereichen der VWL und der Biologie auf den Bereich komplexer IT-Architekturen.

Im Zuge der Entwicklung der Maßzahl wurde zudem deutlich, dass es sich bei Komplexität – speziell der Heterogenität – um ein mehrdimensionales Problem handelt. Neben der bereits im Rahmen der eingeführten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen vorgenommenen Unterteilung in Anzahl und Heterogenität der Komponenten und Beziehungen einer EA, zeigt diese Forschung die Erfordernis einer weiteren Ausdifferenzierung der Heterogenität auf. Die aus dem Feld der Biologie entliehenen Konzepte der Richness und Evenness, als zwei verwandte, jedoch eigenständige Konzepte, komplementieren dabei die integrale Perspektive auf die Heterogenität.

5.9.2 Praktische Implikationen

Aus einer praktischen Perspektive wurde ein flexibles und auf die unterschiedlichsten Kontexte und Fragestellungen anpassbares Maß zur Quantifizierung von Komplexität entwickelt. CIOs und IT-Architekten haben die Möglichkeit, frei eine Architekturebene zu wählen, deren Komplexität sie untersuchen möchten. Auf dieser Basis ist eine bislang nur „gefühlte“ Komplexität unterschiedlicher Bereiche der Unternehmensarchitektur quantifizierbar und vergleichbar.

Wie das dritte Fallbeispiel (vgl. Abschnitt 5.8.3) gezeigt hat, können die errechneten Maßzahlen die Entscheider bei der Weiterentwicklung einer Unternehmensarchitektur unterstützen. Hierdurch kann ein unkontrolliertes Wachstum der Architektur verhindert und eine intensivere Nutzung der bereits bestehenden Architektur unterstützt werden, was zu einer höheren Effizienz sowie einer höheren Flexibilität hinsichtlich der Implementierung neuer Anforderungen führen kann. Die Maßzahl ermöglicht außerdem eine Analyse der bestehenden Architektur, um potenzielle Verbesserungsmöglichkeiten im Hinblick auf die Komplexität zu identifizieren (vgl. Abschnitt 5.8.1).

Darüber hinaus betonten die kooperierenden IT-Architekten den Vorteil des holistischen Ansatzes und der einheitlichen Berechnungsvorschrift der Maßzahl für die unterschiedlichen Kontexte und Fragestellungen. Durch die konsistente Vorgehensweise bei der Definition und Berechnung der Maßzahlen sind diese gegenüber anderen Mitarbeitern sowie dem Management einfacher kommunizierbar.

Um den vorgestellten Ansatz im Unternehmen zu verwenden, ist eine ausreichende Dokumentation der bestehenden Unternehmensarchitektur notwendig. Eine spezifische Modellie-

rung der Daten ist dabei nicht erforderlich.⁶¹ Die typischen, auf gängigen Frameworks wie TOGAF basierenden, Modellierungen sind ohne weiteres verwendbar. In Schmidt et al. (2013) wurde ein Ansatz zur Quantifizierung von Komplexität auf Basis von Architektur-Metamodellen eingeführt, der auf dem hier vorgestellten Ansatz basiert.

5.9.3 Limitationen und weiterer Forschungsbedarf

Eine Limitation des vorgestellten Ansatzes liegt darin, dass Ähnlichkeiten zwischen den Charakteristiken durch die eingeführte Maßzahl zur Quantifizierung von Komplexität nicht berücksichtigt werden. Dies könnte insofern eine interessante Erweiterung darstellen, da insbesondere bei verwandten Charakteristiken, wie z. B. MS SQL oder Oracle MySQL, keine vollständig unterschiedlichen Anforderungen an Personal und Dokumentation bestehen. Eine Weiterentwicklung könnte damit in der Integration des Aspektes der „Nähe der Charakteristiken zueinander“ bestehen.

Weiterer Forschungsbedarf besteht darüber hinaus in der Identifikation der Komponenten und Beziehungen, deren Komplexität typischerweise berechnet werden sollte, um die Komplexität der IT-Architektur unterschiedlicher Unternehmen bestimmen und vergleichen zu können. Dieser Aspekt wäre auch aus praktischer Perspektive z. B. für die Erstellung von Branchenbenchmarks sehr hilfreich.

⁶¹ In der Arbeit Schneider et al. (2015) wurde der Ansatz auch bereits anhand der Architekturdaten von vier Unternehmen unterschiedlicher Industriezweige verwendet.

6 Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen⁶²

In diesem Kapitel werden im Rahmen eines mehrstufigen Forschungsprojektes in Zusammenarbeit mit zwei Banken Designprinzipien für ein Informationssystem erarbeitet, das IT-Architekten bei dem Management – speziell der Planung und Kontrolle – der Komplexität der IT-Architektur unterstützt. Nach einer Motivation der bearbeiteten Forschungsfrage in Unterkapitel 6.1 wird in Unterkapitel 6.2 die verwendete Methodik (Action Design Research) vorgestellt. Im Anschluss daran wird in Unterkapitel 6.3 die im Rahmen des Forschungsprojektes behandelte Problemklasse („Planung und Kontrolle der Komplexität der IT-Architektur“) eingeführt und in Unterkapitel 6.4 das grundlegende iterative Vorgehen dargestellt. In den Unterkapiteln 6.5, 6.6 und 6.7 werden die Ergebnisse der drei aufeinander aufbauenden Teilprojekte präsentiert. Das Kapitel schließt in Unterkapitel 6.8 mit einer Diskussion der theoretischen und praktischen Implikationen des Forschungsprojektes, den mit diesem verbundenen Limitationen sowie weiterem Forschungsbedarf.

6.1 Motivation

In einer Vielzahl von Unternehmen haben sich über die Zeit hoch komplexe IT-Landschaften herausgebildet. Ross et al. (2006) berichten von Banken, deren Systemlandschaften im Laufe der Jahre so komplex wurden, dass es an ein Wunder grenzt, dass diese überhaupt funktionierten. Für die Umsetzung neuer Geschäftsanforderungen wurden die Legacy-Systeme zunehmend „verbastelt“, was zu steigenden Kosten und einer höheren Rigidität der Systeme führte (Ross et al. 2006). Eine Befragung unter mehr als 1100 CIOs und IT-Managern aus Unternehmen unterschiedlicher Größe und Branchen zeigt, dass die Legacy-Problematik der Änderbarkeit der IT-Landschaften vieler Unternehmen entgegensteht (Capgemini 2014). Zudem befinden sich die Unternehmen in einem Umfeld immer schneller aufkommender Anforderungen durch die Kundenseite, aber auch durch die eigenen Mitarbeiter. Während junge Unternehmen häufig die erforderliche Flexibilität mitbringen, die Erwartungen der Kundenseite in Form einer zeitnahen Bereitstellung neuartiger Lösungen – implementiert mit den modernsten Technologien – zu erfüllen, ist dies für Unternehmen mit einer historisch gewachsenen IT-Landschaft eine weitaus schwierigere Aufgabe (Bossert et al. 2014). Vor diesem Hintergrund werden der Anspruch und die Relevanz der Aufgabe des IT-Architekten, die IT-Architektur zu managen, noch wesentlich plastischer. Der IS-Literatur zufolge gehört es zu seinen Aufgaben, die IT-Architektur „im Takt“ mit der Veränderung des Unternehmens zu

⁶² In die erste Phase des dargestellten Forschungsprojektes ist die Arbeit Schütz et al. (2013a) eingeflossen.

halten und gleichzeitig die Komplexität der IT-Architektur zu managen (Guillemette & Paré 2012).

Die wissenschaftliche Literatur gibt hier nur ansatzweise Hilfestellungen. Die Arbeiten von Xia und Lee betonen, dass bei einer Weiterentwicklung der IT-Landschaft stets sowohl die neuen Technologien auf dem Markt als auch die aktuell im Unternehmen vorhandene IT-Landschaft berücksichtigt werden müssen (Lee & Xia 2002; Xia & Lee 2005). Adomavicius et al. (2008) untersuchen in ihrer Arbeit, wie die IT-Manager oder IT-Architekten dabei unterstützt werden können, Prognosen über die künftige Entwicklung einer Technologie zu treffen, um auf dieser Grundlage die zukunftssträchtigste Technologie auswählen zu können. Eine Berücksichtigung der aktuell im Unternehmen vorliegenden IT-Architektur bleibt hierbei hingegen aus, obwohl die Veränderung der bestehenden IT-Architektur eine besonders große Rolle spielt, da diese wiederum die Basis für die zukünftig vorzunehmenden Änderungen darstellt.

Es existiert eine Reihe professioneller EAM-Lösungen, die für das Management von IT-Architekturen verwendet werden. Diese Informationssysteme verfügen jedoch nicht über spezifische Werkzeuge, um die Komplexität der IT-Architektur zu planen und zu kontrollieren. Auch die aktuelle wissenschaftliche Literatur bietet keine Antwort darauf, wie ein Artefakt ausgestaltet sein sollte, welches den Entscheider an dieser Stelle unterstützt.

Im Rahmen des im Folgenden vorgestellten Forschungsprojektes mit zwei Banken werden unter Verwendung der Methode „Action Design Research“ (Sein et al. 2011) IT-Artefakte erarbeitet, die IT-Manager und IT-Architekten bei der Planung und Kontrolle der Komplexität der IT-Architektur unterstützen. Gleichzeitig wird die Wissensbasis durch präskriptives Wissen bzgl. der Form und Funktion von IT-Artefakten zur Planung und Kontrolle von Komplexität in IT-Architekturen erweitert. In diesem Kapitel wird die folgende Forschungsfrage beantwortet:

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

6.2 Methodische Grundlagen

Zur Bearbeitung der in dieser Studie betrachteten Fragestellung wird ein sogenannter „Engaged Scholarship“-Ansatz gewählt (Van de Ven 2007). Van de Ven (2007) diskutiert vier unterschiedliche Typen des „Engaged Scholarship“, zu denen u. a. Action Research sowie Design Science zählen. Jüngst wurde von Sein et al. (2011) mit „Action Design Research“ (ADR) eine Methode vorgeschlagen, welche diese beiden Typen des „Engaged Scholarship“ in einer Methode vereint und bereits bestehende, prominente Vorbilder im Bereich der Entwicklung von Designtheorien berücksichtigt (Lindgren et al. 2004; Markus et al. 2002). In den folgenden Abschnitten werden mit *Design Science Research* (Abschnitt 6.2.2) und *Action Research* (Abschnitt 6.2.3) die beiden methodischen Bestandteile von ADR vorgestellt. Zuvor wird hierfür das Konzept des *Artefaktes* (Abschnitt 6.2.1) eingeführt. Abschließend wird die

in der Studie verwendete Methode des *Action Design Research* mit ihren Phasen und Schritten dargestellt (Abschnitt 6.2.4).

6.2.1 Artefakt

„The engineer, and more generally the designer, is concerned with how things ought to be – how they ought to be in order to attain goals, and to function.“ (Simon 1996, S. 4, 5)

Aus diesem Zitat – aus Simons Diskussion zur Bedeutung des Begriffs *artificial* (Simon 1996) – werden verschiedene für das Verständnis des Begriffs Artefakt relevante Aspekte deutlich: Der Designer sucht nach einer *Lösung für eine spezifische Problemstellung* („to attain goals“). Während des Designprozesses befasst sich der Designer mit der Fragestellung, wie die *Lösung ausgestaltet sein sollte* („how things ought to be“), um *in dem existierenden Kontext* das betrachtete Problem zu lösen („to function“). Diese Punkte greift Simon (1996) erneut im folgenden Zitat auf:

„Fulfillment of purpose or adaptation to a goal involves a relation among three terms: the purpose or goal, the character of the artifact, and the environment in which the artifact performs.“ (Simon 1996, S. 5)

Aus der Tatsache, dass der Designer eine Lösung konstruiert, die eine dingliche Beschaffenheit aufweisen kann („thing“), lässt sich schließen, dass ein Artefakt ein materiell existierendes Objekt darstellen kann. Gleichzeitig entwickelt der Designer ein immaterielles Artefakt in Form der Erkenntnis, wie ein dingliches Artefakt ausgestaltet sein sollte, damit es geeignet ist, das Problem – in dem betrachteten Kontext – zu lösen.⁶³ Der Gesichtspunkt, dass ein künstlich hergestelltes Artefakt (Simon (1996) verwendet auch den Begriff „synthesized“) in der realen Welt samt den ihr zugrunde liegenden Gesetzmäßigkeiten funktionieren soll, weist darauf hin, dass solch ein Artefakt nicht losgelöst von Naturgesetzen oder verhaltenswissenschaftlichen Erkenntnissen gebildet werden kann (Hevner et al. 2004):

„[...] their creation relies on existing kernel theories that are applied, tested, modified, and extended through the experience, creativity, intuition, and problem solving capabilities of the researcher.“ (Hevner et al. 2004, S. 76)

In Anlehnung an Orlikowski & Iacono (2001) wird ein Artefakt als künstliches, von Menschenhand kreierte und aus einem spezifischen Kontext – d. h. zu einem bestimmten Zeitpunkt, an einem bestimmten Ort, für einen inhaltlichen Gegenstand und eine bestimmte Zielgruppe – hervorgegangenes Objekt verstanden. Veränderungen des Kontextes können eine Anpassung des Objektes erforderlich machen. Ein Artefakt ist demzufolge kein von seinem Kontext isoliertes oder statisches Objekt, sondern muss an sich ändernde Gegebenheiten angepasst werden.

Die vorliegende Dissertation ist auf IT-Artefakte – beispielsweise Software – fokussiert, die für einen spezifischen organisationalen Kontext erstellt wurden und den Nutzer bei der Lösung einer spezifischen Problemstellung unterstützen. Dabei folgt die Arbeit dem Verständnis

⁶³ Es handelt sich hierbei um präskriptives Wissen. Dieser Punkt wird im folgenden Abschnitt näher diskutiert.

von Artefakten als sogenannte *ensemble artifacts* (vgl. hierzu Orlikowski & Iacono 2001), wobei Strukturen und Gegebenheiten des organisationalen Kontextes während der Entwicklung und Nutzung des Artefaktes in selbiges eingehen.⁶⁴

6.2.2 Design Science Research

Die designorientierte Forschung hat ihre Wurzeln im Bereich des Ingenieurwesens oder allgemeiner ausgedrückt, in der Erforschung künstlicher, durch Menschenhand erstellter Systeme (Simon 1996). Während sich die Naturwissenschaften damit auseinandersetzen, Phänomene der Realität zu verstehen und diese zu erklären, zielt Design Science darauf ab, Artefakte zu erstellen, die einem bestimmten Verwendungszweck dienen und Menschen dabei unterstützen, ein bestimmtes Ziel zu erreichen (March & Smith 1995; Simon 1996). Wie von Gregor (2002) angeführt, komplementiert die Design Science Forschung damit die der Behavioral Science. Beschäftigt sich letztere im IS-Kontext mit menschlichem Verhalten, beispielsweise im Zuge der Adoption von IT-Systemen (Venkatesh & Davis 2000), setzt die Design Science Forschung den Fokus auf die Ausgestaltung – das Design – der IT-Artefakte zur Lösung eines spezifischen Problems bzw. einer Klasse von Problemen.

Neben der Entwicklung neuartiger IT-Artefakte strebt die Forschung in der Design Science die Identifikation präskriptiven Wissens an (Hevner et al. 2004; March & Smith 1995), das in Form von Designtheorien formalisiert und kommuniziert werden kann (Gregor & Jones 2007). Während der Implementierung eines IT-Artefaktes zur Lösung eines Problems manifestieren sich spezifische Eigenschaften hinsichtlich der Form und Funktion des IT-Artefaktes. Eine Beschreibung der für die Problemlösung relevanten Form und Funktion stellen sogenannte Designprinzipien⁶⁵ dar (Markus et al. 2002; Pries-Heje & Baskerville 2008). Nach Gregor & Jones (2007) handelt es sich bei Designprinzipien um eine wesentliche Komponente einer Designtheorie und demzufolge um einen Teilaspekt präskriptiven Wissens. Durch Anwendung dieses präskriptiven Wissens – der Designprinzipien – ist es möglich, weitere Instanzen von IT-Artefakten für andere Kontexte zu erstellen, die Probleme der gleichen Problemklasse lösen (Purao 2002).

Eine wichtige Charakteristik der Design Science stellt die Evaluation der implementierten Lösung dar, die als Grundlage für die Erweiterung der Wissensbasis dienen soll. In der Literatur wird dabei sowohl eine (Ex-post-)Evaluation des Artefaktes (Hevner 2007; Hevner et al. 2004; March & Smith 1995) als auch die des Designprozesses (Pries-Heje et al. 2008) disku-

⁶⁴ Dieses Verständnis eines IT-Artefaktes ist vergleichbar mit dem Begriff der "Informationssysteme" als sozio-technische Systeme, bestehend aus menschlichen sowie maschinellen Komponenten (Krcmar 2009).

⁶⁵ Neben Beispielen aus der IS-Literatur (Markus et al. 2002; Walls et al. 1992) zeigt auch die Literatur der Domäne der Unternehmensarchitekturen ein grundlegendes Verständnis bzgl. Designprinzipien und der Form, wie diese darzustellen sind, auf (Fischer et al. 2010; Winter & Aier 2011). Greefhorst & Proper (2011), Richardson et al. (1990) und Widjaja & Gregory (2012) schlagen vor, Designprinzipien in Gestalt von drei Facetten darzustellen: (1) ein Statement, welches die grundlegende Idee des Designprinzips transportiert, (2) eine Begründung, warum das Designprinzip befolgt werden sollte und (3) die Implikationen der Umsetzung des Designprinzips. Die vorliegende Arbeit folgt dieser Struktur, verortet die einzelnen Facetten jedoch an unterschiedlichen Bereichen dieses Kapitels. Die Statements sowie die Begründungen der abgeleiteten Designprinzipien werden jeweils in den Ergebnisteilen vorgestellt, wohingegen die Implikationen der Designprinzipien in der Diskussion zu den praktischen Implikationen der Forschungsergebnisse präsentiert werden.

tiert. Wie in Abschnitt 6.2.4 näher erläutert wird, sieht die Methode des in diesem Kapitel vorgestellten Forschungsprojektes eine fortwährende Evaluation des implementierten IT-Artefaktes und der abgeleiteten Designprinzipien bereits während des Entstehungsprozesses vor.

Eine Schwierigkeit der Design Science sehen March & Smith (1995) darin, dass das Verhalten eines entwickelten IT-Artefaktes von dem unmittelbaren Einsatzgebiet des IT-Artefaktes abhängt. Ein unvollständiges oder falsches Verständnis des Einsatzgebietes oder seines Umfelds kann zu einem zweckwidrig ausgestalteten Artefakt führen. Die in Abschnitt 6.2.3 eingeführte Methode Action Research stellt – wie in Abschnitt 6.2.4 näher beleuchtet wird – eine geeignete Ergänzung zur designorientierten Forschung dar, da Action Research die praktische Problemdomäne unmittelbar mit der theoretischen Perspektive und Zielsetzung des Forschers kombiniert.

6.2.3 Action Research

Action Research ist eine Forschungsmethode, die es ermöglicht, Synergien zwischen Praxis und Forschung herzustellen, sodass die Erkenntnisse der Forschung in die Praxis übertragen werden können und gleichzeitig die Praxis die Möglichkeit neuer Erkenntnisse für die Forschung bietet (Avison et al. 1999):

„Action research combines theory and practice (and researchers and practitioners) through change and reflection in an immediate problematic situation within a mutually acceptable ethical framework. Action research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning.“ (Avison et al. 1999 , S. 94)

Davison et al. (2004) führen in ihrer Arbeit fünf Prinzipien für die sogenannte Canonical Action Research (CAR) ein.⁶⁶ Diese Prinzipien sollen Forschern als methodische Richtlinien dienen und sie dabei unterstützen, eine methodisch stringente Durchführung sowie die Relevanz der Forschung sicherzustellen. Gleichzeitig wird anhand dieser Prinzipien der wesentliche Charakter des Action Research deutlich. Davison et al. (2004) schlagen die folgenden Prinzipien für CAR vor (Davison et al. 2004, S. 69):

- (1) Prinzip des Übereinkommens zwischen Forscher und Praxispartner
- (2) Prinzip des zyklischen Prozessmodells
- (3) Prinzip der Theorie
- (4) Prinzip der Veränderung durch Aktion
- (5) Prinzip des Lernens durch Reflektion

Das erste Prinzip ist unmittelbar mit dem Charakter des Action Research als einer kooperativen Forschungsmethode verflochten. Prinzip zwei bezieht sich auf das von Susman &

⁶⁶ „Canonical“ signalisiert hierbei die Berücksichtigung des durch Susman & Evered (1978) entwickelten iterativen, kollaborativen prozessorientierten Modells für Action Research, dessen Anwendung in den Sozialwissenschaften breite Zustimmung erfährt (Davison et al. 2004).

Evered (1978) eingeführte Prozessmodell für Action Research mit den Schritten (a) der Diagnose der betrachteten Problemstellung, (b) der Planung und Vorbereitung der Aktion, (c) der organisationalen Intervention, (d) der Evaluation sowie (e) der Reflektion und des Lernens im Hinblick auf die Folgen der Aktion. Das dritte Prinzip unterstreicht die Zielsetzung des Action Research in Form der Generierung neuen Wissens, wobei bereits die durchgeführten Eingriffe – die Aktion – auf theoretischen Überlegungen basieren sollten. Die Veränderung durch Aktion (Prinzip vier) untermauert den Fokus auf die Betrachtung organisationaler Einflüsse des Eingriffs in Form der durchgeführten Aktion. Prinzip fünf stellt mit „Lernen durch Reflektion“ einen der Schritte des Prozessmodells nach Susman & Evered (1978) explizit in den Vordergrund. Während der Durchführung der einzelnen organisationalen Eingriffe soll hierdurch sichergestellt werden, dass während des Projektes gewonnene Ideen und Erkenntnisse in die weiteren Iterationen einfließen. Weiterhin wird hierdurch auch die fortwährende Reflektion über die (Zwischen-)Ergebnisse und deren Überarbeitung hervorgehoben. Für die Praxisseite stehen an dieser Stelle i. d. R. die praktischen Implikationen im Vordergrund, wobei der Fokus des Forschers (in Kombination mit dem Prinzip der Theorie) auf der iterativen Generierung von Wissen und der damit verbundenen Erweiterung der Wissensbasis steht.

Die im folgenden Abschnitt vorgestellte Methode des Action Design Research verbindet das Paradigma der Design Science mit der Methode Action Research. Wie sich zeigen wird, finden sich die hier vorgestellten Prinzipien auch in Action Design Research wieder.

6.2.4 Action Design Research⁶⁷

Design Research nimmt häufig eine sehr technologieorientierte Perspektive auf das zu erstellende IT-Artefakt ein. Ferner sieht Design Research für die Evaluation des aus dem Entstehungsprozess resultierenden IT-Artefaktes eine gesonderte Phase im Anschluss an die Entwicklung des Artefaktes vor. Dieses Vorgehen führt dazu, dass der organisationale Kontext, in dem das IT-Artefakt eingesetzt werden soll, nur einen sehr begrenzten Einfluss auf die Ausgestaltung des IT-Artefaktes nehmen kann. Dass ein IT-Artefakt vielmehr aus der Interaktion innerhalb des organisationalen Kontextes entsteht, wird durch eine derartige Vorgehensweise nicht ausreichend berücksichtigt (Sein et al. 2011). Genau diese Schwachstelle bestehender Methoden der Design Science Forschung möchten Sein et al. (2011) mit der ADR-Methode beheben. Der Hauptgedanke von ADR ist dabei, dass der organisationale Kontext – sowohl in der Phase der Entstehung als auch des Einsatzes – wesentlich zur Ausgestaltung des IT-Artefaktes beiträgt. Hierzu kombiniert ADR eine organisationale Intervention und deren Beobachtung (die „Action“-Komponente) mit der Forschungsperspektive des Design Science Research (DSR). Der Bedarf an der Einführung einer solchen Methode in die IS-Forschung zeigt sich anhand bekannter DSR Projekte, die sowohl instanziierte Artefakte („material artifacts“) als auch Designtheorien („abstract artefacts“) entwickeln (Gregor & Jones 2007) und zudem organisationale Einflüsse bei Einführung eines instanziierten Artefaktes in dem entsprechenden Kontext diskutieren und berücksichtigen (Lindgren et al. 2004; Markus et al. 2002).

⁶⁷ Vgl. hierzu auch Schütz et al. (2013a)

Im Folgenden werden die ADR-Methode und ihre einzelnen Phasen näher erläutert. ADR ist in vier Phasen (engl. „stages“) strukturiert (vgl. Abbildung 44), wobei jeder dieser Phasen eine Menge von Schritten („tasks“) zugeordnet ist, die ihrerseits eng miteinander verflochten sind. In Tabelle 21 ist angeführt, welche Schritte typischerweise in welcher der Phasen durchlaufen werden.

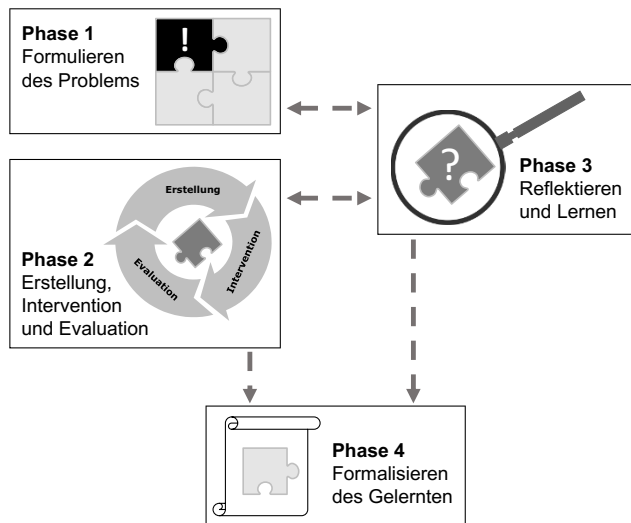


Abbildung 44: Phasen der ADR-Methode nach Sein et al. (2011)

Die erste Phase „*Formulierung des Problems*“ illustriert den Charakter von ADR als einen „Engaged Scholarship“-Ansatz (Van de Ven 2007) anhand der auf einer praktischen Problemstellung gründenden Forschungsfrage. Die zweite Phase „*Erstellung, Intervention und Evaluation*“ (im Englischen „Building, Intervention and Evaluation“, daher im Folgenden abgekürzt mit „BIE“) zeichnet sich durch ihre aufeinanderfolgenden „BIE-Zyklen“ aus. In diesen wird das initiale Design erstellt und das hieraus entstehende IT-Artefakt anschließend im Rahmen organisationaler Eingriffe und einer periodisch in jedem der Zyklen durchgeführten Evaluation überarbeitet. Die dritte Phase „*Reflektieren und Lernen*“ verläuft parallel zu den beiden ersten Phasen. Hier findet ein Reflektieren im Hinblick auf das Design und die erforderlichen Anpassungen des Designs des IT-Artefaktes sowie die Evaluation der – das Artefakt beschreibenden – Designprinzipien statt. Konzeptionell bewegt sich der Forscher hier „von der Erstellung einer Lösung für eine spezifische Instanz hin zu einer Anwendung des Gelernten auf eine weiter gefasste Klasse von Problemen“ (Sein et al. 2011 S. 44). In der vierten Phase „*Formalisieren des Gelernten*“ werden die Ergebnisse generalisiert und damit auf ein abstrakteres Level gehoben, wodurch diese auf eine Klasse von Problemen anwendbar sind. Aufgrund der kontinuierlichen Anpassungen des IT-Artefaktes im Zuge organisationaler Eingriffe und der dabei einfließenden Erfahrungen während des Prozesses der Entwicklung, Verwendung und Verbesserung des Artefaktes, wird ein solches bei ADR als „ensemble“ (vgl. Abschnitt 6.2.1) verstanden (Orlikowski & Iacono 2001; Sein et al. 2011).

Tabelle 21: Phasen und Schritte der ADR-Methode⁶⁸

Phase (P)	Forschungsschritt (S)
Formulierung des Problems (P1)	(P1.S1) Identifiziere und konzeptualisiere den genauen Forschungsbereich
	(P1.S2) Formuliere die initialen Forschungsfragen
	(P1.S3) Stelle das Problem als eine Instanz einer Klasse von Problemen dar
	(P1.S4) Identifiziere die theoretische Basis und bereits bestehende geeignete technologische Lösungen
	(P1.S5) Stelle ein langfristiges Engagement des Unternehmens sicher
	(P1.S6) Definiere Rollen und Verantwortlichkeiten
Erstellung, Intervention und Evaluation (P2)	(P2.S1) Stelle das initiale Ziel für die Wissensbildung heraus
	(P2.S2) Wähle die BIE-Form und passe diese ggf. an
	(P2.S3) Führe die BIE-Zyklen durch
	(P2.S4) Beurteile das Erfordernis weiterer Zyklen, wiederhole
Reflektieren und Lernen (P3)	(P3.S1) Überdenke das Design und führe Designveränderungen durch
	(P3.S2) Überprüfe die Einhaltung der Prinzipien
	(P3.S3) Analysiere die Ergebnisse der Intervention gemäß der gesetzten Ziele
Formalisieren des Gelernten (P4)	(P4.S1) Formalisiere das Gelernte für eine Klasse von Problemen
	(P4.S2) Teile die Ergebnisse und führe eine Überprüfung dieser mit den Praktikern durch
	(P4.S3) Formuliere die Ergebnisse in Form von Designprinzipien
	(P4.S4) Artikuliere das Gelernte im Hinblick auf die gewählten Theorien
	(P4.S5) Formalisiere die Ergebnisse für eine Verbreitung

Die Evaluation des Artefaktes und der Designtheorie finden innerhalb eines kontinuierlichen Prozesses statt. Entgegen der vorherrschenden Überzeugung, dass DSR einer abschließenden Evaluation am Ende des Forschungsprozesses bedarf (wie u. a. vertreten von March & Smith (1995) oder Hevner et al. (2004)), wird der Rigor der Theoriebildung durch die simultane Evaluation während der BIE-Zyklen sichergestellt.

Die ADR-Methode eignet sich besonders für die in diesem Kapitel dargestellte Forschung, da die Zielsetzung in der Lösung eines praktischen Problems innerhalb eines realen Umfelds liegt (Briggs et al. 2011). Hierbei soll ein Artefakt für das Management – speziell die Planung und die Kontrolle – komplexer IT-Architekturen entwickelt und implementiert werden, wobei zur gleichen Zeit über die gesammelten Erfahrungen reflektiert und diese abstrahiert werden sollen. Ziel dieser Abstraktion ist eine sogenannte „Level 2 contribution“ in Form abstrahierter Designprinzipien, basierend auf der „Level 1 contribution“, d. h. dem instanziierten Artefakt oder den instanziierten Designprinzipien (Gregor & Hevner 2013). Zu diesem Zweck wurden drei aufeinanderfolgende kollaborative ADR-Projekte mit zwei führenden deutschen,

⁶⁸ Vgl. hierzu auch Sein et al. (2011) und Schütz et al. (2013a).

international tätigen Universalbanken durchgeführt, für die das Management komplexer IT-Architekturen jeweils ein aktuelles Problem im Architekturmanagement darstellt.

Hinsichtlich der in diesem Kapitel bearbeiteten Forschungsfrage – nach den Designprinzipien, denen ein Informationssystem folgen sollte, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt – liegt der Schwerpunkt der in dieser Forschung erarbeiteten Designprinzipien auf dem Design des Informationssystems. Im folgenden Unterkapitel wird die zugrunde liegende Problemklasse eingeführt.

6.3 Definition der betrachteten Problemklasse „Planung und Kontrolle von Komplexität in IT-Architekturen“⁶⁹

Das Forschungsprojekt basiert auf einer intensiven Zusammenarbeit mit den IT-Architekten der kooperierenden Banken. Im Kontext dieser Zusammenarbeit wurde eine Reihe von Meetings und Workshops zu unterschiedlichen Themenbereichen der Unternehmensarchitekturen mit den IT-Architekten abgehalten. Hierbei stellte sich heraus, dass eine Vielzahl von Problemstellungen, mit denen die IT-Architekten zu tun haben, einen starken Bezug zu einem immer weiter fortschreitenden Wachstum der Komplexität der IT-Architektur aufweisen (z. B. Themen der Architektur Governance, Verfahren zur Architekturbewertung oder die Messung der Komplexität der Architektur) oder sogar das Resultat dessen sind. Die Teilnahme an regelmäßigen Treffen regionaler IT-Architekten aus verschiedenen Industrien zeigte auf, dass der Umgang mit der Komplexität der IT-Architektur nicht ausschließlich eine große Herausforderung für Unternehmen im Finanz- und Bankensektor darstellt, sondern auch für die IT-Bereiche vieler anderer Industriezweige. Demzufolge handelt es sich bei dem Management der Komplexität von IT-Architekturen nicht um eine individuelle Problemstellung eines einzelnen Unternehmens oder einer spezifischen Industrie. Vielmehr kann das Komplexitätsmanagement der IT-Architektur als eine Problemklasse verstanden werden. Zwar gibt es eine Reihe von Lösungen zur Dokumentation des aktuellen Standes einer IT-Architektur, jedoch existiert bislang keine Lösung, die IT-Architekten bei dem Management – speziell der Planung und Kontrolle – der Komplexität der IT-Architektur unterstützt. Hierdurch können einzelne Teilbereiche einer IT-Architektur nur schwer hinsichtlich der Veränderung der dort herrschenden Komplexität untersucht oder Pläne für die Weiterentwicklung der unternehmensweiten IT-Architektur erstellt werden, welche die Komplexität der bestehenden IT-Architektur berücksichtigen.

Vor diesem praktischen Hintergrund⁷⁰ wurde das Erfordernis abgeleitet, die Entwicklung eines Informationssystems zu untersuchen, welches IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt. Diese Herausforderung findet sich in der Formulierung des hier behandelten Problems wieder:

(a) „*Planung und Kontrolle von Komplexität in IT-Architekturen*“ als die untersuchte Problemklasse und

⁶⁹ Vgl. Schütz et al. (2013a)

⁷⁰ Vgl. hierzu „Principle 1: Practice-Inspired Research“ nach Sein et al. (2011), S. 40.

(b) „Planung und Kontrolle der Komplexität der IT-Architektur der Bank“ als eine Instanz der Problemklasse (a) im Kontext der jeweiligen Bank.

Wie bereits in Abschnitt 4.2.2 dieser Dissertation dargestellt, wird angenommen, dass Komplexität nicht per se als etwas Negatives verstanden werden kann, was es zu vermeiden gilt. Vielmehr ist das Ziel, einen angemessenen Grad an Komplexität zu erreichen. Um IT-Architekten bei der Identifikation dieses angemessenen Grades an Komplexität zu unterstützen, sollen im Folgenden Designprinzipien erarbeitet werden, wie ein Informationssystem ausgestaltet sein sollte, das IT-Architekten in ihrer Praxis einsetzen können (vgl. Forschungsfrage 5).

6.4 Grundlegendes Vorgehen im Rahmen des Forschungsprojektes

Zur Durchführung des Forschungsprojektes wurde ein dreistufiges Vorgehen gewählt. Wie bereits angeführt, bildet die Zusammenarbeit mit zwei deutschen Banken die Basis für die im Rahmen der ADR-Projekte erarbeiteten IT-Artefakte sowie der instanziierten bzw. abstrahierten Designprinzipien (DP). Das ADR-Projekt mit der Bank A stellt dabei die erste Stufe des Forschungsprojektes dar. In Kooperation mit zwei IT-Architekten der Bank und einem siebenköpfigen Entwicklungsteam⁷¹ wurde über einen Zeitraum von 16 Wochen ein Prototyp für ein IT-Artefakt zur Lösung des Problems „Planung und Kontrolle der Komplexität der IT-Architektur der Bank A“ entwickelt. Gleichzeitig entstand eine erste Menge abstrakter Designprinzipien, die sich auf die übergeordnete abstrakte Problemklasse beziehen (vgl. Unterkapitel 6.5). Zur Überprüfung der Ergebnisse des ersten ADR-Projektes in einem anderen organisationalen Kontext wurde in der zweiten Stufe ein ADR-Projekt mit einer weiteren Bank B durchgeführt. Ausgehend von den aus dem ersten ADR-Projekt resultierenden abstrakten Designprinzipien wurde auch hier – der ADR-Methode nach Sein et al. (2011) folgend – ein IT-Artefakt implementiert. Seitens der Bank B waren erneut zwei Architekten in das ADR-Projekt involviert. Das Entwicklungsteam bestand aus acht Personen, die den Prototypen über einen Zeitraum von 15 Wochen erstellten (vgl. Unterkapitel 6.6). Im Zuge des zweiten ADR-Projektes emergierten zwei zusätzliche abstrakte Designprinzipien. Diese wurden in der dritten Stufe durch zwei weitere ADR-Zyklen im organisationalen Kontext von Bank A erneut überprüft (vgl. Unterkapitel 6.7). Abbildung 45 gibt einen Überblick über die Struktur des Forschungsprojektes.⁷² Aus dieser Abbildung werden zudem das iterative Vorgehen und der evolutionäre Charakter der entwickelten Designprinzipien deutlich.

⁷¹ Bei den Entwicklern handelte es sich in allen Teilprojekten um Studierende der Wirtschaftsinformatik.

⁷² „IT-COM“, „IT-COM_2“ und „IT-COM_3“ bezeichnen die in dem jeweiligen ADR-Projekt entwickelten IT-Artefakte.

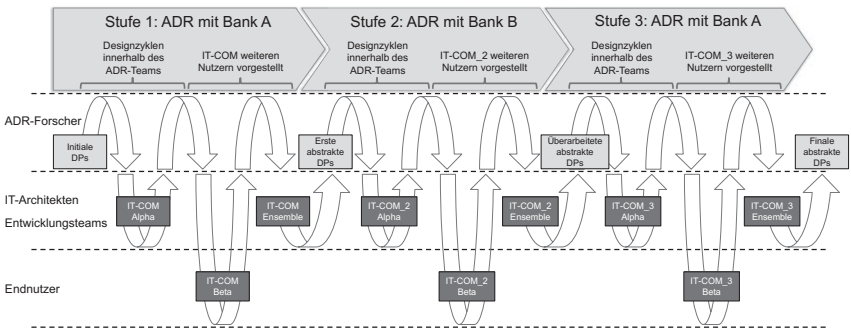


Abbildung 45: Entwicklung der DPs im Rahmen des dreistufigen Forschungsprojektes

Tabelle 22 fasst die Zuordnung der Kooperationspartner zu den einzelnen Stufen sowie die Zusammensetzung des jeweiligen ADR-Teams zusammen.

Tabelle 22: Organisation der Stufen des Forschungsprojektes

Stufe des Forschungsprojektes	Kooperationspartner	Zusammensetzung des ADR-Teams	Projektdauer
1 (Erstes ADR-Projekt)	Bank A	2 Forscher, 2 Architekten, 7 Entwickler	16 Wochen
2 (Zweites ADR-Projekt)	Bank B	2 Forscher, 2 Architekten, 8 Entwickler	15 Wochen
3 (Erweiterung des ersten ADR-Projektes)	Bank A	2 Forscher, 2 Architekten, 2 Entwickler	14 Wochen

In der folgenden Tabelle 23 werden die Verknüpfung⁷³ der einzelnen Teilprojekte anhand der sukzessiv weiterentwickelten Designprinzipien und die jeweiligen Zwischenergebnisse der ADR-Projekte dargestellt.

⁷³ Die Verknüpfung der einzelnen Teilprojekte wird dabei über die fett markierten Passagen hervorgehoben.

Tabelle 23: Verknüpfung und Zwischenergebnisse der Teilprojekte

ADR-Projekt	Theoretische Basis	(Zwischen-) Ergebnisse
Erstes ADR-Projekt (vgl. Unterkapitel 6.5)	Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen (vgl. Unterkapitel 3.2)	<ul style="list-style-type: none"> - Prototyp für Bank A (implementiertes IT-Artefakt) - Instanzierte Designprinzipien für den Kontext von Bank A - Erste Menge abstrahierter Designprinzipien für alle Repräsentanten von Lösungen der untersuchten Problemklasse
Zweites ADR-Projekt (vgl. Unterkapitel 6.6)	Erste Menge abstrahierter Designprinzipien für alle Repräsentanten von Lösungen der untersuchten Problemklasse	<ul style="list-style-type: none"> - Prototyp für Bank B (implementiertes IT-Artefakt) - Instanzierte Designprinzipien für den Kontext von Bank B - Überarbeitete Menge abstrahierter Designprinzipien für alle Repräsentanten von Lösungen der untersuchten Problemklasse
Erweiterung des ersten ADR-Projektes (vgl. Unterkapitel 6.7)	Überarbeitete Menge abstrahierter Designprinzipien für alle Repräsentanten von Lösungen der untersuchten Problemklasse	<ul style="list-style-type: none"> - Überarbeiteter Prototyp für Bank A (implementiertes IT-Artefakt) - Überarbeitete instanzierte Designprinzipien für den Kontext von Bank A - Finale Menge abstrahierter Designprinzipien für alle Repräsentanten von Lösungen der untersuchten Problemklasse

In den folgenden drei Unterkapiteln werden die drei Teilprojekte – die einzelnen ADR-Projekte – vorgestellt. Da das erste Projekt mit Bank A den wesentlichen Grundstein in der Entwicklung der Ergebnisse darstellt, wird dieses in der vorliegenden Arbeit besonders ausführlich beschrieben. Bei der Darstellung der Überarbeitung der Designprinzipien im Rahmen des zweiten ADR-Projektes mit Bank B werden primär die zusätzlich emergierten Designprinzipien dargestellt und diskutiert.

6.5 Erstes ADR-Projekt mit Bank A⁷⁴

In diesem Unterkapitel wird das erste ADR-Projekt in Kooperation mit Bank A präsentiert. Im folgenden Abschnitt 6.5.1 wird zunächst das Vorgehen bei der Durchführung des ADR-Projektes vorgestellt. Im Anschluss daran werden in Abschnitt 6.5.2 die Ergebnisse samt der Genese dieser, von den initialen Designprinzipien hin zu den instanziierten Designprinzipien des IT-Artefaktes, dargestellt. In Unterabschnitt 6.5.2.4 werden schließlich die auf den instanziierten Designprinzipien basierenden abstrakten Designprinzipien abgeleitet.

6.5.1 Vorgehen

Durch die angesprochene Kooperation mit Bank A – speziell mit den IT-Architekten der Bank – wurde das Artefakt „IT-COM“, als ein Informationssystem für die Planung und Kontrolle komplexer IT-Architekturen, entwickelt und implementiert. Dieses Artefakt bildet die Basis der in diesem Unterkapitel dargestellten Theoriebildung.

Das ADR-Team war in drei disjunkte Gruppen strukturiert:⁷⁵ die ADR-Forscher, die IT-Architekten sowie das Software-Entwicklungsteam. Die IT-Architekten zeigten eine hohe Motivation bezüglich der wissenschaftlichen Begleitung bei der Entwicklung der Lösung. Überdies maßen sie dem im Zuge der Anwendung der ADR-Methode erarbeiteten theoretischen Beitrag in Form der Designprinzipien einen hohen Stellenwert bei. Infolgedessen hatten die ADR-Forscher die Möglichkeit, an allen Phasen des Entwicklungsprozesses – von der Planung bis hin zur Übergabe des finalen IT-Artefaktes in Form der Anwendung – teilzuhaben. Zudem hatte das Forscher-Team Zugang zur vollständigen Projektdokumentation und nahm an allen Meetings zwischen den IT-Architekten und dem Software-Entwicklungsteam teil. Hierdurch hatte das Forscher-Team Gelegenheit, alle Änderungswünsche während der Entwicklung des IT-Artefaktes und die Beweggründe für die Änderungen aufzunehmen und zu hinterfragen.

Für die Durchführung dieses ADR-Projektes wurden die folgenden Problemklassen formuliert (vgl. hierzu Unterkapitel 6.3):

- (a) „*Planung und Kontrolle von Komplexität in IT-Architekturen*“ als die untersuchte Problemklasse und
- (b) „*Planung und Kontrolle der Komplexität der IT-Architektur der Bank A*“ als eine Instanz der Problemklasse (a) im Kontext der Bank.

Basierend auf den in Tabelle 21 dargestellten Phasen und Forschungsschritten eines ADR-Projektes werden in Tabelle 24 und Tabelle 25 die Ausprägungen für jeden dieser Schritte für das spezifische ADR-Projekt mit Bank A angeführt (vgl. hierzu Schütz et al. 2013a).

⁷⁴ Vgl. nachfolgend Schütz et al. (2013a).

⁷⁵ Vgl. hierzu „Principle 4: Mutually Influential Roles“ nach Sein et al. (2011), S. 43.

Tabelle 24: Forschungsschritte der ersten und zweiten Phase der ADR-Methode

Forschungsschritt (S)	Form der Umsetzung im ersten ADR-Projekt
(P1.S1) Identifiziere und konzeptualisiere den genauen Forschungsbereich	Der initiale Auslöser der in diesem Kapitel vorgestellten Forschung bestand im Bedürfnis der IT-Architekten der Bank, die Komplexität der IT-Architektur zu messen und zu managen, um die Grundlage für die IT-Architektur beeinflussende Entscheidungen zu verbessern.
(P1.S2) Formuliere die initialen Forschungsfragen	Unter Bezugnahme auf die o. a. Problemklasse wurde die folgende Forschungsfrage formuliert: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?
(P1.S3) Stelle das Problem als eine Instanz einer Klasse von Problemen dar	Die in diesem Forschungsprojekt behandelte Problemklasse lautet: „ <i>Planung und Kontrolle von Komplexität in IT-Architekturen</i> “ (vgl. Unterkapitel 6.3).
(P1.S4) Identifiziere die theoretische Basis und bereits bestehende geeignete technologische Lösungen	Das von Schneberger & McLean (2003) eingeführte theoretische Modell von Komplexität, das in Schütz et al. (2013b) aus einer systemtheoretischen Perspektive auf den Kontext der IT-Architekturen übertragen wurde (vgl. Unterkapitel 3.2) dient als theoretische Basis (vgl. Unterabschnitt 6.5.2.1).
(P1.S5) Stelle ein langfristiges Engagement des Unternehmens sicher	Die Forscher hatten Gelegenheit, an allen Phasen des Entwicklungsprozesses teilzuhaben. Die Zusammenarbeit mit Bank A ging über den Zeitraum des ersten ADR-Projektes hinaus.
(P1.S6) Definiere Rollen und Verantwortlichkeiten	Das ADR-Team war in drei disjunkte Gruppen strukturiert: Forscher, IT-Architekten sowie Software-Entwicklungsteam. Hierdurch wurden theoretische, praktische und technische Perspektiven berücksichtigt.
(P2.S1) Stelle das initiale Ziel für die Wissensbildung heraus	Ziel des Forschungsprojektes ist die Ergänzung der Wissensbasis um Designprinzipien für ein Informationssystem für die „ <i>Planung und Kontrolle von Komplexität in IT-Architekturen</i> “.
(P2.S2) Wähle die BIE-Form und passe diese ggf. an	Da das erwartete Ergebnis des Projektes nicht im Detail spezifiziert war, fand während der Entwicklung des IT-Artefaktes eine fortwährende Evaluation im Zuge der „IT-Dominant“ BIE-Zyklen statt (vgl. Abbildung 46). Hierzu wurde IT-COM zunächst innerhalb des ADR-Teams und anschließend durch eine Auswahl von Endnutzern in der Bank evaluiert.
(P2.S3) Führe die BIE-Zyklen durch	Tabelle 26 gibt einen Überblick über den Designprozess sowie die vier durchgeführten BIE-Zyklen.
(P2.S4) Beurteile das Erfordernis weiterer Zyklen, wiederhole	Mit Scrum wurde im ADR-Projekt eine agile Methode der Softwareentwicklung verwendet. Diese erlaubte eine regelmäßige Evaluation des IT-Artefaktes sowie die Ergänzung weiterer Entwicklungszyklen.

Tabelle 25: Forschungsschritte der dritten und vierten Phase der ADR-Methode

Forschungsschritt (S)	Form der Umsetzung im ersten ADR-Projekt
(P3.S1) Überdenke das Design und führe Designveränderungen während des Projektes durch	Die immer wiederkehrende organisationale Intervention und Evaluation führte sowohl zu einer zyklischen Überprüfung, Überarbeitung und Einführung von Designprinzipien als auch zur Umsetzung neuer Anforderungen an das entstehende IT-Artefakt (vgl. Tabelle 26 und Abschnitt 6.5.2).
(P3.S2) Überprüfe die Einhaltung der Prinzipien	Eine Überprüfung der Einhaltung der Prinzipien fand während der BIE-Zyklen statt (vgl. Abschnitt 6.5.2). Im Zuge der o. a. wiederkehrenden Intervention sowie der Evaluation der emergierenden und anschließend weiterentwickelten Designprinzipien wurde deren Angemessenheit bzgl. der betrachteten Problemklasse verifiziert.
(P3.S3) Analysiere die Ergebnisse der Intervention gemäß der gesetzten Ziele	Während der o. a. wiederkehrenden Intervention und Evaluation wurden neu gewonnene Informationen und Erkenntnisse mit den Zielen und Anforderungen der Problemklasse abgeglichen.
(P4.S1) Formalisiere das Gelernte für eine Klasse von Problemen	Die Ergebnisse wurden durch die Ableitung abstrakter Designprinzipien, die auf andere Instanzen dieser Problemklasse anwendbar sind, generalisiert (vgl. Unterabschnitt 6.5.2.4).
(P4.S2) Teile die Ergebnisse und führe eine Überprüfung dieser mit den Praktikern durch	Die abstrakten Designprinzipien (vgl. Tabelle 29) wurden mit den IT-Architekten diskutiert. Frühe Entwicklungsstufen des Artefaktes IT-COM wurden während der BIE-Zyklen evaluiert und eine Betaversion einer Reihe von Endanwendern vorgestellt und mit diesen diskutiert.
(P4.S3) Formuliere die Ergebnisse in Form von Designprinzipien	Es wurden sieben Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen formuliert. Diese beschreiben das Artefakt IT-COM – implementiert für die Bank A – als eine Instanz einer Lösung für die angeführte Problemklasse (vgl. Tabelle 28 und Tabelle 29).
(P4.S4) Artikuliere das Gelernte im Hinblick auf die gewählten Theorien	Das Gelernte wird im Hinblick auf die verwendeten Theorien in Abschnitt 6.8.1 diskutiert.
(P4.S5) Formalisiere die Ergebnisse für eine Verbreitung	Die Ergebnisse des ersten ADR-Projektes mit der Bank A wurden in Schütz et al. (2013a) publiziert und in der IS-Community auf der ICIS 2013 vorgestellt und diskutiert.

Die Implementierung des IT-Artefaktes in diesem Forschungsprojekt begann „auf der grünen Wiese“. Neben den bankinternen Vorgaben in Form eines Standardkataloges (vgl. Unterabschnitt 2.2.5.3) konnte das Entwicklungsteam ohne weitere Restriktionen verfahren. Betont wurde lediglich das Ziel eines innovativen technischen Designs, das die Problemklasse möglichst vollständig erfasst. Das ADR-Team entschied sich, hierfür eine sogenannte „IT-dominant BIE“ heranzuziehen. Sein et al. (2011) diskutieren mit „IT-dominant BIE“ und „or-

„organization-dominant BIE“ zwei mögliche Ausprägungen der zweiten Phase der ADR-Methode, die sich hinsichtlich des vorrangigen Ursprungs gewonnener Erkenntnisse unterscheiden. Wesentlicher Unterschied bezüglich der Durchführung der ADR-Methode liegt an dieser Stelle darin, dass die Intervention und Evaluation eines relativ frühen Entwicklungsstandes zunächst innerhalb eines begrenzten organisationalen Kontextes durchgeführt wird. Einem größeren Nutzerumfeld wird dabei erst eine ausgereifere Version des IT-Artefaktes zur Verfügung gestellt (vgl. Abbildung 46).

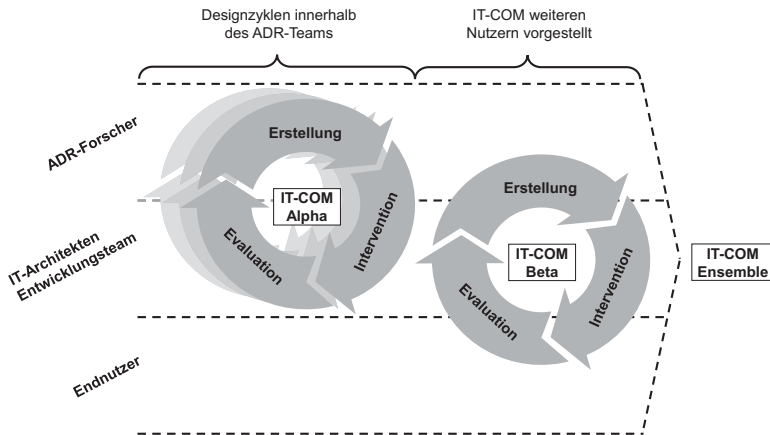


Abbildung 46: Verwendete „IT-Dominant BIE“

Darüber hinaus wurde für die Softwareentwicklung die Methode Scrum (Schwaber & Beedle 2002) verwendet, um bereits in den frühen Phasen des Softwareentwicklungsprozesses eine kontinuierliche Intervention und Evaluation zu ermöglichen. Aufgrund der Tatsache, dass die Ausgestaltung des zu erarbeitenden Informationssystems zu Beginn des Projektes nicht vollständig spezifiziert war, erwies sich die Verwendung von Scrum als äußerst vorteilhaft.

Der Entwicklungsprozess verlief über einen Zeitraum von insgesamt 15 Wochen, unterteilt in sieben zweiwöchige sogenannte „Sprints“. Ein Sprint kann dabei als ein Arbeitszyklus in Scrum verstanden werden. Dieser stellt eine Art „Mini-Projekt“ dar, in dem eine vereinbarte Menge von Anforderungen zu implementieren ist (Schwaber & Beedle 2002). Am Ende jedes Sprints fand ein Meeting zur Evaluation des aktuellen Standes des IT-Artefaktes statt. Das gesamte ADR-Team nahm an diesen Meetings teil. Zusammen mit Planungsmeeting, Kick-Off und der finalen Vorstellung der Ergebnisse wurden insgesamt zehn Meetings abgehalten, jeweils über eine Dauer von ca. 90 Minuten. Zusätzlich zu diesen Meetings wurden vier Workshops (ca. 120 Minuten) und zwei Interviews (ca. 60 Minuten) mit den IT-Architekten durchgeführt, um den bislang hervorgegangenen Stand der Implementierung zu evaluieren und die Angemessenheit der vorgeschlagenen Designprinzipien zu diskutieren.

Das ADR-Projekt wurde in vier BIE-Zyklen unterteilt. Tabelle 26 gibt einen Überblick über den Designprozess und zeigt auf, wie die sieben Sprints den BIE-Zyklen zuzuordnen sind und welche Gruppen an den jeweiligen BIE-Zyklen beteiligt waren.

Tabelle 26: Überblick über die durchgeführten BIE-Zyklen des ersten ADR-Projektes⁷⁶

Beteiligte	BIE-Zyklus 1 Vorschlag der initialen Designprinzipien	BIE-Zyklus 2 Anforderungsanalyse und Mockup-Design (Sprint 1-3)	BIE-Zyklus 3 Komplexitätsmaße und Datenqualität (Sprint 4)	BIE-Zyklus 4 Zeitreihenanalyse und Drill-Down (Sprint 5-7)
ADR-Forscher	X	X	X	X
IT-Architekten	X	X	X	X
Entwicklungsteam		X	X	X
Endnutzer				X

Im ersten BIE-Zyklus wurden die initialen Designprinzipien sowohl aus praktischen als auch theoretischen Überlegungen heraus abgeleitet (vgl. Unterabschnitt 6.5.2.1). Da sich das Entwicklungsteam zu Beginn des Projektes vornehmlich mit technischen Problemstellungen sowie dem Aufsetzen der Entwicklungsumgebung befasste, wurden die ersten drei Sprints gemeinsam in dem zweiten BIE-Zyklus verortet. Hierdurch hatte das Entwicklungsteam ausreichend Zeit, sich vor der Intervention und Evaluation auch den ersten fachlichen Fragestellungen, d. h. der Analyse der Anforderungen und dem Design erster Mockups, zu widmen. Ziel des dritten BIE-Zyklus (Sprint 4) war die Implementierung und Evaluation der Komplexitätsmaße sowie die Untersuchung des Einflusses der Datenqualität auf die Maßzahlen und auf IT-COM als Ganzes. Während der Sprints 5 bis 7, die dem vierten BIE-Zyklus zuzuordnen sind, entstanden die Funktionen der Zeitreihenanalyse (um die Veränderung der Komplexität im Zeitverlauf beobachten und vergleichen zu können) sowie die des Drill-Downs (für eine Analyse auf unterschiedlichen Abstraktionsleveln).

Während der Entwicklung der Alphaversion des IT-Artefaktes (BIE-Zyklen 1 bis 3) fand eine Evaluation des Designs innerhalb des ADR-Teams statt. Dies waren speziell die an dem Projekt beteiligten IT-Architekten sowie die ADR-Forscher. Hierdurch wurden stets sowohl eine praktische als auch eine theoretische Perspektive auf das entstehende IT-Artefakt sichergestellt. In Übereinstimmung mit dem von Sein et al. (2011) vorgeschlagenen Vorgehen lag der Fokus der IT-Architekten dabei auf der Intervention. Weiterhin hinterfragten die IT-Architekten stets kritisch die während des Entwicklungsprozesses emergierenden Designprinzipien im Hinblick auf die betrachtete Probleminstanz und den organisationalen Kontext der Bank. In BIE-Zyklus 4 stellten die IT-Architekten das IT-Artefakt schließlich einer größeren Gruppe potenzieller Anwender vor.⁷⁷ Das hierbei erhaltene Feedback der Anwender wurde anschließend im Rahmen eines gemeinsamen Workshops diskutiert.

⁷⁶ Vgl. hierzu auch Schütz et al. (2013a).

⁷⁷ Vgl. hierzu „Principle 5: Authentic and Concurrent Evaluation“ nach Sein et al. (2011), S. 43.

6.5.2 Ergebnisse des ersten ADR-Projektes

In den folgenden Unterabschnitten werden die Ergebnisse des ersten ADR-Projektes vorgestellt. Auf Basis der in Unterkapitel 6.3 eingeführten Problemklasse werden in Unterabschnitt 6.5.2.1 zunächst die initialen, auf theoretischen Überlegungen basierenden Designprinzipien eingeführt. Im Anschluss wird in Unterabschnitt 6.5.2.2 der Prozess der iterativen Überprüfung und Überarbeitung existierender sowie der Emergenz weiterer Designprinzipien dargestellt. Die aus dem ersten ADR-Projekt resultierende Menge instanziiert Designprinzipien wird in Unterabschnitt 6.5.2.3 präsentiert.

6.5.2.1 Initiale Designprinzipien des ersten ADR-Projektes

Der ADR-Ansatz sieht vor, dass das entwickelte Artefakt auf theoretischen Überlegungen beruht.⁷⁸ Hierzu wurde die in Schütz et al. (2013b) vorgeschlagene Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen (vgl. Unterkapitel 3.2) herangezogen und um den von Schneberger & McLean (2003) diskutierten Aspekt der Veränderungsrate des Systems ergänzt.

Die im Rahmen dieses Forschungsprojektes entwickelten Artefakte sollen die Architekten der Banken bei dem Management der Komplexität der IT-Architektur unterstützen. Demzufolge liegt der Fokus der Untersuchung mit der *IT-Architektur* explizit nur auf einem Subsystem der Unternehmensarchitektur. Die Entwicklung von Artefakten zur Unterstützung bei der Planung und Kontrolle der Komplexität der Geschäftsprozessarchitektur ist nicht Gegenstand dieser Untersuchung. Da die IT-Architektur und die Geschäftsprozessarchitektur eng miteinander verflochten sind, können diese jedoch nicht völlig losgelöst voneinander betrachtet werden (vgl. Abbildung 6). Aus diesem Grund flossen auch Informationen über die Ausgestaltung der Geschäftsprozessarchitektur mit in die IT-Artefakte ein. Um nicht zu suggerieren, dass die entwickelten Artefakte ebenfalls ein Komplexitätsmanagement der Geschäftsprozessarchitektur umfassen, wird im Folgenden eine auf das Subsystem der IT-Architektur beschränkte Konzeptualisierung von Komplexität verwendet (vgl. Abbildung 47).

Entsprechend wird im Folgenden die Konzeptualisierung von Komplexität in IT-Architekturen mit der Veränderungsrate der Architektur (vgl. hierzu Schneberger & McLean 2003) kombiniert. Die initialen Designprinzipien basierten auf Erkenntnissen aus der Literatur sowie aus Diskussionen im Zuge gemeinsamer Workshops mit den IT-Architekten zur Vorbereitung der ersten Entwicklungsphase. Durch dieses Vorgehen wurde sichergestellt, dass bereits die initialen Designprinzipien (DP) sowohl theoretische als auch praktische Perspektiven beinhalten (vgl. Tabelle 27). Dabei soll angemerkt werden, dass DP 1 bis DP 4 auf der Annahme beruhen, dass ein Management der Komplexität der IT-Architektur die Messbarkeit der Komplexität der IT-Architektur voraussetzt (vgl. hierzu Kapitel 5).

⁷⁸ Vgl. hierzu „Principle 2: Theory-Ingrained Artifact“ nach Sein et al. (2011), S. 40, 41.

Tabelle 27: Initiale abstrakte Designprinzipien⁷⁹

Designprinzip	Beschreibung: Für eine Planung und Kontrolle von Komplexität...	Literatur
DP 1: Anzahl und Heterogenität der Komponenten	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Komponenten des Systems bereitstellen.	Henningsson & Hanseth (2011), Klir (2001), Schneberger & McLean (2003), Schütz et al. (2013b), Simon (1962), Widjaja et al. (2012)
DP 2: Anzahl und Heterogenität der Beziehungen	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Beziehungen des Systems bereitstellen.	Henningsson & Hanseth (2011), Klir (2001), Schneberger & McLean (2003), Schütz et al. (2013b), Simon (1962), Widjaja et al. (2012)
DP 3: Veränderungsrate	...sollte das IT-Artefakt die Veränderungsrate des Systems berücksichtigen.	Schneberger & McLean (2003), Simon (1962), Schütz et al. (2013b)
DP 4: Ebenen der Architektur	...sollte das IT-Artefakt die Architekturebenen Datenarchitektur, Anwendungsarchitektur, Infrastrukturarchitektur und Geschäftsprozessarchitektur berücksichtigen.	Schütz et al. (2013b), Richardson et al. (1990), Ross (2003), Spewak & Hill (1993), The Open Group (2011)

Abbildung 47 zeigt, wie sich die initialen Designprinzipien anhand der zugrunde liegenden Konzeptualisierung zuordnen lassen.

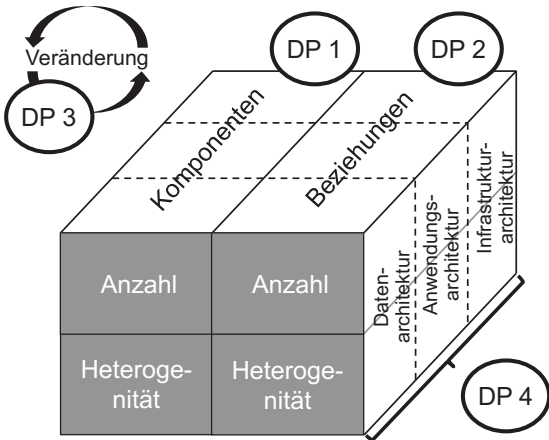


Abbildung 47: Konzeptualisierung von Komplexität in IT-Architekturen⁸⁰

Frühe gemeinsame Workshops mit den IT-Architekten zeigten, dass die verwendete Konzeptualisierung von Komplexität im Einklang mit dem Verständnis der IT-Architekten der Bank stand. Das folgende Zitat des Principal Architects der Bank aus einem der Workshops verdeutlicht dies:

⁷⁹ Vgl. hierzu auch Schütz et al. (2013a).
⁸⁰ Vgl. hierzu auch Schütz et al. (2013b) und Schneberger & McLean (2003).

„Wir mussten uns natürlich erst überlegen, was IT-Komplexität überhaupt ist, welche Eigenschaften von Komplexität wir uns ansehen wollen und wo wir die Objekte im Unternehmen finden, an denen wir die Komplexität festmachen können. [...] Und hier fokussieren wir auf die strukturelle Komplexität eines Systems oder eines Clusters (also mehrere Anwendungen einer Fachdomäne), die aus der Anzahl und der Diversität der Komponenten und aus der Anzahl und Diversität der Beziehungen zwischen den Komponenten resultiert. [...] Der nächste Schritt war dann, zu überlegen, was wollen wir uns denn jetzt tatsächlich anschauen? Ja gut, wir sind Architekten, also betrachten wir uns die ‚Architectural Layers‘, also Daten, Anwendungen und Infrastruktur.“ (Principal Architect Bank A)

6.5.2.2 Iterative Überprüfung und Emergenz von Designprinzipien

In diesem Unterabschnitt werden die Ergebnisse vorgestellt, die – basierend auf den eingeführten initialen Designprinzipien als ein sogenanntes „theory-ingrained artifact“ (Sein et al. 2011) – aus der Durchführung der BIE-Zyklen zwei bis vier resultieren. Während der Implementierung und der parallel stattfindenden Evaluation der jeweiligen Zwischenstände des IT-Artefaktes und der Designprinzipien, konnten drei der vier initial vorgeschlagenen Designprinzipien bestätigt werden (DP 1, 2 und 4). Das Designprinzip *Veränderungsrate* (DP 3) wurde hingegen überarbeitet. Darüber hinaus emergierten im Zuge der BIE-Zyklen drei weitere Designprinzipien.

Um die im Folgenden vorgestellten Ergebnisse und unterstützenden Abbildungen im Kontext der Bank A besser nachvollziehen zu können, wird die grundlegende Organisation der IT-Landschaft zunächst kurz vorgestellt. Diese ist anhand eines Domänen-Modells strukturiert, das hierarchisch in drei Ebenen aufgebaut ist. Die Domänen sind dabei in Subdomänen unterteilt, die wiederum aus einer Menge funktionaler Einheiten bestehen. Jede Anwendung in der IT-Landschaft ist genau einer dieser funktionalen Einheiten zugeordnet. Abbildung 48 zeigt beispielhaft die Darstellung einer Reihe von Komplexitätskennziffern der IT-Architektur der Bank und hebt hervor, an welchen Stellen sich die einzelnen Designprinzipien in dem IT-Artefakt wiederfinden. Die in Abbildung 48 bereits dargestellten, bislang jedoch noch nicht eingeführten Designprinzipien 5 bis 7 stellen die während der durchgeführten BIE-Zyklen emergierten Designprinzipien dar, auf die in den folgenden Sektionen näher eingegangen wird. Die Beschriftung der linken Spalte in Abbildung 48 zeigt dabei die Domänenstruktur auf. Beispielhaft wurde die Domäne „Data Warehouse“ erweitert.⁸¹ Diese besteht aus genau einer Subdomäne und diese wiederum aus drei funktionalen Einheiten. Die in den Spalten „Quantity“, „Diversity“ und „Interdependency“ dargestellten Werte sind die berechneten Komplexitätsmaße.

⁸¹ Die weiteren Domänen sind in der Abbildung unkenntlich gemacht.

Homepage | Contact | Imprint | Administration | Logout: admin

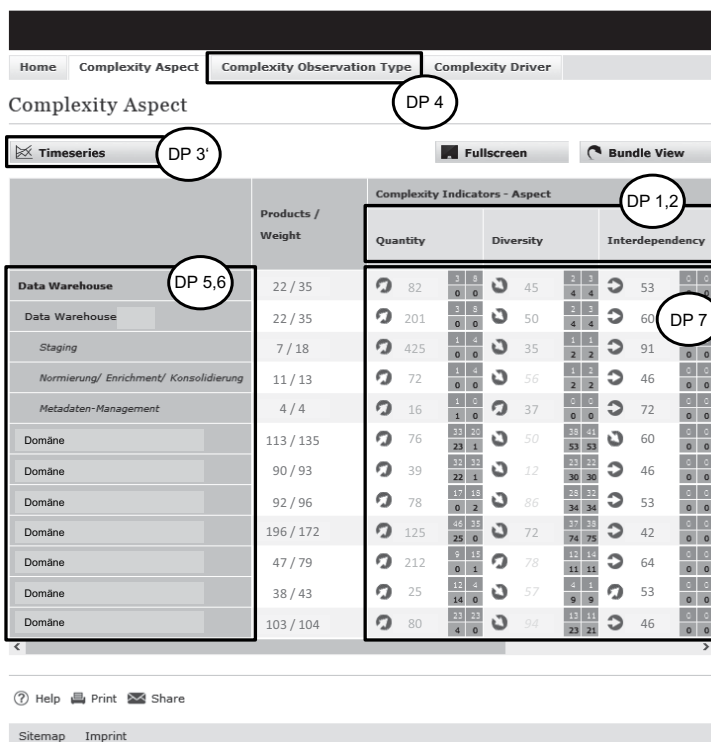


Abbildung 48: Verortung der Designprinzipien in IT-COM⁸²

Die in IT-COM implementierten Komplexitätsmaße der Kategorie „Quantity“ sind dabei anzahlorientiert, die der Kategorie „Diversity“ heterogenitätsbasiert und die der „Interdependency“ beziehen sich auf die Beziehungen zwischen den betrachteten Anwendungen. Unter dem Reiter „Complexity Observation Type“ am oberen Rand der Abbildung befinden sich die Ebenen der IT-Architektur (vgl. Abschnitt 2.2.4). Die in Abbildung 48 angeführten Begriffe sind im Sprachgebrauch der Bank etabliert und weichen daher teilweise von den in der vorliegenden Arbeit verwendeten Begriffen leicht ab. Die „Complexity Aspects“ entsprechen den initialen Designprinzipien *Anzahl und Heterogenität der Komponenten* (DP 1) und *Anzahl und Heterogenität der Beziehungen* (DP 2) und der „Complexity Observation Type“ dem Designprinzip *Ebenen der IT-Architektur* (DP 4).

⁸² Vgl. hierzu auch Schütz et al. (2013a). Die in der Abbildung angegebenen Werte der Kennzahlen wurden durch fiktive Werte ersetzt.

6.5.2.2.1 Überarbeitung des Designprinzips „Veränderungsrate“

Wie in Unterabschnitt 6.5.2.1 eingeführt, stellte die in Unterkapitel 3.2 vorgeschlagene systemtheoretische Konzeptualisierung von Komplexität in IT-Architekturen die theoretische Basis für die Entwicklung des IT-Artefaktes dar. Während sich DP 1, 2 und 4 als adäquat erwiesen, führte das Feedback der IT-Architekten im Kontext des vierten BIE-Zyklus zu einer Überarbeitung des Designprinzips *Veränderungsrate* (DP 3). Dabei betonten die IT-Architekten die Relevanz einer Einbeziehung der Faktoren der „Veränderung“ und der „Zeit“ in das Komplexitätsmanagement der IT-Architektur:

„Veränderung ist auf jeden Fall eine relevante Dimension in der Komplexitätsbewertung. Wir reden von einem Problem, das heißt Wachstum. Wachsen einer Anwendungslandschaft ist einhergehend mit dem Wachstum von Komplexität. [...] [Wir benötigen die Möglichkeit,] zu sehen, in welchen Bereichen sie wächst und in welchen eher nicht. Auch um zu verstehen, welche Business-Treiber das Wachstum verursachen und um bewerten zu können, ob die Veränderung der Komplexität angemessen ist oder nicht.“ (Principal Architect Bank A)

Das hier diskutierte Designprinzip DP 3 basierte auf der Idee von Schneberger & McLean (2003), die Veränderungsrate der Architektur (aus dem engl. „rate of change“) zu berücksichtigen. Eine Änderungsrate der Architektur bezeichnet dabei einen gewissen Umfang von Änderungen in einem bestimmten Zeitabschnitt. Diese Betrachtung erschien den IT-Architekten als nicht ausreichend und nur schwer operationalisierbar.

Eine Schwierigkeit stellte die Tatsache dar, dass bei einer Veränderungsrate für jede Änderung hätte entschieden werden müssen, inwieweit diese einen Einfluss auf die strukturelle Komplexität der IT-Architektur hat:

„Du kannst 10 Releaseprojekte an einem Anwendungssystem durchführen, ohne dass diese sich – alleine durch die Tatsache, dass du eine Änderung vorgenommen hast – auch nur ansatzweise auf die Komplexität auswirken. Du hast z. B. eine weitere Eingabemaske aufgenommen, ein Feld hinzugenommen oder ein Feld ausgebaut. Das sind alles Änderungen, aber die müssen sich noch lange nicht auf die Komplexität auswirken. Nimmst du jedoch weitere Schnittstellen oder weitere Systeme auf, dann kann sich das sehr wohl auf die Komplexität auswirken. Also die Tatsache, dass an einem System Änderungen vorgenommen werden, ist für sich genommen nicht ausreichend.“ (Principal Architect Bank A)

Um dieses Problem zu berücksichtigen und das Designprinzip weiter zu schärfen, sollte der Fokus auf die Veränderung der Komplexitätskennzahlen im Zeitverlauf gelegt werden:

„Ich glaube, die Änderungsrate der Architektur ist nur aussagefähig, wenn man sie mit anderen Werten, also Komplexitätsindikatoren, in Beziehung setzt. [...] Die Änderungsrate der Architektur alleine ist meiner Meinung nach kein relevanter Wert. Eher die Veränderung der Indikatoren über die Zeit hinweg, also eine Zeitreihenbetrachtung.“ (Principal Architect Bank A)

Hierdurch konnten sowohl eine Identifikation der Menge durchgeführter Änderungen als auch eine aufwändige Überprüfung einzelner Änderungen, hinsichtlich deren Wirkung auf die strukturelle Komplexität der Architektur, vermieden werden:

„Wir sollten uns letztendlich auf die historische Betrachtung fokussieren. Wo kommst du her, wo stehst du heute, wo möchtest du hin? [...] Gut, wir wollen [die strukturelle Komplexität der IT-Architektur] ja quartalsweise messen, also müssen wir dann auch die Quartale gegeneinander abgleichen. Und im Endeffekt diese auf einem Zeitstrahl darstellen, um die Gesamtentwicklung der Komplexität der Anwendungslandschaft abzubilden.“ (Principal Architect Bank A)

Die hierbei gewählte „historische“ Perspektive auf die Veränderung der Komplexität im Zeitverlauf kann als „Verlauf einer Kurve“ verstanden werden (im Gegensatz zu der initialen Perspektive auf die „Steigung einer Kurve“). Im Zuge der Evaluation während des vierten BIE-Zyklus wurde der Fokus daher auf die Nachvollziehbarkeit der strukturellen Komplexität der IT-Architektur in Form von Zeitreihen gelegt und das Designprinzip entsprechend angepasst (vgl. DP 3' in Tabelle 28 und Tabelle 29). Die Umsetzung des angepassten Designprinzips *Nachvollziehbarkeit von Veränderung im Zeitverlauf* (DP 3')⁸³ im IT-Artefakt IT-COM ist Abbildung 49 zu entnehmen. In dieser Abbildung ist beispielhaft die Veränderung der Schnittstellenkomplexität der Anwendungen vier beliebig gewählter Domänen dargestellt.

Time Series Analysis

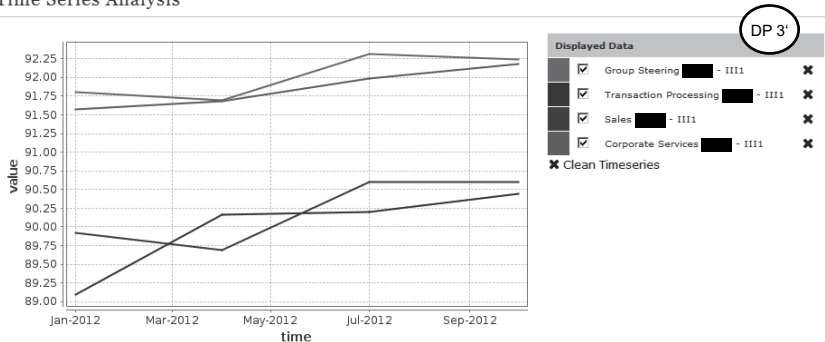


Abbildung 49: Zeitreihenanalyse in IT-COM⁸⁴

6.5.2.2.2 Emergenz des Designprinzips „Datenqualität“

Bereits von Beginn der Entwicklung an (ab BIE-Zyklus 2) beschäftigte sich das Entwicklungsteam intensiv mit dem Thema „Quelldaten“. Die unterschiedlichen Quellsysteme (u. a. SAP oder das ARIS-Repository) führten zu einer hohen Heterogenität der zu berücksichtigenden Datenformate. Ein weiterer Aspekt war die Datenqualität. Im Rahmen der Überprüfung der für die Implementierung zur Verfügung gestellten Datensätze zeigte sich, dass die

⁸³ Das instanziierte DP 3' lautet: „IT-COM ermöglicht die Nachvollziehbarkeit der Veränderung der berechneten Komplexitätskennzahlen über die Zeit hinweg anhand von Zeitreihen.“

⁸⁴ Vgl. hierzu auch Schütz et al. (2013a). Bei den der Abbildung zugrunde liegenden Daten handelt es sich um Testdaten.

Qualität der Daten (hinsichtlich der Vollständigkeit der Datensätze) zwischen den einzelnen Liefersystemen stark variierte. Dies fiel auch den IT-Architekten primär während der Überprüfung der implementierten Kennzahlen zur Ableitung erster Handlungsempfehlungen in BIE-Zyklus 3 auf. Da alle Kennzahlen zur Quantifizierung der strukturellen Komplexität auf den eingelesenen Architekturdaten basierten, musste der Aspekt der Datenqualität in der weiteren Entwicklung des IT-Artefaktes durch das ADR-Team berücksichtigt werden.

Aus den Diskussionen mit den IT-Architekten in BIE-Zyklus 3 resultierte, dass ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen den Nutzer auf die Vollständigkeit und Verlässlichkeit der Daten, die den berechneten Kennzahlen zugrunde liegen, aufmerksam machen sollte. Da die Kennzahlen eine Entscheidungsgrundlage für das Management von IT-Architekturen darstellen sollen, wurde dieser Punkt besonders von den IT-Architekten hervorgehoben:

„Wir bekommen die Daten aus verschiedenen Quellen. [...] Und da haben wir in einigen Fällen festgestellt, mit der Qualität ist das so eine Sache. [...] Wir wollen anhand der Komplexitätskennzahlen Entscheidungen treffen. Wenn man Entscheidungen treffen will, sollte man dies auf einer möglichst verlässlichen Datengrundlage tun. Und die ist natürlich verlässlicher, je besser der Pflegestand ist. Deswegen weisen wir auf dem Dashboard darauf hin. Nach dem Motto, wenn du 90 % Erfassungsstand hast, dann hast du [bei hoher Komplexität] mit Sicherheit ein Problem. Hast du schlechte Werte [bzgl. der Komplexität], aber nur 30 % Erfassungsstand, dann könntest du ein Problem haben, aber vielleicht auch nicht, wenn du die anderen 70 % auch berücksichtigen könntest. Um den Nutzer des Tools auf den Erfassungsstand der zugrunde liegenden Daten aufmerksam zu machen, erhält er in IT-COM also ein Signal, wie verlässlich die Kennzahl dahingehend ist.“ (Principal Architect Bank A)

Dies gab Anlass dazu, das (instanziierte) Designprinzip *Datenqualität* (DP 7) einzuführen: „IT-COM macht den Nutzer visuell auf die Qualität der Daten, auf denen die dargestellten Kennzahlen beruhen, aufmerksam.“ Um auf Kennzahlen mit geringer Datenqualität hinzuweisen, wurde eine visuelle Lösung implementiert, durch die dem Nutzer der Grad der jeweiligen Datenqualität einer Kennzahl kenntlich gemacht wird. Kennzahlen mit einer hohen Datenqualität werden in dunklen Zeichen, Kennzahlen mit einer mittleren Datenqualität in deutlich helleren Zeichen und solche mit einer niedrigen Datenqualität zusätzlich kursiv dargestellt (vgl. Abbildung 48). Durch diese Form der Kennzeichnung wird der Nutzer darauf aufmerksam gemacht, welche der Kennzahlen eine verlässliche Basis für eine Entscheidung darstellen und welche er im Zweifel – im Hinblick auf die Datenqualität – nochmals genauer hinterfragen sollte. Der IT-Architekt fasst das im Rahmen eines internen Workshops mit potenziellen Nutzern (in BIE-Zyklus 4) erhaltene Feedback wie folgt zusammen:

„Auch die Teilnehmer der internen Workshops haben betont, dass der Hinweis auf die Verlässlichkeit der einzelnen Werte sehr wichtig ist.“ (Principal Architect Bank A)

Eine Führungskraft im Architekturbereich unterstrich diesen Punkt im Zuge eines Workshops:

„Den Impuls muss man immer mitgeben. Ich muss wissen, wie viel Prozent der Daten ich habe.“ (Abteilungsleiter eines Teilbereiches der Architektur Bank A)

6.5.2.2.3 Emergenz des Designprinzips „Offenheit“

Im Rahmen eines Meetings zur Evaluation des aktuellen Standes des IT-Artefaktes in BIE-Zyklus 4 kam ein weiterer Punkt zur Sprache. Während der Diskussionen zwischen dem Entwicklungsteam und den IT-Architekten, wie der Zugriff auf das IT-Artefakt IT-COM umgesetzt werden sollte, kam die Frage auf, welche Mitarbeiter Zugriff auf IT-COM benötigen. Im Vorfeld des ADR-Projektes war der direkte Zugang zu Informationen bezüglich der Ausgestaltung der IT-Architektur im Wesentlichen dem Kreis der IT-Architekten vorbehalten. Ein Projektmanager, der für ein Projekt zuständig war, bei dem Entscheidungen mit Einfluss auf die IT-Architektur getroffen werden mussten, verfügte in der Regel nicht über einen direkten Zugang zu Architekturinformationen der von dem Projekt betroffenen Anwendungen oder Infrastrukturkomponenten. Durch die Evaluation des IT-Artefaktes wurde deutlich, dass der Zugriff auf IT-COM nicht per se auf einen spezifischen Zuständigkeitskreis beschränkt werden sollte. Vielmehr sollte ein Informationssystem zur Planung und Kontrolle der Komplexität der IT-Architektur allen Mitarbeitern der Bank zugänglich sein, deren Aufgabenbereiche Architekturentscheidungen berühren. Das folgende Zitat eines der IT-Architekten verdeutlicht diesen Punkt:

„IT-COM ist für uns ein ‚IT for IT‘-Tool, das für alle Bereiche verfügbar sein sollte, die mit IT zu tun haben. [...] Wir wollen also per se nicht eine Zielgruppe ausschließen. Die Kennzahlen sollten allen in irgendeiner Weise davon betroffenen Mitarbeitern zugänglich sein.“ (Principal Architect Bank A)

Die zu Anfang des Projektes durch das ADR-Team getroffene Entscheidung, das Informationssystem als eine webbasierte Anwendung zu implementieren, kam dieser Entwicklung zugute. So ermöglichte diese Technologie einen einfachen und schnell umzusetzenden Zugriff auf IT-COM über das Intranet der Bank. Auch nicht unmittelbar mit der IT-Architektur beauftragte potenzielle Endnutzer in der Bank beurteilten den direkten Zugriff auf Architekturinformationen sowie die berechneten Kennzahlen als einen wesentlichen Fortschritt. Einer der IT-Architekten berichtet hierzu von dem Feedback des *Head of Payments* wie folgt:

„Ein Kollege kämpft schon seit längerem dafür, seinen ‚Zoo‘ ein bisschen kleiner zu bekommen – zu standardisieren –, hat dies über die Jahre hinweg aufgrund der Budgetsituation aber nicht geschafft. Und jetzt konnte er anhand von IT-COM sehen, dass in den letzten Jahren die Standard-Compliance sogar noch weiter abgenommen hat. [...] Da hat er gesagt: ‚ja ok, wenn ich jetzt die Information habe, dass es sogar noch weiter angestiegen ist, dann habe ich nun vielleicht ein besseres Argument, hierfür entsprechendes Budget zu bekommen‘.“ (Principal Architect Bank A)

Ein weiterer Aspekt im Kontext von DP 5 sind die sogenannten Architektur-Checks, die in der Bank für jede Änderungsanfrage mit Einfluss auf die Ausgestaltung der IT-Architektur durchzuführen sind. Hierbei hat der Projektverantwortliche zu beurteilen, welchen Einfluss

die geplante Änderung auf u. a. die Komplexität der IT-Architektur hat. Die Projektverantwortlichen standen hier regelmäßig vor dem Problem, die erforderlichen Informationen zu identifizieren und aus den unterschiedlichsten Systemen und bei diversen Ansprechpartnern anzufordern und zu erfassen. Außerdem waren die Auswirkungen einer Änderung auf die unterschiedlichen Teilbereiche der Architektur häufig nur schwer zu ermitteln oder wurden lediglich aus „dem Bauch heraus“ beurteilt. Das im Zuge des ADR-Projektes entwickelte Informationssystem IT-COM erleichtert diese Aufgabe deutlich und stärkt gleichzeitig die Position der Entscheider, da die verwendeten Kennzahlen nicht individuell entworfen werden müssen, sondern aus einem System stammen.

Die Erkenntnisse aus diesen Diskussionen führten zur Einführung des Designprinzips *Offenheit* (DP 5): „Der Zugriff auf IT-COM soll nicht auf eine spezifische Zielgruppe eingeschränkt werden. Alle Mitarbeiter, die Entscheidungen mit Einfluss auf die IT-Architektur treffen, sollen Zugriff auf IT-COM haben.“

6.5.2.2.4 Emergenz des Designprinzips „Abstraktionsgrad und Drill-Down“

Die Diskussionen um DP 5 in BIE-Zyklus 4 brachten zudem ein weiteres Designprinzip hervor, das in einem engen Zusammenhang mit DP 5 steht. Wie bereits angeführt, wurde die Zielgruppe des entstehenden IT-Artefaktes im Verlauf des ADR-Projektes deutlich erweitert. Gleichzeitig veränderten sich dadurch auch die Anforderungen an die durch das Informationssystem bereitgestellten Analysemöglichkeiten. Da nun Mitarbeiter aus den unterschiedlichsten Aufgabenfeldern und mit verschiedenen Perspektiven auf eine IT-Architektur in der Lage sein mussten, mit dem IT-Artefakt umzugehen, wurde ein stärkerer Fokus auf die Flexibilität möglicher Analysen gelegt. So stehen für einen Anwendungsverantwortlichen die Kennzahlen der Anwendungen, für die er zuständig ist, im Mittelpunkt des Interesses. Für einen Projektmanager sind die Anwendungen von besonderem Interesse, die von dem von ihm verantworteten Projekt betroffen sind. Ein CIO hingegen wird in der Regel keine spezifischen Detailinformationen auf Anwendungsebene benötigen. In diesem Fall wäre eine „Aussage“ über die gesamte IT-Architektur oder zumindest ein wesentlich höherer Abstraktionsgrad erforderlich.

Neben der erweiterten, heterogenen Zielgruppe des IT-Artefaktes zeigte sich, dass die Endnutzer eine Möglichkeit benötigen, die Ursachen einer hohen Komplexität ermitteln zu können:

„Auf Ebene der gesamten Anwendungslandschaft ist die aggregierte Kennzahl nichts anderes, als ein Indikator. Du könntest an der Stelle jetzt Handlungsentscheidungen ableiten nach dem Motto: ‚Hier muss ich etwas tun‘. Aber was dieses ‚etwas‘ ist, das wirst du nur herausbekommen, wenn du wirklich ganz nach unten runtergehst. Weil, wenn du z. B. erkennst, dass die Standardkonformität in einer Domäne nicht gut ist, dann musst du untersuchen, an welcher Stelle. Die Domäne umfasst vielleicht 50 Anwendungssysteme, aber welches dieser 50 Systeme jetzt letztendlich nicht standardkonform ist, das kannst du aus der aggregierten Zahl ja nicht ablesen. Deshalb brauchst du den Drill-Down. So kommst du bis zur Produktverantwortung und die müssen dann dort untersuchen, woran es liegt.“ (Principal Architect Bank A)

So könnte es durchaus möglich sein, dass auf einem hohen Aggregationsgrad auf Ebene der Domänen eine hohe Komplexität ausgewiesen wird, jedoch eine Analyse der darunterliegenden Substrukturen (z. B. der Subdomänen) ein sehr homogenes und somit weniger komplexes Bild zeigt. Auf der anderen Seite ist natürlich auch denkbar, dass tatsächlich eine vermeidbare Komplexität vorliegt. Um diese zu identifizieren, benötigt der Nutzer die Möglichkeit, die Ursache herauszuarbeiten:

„Wenn ich nur Kennzahlen auf Ebene der Gesamtbank habe, dann hilft es mir auch nicht weiter. Ich muss es weiter herunterbrechen können. Sonst kann ich nicht sagen, wo ich anfangen muss zu konsolidieren.“ (Abteilungsleiter eines Teilbereiches der Architektur Bank A)

Ein weiteres Einsatzgebiet eines solchen Drill-Downs liegt in der Überprüfung der Wirksamkeit von Maßnahmen. Wird beispielsweise eine Maßnahme durchgeführt, die sich nicht auf die IT-Architektur als Ganzes, sondern auf einen spezifischen Teilbereich bezieht, so benötigt der Nutzer die Möglichkeit, eben diesen auf Veränderungen zu untersuchen.

Auf Grundlage der angeführten Überlegungen wurde Designprinzip *Abstraktionsgrad und Drill-Down* (DP 6) eingeführt: „IT-COM erlaubt den Nutzern, das Abstraktionslevel im Hinblick auf die technische Domänenstruktur frei auszuwählen (Domäne, Subdomäne oder funktionale Einheit). Zudem können die Nutzer mithilfe eines „Drill-Downs“ in tiefere Ebenen die Ursachen von Komplexität auf einer höheren Ebene der Struktur untersuchen.“

6.5.2.3 Instanziierte Designprinzipien des ersten ADR-Projektes

In der folgenden Tabelle 28 werden die instanziierten Designprinzipien für ein Informationssystem, das die IT-Manager und IT-Architekten der Bank bei dem Management der Komplexität von IT-Architekturen unterstützt, zusammengefasst. Sowohl diese instanziierten Designprinzipien als auch das implementierte IT-Artefakt IT-COM stellen sogenannte „material artifacts“ (Gregor & Jones 2007) und somit „Level 1 contributions“ (Gregor & Hevner 2013) aus der Durchführung des ADR-Projektes dar.

Tabelle 28: Instanzierte Designprinzipien des ersten ADR-Projektes⁸⁵

Designprinzip	Instanziierungen der Designprinzipien in der Bank
DP 1: Anzahl und Heterogenität der Komponenten	Das implementierte Informationssystem IT-COM berücksichtigt die Anzahl der Datenbankobjekte, Use Cases (Geschäftsfunktionalität die eine Anwendung unterstützt), Vorgänge in Prozessen, Benutzer, Abteilungen, funktionalen Implementierungen und der Anforderungen an die Infrastruktur sowie die Heterogenität der in der Bank verwendeten Technologien.
DP 2: Anzahl und Heterogenität der Beziehungen	IT-COM berücksichtigt die Anzahl der Schnittstellen und die Heterogenität der Implementierungstypen der Schnittstellen sowie das Verhältnis zwischen den Schnittstellen innerhalb und außerhalb des Systems.
DP 3': Nachvollziehbarkeit von Veränderung im Zeitverlauf	IT-COM ermöglicht die Nachvollziehbarkeit der Veränderung der berechneten Komplexitätskennzahlen über die Zeit hinweg anhand von Zeitreihen.
DP 4: Ebenen der Architektur	IT-COM berücksichtigt, ausgehend von der Anwendungsarchitektur, die Ebenen der Daten- und Infrastrukturarchitektur sowie die funktionale Ebene der Architektur (Business-Perspektive). Durch die Zuordnung der Komponenten-Ausprägungen der Architekturebenen zu den einzelnen Anwendungen werden die Beziehungen zwischen den einzelnen Ebenen modelliert.
DP 5: Offenheit	Der Zugriff auf IT-COM soll nicht auf eine spezifische Zielgruppe eingeschränkt werden. Alle Mitarbeiter, die Entscheidungen mit Einfluss auf die IT-Architektur treffen, sollen Zugriff auf IT-COM haben.
DP 6: Abstraktionsgrad und Drill-Down	IT-COM erlaubt den Nutzern, das Abstraktionslevel im Hinblick auf die technische Domänenstruktur frei auszuwählen (Domäne, Subdomäne oder funktionale Einheit). Zudem können die Nutzer mithilfe eines „Drill-Downs“ in tiefere Ebenen die Ursachen von Komplexität auf einer höheren Ebene der Struktur untersuchen.
DP 7: Datenqualität	IT-COM macht den Nutzer visuell auf die Qualität der Daten, auf denen die dargestellten Kennzahlen beruhen, aufmerksam.

6.5.2.4 Abstrahierte Designprinzipien des ersten ADR-Projektes

Die in Tabelle 28 dargestellten instanziierten Designprinzipien beziehen sich auf das in Unterkapitel 6.3 eingeführte Problem „*Planung und Kontrolle der Komplexität der IT-Architektur der Bank A*“, als Instanz der Problemklasse „*Planung und Kontrolle von Komplexität in IT-Architekturen*“. Um alle Repräsentanten von Lösungen dieser Problemklasse zu erfassen, wird die instanziierte Lösung generalisiert und somit eine Menge abstrakter Designprinzipien vorgeschlagen (vgl. Tabelle 29).

⁸⁵ Vgl. hierzu auch Schütz et al. (2013a).

Tabelle 29: Aus dem ersten ADR-Projekt resultierende abstrakte Designprinzipien⁸⁶

Designprinzip	Beschreibung: Für eine Planung und Kontrolle von Komplexität...	Begründung	Quelle
DP 1: Anzahl und Heterogenität der Komponenten	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Komponenten des Systems bereitstellen.	Die Anzahl und Heterogenität der Komponenten sind wesentlicher Bestandteil der Konzeptualisierung von Komplexität in IT-Architekturen.	Literatur (vgl. Tabelle 27); bestätigt im Zuge der BIE-Zyklen
DP 2: Anzahl und Heterogenität der Beziehungen	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Beziehungen des Systems bereitstellen.	Die Anzahl und Heterogenität der Beziehungen sind wesentlicher Bestandteil der Konzeptualisierung von Komplexität in IT-Architekturen.	Literatur (vgl. Tabelle 27); bestätigt im Zuge der BIE-Zyklen
DP 3': Nachvollziehbarkeit von Veränderung im Zeitverlauf	...sollte das IT-Artefakt dem Benutzer die Nachvollziehbarkeit von Veränderungen der Komplexität der IT-Architektur im Zeitverlauf ermöglichen (z. B. in Form von Zeitreihen).	Informationen bzgl. der Veränderung der Komplexität der IT-Architektur ermöglichen die Nachvollziehbarkeit der Wirkung von Maßnahmen des Architekturmanagements.	Literatur (vgl. Tabelle 27); überarbeitet in BIE-Zyklus 4 (vgl. Section 6.5.2.2.1)
DP 4: Ebenen der Architektur	...sollte das IT-Artefakt die Architekturebenen Datenarchitektur, Anwendungsarchitektur und Geschäftsprozessarchitektur berücksichtigen.	Die EAM Wissensbasis sieht eine differenzierte Betrachtung der einzelnen Ebenen einer Architektur im Zuge von Designentscheidungen vor.	Literatur (vgl. Tabelle 27); bestätigt im Zuge der BIE-Zyklen
DP 5: Offenheit	...sollte der Zugriff auf das IT-Artefakt und somit Architekturinformationen und die Komplexitätskennziffern nicht auf einen spezifischen Nutzerkreis beschränkt werden. Alle Mitarbeiter, die Entscheidungen mit Einfluss auf die IT-Architektur treffen, sollten über einen Zugriff verfügen.	Informationen, die für alle Interessengruppen im Unternehmen verfügbar sind, erhöhen die Qualität des Managements der Komplexität der IT-Architektur.	Emergiert in BIE-Zyklus 4 (vgl. Section 6.5.2.2.3)
DP 6: Abstraktionsgrad und Drill-Down	...sollte das IT-Artefakt eine Analyse der IT-Architektur auf verschiedenen Abstraktionsleveln ermöglichen.	Informationen auf verschiedenen Abstraktionsleveln ermöglichen die Identifikation der Ursachen der Komplexität sowie die Ableitung von Handlungsbedarf.	Emergiert in BIE-Zyklus 4 (vgl. Section 6.5.2.2.4)
DP 7: Datenqualität	...sollte das IT-Artefakt den Nutzer auf die den Kennzahlen zugrunde liegende Datenqualität aufmerksam machen.	Bewusstsein bezüglich der Richtigkeit und Verlässlichkeit der Informationen erhöht die Qualität der Entscheidungen.	Emergiert in BIE-Zyklus 3 (vgl. Section 6.5.2.2.2)

⁸⁶ Vgl. hierzu auch Schütz et al. (2013a).

6.6 Zweites ADR-Projekt mit Bank B

Auch die IT-Landschaft von Bank B ist über die Jahre hinweg sehr stark angewachsen und weist eine hohe Heterogenität an Technologien und bankfachlichen Lösungen auf. Bedingt durch Zukäufe weiterer Finanzinstitute, die in die IT-Landschaft der Bank integriert wurden, hat sich eine Vielzahl an Redundanzen in der IT-Landschaft herausgebildet. Da das Management der Komplexität von IT-Architekturen demzufolge eine wesentliche Aufgabe der IT der Bank B darstellt, bestand auch hier ein großes Interesse an der Teilnahme an dem Forschungsprojekt. Wie bereits in Unterkapitel 6.4 angeführt, wurde – basierend auf den aus dem ersten ADR-Projekt abgeleiteten abstrakten Designprinzipien – ein zweites ADR-Projekt mit der Bank B durchgeführt. In diesem Unterkapitel werden das Vorgehen (Abschnitt 6.6.1) sowie die Ergebnisse dieses zweiten ADR-Projektes mit der Bank B (Abschnitt 6.6.2) vorgestellt.

6.6.1 Vorgehen

In Zusammenarbeit mit der Bank B wurde das IT-Artefakt „IT-COM_2“ entwickelt, das die Architekten der Bank bei dem Komplexitätsmanagement der IT-Architektur unterstützen soll (vgl. Problemklasse in Unterkapitel 6.3). Hierfür konnte ein mit dem ersten ADR-Projekt nahezu identisches Setup installiert werden. Seitens der Bank B nahmen zwei Mitarbeiter an dem ADR-Projekt teil. Thematisch waren die Mitarbeiter im Unternehmen dabei dem Bereich der Anwendungsarchitektur und dem Portfoliomanagement zugeordnet. Hierdurch gab es bereits von Beginn an zwei unterschiedliche fachliche Perspektiven auf das entstehende IT-Artefakt. Die beiden Mitarbeiter der Bank stellten dabei eine der drei – auch in diesem Projekt – disjunkten Gruppen des ADR-Teams dar. Die ADR-Forscher und das Software-Entwicklungsteam komplettierten die personelle Projektzusammenstellung (vgl. hierzu auch Tabelle 22).

Wie bereits im ersten ADR-Projekt, hatten die ADR-Forscher jederzeit die Möglichkeit, die vollständige Projektdokumentation einzusehen, den aktuellen Entwicklungsstand des entstehenden IT-Artefaktes in Augenschein zu nehmen und an allen Meetings während des Entstehungsprozesses des IT-Artefaktes teilzunehmen. Hierdurch waren alle Veränderungen an dem Artefakt stets transparent und deren Ursachen nachvollziehbar, was einen wesentlichen Aspekt der parallel erfolgenden Überprüfung und Erweiterung der Designprinzipien darstellt.

Das Artefakt wurde auch im Kontext der Bank B von Grund auf neu entwickelt und der Fokus lag ebenfalls auf einem innovativen technischen Design, das die betrachtete Problemklasse möglichst vollumfänglich erfassen sollte. Aus diesem Grund wurde auch im Rahmen des zweiten ADR-Projektes eine „IT-dominant BIE“ herangezogen (vgl. Abbildung 50; für weitere Details vgl. Abschnitt 6.5.1 und Sein et al. (2011)).

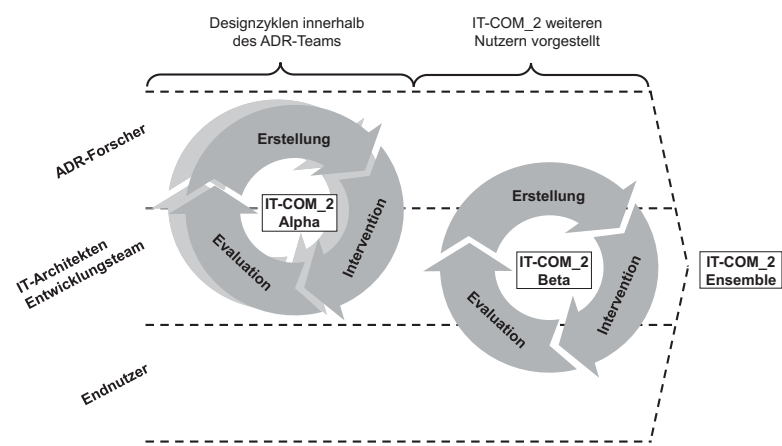


Abbildung 50: „IT-Dominant BIE“ des zweiten ADR-Projektes

Als Softwareentwicklungsmethode wurde erneut Scrum verwendet. Der Softwareentwicklungsprozess von insgesamt 15 Wochen verteilte sich auf sechs Sprints. Die Aufteilung der Sprints auf die einzelnen – in diesem Fall drei – BIE-Zyklen des zweiten ADR-Projektes sind in der folgenden Tabelle 30 dargestellt.

Tabelle 30: Überblick über die durchgeführten BIE-Zyklen des zweiten ADR-Projektes

Beteiligte	BIE-Zyklus 1	BIE-Zyklus 2	BIE-Zyklus 3
	Kennzahlenentwicklung und Mockup-Design (Sprint 1-2)	Visualisierung (Sprint 3-4)	Simulation (Sprint 5-6)
ADR-Forscher	X	X	X
Praktiker	X	X	X
Entwicklungsteam	X	X	X
Endnutzer			X

Im ersten BIE-Zyklus wurden zunächst die im Unternehmen vorhandenen Architekturinformationen durch das Entwicklungsteam untersucht und Kennzahlen zur Quantifizierung der Komplexität der IT-Architektur entwickelt. Zudem wurden – informiert durch die aus dem ersten ADR-Projekt resultierenden instanziierten und abstrakten Designprinzipien – erste Mockups entworfen, die im Anschluss mit den beiden Mitarbeitern der Bank diskutiert wurden. Wie sich die abstrakten Designprinzipien aus dem ersten ADR-Projekt in dem für Bank B entwickelten Artefakt wiederfinden, wird in Sektion 6.6.2.2.1 vorgestellt. Im Rahmen des zweiten BIE-Zyklus setzte das Entwicklungsteam die erarbeiteten Kennzahlen um und widmete sich intensiv den unterschiedlichen durch die Mitarbeiter der Bank geäußerten Wünschen an die Form der Darstellung der Informationen im IT-Artefakt. Für die ADR-Forscher lag in dem zweiten BIE-Zyklus in genau diesem Aspekt, d. h. den sehr unterschiedlichen Anforderungen an die Form der *Visualisierung* der Kennzahlen, die erste wesentliche Neuerung

(vgl. Sektion 6.6.2.2.2). Eine weitere Neuerung lag in der Funktion der *Simulation* von Änderungen an der IT-Architektur, die im dritten BIE-Zyklus zu Tage trat (vgl. Sektion 6.6.2.2.3).

Während der ersten beiden BIE-Zyklen fand die Evaluation des entstehenden IT-Artefaktes ausschließlich innerhalb des ADR-Teams statt, d. h. durch die beiden Mitarbeiter der Bank sowie die ADR-Forscher. Letztere nahmen hierbei eine deutlich passivere Rolle ein, um sicherzustellen, dass die Form der Weiterentwicklung des IT-Artefaktes möglichst vollständig auf den Anforderungen der Bank beruht. Im Rahmen des dritten und letzten BIE-Zyklus des zweiten ADR-Projektes fand eine Evaluation in einem auf weitere Mitarbeiter erweiterten organisationalen Kontext in Form eines Workshops statt.

6.6.2 Ergebnisse des zweiten ADR-Projektes

Im Folgenden werden die Ergebnisse des ADR-Projektes mit der Bank B vorgestellt. Das implementierte IT-Artefakt wird dabei auch als „IT-COM_2“ bezeichnet. IT-COM_2 wurde nicht auf der technischen Basis des bestehenden IT-Artefaktes IT-COM entwickelt. Vielmehr handelt es sich bei IT-COM_2 um ein vollständig neu implementiertes IT-Artefakt.

6.6.2.1 Initiale Designprinzipien des zweiten ADR-Projektes

Als initiale Designprinzipien dienten die abstrakten Designprinzipien, die aus der Durchführung des ersten ADR-Projektes mit Bank A resultierten. Hierbei handelt es sich demnach um theoriebasierte und durch die Erfahrungen aus dem ersten ADR-Projekt erweiterte Designprinzipien. Diese wurden dem Entwicklungsteam im Rahmen eines gemeinsamen Workshops in Vorbereitung des zweiten ADR-Projektes vorgestellt und erläutert. Hierdurch wurde ermöglicht, dass diese ersten abstrakten Designprinzipien unmittelbar von den Entwicklern berücksichtigt und durch die Praktiker im Hinblick auf deren Angemessenheit für den organisationalen Kontext von Bank B überprüft werden konnten.

6.6.2.2 Iterative Überprüfung und Emergenz von Designprinzipien

Im Zuge der durchgeführten BIE-Zyklen des zweiten ADR-Projektes wurden somit sowohl die bereits bestehenden abstrakten Designprinzipien überprüft als auch neue, zusätzliche Designprinzipien identifiziert. In der folgenden Sektion wird zunächst dargestellt, inwiefern sich die bereits bekannten Designprinzipien im neu entwickelten IT-Artefakt IT-COM_2 wiederfinden. Anschließend wird in den Sektionen 6.6.2.2.2 und 6.6.2.2.3 näher erläutert, welche neuen Designprinzipien während des ADR-Projektes mit Bank B emergierten.

6.6.2.2.1 Verortung der bereits bekannten Designprinzipien des ersten ADR-Projektes

Das entwickelte IT-Artefakt IT-COM_2 weist mit Ausnahme des Designprinzips *Datenqualität* instanziierte Ausprägungen aller abstrakten Designprinzipien des ersten ADR-Projektes auf. Dabei ist anzumerken, dass die Datenqualität – in Form eines sehr heterogenen Pflegestandes der Architekturdaten – auch ein grundlegendes Problem des zweiten ADR-Projektes darstellte. Dies führte dazu, dass einige der im ersten BIE-Zyklus des Projektes entwickelten

Kennzahlen nicht berechnet werden konnten und ein breiter Einsatz des Prototypen im Unternehmen – zum Zeitpunkt des Forschungsprojektes – daher noch nicht für sinnvoll erachtet wurde. Die Anwendung und Evaluation im erweiterten Nutzerkreis im dritten BIE-Zyklus beschränkte sich demnach auf Workshops. In diesen wurden die Funktionen von den Entwicklern und den beiden an dem ADR-Projekt beteiligten Mitarbeitern potenziellen weiteren Nutzern anhand von Anwendungsszenarien vorgestellt und weitere mögliche Einsatzfelder des IT-Artefaktes diskutiert. Die Datenqualität stellte hier also einen limitierenden Faktor dar. Das Fehlen einer Instanziierung des Designprinzips kann in diesem Projekt demzufolge damit erklärt werden, dass aufgrund der grundlegenden Datenproblematik noch nicht ausreichend Augenmerk auf die mit diesem Designprinzip verbundene Problemstellung gelegt werden konnte.

Abbildung 51 zeigt eine Übersicht über eine Reihe der in IT-COM_2 implementierten Kennzahlen (in Spalte „Tupelname“).

Application Architecture						
Tupelname	Name	N	% Δ	H	% Δ	[T]
App-Sending		10	0.0	1.07	0.0	4
App-Receiving		78	0.0	1.36	0.0	5
Business Architecture						
Tupelname	Name	N	% Δ	H	% Δ	[T]
DomainSFlow		146	0.0	1.01	0.0	4
Technical Architecture						
Tupelname	Name	N	% Δ	H	% Δ	[T]
BaseC2		0	0.0	0.0	0.0	0
App-BaseC		0	0.0	0.0	0.0	0
App-BaseC		23	0.0	1.68	0.0	6

Abbildung 51: Designprinzipien IT-COM_2 in der Architekturansicht

Wie Abbildung 51 zu entnehmen ist, stellt IT-COM_2 die Anzahl „N“ und die Heterogenität „H“ der jeweils betrachteten Komponenten bzw. Beziehungen als Kennzahl bereit. Dies entspricht den abstrakten Designprinzipien DP 1 und DP 2. Weiterhin ist der Abbildung zu entnehmen, dass die zur Verfügung gestellten Kennzahlen die verschiedenen Ebenen der Architektur abdecken (vgl. DP 4). Anhand der teilweise nicht mit Werten belegten Kennzahlen ist zu erkennen, dass das Designprinzip DP 7 (Datenqualität) grundsätzlich eine Rolle spielt. Bei Auswahl einer der implementierten Kennzahlen erhält der Nutzer weitere detaillierte Informationen. Abbildung 52 sind diese für die unterste in Abbildung 51 dargestellte Kennzahl „App-BaseC“ zu entnehmen. „App-BaseC“ stellt die Komplexität der Betriebssysteme (Basiskomponente „Operating System“) für alle Anwendungen eines bestimmten Bereiches (hier der – anonymisierte – „Bereich 3“) dar.

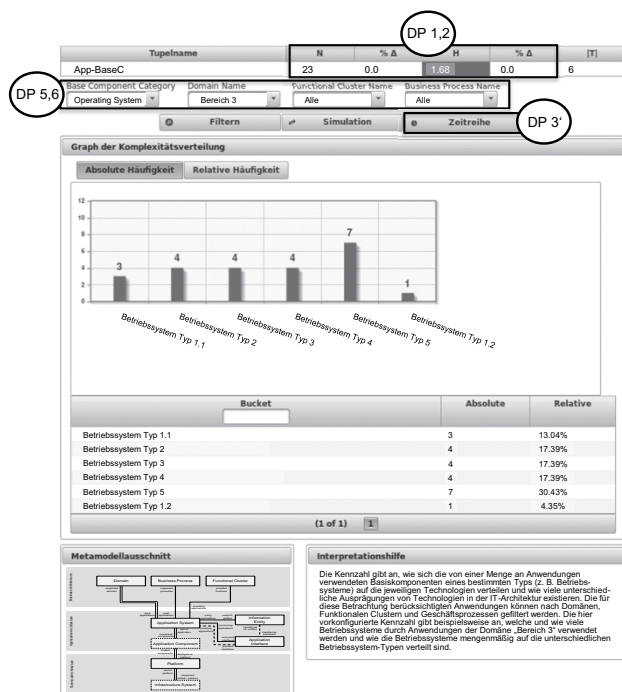


Abbildung 52: Detailsicht der Kennzahlen in IT-COM_2

Im oberen Bereich von Abbildung 52 werden dem Nutzer erneut die berechneten Kennzahlen angezeigt (vgl. DP 1 und DP 2). Unmittelbar darunter hat der Nutzer die Möglichkeit, die Auswahl auf spezifische untergeordnete organisationale Bereiche (auf Ebene der sogenannten Functional Cluster sowie der Geschäftsprozesse) einzuschränken, um die Ursache eines bestimmten Grades an Komplexität näher zu untersuchen. Anwender aus den verschiedensten Bereichen des Unternehmens können den für sie relevanten Teilbereich auswählen und durch die Möglichkeit des Drill-Downs in die einzelnen untergeordneten Teilbereiche Schwachstellen bzw. Handlungsoptionen lokalisieren. Dies entspricht den Designprinzipien *Offenheit* (DP 5) und *Abstraktionsgrad und Drill-Down* (DP 6). Unmittelbar unter der Auswahl für den Drill-Down befindet sich die Funktion der „Zeitreihe“, durch die dem Anwender die Veränderung der betrachteten Kennzahlen im Zeitverlauf angezeigt wird.⁸⁷ Dies entspricht unmittelbar Designprinzip *Nachvollziehbarkeit von Veränderung im Zeitverlauf* (DP 3'). Auf die Details im unteren Bereich von Abbildung 52 wird in der folgenden Sektion 6.6.2.2.2 näher eingegangen.

⁸⁷ Da in der verfügbaren Version von IT-COM_2 lediglich Daten für einen Zeitpunkt vorliegen und somit keine Veränderungen sichtbar sind, wird an dieser Stelle auf die Darstellung eines Screenshots für die Zeitreihe verzichtet.

6.6.2.2.2 Emergenz des Designprinzips „Visualisierung“

Im zweiten ADR-Projekt wurde von Beginn an deutlich, dass die Lösung nicht ausschließlich auf den Nutzerkreis der IT-Architekten beschränkt sein darf. Dies zeigte sich bereits anhand der von den beteiligten Praktikern betreuten fachlichen Bereiche in der Bank. Bewegt sich einer der Praktiker in einem eher technisch orientierten Kontext der Anwendungsarchitektur, liegt der Themenbereich des zweiten teilnehmenden Mitarbeiters der Bank B im Portfoliomanagement. Hierdurch wurden im Projektverlauf auch die unterschiedlichen Anforderungen an die Darstellung der Ergebnisse der Berechnungen sehr schnell deutlich. Neben der stark zahlenorientierten Darstellung des IT-Artefaktes des ersten ADR-Projektes wurden im Verlauf des zweiten ADR-Projektes zusätzliche Anforderungen vorgebracht, die zum einen der Unterstützung des Verständnisses der berechneten Kennziffern und zum anderen der Verwendung der Informationen für die interne Kommunikation dienten.

„Das Modell ist grundsätzlich nicht schwer zu erklären. Insofern ist es auch für die interne Kommunikation – sobald man erläutert hat, wie es zu interpretieren ist – gut verwendbar. Man muss aber natürlich ein grundlegendes Verständnis dafür haben, wie die Kennzahlen zustande kommen und was sie bedeuten.“ (Architekt Bank B)

Im Zuge der Workshops im dritten BIE-Zyklus wurde deutlich, dass sich die Interpretation der Kennzahlen für nicht unmittelbar an dem Projekt beteiligte Mitarbeiter der Bank B deutlich schwerer gestaltete. Hieraus wurde abgeleitet, dass das IT-Artefakt dem Benutzer (a) zusätzliche Informationen im Hinblick auf die Bedeutung der Kennzahlen und (b) eine Alternative zu der zahlengetriebenen Darstellungsform bereitstellen sollte. Für den ersten Punkt (a) wurden in IT-COM_2 sowohl eine visuelle als auch eine textuelle Unterstützung für ein besseres Verständnis der jeweiligen Kennzahl integriert. Wie Abbildung 53 zeigt, wurden zum einen die für die Berechnung der Kennzahlen benötigten Objekte im Architektur-Metamodell⁸⁸ optisch hervorgehoben und zum anderen eine Interpretationshilfe im Form eines kennzahlenspezifischen Informationstextes hinzugefügt. Einer der beteiligten Architekten begründete diese beiden Hilfestellungen wie folgt:

„Dies stellt eine Art Definition der spezifischen Kennzahl dar, die aufzeigt, was genau gemessen wird. Auch Mitarbeiter, die das Tool noch nicht kennen, müssen die Chance haben, nachvollziehen und verstehen zu können, was man als Kennzahl festgelegt hat und was sie bedeutet.“ (Architekt Bank B)

⁸⁸ Das tatsächliche Architektur-Metamodell von Bank B wurde für die Abbildungen durch ein fiktives Metamodell ersetzt.

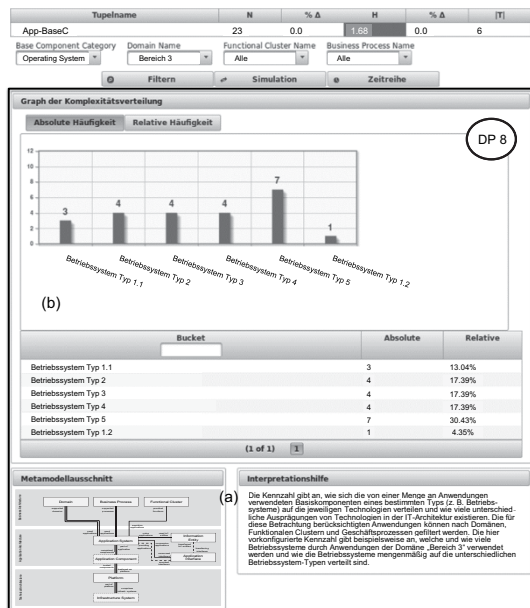


Abbildung 53: Zusätzliche Visualisierungen der Kennzahlen in IT-COM_2

Punkt (b) wurde ebenfalls durch Hinzufügen von zwei zusätzlichen Elementen im IT-Artefakt begegnet. Die für die Berechnung der jeweiligen Kennzahlen betrachteten Komponenten bzw. Relationen werden in IT-COM_2 sowohl anhand eines Balkendiagramms als auch in Tabellenform visualisiert. Der Benutzer erhält hierdurch mehr Informationen über das Zustandekommen der jeweiligen Kennzahl und kann frei wählen, welche Form der Darstellung er für die weitere interne Kommunikation heranzieht.

„Eine Excel-ähnliche Darstellung ist natürlich auch ein Instrument, um sich einen Überblick über die Kennzahlen zu verschaffen. Aber eine graphische Darstellung ist natürlich interessanter und einfacher, um ein Verständnis dafür zu bekommen, wie die Verteilung aussieht. Das kann man dadurch schön visualisieren. Und man sieht auf einen Blick, wie sich die Komplexität für diese Dimension ausprägt. Wenn ich beispielsweise sehe, dass eine relativ hohe Verschiedenartigkeit vorliegt und gleichzeitig eine Verteilung, die ziemlich gleichverteilt ist, dann weiß ich – das geübte Auge sieht sofort –, die Komplexität könnte kaum höher sein. Wenn ich aber sehe, dass dort irgendwelche Peaks sind und die Anzahl unterschiedlicher Ausprägungen gleichzeitig relativ überschaubar ist, dann sehe ich sofort, dass ich für diese Dimension schon ganz gut mit der Komplexität dastehe.“
(Architekt Bank B)

Für den Architekten erweist sich die visuelle Unterstützung mittels der Balkendiagramme demnach als eine Möglichkeit, schnell weitere Schlüsse im Hinblick auf den Grad der Komplexität des betrachteten Ausschnittes der IT-Architektur zu ziehen. In der tabellarischen Dar-

stellungsform sehen die Praktiker dabei eine Alternative zum Balkendiagramm, die zwar die gleichen Informationen transportiert, jedoch für manche Kollegen die geeignetere Form der Visualisierung darstellt. Das entwickelte Informationssystem soll dabei beide Möglichkeiten bereitstellen.

„Es ist dieselbe Information, einmal tabellarisch, einmal eher visuell. Es ist lediglich eine andere Art der Repräsentation. Manche Leute brauchen eher die eine, andere die andere Form. Ich würde immer beides zulassen und nicht eine der beiden vorgeben. Ich bin eher der visuelle Typ, könnte aber nicht sagen, das eine ist besser als das andere.“ (Architekt Bank B)

Auf Grundlage der hier angeführten Überlegungen wurde das instanziierte Designprinzip *Visualisierung* (DP 8) eingeführt: „IT-COM_2 stellt dem Benutzer – entsprechend der individuellen Anforderungen – verschiedene Darstellungsformen der relevanten Informationen bereit“.

Dieses Designprinzip bezieht sich dabei nicht lediglich auf die hier angeführten Formen der Visualisierung, sondern auf die Gesamtheit des entwickelten Informationssystems, das ermöglichen soll, die Komplexität der IT-Architektur greifbar und managebar zu machen. Es umfasst demzufolge insbesondere auch die verfügbaren Zeitreihen zur Visualisierung der Veränderung der berechneten Kennzahlen im Zeitverlauf.

6.6.2.2.3 *Emergenz des Designprinzips „Simulation“*

Ein weiteres Designprinzip bezieht sich u. a. auch auf die eben angeführten Zeitreihen und somit auf das Designprinzip *Nachvollziehbarkeit von Veränderung im Zeitverlauf* (DP 3'). Im Zuge des dritten BIE-Zyklus des zweiten ADR-Projektes wurde deutlich, dass in DP 3' sowie DP 5 (Offenheit) implizit die Möglichkeit des Testens von Änderungen an der Architektur enthalten ist. Soll auf die Entwicklung der Veränderung im Zeitverlauf Einfluss genommen werden, so benötigen die Nutzer die Möglichkeit, verschiedene Handlungsoptionen und deren Wirkung auf die Komplexität zu simulieren. Eine solche Möglichkeit zur Simulation eines Eingriffs in die IT-Architektur war in IT-COM (dem aus dem ersten ADR-Projekt resultierenden IT-Artefakt) nicht explizit vorgesehen. Hier konnten lediglich „künstlich angepasste“ Architekturinformationen in den Prototypen geladen und somit eine Änderung an der IT-Architektur simuliert werden. Im Rahmen des zweiten ADR-Projektes wurde die Möglichkeit zur Simulation hingegen als eine eigene Anforderung an das IT-Artefakt formuliert und umgesetzt. Der Architekt sagte hierzu folgendes:

„Die Simulation dient dazu, darzustellen, was mit der IT-Architektur und deren Komplexität geschehen würde, wenn ich eine bestimmte Änderung – beispielsweise im Zuge von Projekten – realisiere. Verringert sich durch eine Änderung die absolute bzw. relative Häufigkeit oder die Anzahl der verschiedenen Optionen [z. B. Technologien]? Und welchen Wert für die Komplexität würde ich dann bekommen? [...] Hiermit kann ich verschiedene Zielszenarien durchgehen und schauen, wie es aussehen würde, wenn ich diese realisiere.“ (Architekt Bank B)

Abbildung 54 ist die Umsetzung dieser Funktionalität in IT-COM_2 zu entnehmen. In dem hier dargestellten Beispiel wird eine Änderung an dem bereits in den vorangegangenen Abbildungen betrachteten Architekturbereich vorgenommen, indem eine der Betriebssystem-Instanzen (Typ 1.2) auf die dominantere Ausprägung (Typ 1.1) migriert wird. An dieser Stelle könnten noch weitere Betriebssysteme migriert oder auch zusätzliche Betriebssysteme ergänzt werden.

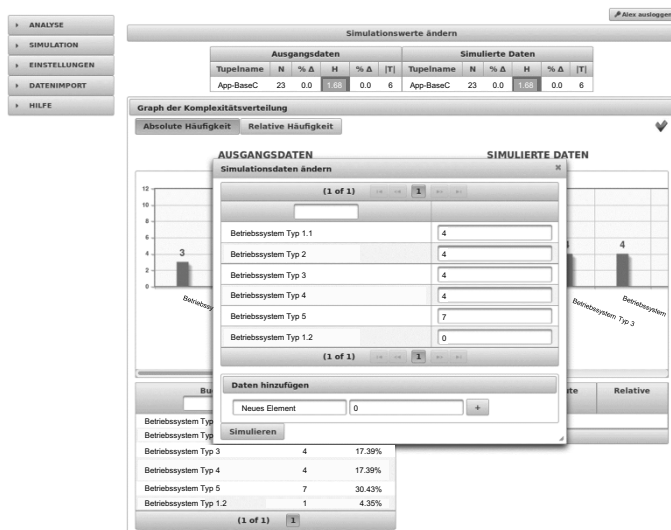


Abbildung 54: Simulation von Änderungen in IT-COM_2

Bei Bestätigung der Veränderung der IT-Architektur werden die jeweils betroffenen Kennzahlen neu berechnet und dem Nutzer angezeigt. Eine solche Veränderung ist der folgenden Abbildung 55 zu entnehmen.



Abbildung 55: Darstellung der Simulationsergebnisse in IT-COM_2

Im oberen Bereich von Abbildung 55 ist zu erkennen, dass sich die betrachtete Heterogenität der Betriebssysteme durch den angeführten Eingriff in die IT-Architektur von 1,68 auf 1,58 (H) reduziert und dabei auch eine der Technologien ([T]) in diesem Bereich entfällt.

„Insofern bietet mir die Simulation die Möglichkeit, eine konkret geplante Maßnahme zu illustrieren und einer Idee – die ich habe – Ausdruck zu verleihen. Ich kann sie damit darstellen, ich kann sie dokumentieren und ich kann sie präsentieren: ‚Wir haben die folgenden Projekte geplant und Handlungsoptionen verprobt, bei denen sich die Komplexität wie folgt verändert.‘ Das ist für mich der wesentliche Mehrwert der Simulation.“
(Architekt Bank B)

Auf Grundlage der hier gewonnenen Erkenntnisse wurde das instanziierte Designprinzip *Simulation* (DP 9) eingeführt: „IT-COM_2 stellt dem Benutzer die Möglichkeit zur Simulation der Auswirkungen von Veränderungen der IT-Architektur bereit“.

6.6.2.3 Instanziierte Designprinzipien des zweiten ADR-Projektes

Eine Übersicht der instanziierten Designprinzipien des im zweiten ADR-Projekt implementierten IT-Artefaktes IT-COM_2 wird in der folgenden Tabelle 31 angeführt. Entsprechend dem ersten ADR-Projekt stellen sowohl das IT-Artefakt IT-COM_2 als auch die instanziierten Designprinzipien für den Kontext der Bank B „material artifacts“ (Gregor & Jones 2007) und somit „Level 1 contributions“ (Gregor & Hevner 2013) aus der Durchführung des ADR-Projektes dar.

Tabelle 31: Instanzierte Designprinzipien im Rahmen des zweiten ADR-Projektes

Designprinzip	Instanziierungen der Designprinzipien in der Bank
DP 1: Anzahl und Heterogenität der Komponenten	Das implementierte Informationssystem IT-COM_2 berücksichtigt die Anzahl und Heterogenität unterschiedlicher Ausprägungen von Basiskomponenten (z. B. Datenbanken, Middleware, Betriebssysteme oder Server-Hardware)
DP 2: Anzahl und Heterogenität der Beziehungen	IT-COM_2 berücksichtigt die Anzahl der Schnittstellen und die Heterogenität hinsichtlich der Domänenzugehörigkeit der Zielapplikation der Schnittstellenbeziehung. Zudem gibt IT-COM_2 an, wie häufig eine Anwendung die Funktion des Senders bzw. Empfängers inne hat (Anzahl) und wie diese Schnittstellen auf die jeweils an dem Informationsaustausch beteiligten Applikationen verteilt sind (Heterogenität).
DP 3': Nachvollziehbarkeit von Veränderung im Zeitverlauf	IT-COM_2 ermöglicht die Nachvollziehbarkeit der Veränderung der berechneten Komplexitätskennzahlen der IT-Architektur über die Zeit hinweg anhand von Zeitreihen.
DP 4: Ebenen der Architektur	IT-COM berücksichtigt, ausgehend von der Anwendungsarchitektur, die Ebenen der Infrastrukturarchitektur sowie der Geschäftsarchitektur.
DP 5: Offenheit	Der Zugriff auf IT-COM_2 soll nicht auf eine spezifische Zielgruppe eingeschränkt werden. Alle Mitarbeiter, die Entscheidungen mit einem Einfluss auf die IT-Architektur treffen, sollen Zugriff auf IT-COM_2 haben.
DP 6: Abstraktionsgrad und Drill-Down	IT-COM_2 erlaubt den Nutzern, das Abstraktionslevel durch das Setzen zusätzlicher Filter weiter zu verfeinern. Zur Verfügung stehen hierfür die Domänen, diese weiter präzisierende funktionale Cluster sowie die Geschäftsprozesse. Hiermit können die Nutzer mithilfe eines „Drill-Downs“ in tiefere Ebenen die Ursachen von Komplexität auf einer höheren Ebene der Struktur untersuchen.
DP 8: Visualisierung	IT-COM_2 stellt dem Benutzer – entsprechend der individuellen Anforderungen – verschiedene Darstellungsformen der relevanten Informationen bereit. Hierdurch erhalten auch Mitarbeiter, die sich bislang noch nicht mit dem Tool beschäftigt haben, ein Grundverständnis für die bereitgestellten Daten und verschiedene Formen der Ergebnisse für die Weiterverwendung im Zuge der internen Kommunikation.
DP 9: Simulation	IT-COM_2 stellt dem Benutzer die Möglichkeit zur Simulation der Auswirkungen von Veränderungen der IT-Architektur bereit.

6.6.2.4 Abstrahierte Designprinzipien des zweiten ADR-Projektes

Die Darstellung der abstrahierten Designprinzipien beschränkt sich an dieser Stelle auf die während der Durchführung des zweiten ADR-Projektes zusätzlich emergierten Designprinzipien. Daher handelt es sich bei der folgenden Tabelle 32 um eine Ergänzung der bereits in Tabelle 29 eingeführten abstrakten Designprinzipien:

Tabelle 32: Aus dem zweiten ADR-Projekt resultierende abstrakte Designprinzipien

Designprinzip	Beschreibung/Statement: Für eine Planung und Kontrolle von Komplexität...	Begründung	Quelle
DP 8: Visualisierung	...sollte das IT-Artefakt – entsprechend der individuellen Anforderungen der Benutzer – verschiedene Darstellungsformen der relevanten Informationen bereitstellen.	Adäquate Darstellungsformen der bereitgestellten Kennzahlen und ergänzende Informationen erleichtern das Verständnis sowie die Kommunizierbarkeit entscheidungsrelevanter Informationen.	Emergiert in BIE-Zyklus 2 (vgl. Sektion 6.6.2.2.2)
DP 9: Simulation	...sollte das IT-Artefakt eine Simulation der Auswirkungen von Veränderungen der IT-Architektur ermöglichen.	Die Simulation von Veränderungen an der IT-Architektur ermöglicht dem Benutzer die Überprüfung unterschiedlicher Handlungsalternativen und deren Auswirkungen auf die Komplexität der IT-Architektur.	Emergiert in BIE-Zyklus 3 (vgl. Sektion 6.6.2.2.3)

6.7 Erweiterung der Ergebnisse des ersten ADR-Projektes

Die in Unterkapitel 6.6 überarbeiteten abstrakten Designprinzipien wurden in der dritten Stufe des Forschungsprojektes (im Folgenden wird diese auch als drittes ADR-Projekt bezeichnet) durch zwei weitere ADR-Zyklen im organisationalen Kontext von Bank A erneut überprüft. Ziel war hierbei zum einen zu untersuchen, inwieweit die im Kontext von Bank B emergierten Designprinzipien (DP 8 und DP 9) auch auf den Kontext von Bank A zutreffen. Zum anderen sollten im Rahmen zweier zusätzlicher BIE-Zyklen die bislang durch die IT-Architekten gesammelten Erfahrungen mit dem IT-Artefakt eingebracht werden. Im folgenden Abschnitt 6.7.1 wird zunächst das Vorgehen im Zuge des dritten ADR-Projektes eingeführt. Im Anschluss daran werden in Abschnitt 6.7.2 die aus den zusätzlichen BIE-Zyklen im Kontext von Bank A gewonnenen Erkenntnisse dargestellt.

6.7.1 Vorgehen

Basierend auf dem bestehenden IT-Artefakt IT-COM sowie den aus dem zweiten ADR-Projekt resultierenden abstrakten Designprinzipien, wurde eine Überarbeitung des IT-Artefaktes und damit auch eine erneute Überprüfung der abstrakten Designprinzipien vorgenommen. Auf Praxisseite waren die beiden IT-Architekten des ersten ADR-Projektes an der Durchführung der zusätzlichen BIE-Zyklen involviert. Im Rahmen des finalen Workshops nahm zudem ein Abteilungsleiter eines Teilbereiches der Architektur teil. Das Entwicklungsteam bestand aus zwei Entwicklern, die auch bereits an der Implementierung des ersten IT-Artefaktes IT-COM beteiligt waren. Dies ermöglichte einen schnellen Wiedereinstieg in die Designzyklen. Aufgrund der geringeren Teamstärke wurde die Dauer eines Sprints auf durchschnittlich drei Wochen pro Sprint erweitert, wodurch sich eine Gesamtzahl von vier Sprints für die Weiterentwicklung des IT-Artefaktes ergab. Diese wurden gleichmäßig auf zwei BIE-

Zyklen verteilt. Tabelle 33 gibt einen Überblick über die beiden BIE-Zyklen und die jeweils beteiligten Gruppen.

Tabelle 33: Überblick über die durchgeführten BIE-Zyklen des dritten ADR-Projektes

Beteiligte	BIE-Zyklus 1	BIE-Zyklus 2
	Kennzahlendesign und Modularität (Sprint 1-2)	Visualisierung und Simulation (Sprint 3-4)
ADR-Forscher	X	X
Praktiker	X	X
Entwicklungsteam	X	X
Endnutzer		X

Inhaltlich wurde im ersten BIE-Zyklus eine Anpassung der Kennzahlendarstellung durchgeführt. Darüber hinaus kam mit der Modularität eine neue, zusätzliche Perspektive auf die Problemstellung des Komplexitätsmanagements von IT-Architekturen hinzu. Im zweiten BIE-Zyklus wurden die Designprinzipien der Visualisierung und der Simulation umgesetzt. Die Evaluation im Zuge des zweiten BIE-Zyklus fand dabei in Form eines Workshops mit den IT-Architekten und einem Abteilungsleiter aus dem Bereich der IT-Architektur statt.

6.7.2 Ergebnisse des dritten ADR-Projektes

In diesem Abschnitt werden die Ergebnisse der Durchführung des dritten ADR-Projektes zur Überprüfung und Weiterentwicklung des IT-Artefaktes sowie der Designprinzipien vorgestellt. Entsprechend der bisherigen Struktur werden hierzu in Unterabschnitt 6.7.2.1 zunächst die initialen Designprinzipien des ADR-Projektes eingeführt. Im Anschluss daran wird in Unterabschnitt 6.7.2.2 die iterative Überprüfung der Designprinzipien während der beiden BIE-Zyklen dargestellt, wobei auch an dieser Stelle zwischen der Verortung bereits bekannter und der Emergenz zusätzlicher Designprinzipien unterschieden wird. Die Manifestierung dieser Designprinzipien in IT-COM_3 wird in Unterabschnitt 6.7.2.3 aufgezeigt.

6.7.2.1 Initiale Designprinzipien des dritten ADR-Projektes

Wie bereits in Unterabschnitt 6.5.2.1 angeführt, basiert die Entwicklung eines Artefaktes nach der ADR-Methode auf einer theoretischen Grundlage.⁸⁹ Diese setzt sich im Fall des dritten ADR-Projektes aus den abstrakten Designprinzipien des ersten ADR-Projektes (vgl. Unterabschnitt 6.5.2.4), den im Rahmen des zweiten ADR-Projektes hinzugekommenen abstrakten Designprinzipien (vgl. Unterabschnitt 6.6.2.4) sowie theoretischen Überlegungen zu den durch die IT-Architekten gesammelten Erfahrungen des Einsatzes von IT-COM zusammen.

Bereits parallel zur Durchführung des zweiten ADR-Projektes mit Bank B kam es zu regelmäßigen Treffen mit den Architekten der Bank A, um über die weiteren Erfahrungen mit dem IT-Artefakt zu sprechen. Diese gaben dabei stets intern gesammeltes Feedback ihrer Kollegen

⁸⁹ Vgl. hierzu „Principle 2: Theory-Ingrained Artifact“ nach Sein et al. (2011), S. 40, 41.

sowie Rückmeldungen von IT-Architekten anderer Unternehmen, denen sie das IT-Artefakt vorstellen konnten, wieder.⁹⁰ In diesen Gesprächen mit den IT-Architekten wurde zunehmend deutlich, dass neben den bereits existierenden Designprinzipien eine weitere, die Struktur des Systems IT-Architektur beschreibende Perspektive ergänzt werden sollte.

Hintergrund ist dabei, dass ein großes und heterogenes System, das jedoch bezüglich der Interaktion der Komponenten untereinander „gut geschnitten“ und modular aufgebaut ist, relativ gut zu managen ist. Schwieriger ist dies hingegen im Falle eines Systems, dessen Anwendungen die gleiche Anzahl und Heterogenität aufweisen, diese jedoch Informationen mit einer Vielzahl von Anwendungen in den verschiedensten Bereichen des Unternehmens austauschen. Auch auf organisatorischer Ebene kann eine solche Perspektive relevant sein. Einer der IT-Architekten äußerte sich hierzu wie folgt:

„Eine falsche Zuordnung von Komponenten kann im Hinblick auf die Flexibilität des Systems katastrophale Folgen haben. Wenn z. B. Abhängigkeiten kreuz und quer zwischen den Bereichen verteilt sind, kann dies Abstimmungen im Rahmen von Änderungen wesentlich erschweren. Wenn eine organisatorische Zuordnung der Verantwortlichkeiten – oder auch der Budgetverantwortlichkeiten – so geschnitten ist, dass für eine Anwendung 15 verschiedene Verantwortlichkeiten existieren, dann erreichst du nichts mehr.“
(Enterprise Architect Bank A)

Feedback der IT-Architekten im Vorfeld des dritten ADR-Projektes bot die Möglichkeit, bestehende theoretische Konzepte in der Literatur zu sichten und im Zuge des dritten ADR-Projektes einzubringen. Hierbei zeigte sich, dass das Konzept der *Modularität eines Systems* die durch die Architekten diskutierte Problemstellung adäquat berücksichtigt. Im Folgenden wird dieses theoretische Konzept eingeführt. Baldwin & Clark (2000) definieren die Begriffe „Modul“ und „Modularität“ wie folgt:

„[...] [A Module is] a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units.“ (Baldwin & Clark 2000, S. 63)

“Modularity is a very general set of principles for managing complexity. By breaking up a complex system into discrete pieces—which can then communicate with one another only through standardized interfaces within a standardized architecture—one can eliminate what would otherwise be an unmanageable spaghetti tangle of systemic interconnections.” (Baldwin & Clark 2000, S. 63 f.)

Durch Integration (z. B. fachlich) zusammengehöriger Komponenten zu einem Modul und der Definition klarer Schnittstellen, über welche die Kommunikation des Moduls mit den weiteren Komponenten des Systems realisiert wird, kann ein Teil der Komplexität des Systems in einem Modul zusammengefasst werden (Baldwin & Clark 2000). Eine derart integrierte Form der Komponenten der IT-Architektur – durch Module – erlaubt es, effizient Änderungen an

⁹⁰ Die IT-Architekten stellten das IT-Artefakt IT-COM u. a. im Rahmen eines Treffens der Enterprise Architekten der Region vor.

der Struktur des Systems vorzunehmen. Langlois (2002) führt Modularität daher als grundlegendes Prinzip für das Management der Komplexität von Systemen an.

Modularisierung findet bereits in den unterschiedlichsten Kontexten, wie der Konstruktion von Flugzeugen (Woolsey 1994), der Automobilindustrie (Tully & Welsh 1993), in der Entwicklung von Mikrocomputern (Langlois & Robertson 1992) oder der Softwareentwicklung (Parnas et al. 1985), Anwendung. Simon (1962) führte die Idee der Modularität im Zuge der Diskussion der Konzepte der sogenannten *hierarchic and nearly decomposable systems* ein. Ein hierarchisches System ist dabei ein technisches oder organisatorisches System, bestehend aus interagierenden Subsystemen. Bei diesen Subsystemen handelt es sich wiederum um hierarchische Systeme, die solange in Subsysteme aufgeteilt werden können, bis “[...] *some lowest level of elementary subsystem* [...]” (Simon 1962, S. 468) erreicht ist. Der Kern des Konzeptes der *nearly decomposable systems* liegt darin, dass Interaktionen von Komponenten innerhalb eines Subsystems stärker ausgeprägt sind, als die Interaktion zwischen den einzelnen Subsystemen. Aus dieser Idee der Stärke der Interaktion innerhalb bzw. zwischen Subsystemen hat sich in der Literatur zur Modularität das Konzept der losen Kopplung herausgebildet (Sanchez & Mahoney 1996; Schilling 2000). Die Berücksichtigung dieses Konzeptes erleichtert die Durchführung von Änderungen, indem z. B. der in obigem Zitat angesprochene Abstimmungsaufwand zwischen den Verantwortlichen einzelner Komponenten auf einen Teilbereich (das Subsystem) beschränkt wird. Ein Änderungsprozess kann somit deutlich dezentralisiert werden (Schilling 2000). Einer der IT-Architekten äußerte sich hierzu wie folgt:

„Völlig saubere, disjunkte 1:1-Zuordnungen bekommst du niemals hin. Aber das ganze so zu schneiden, dass ein Abstimm-Overhead gering bleibt oder Abhängigkeiten allgemein möglichst innerhalb dieser einen Einheit konzentriert sind, das ist das Ziel. Das ist auch dieser Aspekt der losen Kopplung: Möglichst viel mit sich selbst beschäftigt sein, möglichst wenig und möglichst gezielt und geplant mit den Komponenten außerhalb des Subsystems.“ (Enterprise Architect Bank A)

Neben der angeführten Abstimmungs-Thematik spielt auch das für die Durchführung von Änderungen erforderliche Fachwissen eine Rolle. Sind die Module fachlich gut geschnitten, wird für die Durchführung von Änderungen an der Struktur oder der Implementierung einzelner Komponenten des Moduls primär modulspezifisches Fachwissen benötigt. Hingegen ist Fachwissen im Hinblick auf die Struktur oder die Implementierung anderer Module nicht (oder nur geringfügig) erforderlich (Parnas et al. 1985).

In der Literatur wird betont, dass die Erstellung eines u. a. hinsichtlich der Modularität optimal ausgestalteten Systems eine sehr schwierige Aufgabe darstellt. Ethiraj & Levinthal (2004) schreiben hierzu:

„[...] the problem of complex system design is not an exact science that will always result in an optimal solution.“ (Ethiraj & Levinthal 2004, S. 162)

Ethiraj & Levinthal (2004) führen in ihrer Simulationsstudie die folgenden vier Gütekriterien ein, anhand derer der Designer die Angemessenheit der Modularität eines Systems beurteilen kann:

- (1) die Angemessenheit der Anzahl von Modulen,
- (2) die Angemessenheit der Zuordnung der Designelemente auf die Module,
- (3) die Angemessenheit der Interaktionen der Designelemente innerhalb der Module und
- (4) die Angemessenheit der Schnittstellen oder Interaktionen zwischen Modulen.

Die theoretische Perspektive auf die Modularität des Systems sowie die hier angegebenen Gütekriterien stellten die Basis für die Berücksichtigung des angeführten Problems der IT-Architekten dar.

6.7.2.2 Iterative Überprüfung der Designprinzipien

Wie die folgenden beiden Sektionen zeigen, emergierten im Kontext des dritten ADR-Projektes mit der Bank A mit *Modularität* und *Anpassbarkeit* zwei zusätzliche Designprinzipien. Die beiden Designprinzipien, die sich aus dem zweiten ADR-Projekt mit Bank B (*Visualisierung* und *Simulation*) ergaben, wurden ebenfalls im Kontext von Bank A bestätigt (vgl. Sektionen 6.7.2.2.3 und 6.7.2.2.4).

6.7.2.2.1 Emergenz des Designprinzips „Modularität“

Wie bereits in Unterabschnitt 6.7.2.1 angeführt, kristallisierte sich bereits vor Beginn des dritten ADR-Projektes das Fehlen einer Möglichkeit zur Beurteilung der Struktur des Systems heraus. Um diesen Gesichtspunkt zu berücksichtigen, wurde in BIE-Zyklus 1 das in Unterabschnitt 6.7.2.1 vorgestellte theoretische Konzept der Modularität in das IT-Artefakt integriert. Die Umsetzung dieser Anforderung orientierte sich am Verständnis von Modularität nach Ethiraj & Levinthal (2004) sowie den in ihrer Simulationsstudie eingeführten vier Dimensionen zur Beurteilung der Angemessenheit der Modularität eines Systems. Zielsetzung bei der Umsetzung war demzufolge, dass die vier Gütekriterien bzgl. der Modularität des Systems durch den IT-Architekten überprüfbar sein sollten. Der folgenden Abbildung 56 ist die Umsetzung in IT-COM_3 zu entnehmen.

Der Nutzer hat die Möglichkeit, ein System (oder auch ein Subsystem) auszuwählen, dessen Modularität er im Detail untersuchen möchte. Hier stehen ihm sowohl die Domänen als auch die Subdomänen für eine Analyse zur Verfügung. Im linken Bereich der Abbildung 56 kann anhand der Anzahl der Balken – die jeweils eine funktionale Einheit und somit ein Modul repräsentieren – das erste Gütekriterium beurteilt werden, d. h. die Angemessenheit der Anzahl der Module. Die Höhe der Balken des linken Diagramms zeigt die Anzahl der jeweils zugeordneten Anwendungen auf und dient somit der Beurteilung des zweiten Gütekriteriums bzgl. der Zuordnung der Designelemente zu den Modulen. Zur Beurteilung des dritten und vierten Gütekriteriums sind die Balkendiagramme auf der rechten Seite aufgeführt. Dem obersten Diagramm ist die Anzahl der Schnittstellen zwischen den Anwendungen innerhalb der funktionalen Einheiten zu entnehmen, was dem dritten Gütekriterium entspricht. Im mittleren Diagramm ist dementsprechend die Anzahl der Schnittstellen der funktionalen Einheiten zu Anwendungen außerhalb der jeweiligen funktionalen Einheit angeführt.

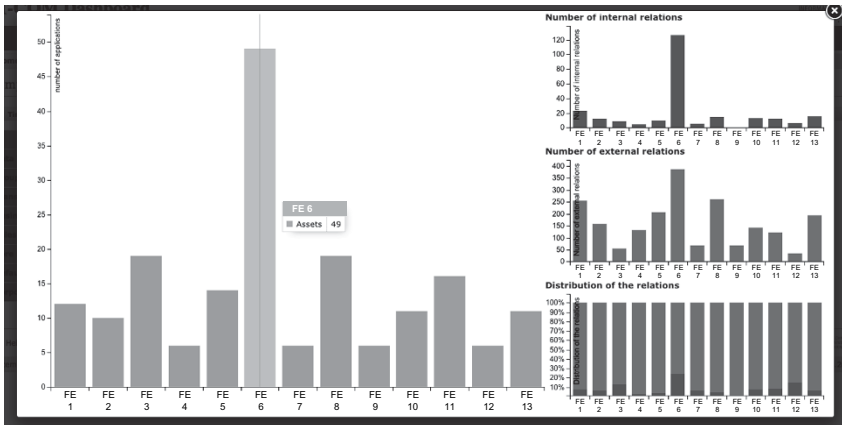


Abbildung 56: Darstellung von Modularität in IT-COM_3

Um optische Größeneffekte der beiden Diagramme zu relativieren, zeigt das unterste Diagramm das Verhältnis zwischen der Anzahl interner und externer Schnittstellen der Module auf. Diese Form der Darstellung ermöglicht es dem Nutzer, direkt auffällige funktionale Einheiten zu identifizieren, die einer weiteren Betrachtung unterzogen werden sollten:

„Der Vorteil ist, dass ich sofort sehe, welche der funktionalen Einheiten ich mir näher ansehen sollte.“ (Principal Architect)

Der Nutzer kann dabei definieren, welches die relevanten Designelemente sind und auf Grundlage welcher Form der Interaktionen die Kommunikation zwischen den Designelementen festzumachen ist (z. B. anhand der technischen Schnittstellen zwischen den Anwendungen oder anhand der Zuordnung des für die Anwendung verantwortlichen Mitarbeiters). Dieser generische Ansatz ermöglicht dem Nutzer, unterschiedliche Perspektiven auf die Modularität einzunehmen, ohne dass sich die zugrunde liegende Logik ändert.

Der Aspekt der Modularität manifestierte sich noch an einer weiteren Stelle von IT-COM_3, wobei sich diese Darstellung auf das dritte und vierte Gütekriterium beschränkt. Mit Abbildung 57 wurde eine universelle Darstellung aufgenommen, die gleichzeitig eine Übersicht über alle Domänen und die Verknüpfungen ihrer Subsysteme aufzeigt.

Am Rand des Kreises sind die funktionalen Einheiten (FEs) angegeben, jeweils gruppiert nach deren Zugehörigkeit zu den Subdomänen. Die Linien stellen die Schnittstellen zwischen Anwendungen der funktionalen Einheiten dar, wobei die Dicke der Verbindungen Auskunft über die Anzahl der Schnittstellen zwischen den Anwendungen der funktionalen Einheiten gibt. Bei Vorliegen einer sehr modularen Struktur auf Ebene der Subdomänen würden die Schnittstellen grundsätzlich primär innerhalb der Subdomänen verlaufen. Im Hinblick auf die Darstellung in Abbildung 57 würde dies einer Bündelung der Schnittstellen an den Rändern des Kreises und innerhalb der abgesetzten Gruppen entsprechen. Zu erwartende Ausnahmen stellen funktionale Einheiten dar, denen Anwendungen zugeordnet sind, die zentrale Dienste

anbieten und somit bereichsübergreifend Informationen bereitstellen (z. B. ein Data Warehouse).

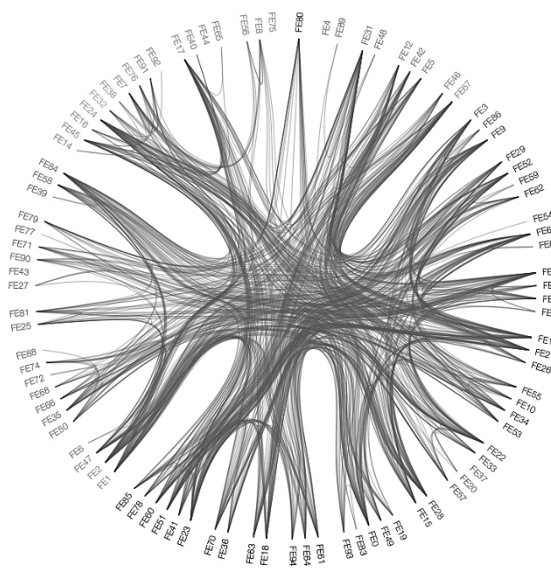


Abbildung 57: Zweite Form der Visualisierung der Modularität in IT-COM_3⁹¹

Einer der Architekten fasst diese Darstellungsform der Modularität wie folgt zusammen:

„Dies ist eine Gesamtdarstellung der kompletten Anwendungsarchitektur. [...] Hier kann man erkennen, von und zu welcher Domäne, welchen Subdomänen und funktionalen Einheiten wie viele Schnittstellen hinführen. [...] Eigentlich soll eine lose Kopplung entlang der Domänen erreicht werden. [...] Eine ‚ausgedünntere Mitte‘ würde dieser losen Kopplung entsprechen“. (Principal Architect)

Die angeführten Überlegungen führten zur Einführung des Designprinzips *Modularität* (DP 10): „IT-COM_3 erlaubt die Beurteilung der Güte der Modularität des betrachteten Systems“.

6.7.2.2.2 Emergenz des Designprinzips „Anpassbarkeit“

Eine weitere Erfahrung während der ersten Nutzung von IT-COM in der Bank lag darin, dass einige potenzielle Nutzer, die nicht an den ersten Workshops teilgenommen hatten, mit der Masse an Kennzahlen überfordert waren. Da zudem viele der Kennzahlen der ersten Version teilweise auf sehr spezifischen Berechnungsvorschriften beruhten, war das Verständnis der Kennzahlen häufig nur mit Hilfe einer unterstützenden Dokumentation möglich:

⁹¹ Für die Erstellung der Abbildung wurden Testdaten verwendet.

„Ja, wer hat aber die 22 Kennzahlen im Kopf und weiß, welche These dahinterstand, wie sie sich genau zusammensetzen und wie sie genau berechnet werden? Hier brauchte man ja eine einwöchige Schulung für die Kollegen, damit sie das gesamte Modell mit allen Kennzahlen verstehen konnten. Und hier ist es nun deutlich schneller und intuitiver zugänglich.“ (Principal Architect)

Dieser Punkt wurde bei der Überarbeitung des IT-Artefaktes in zweierlei Hinsicht angegangen. Zum einen wurde eine einheitliche Form der Quantifizierung struktureller Komplexität von IT-Architekturen, basierend auf den in Kapitel 5 eingeführten Kennzahlen, ergänzt. Gleichzeitig hat der Nutzer in IT-COM_3 die Möglichkeit, die von ihm benötigte Kennzahl frei auszuprägen (hier limitiert lediglich der vorhandene Datenbestand). Wie Abbildung 58 zu entnehmen ist, kann der Nutzer je nach Fragestellung die Komponente und die Beziehung sowie jeweils das Kriterium, nach dem die Komponenten bzw. die Beziehungen unterschieden werden sollen, auswählen. In dem in Abbildung 58 angeführten Beispiel wurden die Komponente „Application“ mit dem Unterscheidungskriterium „Vendor“ sowie die Beziehung „Interface“ mit dem Unterscheidungskriterium „Implementation“ gewählt.

Home Complexity Complexity - Scenario Complexity - Scatter View Complexity - Relation Circle Complexity Aspects

Complexity

✉ Timeseries

Choose data for simplified view of heterogeneity

Please choose a thing Applications

Heterogeneity measure for Applications Vendor

Please choose a Relation Interfaces

Heterogeneity measure for Interfaces Implementation

Submit

		Thing: Applications			
		n	H: Vendor	Rel: Interface	
				n	H: Implementation
Domäne	33	0,37	326	1,29	
Domäne	113	2,14	1.028	1,2	
Domäne	90	1,53	823	0,99	
Domäne	92	2,31	1.534	1,05	
Domäne	196	3,13	1.362	1,17	
Domäne	47	1,67	662	1,16	
Domäne	38	2,98	47	1,03	
Domäne	103	3,37	453	1,2	

Abbildung 58: Ausprägung und Darstellung der Kennzahlen in IT-COM_3⁹²

Die Herangehensweise bei der Analyse ist demnach deutlich unterschiedlich zu der des ersten IT-Artefaktes. Anstatt den Nutzer mit einer Vielzahl von Kennzahlen visuell und kognitiv zu „überfluten“, bestimmt die Fragestellung die darzustellenden Komplexitätskennzahlen:

„Das spannende ist hier nun, dass man sich aus der riesigen Tapete Einzelfragestellungen herauspicken kann. Ich schaue nur auf Programmiersprachen, nur auf Daten-

⁹² Bei den der Abbildung zugrunde liegenden Daten handelt es sich um Testdaten.

banksysteme und und und... Hier kann ich schön eine isolierte Fragestellung betrachten, ohne von der Gesamtwolke erschlagen zu werden.“ (Abteilungsleiter eines Teilbereiches der Architektur Bank A)

Der Schritt von den fest vorgegebenen Kennzahlen, hin zu einer – auf die aktuell betrachtete Fragestellung angepasste – frei ausprägbaren Kennzahl wurde hier als besonders wertvoll empfunden:

„Das ist das Generische an dem Tool. Du kannst es nun als Werkzeugkasten verwenden für nahezu beliebige Fragestellungen.“ (Enterprise Architect Bank A)

Hierin besteht auch eine wesentliche Änderung in der Herangehensweise: Nicht die bestehende Menge der Kennzahlen gibt vor, welche Fragen beantwortet werden, sondern die Kennzahl kann aus einer Fragestellung heraus ausgeprägt werden.⁹³ Als mögliche Fragestellung zeigte sich im Zuge eines Workshops u. a. die Untersuchung, in welchen Bereichen eine Datenbankkonsolidierung besonders sinnvoll erscheint. Hierfür würde der Fokus auf die Datenbankheterogenität gelegt. Weitere interessante Fragestellungen wurden in der Identifikation von Sourcing-Optionen (durch Betrachtung von Bereichen in denen die Programmiersprachen besonders homogen sind) oder von Möglichkeiten zur Reduktion der Komplexität der IT-Landschaft (z. B. durch Analyse der Schnittstellen) gesehen.

Wie bereits oben angeführt, basieren die Kennzahlen auf dem in Kapitel 5 eingeführten holistischen Ansatz zur Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen. In IT-COM_3 wird dem Nutzer zunächst als Heterogenitätsmaß die Entropie angezeigt, bei der es sich (wie ausführlich in Unterkapitel 5.6 diskutiert) um eine sogenannte Dual-Konzept Maßzahl handelt (vgl. in Abbildung 58 die Spalten „H: Vendor“ und „H: Implementation“). Der Nutzer kann sich an dieser Stelle einfach auch die weiteren definierten Maßzahlen für die Evenness und Richness anzeigen lassen, um die vorliegende Heterogenität und deren Ursache besser verstehen zu können (vgl. Abbildung 59). Auch die Fokussierung auf eine einheitliche Form der Berechnung wurde als Mehrwert empfunden:

„Auf mich wirkt es nun deutlich konzentrierter. Diese 22 Kennzahlen der Version 1.0 sind einfach sehr erschlagend. Die Konzentration auf das Entropie-Maß und die einzelnen Ausprägungen entsprechend des Modells, das erscheint mir konzentrierter und damit auch deutlich handhabbarer.“ (Principal Architect)

Gleichzeitig bleiben die Möglichkeiten des Drill-Down zur Unterstützung des Nutzers bei der Identifikation der Ursachen der Komplexität erhalten (vgl. Abbildung 59).

⁹³ Hierbei ist zu beachten, dass die Menge der erstellbaren Kennzahlen immer durch die Menge der vorhandenen Daten im Unternehmen sowie deren Pflegezustand limitiert wird.

		Thing: Applications				
		n	H: Vendor			Rel: Interface
			Entropy	Evenness	Richness	n
Domäne	M	33	0,37	0,53	2	326
Subdomäne	M	33	0,37	0,53	2	326
Funktionale Einheit		10	0	0	1	108
Funktionale Einheit		16	0	0	1	160
Funktionale Einheit		7	0,68	0,99	2	58

Abbildung 59: Kennzahlen und Drill-Down in IT-COM_3⁹⁴

Die Diskussionen während der BIE-Zyklen 1 und 2 führten dabei zur Einführung des neuen Designprinzips *Anpassbarkeit* (DP 11): „IT-COM_3 stellt dem Nutzer an spezifische Fragestellungen anpassbare Ausprägungen von Kennzahlen bereit.“ Dies wurde auch durch einen der IT-Architekten explizit in den Vordergrund gestellt.

„Die neu gewonnene Flexibilität und diese Toolbox-Eigenschaft des Systems sind für mich auch ein neues Designprinzip.“ (Enterprise Architect Bank A)

6.7.2.2.3 Integration des Designprinzips „Visualisierung“

Neben der zuvor eingeführten zusätzlichen Designprinzipien bestätigte sich im Zuge des dritten ADR-Projektes auch das im Rahmen des zweiten Projektes mit Bank B emergierte Designprinzip *Visualisierung*. Steht bei dem in Sektion 6.7.2.2.2 eingeführten Designprinzip *Anpassbarkeit* die Ausgestaltung der Kennzahl im Sinne einer Ausprägung einer generischen Kennzahl für eine spezifische Fragestellung im Fokus, so ist das in dieser Sektion diskutierte Designprinzip eine logische Folge des bereits bestehenden Designprinzips der „Offenheit“. Im Rahmen des ersten ADR-Projektes erweiterte sich die Zielgruppe des IT-Artefaktes von den IT-Architekten hin zu allen Mitarbeitern, die Entscheidungen mit einem Einfluss auf die IT-Architektur treffen. Bei dem Designprinzip *Offenheit* (DP 5) wurde jedoch nicht berücksichtigt, dass Mitarbeiter in unterschiedlichen Aufgabenbereichen bzw. Stufen des Managements eine unterschiedliche Form der Darstellung benötigen. In der Diskussion mit dem Abteilungsleiter der IT-Architekten der Bank A erwies sich dieser Punkt ebenfalls als äußerst relevant.

„Durch diese Zahlen-Sicht kann man sich wunderbar durchfräsen, wenn man Experte ist, aber wenn ich unvoreingenommen auf das Dashboard schaue, dann sehe ich den Wald vor lauter Bäumen nicht.“ (Abteilungsleiter eines Teilbereiches der Architektur Bank A)

Durch den bereits in der vorangegangenen Sektion erläuterten – an der Fragestellung orientierten – Aufbau der in IT-COM_3 dargestellten Kennzahlen, wurde die Menge unterschiedlicher Kennzahlen, die sich nicht auf die aktuelle Fragestellung beziehen, deutlich reduziert. Entgegen der Umsetzung in IT-Artefakt IT-COM_2 – im Kontext der Bank B – wurde in IT-COM_3 auf Erläuterungen der Zusammenstellung der Kennzahl anhand eines

⁹⁴ Bei den der Abbildung zugrunde liegenden Daten handelt es sich um Testdaten.

Meta-Modells oder eine textuelle Interpretationshilfe verzichtet. Dies wurde hier als nicht erforderlich angesehen, da der Nutzer die Kennzahl selbst ausprägt (vgl. Abbildung 58).

Zur weiteren visuellen Unterstützung wurde, analog zur Modularisierung, eine auf Balkendiagrammen basierende Darstellung der Kennzahlen gewählt. Abbildung 60 zeigt beispielhaft die Visualisierung der Verteilung unterschiedlicher Schnittstellen-Implementierungstypen der einer funktionalen Einheit zugeordneten Anwendungen.

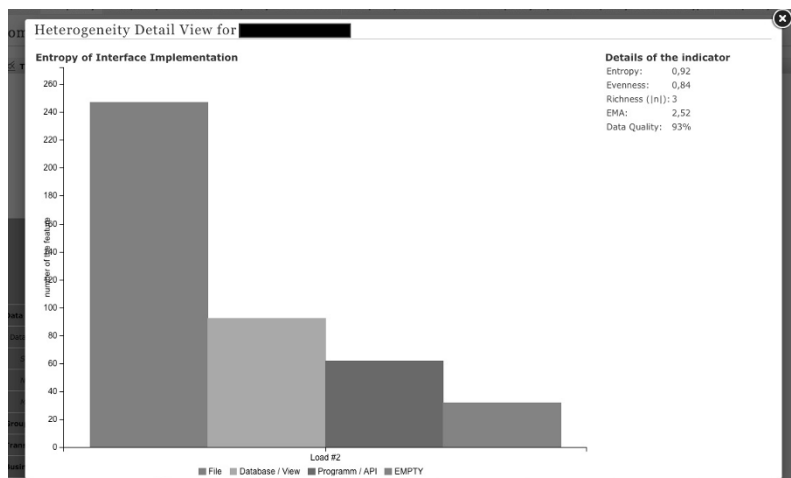


Abbildung 60: Visualisierung der Kennzahlen in IT-COM_3

Zusätzlich zu den Balkendiagrammen erhält der Nutzer Informationen bzgl. der berechneten Kennzahlen. Darüber hinaus hat der Nutzer in IT-COM_3 die Möglichkeit, beliebig ausgeprägte Kennzahlen anhand eines Scatter Plots miteinander zu kombinieren, um evtl. bestehende Korrelationen zu erkennen (vgl. Abbildung 61).

Select your data for scatter plot

Entry node: [redacted]

Data Combination: Applications: Vendor

Measure: Richness

Data for x-axis

Data Combination: Interface: Protocol

Measure: Richness

Data for y-axis

Submit

Abbildung 61: Auswahl der Dimensionen des Scatter Plots in IT-COM_3

War eine mögliche Ursache einmal ausgemacht und der IT-Architekt wollte eine Empfehlung für eine bestimmte Maßnahme ausarbeiten, fehlte es ihm an der Möglichkeit in IT-COM, den Zusammenhang identifizierter Ausprägungen der berechneten Komplexität untereinander oder mit Zielgrößen wie Kosten und Incidents darzustellen. IT-COM_3 bietet dem Architekten die Möglichkeit zusätzlicher Auswertungen. In frei ausprägbaren x-y-Scatter Plots können beliebige Paare an Kennzahlen miteinander verglichen werden (vgl. Abbildung 62 für einen beispielhaften Scatter Plot). Jeder der Knoten stellt dabei eine der funktionalen Einheiten dar und die Größe des Knotens symbolisiert die Anzahl der jeweils zugeordneten Anwendungen.

Die Stärke dieser Form der Visualisierung liegt dabei in der Freiheit des Architekten, jede der Kennzahlen mit anderen zu plotten und die Zusammenhänge so aufbereiten zu können, dass diese im Zuge der internen Kommunikation (z. B. im Rahmen einer internen Präsentation) verwendet und von Dritten einfach verstanden werden können.

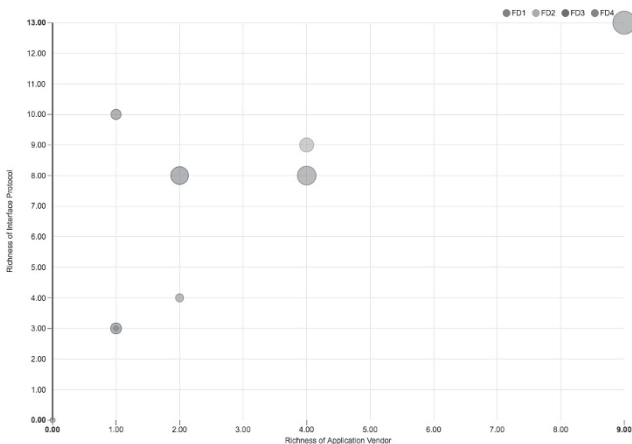


Abbildung 62: Beispielhafter Scatter Plot in IT-COM_3

Einen weiteren Kritikpunkt des ersten IT-Artefaktes IT-COM stellte die fehlende Möglichkeit des direkten Zugriffs auf die Datenbasis dar. Möchte ein Architekt die Ursachen eines identifizierten Grades an Komplexität untersuchen, so besteht häufig die Notwendigkeit, neben aggregierten Kennzahlen auch die Daten im Detail zu sichten. Bislang waren diese nicht über das IT-Artefakt verfügbar und mussten aufwändig aus gesonderten Systemen mittels spezieller Reporting-Funktionen generiert werden. Dies setzte zudem auch den Zugriff auf die hierfür erforderlichen Systeme voraus. Um diesen Medienbruch zu umgehen, versorgt IT-COM_3 den IT-Architekten mit Detailinformationen der betrachteten Komponenten (vgl. Abbildung 63).

		Application								Database				Operating Systems			
		Vendor	Programming Language	Product Type	Interface Intern		Interface Extern		Responsibility Intern	Responsibility Extern	Type	Class	Type	Version			
					In	Out	In	Out									
			Hyperion Esscomand; PL/1; C#, .NET, Uni- Shell; JCL, JAVA, SAS; SQL	Eigenentwicklung	0	0	0	0	1	1	MS SQL; Oracle; Sybase; Hyperion Essbase; DB2	Unix; Windows	Windows	Windows 2003			
			Hyperion Esscomand; PL/1; C#, .NET, Uni- Shell; JCL, JAVA, SAS; SQL	Eigenentwicklung	0	0	2	3	1	2	MS SQL; Oracle; Sybase; Hyperion Essbase; DB2	z/OS	IBM	IBM z/OS			
			Hyperion Esscomand; PL/1; C#, .NET, Uni- Shell; JCL, JAVA, SAS; SQL	Eigenentwicklung	0	0	0	0	1	1	MS SQL; Oracle; Sybase; Hyperion Essbase; DB2	Windows	Windows	Windows 2003			
			ASAP; Accelerated Java für ASAP-Developer; ASAP Objects; J2 Development with ASAP Web Dynpro; Advanced Web Dynpro for ASAP	Kaufprodukt	0	0	0	0	1	15	Oracle	Unix	SOLARIS	SOLARIS			

Abbildung 63: Detailinformationen in IT-COM_3

Zu dem Anwendungsfeld dieser Funktion äußerte sich ein IT-Architekt wie folgt:

„Wenn man z. B. die Datenbanken analysiert, dann muss man eben manchmal wissen, welche das denn nun genau sind. Dass man dies wirklich in dem Detail-View direkt sehen kann, ohne nochmal das Medium wechseln zu müssen, um dann in [System A] einen entsprechenden Report zu ziehen, das ist klasse. Weil in [System A] bekommt man die Information auch nicht direkt. Man kann die einzelnen Applikation sehen, aber nicht gebündelt auf alle Anwendungen einer funktionalen Gruppe zugreifen. In IT-COM_3 nun schon. Alleine das ist für die Suche ein echter Mehrwert. Für die Kollegen, die sich mit solchen Fragestellungen beschäftigen, für die ist das sehr hilfreich.“ (Principal Architect)

Die angeführten Überlegungen führten auch für den Kontext von Bank A zur Einführung des instanziierten Designprinzips *Visualisierung* (DP8): „IT-COM_3 stellt dem Benutzer verschiedene Darstellungsformen der relevanten Informationen bereit.“

6.7.2.2.4 Integration des Designprinzips „Simulation“

In den instanziierten und abstrakten Designprinzipien des ersten ADR-Projektes beinhalteten die Designprinzipien *Nachvollziehbarkeit von Veränderung im Zeitverlauf* (DP 3') und *Offenheit* (DP 5) implizit bereits einen Aspekt, der sich erst im Verlauf des zweiten ADR-Projektes mit der Bank B explizit mit dem Designprinzip *Simulation* (DP 9) in einem eigenen Designprinzip manifestierte. Die Möglichkeit, Veränderungen an der IT-Architektur im Zeitverlauf nachzuvollziehen und das IT-Artefakt gleichzeitig zur Entscheidungsunterstützung bei anstehenden Eingriffen in die IT-Architektur zu nutzen, impliziert die Möglichkeit, eine Änderung der IT-Architektur testen zu können.

Im Rahmen des zweiten BIE-Zyklus bestätigte sich das Erfordernis einer solchen Funktion auch im dritten ADR-Projekt. Die Architekten der Bank äußerten sich hierzu wie folgt:

„Die Szenarien sind wichtig und werden in Zukunft sicher auch immer wichtiger werden. Die Szenarien setzen aber natürlich auch voraus, dass du über einen gewissen Planungsstand verfügst, um überhaupt die Parameter in einem Zukunftsszenario zu verändern.“ (Enterprise Architect Bank A)

„Dem alten CIO war dies enorm wichtig. Der wollte genau wissen, welche Lösungsvariante die beste ist. Das wurde damals in Excel gemacht.“ (Principal Architect)

In dem IT-Artefakt IT-COM_3 können Architekten beliebige Komponenten löschen, verändern und hinzufügen (vgl. Abbildung 64). Die durchgeführten Änderungen schlagen sich entsprechend auf die Berechnung der Komplexitätskennzahlen nieder. Hierdurch hat der Architekt die Möglichkeit, die Effekte beliebiger Eingriffe in die IT-Architektur zu simulieren, um die Veränderung der Kennzahlen zur Entscheidungsunterstützung heranzuziehen.

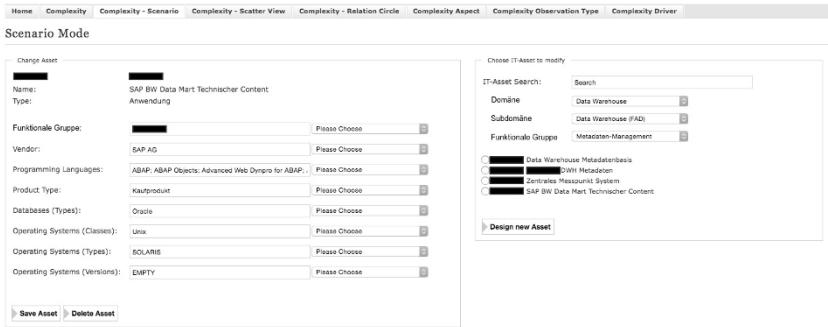


Abbildung 64: Simulation in IT-COM_3

Demzufolge wurde auch für den Kontext von Bank A ein instanziiertes Designprinzip *Simulation* (DP 9) eingeführt: „IT-COM_3 stellt dem Benutzer die Möglichkeit zur Simulation der Auswirkungen von Veränderungen der IT-Architektur bereit.“

6.7.2.3 Instanziierte Designprinzipien nach den zusätzlichen BIE-Zyklen

In der folgenden Tabelle 34 werden die Veränderungen der bestehenden Instanzierungen der Designprinzipien des ersten ADR-Projektes sowie die Instanzierungen der in den beiden folgenden ADR-Projekten neu hinzugekommenen Designprinzipien vorgestellt.

Tabelle 34: Instanziierte Designprinzipien im Rahmen des dritten ADR-Projektes

Designprinzip	Instanzierungen der Designprinzipien in der Bank
DP 1: Anzahl und Heterogenität der Komponenten	Zusätzlich zu dem bereits in Tabelle 28 eingeführten instanziierten Designprinzip DP 1 stellt das IT-Artefakt IT-COM_3 die Anzahl und Heterogenität unterschiedlicher Ausprägungen von Komponenten bereit.
DP 2: Anzahl und Heterogenität der Beziehungen	Zusätzlich zu dem bereits in Tabelle 28 eingeführten instanziierten Designprinzip DP 2 stellt das IT-Artefakt IT-COM_3 die Anzahl und Heterogenität unterschiedlicher Ausprägungen von Beziehungen bereit.
DP 8: Visualisierung	IT-COM_3 stellt dem Benutzer verschiedene Darstellungsformen der relevanten Informationen bereit. Hierdurch erhalten auch Mitarbeiter, die sich bislang noch nicht mit dem Informationssystem beschäftigt haben, ein Grundverständnis für die bereitgestellten Daten und verschiedene Formen der Ergebnisse für die Weiterverwendung im Zuge der internen Kommunikation.
DP 9: Simulation	IT-COM_3 stellt dem Benutzer die Möglichkeit zur Simulation der Auswirkungen von Veränderungen der IT-Architektur bereit.
DP 10: Modularität	IT-COM_3 erlaubt die Beurteilung der Güte der Modularität des betrachteten Systems.
DP 11: Anpassbarkeit	IT-COM_3 stellt dem Nutzer an spezifische Fragestellungen anpassbare Ausprägungen von Kennzahlen bereit.

6.7.2.4 Abstrakte Designprinzipien nach den zusätzlichen BIE-Zyklen

Wie bereits in Unterabschnitt 6.6.2.4 beschränkt sich die Darstellung der abstrahierten Designprinzipien an dieser Stelle auf die während der Durchführung des dritten ADR-Projektes zusätzlich emergierten Designprinzipien. Daher handelt es sich bei der folgenden Tabelle 35 um eine Ergänzung der bereits eingeführten abstrakten Designprinzipien.

Tabelle 35: Aus den Forschungsergebnissen abgeleitete abstrakte Designprinzipien

Design- prinzip	Beschreibung/Statement: Für eine Planung und Kon- trolle von Komplexität...	Begründung	Quelle
DP 10: Modu- larität	...sollte das IT-Artefakt Infor- mationen zur Beurteilung der Angemessenheit der Modu- larität der IT-Architektur be- reitstellen.	Eine nicht angemessene Modularität der IT-Architektur erschwert die orga- nisationale Abstimmung und die fach- liche Umsetzung bei Änderungen der IT-Architektur.	Ethiraj & Levinthal (2004), Simon (1962); Emergiert in BIE-Zyklus 1 (vgl. Sektion 6.7.2.2.1)
DP 11: Anpass- barkeit	...sollte das IT-Artefakt an spezifische Fragestellungen im Rahmen des Komplexitäts- managements anpassbar sein.	Wechselnde Fragestellungen erfordern situativ unterschiedliche Ausprägungen von Kennzahlen. Das IT-Artefakt sollte nicht der limitierende Faktor sein.	Emergiert in BIE- Zyklus 1 (vgl. Sekti- on 6.7.2.2.2)

6.8 Zusammenfassung und Diskussion der Ergebnisse

In diesem Kapitel wurde die fünfte Forschungsfrage der vorliegenden Dissertation bearbeitet:

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

Im Rahmen eines Forschungsprojektes, bestehend aus drei aufeinander folgenden ADR-Projekten in Kooperation mit zwei Banken, wurden Informationssysteme (IT-Artefakte) zur Planung und Kontrolle von Komplexität in IT-Architekturen entwickelt und in diesem Kontext sowohl instanziierte als auch abstrahierte Designprinzipien abgeleitet. Im folgenden Abschnitt 6.8.1 werden die aus der Durchführung des Forschungsprojektes resultierenden abstrakten Designprinzipien zunächst noch einmal konsolidiert dargestellt und die theoretischen Implikationen des Forschungsprojektes diskutiert. Im Anschluss daran werden in Abschnitt 6.8.2 die praktischen Implikationen anhand der beobachteten Konsequenzen des Einsatzes der IT-Artefakte im organisationalen Kontext der beiden Banken aufgezeigt. Abschnitt 6.8.3 beschließt das Unterkapitel mit einer Diskussion der Limitationen sowie des weiteren Forschungsbedarfs.

6.8.1 Theoretische Implikationen

Die im Zuge der ADR-Projekte implementierten Artefakte IT-COM_2 und IT-COM_3 stellen Lösungen der spezifischen Probleme der beiden teilnehmenden Banken dar. Diese Artefakte beziehen sich auf das in Unterkapitel 6.3 eingeführte Problem „Planung und Kontrolle der

Komplexität der IT-Architektur der Bank“ als Instanzen der Problemklasse „*Planung und Kontrolle von Komplexität in IT-Architekturen*“ im Kontext der Bank A bzw. der Bank B. Da es sich bei IT-COM_2 und IT-COM_3 um Software und somit real existierende Artefakte für jeweils einen spezifischen Kontext handelt, stellen diese eine sogenannte „Level 1 contribution“ (Gregor & Hevner 2013) dar (vgl. Abbildung 65). Gleiches gilt für die im Zuge der Durchführung der ADR-Projekte instanziierten Designprinzipien, welche die Basis der abstrahierten Designprinzipien bilden. Die abstrahierten Designprinzipien (vgl. Tabelle 36 und Tabelle 37) repräsentieren eine sogenannte naszierende Designtheorie und damit eine „Level 2 contribution“ (Gregor & Hevner 2013).

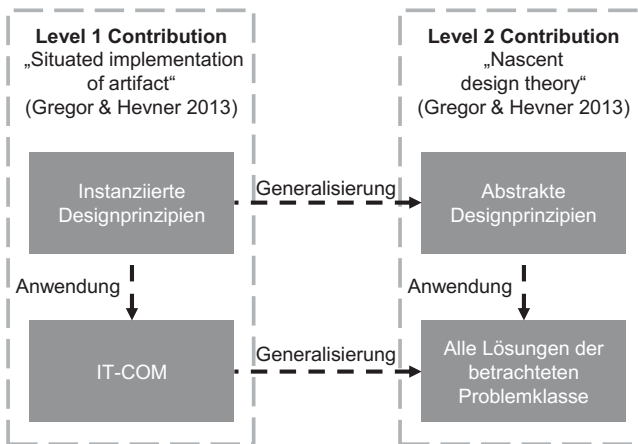


Abbildung 65: Beiträge des Forschungsprojektes⁹⁵

Wie bereits erläutert, handelt es sich bei IT-COM_2 und IT-COM_3 um Lösungen für den spezifischen Kontext der jeweiligen Bank und damit zwei Repräsentanten der Menge von Lösungen für die Problemklasse „*Planung und Kontrolle von Komplexität in IT-Architekturen*“. Um alle Repräsentanten von Lösungen dieser Problemklasse abdecken zu können, werden die instanziierten Lösungen generalisiert. Hierzu werden die ermittelten instanziierten Designprinzipien abstrahiert und somit eine Menge abstrakter Designprinzipien vorgeschlagen. Diese Menge generalisierter Designprinzipien (vgl. Tabelle 36 und Tabelle 37) ist die Antwort auf die in diesem Kapitel behandelte Forschungsfrage, welchen Designprinzipien ein Informationssystem folgen sollte, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt.

Im Hinblick auf die Theoriebildung unterscheiden Gregor & Jones (2007) zwischen (a) Instanzierungen (real existenten Artefakten), repräsentiert durch ein instanziiertes Produkt oder eine Methode, und (b) Theorien (abstrakten Artefakten), wie u. a. Designtheorien oder -prinzipien. Aus dieser Perspektive stellen sowohl die implementierten IT-Artefakte (IT-COM_2 und IT-COM_3) als auch die instanziierten Designprinzipien (vgl. Tabelle 28, Tabel-

⁹⁵ Vgl. hierzu auch Schütz et al. (2013a).

le 31 und Tabelle 34) real existente, instanziierte Artefakte dar. Von den abstrakten Designprinzipien (vgl. Tabelle 36 und Tabelle 37) existiert hingegen kein stoffliches Abbild in der realen Welt, sondern sie stellen ein abstraktes Artefakt bzw. Elemente einer Designtheorie nach Gregor & Jones (2007) dar.

Tabelle 36: Aus dem Forschungsprojekt abgeleitete abstrakte Designprinzipien (1)

Designprinzip	Beschreibung/Statement: Für eine Planung und Kontrolle von Komplexität...	Begründung
DP 1: Anzahl und Heterogenität der Komponenten	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Komponenten des Systems bereitstellen.	Die Anzahl und Heterogenität der Komponenten sind wesentlicher Bestandteil der Konzeptualisierung von Komplexität in IT-Architekturen.
DP 2: Anzahl und Heterogenität der Beziehungen	...sollte das IT-Artefakt Informationen über die Anzahl und die Heterogenität der Beziehungen des Systems bereitstellen.	Die Anzahl und Heterogenität der Beziehungen sind wesentlicher Bestandteil der Konzeptualisierung von Komplexität in IT-Architekturen.
DP 3': Nachvollziehbarkeit von Veränderung im Zeitverlauf	...sollte das IT-Artefakt dem Benutzer die Nachvollziehbarkeit von Veränderungen der Komplexität der IT-Architektur im Zeitverlauf ermöglichen (z. B. in Form von Zeitreihen).	Informationen bzgl. der Veränderung der Komplexität der IT-Architektur ermöglichen die Nachvollziehbarkeit der Wirkung von Maßnahmen des Architekturmanagements.
DP 4: Ebenen der Architektur	...sollte das IT-Artefakt die Architektur-ebenen Datenarchitektur, Anwendungsarchitektur, Infrastrukturarchitektur und Geschäftsprozessarchitektur berücksichtigen.	Die EAM Wissensbasis sieht eine differenzierte Betrachtung der einzelnen Ebenen einer Architektur im Zuge von Designentscheidungen vor.
DP 5: Offenheit	...sollte der Zugriff auf das IT-Artefakt und somit Architekturinformationen und die Komplexitätskennziffern nicht auf einen spezifischen Nutzerkreis beschränkt werden. Alle Mitarbeiter, die Entscheidungen mit Einfluss auf die IT-Architektur treffen, sollten über einen Zugriff verfügen.	Informationen, die für alle Interessengruppen im Unternehmen verfügbar sind, erhöhen die Qualität des Managements der Komplexität der IT-Architektur.
DP 6: Abstraktionsgrad und Drill-Down	...sollte das IT-Artefakt eine Analyse der IT-Architektur auf verschiedenen Abstraktionsleveln ermöglichen.	Informationen auf verschiedenen Abstraktionsleveln ermöglichen die Identifikation der Ursachen der Komplexität sowie die Ableitung von Handlungsbedarf.
DP 7: Datenqualität	...sollte das IT-Artefakt den Nutzer auf die den Kennzahlen zugrunde liegende Datenqualität aufmerksam machen.	Bewusstsein bezüglich der Richtigkeit und Verlässlichkeit der Informationen erhöht die Qualität der Entscheidungen.

Tabelle 37: Aus dem Forschungsprojekt abgeleitete abstrakte Designprinzipien (2)

Designprinzip	Beschreibung/Statement: Für eine Planung und Kontrolle von Komplexität...	Begründung
DP 8: Visualisierung	...sollte das IT-Artefakt – entsprechend der individuellen Anforderungen der Benutzer – verschiedene Darstellungsformen der relevanten Informationen bereitstellen.	Adäquate Darstellungsformen der bereitgestellten Kennzahlen und ergänzende Informationen erleichtern das Verständnis sowie die Kommunizierbarkeit entscheidungs-relevanter Informationen.
DP 9: Simulation	...sollte das IT-Artefakt eine Simulation der Auswirkungen von Veränderungen der IT-Architektur ermöglichen.	Die Simulation von Veränderungen an der IT-Architektur ermöglicht dem Benutzer die Überprüfung unterschiedlicher Handlungsalternativen und deren Auswirkungen auf die Komplexität der IT-Architektur.
DP 10: Modularität	...sollte das IT-Artefakt Informationen zur Beurteilung der Angemessenheit der Modularität der IT-Architektur bereitstellen.	Eine nicht angemessene Modularität der IT-Architektur erschwert die organisationale Abstimmung und die fachliche Umsetzung bei Änderungen der IT-Architektur.
DP 11: Anpassbarkeit	...sollte das IT-Artefakt an spezifische Fragestellungen im Rahmen des Komplexitätsmanagements anpassbar sein.	Wechselnde Fragestellungen erfordern situativ unterschiedliche Ausprägungen von Kennzahlen. Das IT-Artefakt sollte nicht der limitierende Faktor sein.

Neben den vorgeschlagenen abstrakten Designprinzipien und den entwickelten Artefakten IT-COM_2 und IT-COM_3, die eine Lösung zu dem Problem der jeweils betrachteten Bank darstellen, wurde die in Unterkapitel 3.2 eingeführte systemtheoretische Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen⁹⁶ validiert. Durch die kontinuierliche Evaluierung während der BIE-Zyklen wurden die Artefakte IT-COM_2 und IT-COM_3, die Designprinzipien und die Konzeptualisierung fortwährend überprüft. Die resultierenden Designprinzipien DP 1, 2 und 3' heben die ganzheitliche, systemtheoretische Perspektive auf IT-Architekturen hervor und DP 4 die Eignung dieser Perspektive für eine Anwendung auf die betrachteten Ebenen einer IT-Architektur. Die Ergebnisse dieses Forschungsprojektes weisen demnach auf die Angemessenheit der vorgeschlagenen Konzeptualisierung von Komplexität in IT-Architekturen für die betrachtete Problemklasse hin.

Die Emergenz der Designprinzipien DP 5, DP 6, DP 8, DP 9 und DP 11 weist darauf hin, dass ein Komplexitätsmanagement in IT-Architekturen nicht ausschließlich ein relevantes Thema für die IT-Architekten im Unternehmen darstellt, sondern alle Mitarbeiter betrifft, deren Entscheidungen und Handeln einen Einfluss auf die Ausgestaltung der IT-Architektur haben. In

⁹⁶ Die Validierung beschränkte sich hierbei auf das Subsystem der IT-Architektur. Die Komplexität der Geschäftsprozessarchitektur war nicht Gegenstand der Untersuchung.

der IS-Literatur werden theoretische Profile einer IT-Funktion in Unternehmen vorgeschlagen. Guillemette & Paré (2012) führen den „Architecture Builder“ (vgl. Abschnitt 2.2.6) als die Funktion an, deren Aufgabe es ist, die IT-Infrastruktur zur Unterstützung der Geschäftsprozesse zu gestalten und zu managen sowie deren Komplexität zu reduzieren (Guillemette & Paré 2012, S. 532). Cross et al. (1997) weisen die Aufgabe, die Komplexität der Architektur zu reduzieren, dem sogenannten „Infrastructure Planner“ zu. Über diese Ansätze hinaus weisen die Ergebnisse dieses Forschungsprojektes darauf hin, dass das Management der Komplexität der IT-Architektur eine Aufgabe darstellt, die weitaus mehr unterschiedliche Interessengruppen betrifft. Demnach sollten nicht ausschließlich IT-Architekten, sondern Mitarbeiter aus verschiedenen Bereichen an den entsprechenden Managementprozessen beteiligt werden.

Designprinzip DP 10 weist darauf hin, dass die Struktur des Systems ebenfalls eine wesentliche Rolle bei dem Management der Komplexität von IT-Architekturen spielt. In Anlehnung an die durch Ethiraj & Levinthal (2004), Simon (1962) und Simon (1996) eingeführte Perspektive auf die Modularität eines Systems stellte sich während der Durchführung des Forschungsprojektes heraus, dass die Ausgestaltung des Systems und seiner Subsysteme – im Hinblick auf die Zuordnung der Designelemente und deren Interaktion – einen relevanten Einfluss auf die Durchführbarkeit von Änderungen und damit auf die Flexibilität eines Systems hat. Nachdem im Rahmen der in Kapitel 4 vorgestellten qualitativ-empirischen Studie die IT-Standardisierung und die Migration in die Komplexitäts-Perspektive integriert wurden, zeigt sich in diesem Forschungsprojekt, dass sich auch Modularität in dieser Perspektive wiederfindet. Dies deutet darauf hin, dass es sich bei der Komplexitäts-Perspektive um eine übergeordnete Perspektive auf das System „IT-Architektur“ handelt, in die bestehende Ansätze aus der Literatur integriert werden können.

6.8.2 Praktische Implikationen

Die entwickelten Artefakte IT-COM_2 und IT-COM_3 stellen Lösungen für die formulierten Probleme der Banken A bzw. B (vgl. Unterkapitel 6.3) dar. Die implementierten Informationssysteme für die „Planung und Kontrolle der Komplexität der IT-Architektur der Bank A“ bzw. die „Planung und Kontrolle der Komplexität der IT-Architektur der Bank B“ – basierend auf den vorgeschlagenen Designprinzipien – erlauben den IT-Architekten, die Aufgaben eines unternehmensweiten Managements der IT-Komplexität anzugehen.

Detaillierte praktische Implikationen können zudem aus den beobachteten Konsequenzen des Einsatzes von IT-COM im organisationalen Kontext der Banken gewonnen werden. In der folgenden Tabelle 38 werden diese für die Einführung der IT-Artefakte bei einer Auswahl an Endbenutzern (IT-Architekten, Projektmanagern, Anwendungsentwicklern sowie Entscheidung in der Bank) zusammengefasst.

Tabelle 38: Implikationen der BIE und IT-COM für die Banken A bzw. B

Design- prinzip	Implikationen für die Organisation	Beobachtet bei Bank...	
		A	B
DP 1-4	Die verwendete Konzeptualisierung von Komplexität in IT-Architekturen beeinflusste die Perspektive der IT-Architekten sowie der weiteren beteiligten Interessengruppen auf das Problem des Managements von Komplexität.	X	X
DP 5: Offenheit & DP 9: Simulation	Die Nutzer (u. a. Entscheider) haben die Möglichkeit, auf die Daten der aktuellen IT-Architektur zuzugreifen. Sie können den Einfluss von Änderungen an der IT-Architektur für einen Teilbereich oder die IT-Architektur als Ganzes überprüfen. Diese zusätzliche Transparenz hat das Potenzial, die Qualität der IT-Architekturentscheidungen mittel- und langfristig zu verbessern.	X	X
DP 6: Abstrak- tionsgrad und Drill- Down	Die Nutzer haben die Möglichkeit, die IT-Architektur aus verschiedenen Perspektiven und auf unterschiedlichen Abstraktionsleveln zu analysieren. Dies ermöglicht ihnen, den Einfluss von Änderungen auf kleine Subsysteme (z. B. ihre funktionale Einheit) oder auf ein größeres System (z. B. die Domäne) zu überprüfen. Diese wechselnde Perspektive erlaubt den IT-Architekten eine Diskussion, inwiefern ein lokales Optimum (auf Subsystemebene) einem globalen Optimum (z. B. auf Ebene des Gesamtsystems) zu bevorzugen ist.	X	X
DP 7: Daten- qualität	Die Einführung eines Informationssystems für die Planung und Kontrolle von Komplexität in IT-Architekturen erhöht den Bedarf hinsichtlich der Verfügbarkeit und Korrektheit der Daten für die Berechnung der Kennzahlen. Fehlerhafte oder unvollständige Daten stellen eine wesentlich weniger belastbare Basis für die Berechnung der Kennzahlen dar und erhöhen das Risiko suboptimaler oder sogar falscher Entscheidungen.	X	(X) ⁹⁷
DP 8: Visuali- sierung	Durch die Verwendung des Informationssystems stehen den Nutzern verschiedene Darstellungsformen der Architektur- und Komplexitätsinformationen zur Verfügung. Diese berücksichtigen unterschiedliche individuelle Erfahrungen im Themenfeld der Architektur sowie der strukturellen Komplexität. Hierdurch wird den diversen Nutzergruppen ein Zugang zu der Thematik gewährt sowie die Möglichkeit zur Integration dieser in praktische Aufgabenstellungen im Unternehmen gegeben. Die unterschiedlichen Formen der Visualisierung ermöglichen eine zielgruppenadäquate Weiterverwendung der Informationen.	X	X
DP 10: Modulari- tät	Die Nutzer können die Angemessenheit der Modularität der IT-Architektur hinsichtlich verschiedener Dimensionen beurteilen. Bei der Zuordnung der Anwendungen zu bestimmten funktionalen Einheiten – und somit auch zu den Subdomänen und Domänen – können der fachliche Kontext der Anwendung oder auch die Zuständigkeit der Mitarbeiter für die jeweilige Anwendung berücksichtigt und dadurch ein möglichst reiner Schnitt der Domänen realisiert werden.	X	-
DP 11: Anpass- barkeit	Die Nutzer können die für ihre Fragestellungen relevanten Kennzahlen frei ausprägen. Hierdurch besteht die Möglichkeit, z. B. die für ein Risikoreporting relevanten Kennzahlen zur Beurteilung der Komplexität der IT-Architektur festzulegen und die Berechnungen entsprechend durchzuführen.	X	(X) ⁹⁸

⁹⁷ Vgl. hierzu Sektion 6.6.2.2.1.⁹⁸ Im IT-Artefakt IT-COM_2 müssen diese individuell implementiert werden. Zwar lässt das zugrunde liegende Modell diese Ausprägung grundsätzlich zu, jedoch ist bislang keine programmseitige Unterstützung zur Ausprägung der Kennzahlen vorgesehen.

In beiden Organisationen hat die Verfügbarkeit eines Werkzeugs zur Quantifizierung das Thema des Komplexitätsmanagements von IT-Architekturen weiter vorangebracht. So untersucht Bank A, in welchen Bereichen des Unternehmens und in welchen Prozessen das IT-Artefakt für die Messung und Analyse der Komplexität im Kontext unterschiedlicher Fragestellungen sowie der anschließenden Planung und Durchführung möglicher Maßnahmen eingesetzt werden kann. Auch im Kontext der zunehmenden Anforderungen seitens der Bankenaufsicht kann der Einsatz des Informationssystems sehr wertvoll sein. Die Architekten führen an, dass die Bankenaufsicht IT-Risiken – neben Kreditrisiken oder Marktrisiken der Institute – mittlerweile als maßgebliche Risiken einer Bank ansieht. Hierzu zählt auch die zunehmend entstehende „unnötige IT-Komplexität“, die in den Gesprächen mit den Banken sehr kritisch gesehen wird.⁹⁹ Hier wird in Bank A untersucht, welche Fragestellungen und spezifischen Ausprägungen der Kennzahlen für ein IT-Risikoreporting besonders geeignet sind.

Auch bei Bank B soll untersucht werden, welche Kennzahlen typischerweise herangezogen werden sollten, wenn die Veränderung der Komplexität zu beurteilen ist. Damit soll verhindert werden, dass die Kennzahl genau so ausgeprägt wird, dass sie besonders gut zu der „gewünschten“ Entscheidung des Nutzers passt. Vielmehr soll eine möglichst objektive Bewertung die Grundlage zukünftiger Entscheidungen darstellen.

„Also wir sind jetzt gerade in der Diskussion, wie wir das Modell, das wir gemeinsam erarbeitet haben, verwenden können. Hierfür müssen wir die Dimensionen festlegen, an denen wir die Komplexität eines IT-Systems festmachen können. [...] Zudem darf auch niemand tricksen können, indem er z. B. die Anzahl der Anwendungen verändert und einfach Systeme oder Komponenten zusammenfasst und diese unter einer Applikation subsummiert und somit statt fünf nur noch eine Applikation vorliegt. Derartige Eingriffe müssen verhindert werden. Wenn die Organisation dies konterkariert, um besser dazustehen, dann macht der ganze Mechanismus keinen Sinn mehr. Daher muss vereinbart werden, woran Komplexität messbar gemacht werden soll.“ (Architekt Bank B)

Neben der Implementierung der Informationssysteme führte die Durchführung des Forschungsprojektes auch zu einer weiteren Verbreitung der Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen (vgl. Unterkapitel 3.2) in der Praxis, die ein einheitliches Verständnis des Problems der Komplexität und eine einheitliche Sprache gewährleistet.

⁹⁹ Besonders für viele Finanzdienstleister – deren Geschäftstätigkeit häufig auf historisch gewachsenen IT-Landschaften beruht – stellt die zunehmende Komplexität der IT-Landschaft ein wesentliches Risiko dar. Die Maßgaben der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) betonen neben den bankfachlichen Risiken, wie Adressenausfall-, Marktpreis- und Liquiditätsrisiken auch die Relevanz operationeller Risiken (BaFin 2012), welche sogenannte IT-Risiken umfassen (Kokert & Held 2013). Da eine zunehmende Komplexität der IT-Landschaft die Stabilität der IT-Systeme negativ beeinflussen kann, ist diese als ein IT-Risiko zu werten. Für derartige Risiken haben die Finanzinstitute angemessene Vorkehrungen zu treffen, die eine Beurteilung, Steuerung und Überwachung der Risikosituation zulassen (BaFin 2012).

6.8.3 *Limitationen und weiterer Forschungsbedarf*

Wie jedes Forschungsprojekt weist auch dieses gewisse Limitationen auf. Da der ADR-Ansatz häufig von einer besonders intensiven Zusammenarbeit mit einem Unternehmen geprägt ist, profitiert die Forschung auf der einen Seite von der tiefen Analyse innerhalb des organisationalen Kontextes, auf der anderen Seite bestehen jedoch die gleichen Limitationen wie die der Single-Case-Ansätze im Bereich der qualitativen Forschung. Obgleich grundsätzlich die Möglichkeit besteht, Ergebnisse einer Single-Case-Studie zu generalisieren (Lee & Baskerville 2003), wird eine Generalisierbarkeit durch die Replikation auf weitere organisationale Kontexte unterstützt (Eisenhardt 1989; Yin 2014). Um diese Problematik zu berücksichtigen, wurden die Ergebnisse der ersten ADR-Studie in Form eines weiteren ADR-Projektes in einem weiteren organisationalen Kontext repliziert. Hierdurch konnte der theoretische Beitrag des Forschungsprojektes gestärkt werden. Dennoch ist anzumerken, dass es sich bei beiden Unternehmen um Banken handelt, die sich somit in der gleichen Branche befinden. Da es sich – wie bereits bei der Einführung der Problemklasse in Unterkapitel 6.3 dargestellt – hierbei um eine Problemklasse handelt, die auch für Unternehmen anderer Branchen eine Rolle spielt, wäre eine Übertragung der Designprinzipien auf weitere organisationale Kontexte anderer Branchen interessant.

In der IS-Literatur existiert eine Reihe weiterer Designprinzipien im Themenfeld der Unternehmensarchitekturen (Greefhorst & Proper 2011; Haki & Legner 2013; Richardson et al. 1990). Eine Untersuchung, wie die in diesem Forschungsprojekt erarbeiteten Designprinzipien mit den bestehenden Designprinzipien im Kontext der Unternehmensarchitekturen zusammenhängen und inwiefern diese integriert werden können, wäre ein interessanter Ansatzpunkt für weitere Forschung.

Aus methodischer Perspektive erwies sich Scrum als eine für den ADR-Ansatz ausgesprochen passende Methode der Softwareentwicklung. Das agile Konzept und speziell die einzelnen Entwicklungs-Sprints unterstützten die Durchführung des Forschungsprojektes entsprechend der Grundideen des ADR-Ansatzes. Weitere Forschung könnte die Verwendung anderer Methoden der Softwareentwicklung in Kombination mit ADR untersuchen.

7 Fazit und Implikationen

Gegenstand der vorliegenden Dissertation war die Untersuchung des Phänomens der Komplexität von Unternehmens-/IT-Architekturen. Hierbei wurde der Fokus auf Komplexität als eine strukturelle Eigenschaft der Unternehmens-/IT-Architekturen gelegt und eine systemtheoretische Perspektive eingenommen. Im folgenden Unterkapitel 7.1 werden zunächst die durch die Anwendung unterschiedlicher Methoden erarbeiteten Ergebnisse der einzelnen Studien zusammengefasst. Anschließend werden in Unterkapitel 7.2 die wesentlichen theoretischen Beiträge diskutiert, wobei der Fokus hierbei auf den übergeordneten Beiträgen aus der Gesamtheit der durchgeführten Studien liegt. In Unterkapitel 7.3 werden daran anknüpfend die praktische Relevanz der Beiträge diskutiert und Handlungsempfehlungen abgeleitet. Abschließend werden in Unterkapitel 7.4 weitere Forschungsmöglichkeiten aufgezeigt.

7.1 Zusammenfassung

Wie in der in Kapitel 3 durchgeführten strukturierten Literaturrecherche gezeigt wurde, existiert in der aktuellen IS-Literatur keine holistische Konzeptualisierung struktureller Komplexität, die auf alle Ebenen einer Unternehmens-/IT-Architektur (vgl. Abschnitt 2.2.3 und 2.2.4) anwendbar ist. Diesem Punkt wurde durch die erste Forschungsfrage Rechnung getragen:

Forschungsfrage 1: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen konzeptualisiert werden?

Durch Analyse der im Rahmen der strukturierten Literaturrecherche (Unterkapitel 3.1) identifizierten und als relevant eingestuft 45 Publikationen, wurde in Unterkapitel 3.2 eine systemtheoretische Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen erarbeitet. Die strukturelle Komplexität wurde dabei als die Anzahl und die Heterogenität der Komponenten und Beziehungen eines Systems definiert, wobei sowohl die gesamte Unternehmens-/IT-Architektur als auch die einzelnen Architekturebenen als zu betrachtendes System herangezogen werden können (vgl. Abschnitt 3.2.3). Dieses Verständnis struktureller Komplexität bildete die Grundlage der im weiteren Verlauf der Dissertation durchgeführten Studien.

Die zweite und dritte Forschungsfrage zielten darauf ab, ein besseres Verständnis der Entstehung struktureller Komplexität in IT-Architekturen zu erlangen und die Auswirkungen struktureller Komplexität auf Zieleigenschaften einer IT-Architektur wie IT-Effizienz und IT-Flexibilität zu untersuchen:

Forschungsfrage 2: Wie entsteht strukturelle Komplexität von IT-Architekturen?

Forschungsfrage 3: Welche Auswirkungen hat die strukturelle Komplexität auf Zieleigenschaften der IT-Architektur?

Im Rahmen einer qualitativ-empirischen Fallstudie mit einer Unternehmensgruppe aus der chemischen Industrie (vgl. Kapitel 4) wurde eine Reihe von Auslösern und Mechanismen, die zu einer Entstehung struktureller Komplexität führen, identifiziert. Darüber hinaus wurde eine Menge von Zieleigenschaften einer IT-Architektur, die durch strukturelle Komplexität beeinflusst werden, ermittelt. Bei der Untersuchung der Entstehung struktureller Komplexität wurde konzeptionell eine Unterscheidung in *erforderliche* und *unnötige* Komplexität vorgenommen. Dies erlaubte die Betrachtung des Einflusses eingehender Anforderungen auf die erforderliche bzw. die sogenannte *minimale IT-Komplexität*. In einem nächsten Schritt konnte untersucht werden, warum es zu einer Abweichung von dieser minimalen IT-Komplexität in Form von *unnötiger IT-Komplexität* kommt. Im Kontext der durchgeführten Fallstudie entstanden derartige Abweichungen aufgrund verteilter Kompetenzen und Möglichkeiten bei der Auswahl und Umsetzung technischer Lösungen. Begünstigt wurde die Entstehung unnötiger Komplexität zudem durch die fehlende Architekturmanagement-Funktion und die unzureichende Dokumentation der bestehenden Architektur.

Anhand einer Reihe von IT-Projekten der Unternehmensgruppe konnte das Zusammenspiel zwischen den identifizierten Einflussfaktoren auf die minimale IT-Komplexität und den Mechanismen, die zu unnötiger Komplexität führen, veranschaulicht werden. Die projektspezifische Ausgestaltung dieser Einflussfaktoren und Mechanismen führte dabei zu einer unterschiedlichen Intensität des Einsatzes von Architekturmaßnahmen wie Standardisierung oder Migration. Dies schlug sich entsprechend in einer differierenden Wirkung auf IT-Effizienz und IT-Flexibilität nieder. Neben diesen Zielgrößen des Architekturmanagements (vgl. Unterabschnitt 2.2.5.2) wurde mit der Verfügbarkeit des Systems eine weitere relevante Zieleigenschaft des Systems identifiziert.

Um die Veränderung und Entwicklung der strukturellen Komplexität der IT-Architektur im Zeitverlauf beobachten zu können, bedarf es einer Möglichkeit, diese strukturelle Komplexität zu quantifizieren. Diese Problemstellung wurde in der vorliegenden Dissertation anhand der vierten Forschungsfrage bearbeitet:

Forschungsfrage 4: Wie kann die strukturelle Komplexität von Unternehmens-/IT-Architekturen quantifiziert werden?

Basierend auf der eingeführten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen wurde in Kapitel 5 eine Vorgehensweise zur Quantifizierung von Komplexität in Unternehmens-/IT-Architekturen entwickelt, die auf alle Ebenen (bzw. Subsysteme) einer Unternehmens-/IT-Architektur anwendbar ist. Diese basiert mit der *Entropie* auf einem gut erforschten Maß, das bereits in zahlreichen wissenschaftlichen und praktischen Bereichen zum Einsatz kommt. Ergänzend wurden mit der *Evenness* und der *Richness* zwei Konzepte aus dem Bereich der Biologie auf den Kontext der Komplexität von Unternehmens-/IT-Architekturen übertragen. Die Anwendbarkeit der sich komplementierenden Maßzahlen wurde anhand von drei Anwendungsbeispielen mit Unternehmen aus unterschiedlichen Branchen veranschaulicht.

Wie die in Kapitel 4 durchgeführte qualitativ-empirische Fallstudie bestätigt hat, ist die Dokumentation der IT-Architektur ein wesentlicher Bestandteil des Architekturmanagements (vgl. hierzu auch Schmidt & Buxmann (2011)). Die Dokumentation der bestehenden IT-Architektur alleine stellt für sich genommen jedoch noch keine ausreichende Unterstützung des Architekten bei dem Management der Komplexität der IT-Architektur dar. Geleitet durch die fünfte Forschungsfrage wurde untersucht, nach welchen Designprinzipien ein Informationssystem für das Management – speziell die Planung und Kontrolle – von Komplexität ausgestaltet sein sollte, um den Architekten bei dieser Aufgabe zu unterstützen:

Forschungsfrage 5: Welchen Designprinzipien sollte ein Informationssystem folgen, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt?

Im Zuge eines Forschungsprojektes, bestehend aus drei aufeinander folgenden Teilprojekten in Kooperation mit zwei Banken, wurden unter Verwendung der ADR-Methode (Sein et al. 2011) Informationssysteme entwickelt, die IT-Manager und -Architekten bei dem Management der Komplexität unterstützen. Parallel zur Entwicklung der Informationssysteme wurden in enger Abstimmung mit den beteiligten Architekten Designprinzipien formuliert. Diese beschreiben, wie das implementierte Informationssystem ausgestaltet sein sollte, um das betrachtete Problem „*Planung und Kontrolle von Komplexität in IT-Architekturen*“ zu lösen. Am Ende jedes Teilprojektes wurden die für die Kontexte der Banken abgeleiteten Designprinzipien abstrahiert, um diese auch auf weitere Kontexte und Instanzen der betrachteten Problemklasse anwenden zu können. Ausgehend von der eingeführten systemtheoretischen Konzeptualisierung von Komplexität in Unternehmens-/IT-Architekturen wurden im Rahmen des Forschungsprojektes zwei Prototypen sowie elf abstrakte Designprinzipien entwickelt (vgl. Tabelle 36 und Tabelle 37).

7.2 Wissenschaftlicher Beitrag der Arbeit

Der übergeordnete Beitrag der Arbeit liegt in der Einführung einer neuen Perspektive auf das Management von IT-Architekturen. Mit der Komplexitäts-Perspektive wird das Verständnis bzgl. des Managements von IT-Architekturen um einen in diesem Kontext bislang überraschenderweise lediglich am Rande beachteten Aspekt erweitert: Die Komplexität von Systemen. Der systemtheoretische Charakter dieser Perspektive erlaubt dabei eine Abstraktion der Elemente auf den verschiedenen Ebenen einer IT-Architektur. Dies ermöglicht eine Integration der Ebenen der IT-Architektur sowie die Übertragung der Ansätze und Denkweisen aus der Systemtheorie auf IT-Architekturen (vgl. hierzu u. a. die in Abschnitt 2.1.1 eingeführten Systemkonzepte nach Ropohl (2009) und die Arbeit von Simon (1996)). Auch die Komplexitäts-Perspektive weist im Zuge der durchgeführten Studien einen integrativen Charakter auf. So lassen sich bestehende Ansätze zu dem Management von IT-Architekturen wie die Transformation einer gegebenen Architektur (As-Is) in eine Zielarchitektur (To-Be) (Aier et al. 2012), die Standardisierung (Boh & Yellin 2006; Weitzel et al. 2006) und die Modularität (Baldwin & Clark 2000; Ethiraj & Levinthal 2004; Simon 1962) in die Komplexitäts-Perspektive integrieren und miteinander in Verbindung bringen (vgl. Kapitel 4 und 6). Im

Folgenden werden darauf aufbauend die wesentlichen theoretischen Beiträge der Arbeit diskutiert.

Die in Kapitel 3 durchgeführte Literaturrecherche hat gezeigt, dass in der IS-Literatur bereits eine Reihe von Arbeiten zum Thema der Komplexität existiert, diese jedoch nicht auf einem einheitlichen Verständnis des Konzeptes „Komplexität“ basieren. Analog zu den in Abschnitt 2.1.2 erläuterten Überlegungen von Bertalanffy (1968), die ihn zu der Einführung der allgemeinen Systemtheorie bewegten, ist auch in der IS-Forschung zu Komplexität die Tendenz zu erkennen, dass Forschungsergebnisse in einzelnen, spezifischen Teilbereichen existieren, diese jedoch nur schwer auf andere Kontexte übertragbar sind. Diesem Punkt widmete sich die erste Forschungsfrage der vorliegenden Dissertation. In Unterkapitel 3.2 wurde die erste holistische – auf alle Ebenen einer Unternehmens-/IT-Architektur anwendbare – systemtheoretische Konzeptualisierung struktureller Komplexität vorgeschlagen. Durch den abstrahierenden, systemtheoretischen Charakter der Konzeptualisierung kann diese für die Forschung in einzelnen Teilbereichen der IT-Architektur herangezogen werden. Dies erleichtert eine Überprüfung der Übertragbarkeit spezifischer Forschungsergebnisse auf die weiteren Teilbereiche der IT-Architektur. Gleichzeitig erlaubt der abstrakte Charakter der Konzeptualisierung deren Anwendbarkeit über die Grenzen einzelner Ebenen der IT-Architektur hinaus.

Der theoretische Beitrag der in Kapitel 4 vorgestellten qualitativ-empirischen Studie zur Bearbeitung der zweiten und dritten Forschungsfrage liegt neben der Identifikation der Ursachen und Auswirkungen struktureller Komplexität (vgl. Abschnitt 4.5.1) in der Entwicklung eines tieferen Verständnisses des Konzeptes der Komplexität. Die Studie zeigt dabei auf, dass Komplexität differenziert betrachtet werden sollte. Es wird deutlich, dass Komplexität nicht per se etwas Schlechtes ist, was es zu vermeiden oder zu reduzieren gilt, sondern dass ein gewisser Grad an Komplexität erforderlich und demzufolge erwünscht ist. Wie die qualitative Studie weiter aufzeigt, kann über den Grad erforderlicher Komplexität hinausgehende Komplexität einen Einfluss auf wirtschaftliche Kenngrößen nehmen und auf diese sowohl positiv als auch negativ wirken:

1) Eine zunehmende unnötige Komplexität kann zu einer abnehmenden Effizienz des Einsatzes von IT sowie zu einer abnehmenden Reaktionsfähigkeit bei der Durchführung von Änderungen führen. Hier sind vor allem historisch gewachsene Legacy-Systeme zu nennen, die aus einer Vielzahl von Systemen und Schnittstellen bestehen. Deren Betrieb ist teuer (geringe IT-Effizienz) und Änderungen sind meist mit relativ großem Aufwand verbunden (geringe IT-Flexibilität).

2) Auf der anderen Seite kann eine zu restriktive Entwicklung der IT-Architektur mit dem Fokus auf einer möglichst geringen Heterogenität der eingesetzten Technologien zwar zu einer geringen Komplexität führen, zugleich aber auch bei der Einführung innovativer Produkte auf Basis moderner Technologien hinderlich sein. Hier würde demnach eine gezielt höhere Komplexität – durch Zulassen neuer Technologien – zu einer höheren IT-Flexibilität führen.

Hieraus werden auch zwei unterschiedliche Facetten der IT-Flexibilität deutlich. Wie 1) zeigt, kann ein System, das aus einer Vielzahl heterogener und interagierender Komponenten besteht, die Durchführung von Änderungen erschweren, da viele unterschiedliche Technologien

und die Abhängigkeiten der Komponenten berücksichtigt werden müssen. Darüber hinaus können unerwartete Seiteneffekte die Änderung eines solchen Systems zusätzlich behindern. Die Größe des Systems, die Menge unterschiedlicher Technologien und die Abhängigkeiten können demzufolge zu einer geringeren IT-Flexibilität führen. Punkt 2) weist hingegen darauf hin, dass eine zu geringe Heterogenität der Komponenten und Beziehungen die IT-Flexibilität negativ beeinflussen kann, da Anforderungen, die mit nicht „zugelassenen“ Technologien schnell und einfach umzusetzen wären, mit den bestehenden Technologien zeitintensiv nachgebaut werden müssen.

Diese Erkenntnisse unterstreichen, dass die bestehenden Strategien und Maßnahmen des Architekturmanagements – aufgrund des dort vorherrschenden Fokus auf die Reduktion von Kosten und Heterogenität – nicht ausreichen, um eine adäquate Weiterentwicklung der IT-Architektur zu gewährleisten. Vielmehr bedarf es einer Strategie, die bei der Ermittlung eines angemessenen Grades an Heterogenität unterstützt und gleichzeitig eine potenziell geringere Effizienz in Kauf nimmt.

Die im Rahmen der qualitativ-empirischen Studie getroffenen Annahmen sowie die herangezogene Vorgehensweise, die Weiterentwicklung der IT-Architektur zu hinterfragen, haben sich als hilfreicher Ansatz erwiesen. Durch die Differenzierung struktureller Komplexität nach erwünschter und unnötiger Komplexität sowie der anschließenden Reflektion über die Gründe, warum es im Zuge von Eingriffen in die Architektur zu einer Abweichung in Form der unnötigen Komplexität kommt, können sowohl Forschung als auch Praxis zu neuen Erkenntnissen gelangen. Im Kontext der durchgeführten Fallstudie bestand ein wesentlicher Grund für die Entstehung unnötiger Komplexität u. a. im Fehlen einer mit ausreichenden Kompetenzen ausgestatteten Architekturmanagement-Funktion. Nach der Einführung einer solchen Funktion können weitere Ursachen zutage treten. Mögliche Ursachen könnten beispielsweise darin bestehen, wie die Architekturmanagement-Funktion organisational im Unternehmen eingegliedert ist, mit welchen Kompetenzen sie ausgestattet ist oder ob alle architekturrelevanten Prozesse identifiziert wurden. Weitere denkbare Ursachen könnten auch in der fehlenden oder nicht adäquaten Definition von Architektur-Richtlinien liegen. Ein fortwährendes Hinterfragen der Ursachen unnötiger Komplexität kann demnach zu weiteren Erkenntnissen hinsichtlich der Entstehung und Auswirkungen struktureller Komplexität führen.

Durch die in Kapitel 5 eingeführte – auf der systemtheoretischen Konzeptualisierung basierenden – Form der Quantifizierung von Komplexität wurde die Messbarkeit dieses Phänomens aufgezeigt. Ebendiese Messbarkeit von Komplexität ist Voraussetzung für die in Kapitel 6 erarbeiteten Beiträge. Hier wurde präskriptives Wissen in Gestalt abstrakter Designprinzipien abgeleitet, das beschreibt, wie ein Informationssystem ausgestaltet sein sollte, das IT-Manager und IT-Architekten bei dem Management – insbesondere der Planung und Kontrolle – der Komplexität von IT-Architekturen unterstützt. Die im Rahmen des Forschungsprojektes erstellten IT-Artefakte schaffen IT-Managern und IT-Architekten die erforderliche Transparenz über den aktuell vorliegenden Grad an Komplexität. Darüber hinaus bieten die IT-Artefakte die Möglichkeit, die Ursachen einer als zu hoch erachteten Komplexität zu untersuchen und mögliche Handlungsalternativen sowie deren Auswirkungen auf die Komplexi-

tät der Architektur zu überprüfen.¹⁰⁰ Zwar beziehen sich die im Rahmen des Forschungsprojektes entwickelten Designprinzipien auf Eigenschaften des zu erstellenden IT-Artefaktes, sie beinhalten jedoch auch implizit Wissen über das Management von Komplexität in IT-Architekturen im Allgemeinen. So unterstreichen die Designprinzipien DP 1 bis 4 die Angemessenheit der systemtheoretischen Perspektive auf das Problem des Komplexitätsmanagements. Ebenfalls einen systemischen Charakter weist DP 10 (Modularität) auf, hebt aber gleichzeitig hervor, dass es sich bei dem Management der Komplexität von IT-Architekturen selbst um ein Designproblem handelt (vgl. hierzu Ethiraj & Levinthal 2004; Schilling 2000; Simon 1996). Bezüglich der Frage, wessen Aufgabe im Wesentlichen darin besteht, ein den Anforderungen genügendes Design zu entwickeln, sieht die Literatur den IT-Architekten (bzw. äquivalente Bezeichnungen wie „Architecture Builder“ oder „Infrastructure Planner“) vor (Cross et al. 1997; Guillemette & Paré 2012). Im Zuge des Forschungsprojektes mit den beiden Banken wurde deutlich, dass dieser Fokus zu eng gesetzt ist. Die Designprinzipien 5, 6, 8, 9 und 11 weisen darauf hin, dass das Management der Komplexität von IT-Architekturen einen weit größeren Personenkreis betrifft. Alle Mitarbeiter, deren Entscheidungen Einfluss auf die Ausgestaltung der IT-Architektur haben, sollten an einem Architekturmanagement beteiligt werden.

7.3 Beitrag der Arbeit für die Unternehmenspraxis

Die Ergebnisse der vorliegenden Dissertation unterstützen die Praxis bei dem Management der IT-Architektur in einem sich ständig im Wandel befindlichen Umfeld. Mit dem Fokus auf der Komplexität der IT-Architektur wurde in dieser Arbeit eine Perspektive eingeführt, die bereits bestehende Ansätze komplementiert.

Der Begriff der „Komplexität“ wird in Unternehmen häufig unterschiedlich verstanden und verwendet. Die in Kapitel 3 vorgestellte systemtheoretische Konzeptualisierung struktureller Komplexität von Unternehmens-/IT-Architekturen kann herangezogen werden, um ein einheitliches Verständnis von Komplexität – und dies auch über verschiedene Architekturebenen hinweg – sicherzustellen. Ein solches ist erforderlich, um eine geordnete Weiterentwicklung der IT-Architektur zu gewährleisten und ein unkontrolliertes Wachstum der IT-Landschaft zu verhindern.

Die in Kapitel 4 vorgestellte qualitativ-empirische Studie hat gezeigt, dass Zusammenhänge zwischen der strukturellen Komplexität der IT-Architektur und der für Unternehmen relevanten Zieleigenschaften wie Effizienz, Flexibilität und Stabilität der IT bestehen. So führen u. a. die Anschaffung und der Betrieb fachlich redundanter Lösungen zu einer geringeren IT-Effizienz, da Kostensynergien ungenutzt bleiben. Weiterhin kann die zunehmende Komplexität aufgrund entstehender Abhängigkeiten zwischen heterogenen Lösungen zu einer geringeren IT-Flexibilität führen. Des Weiteren gibt die Studie Hinweise auf die Entstehung struktureller Komplexität in IT-Architekturen. Hierbei ist zu unterscheiden zwischen (1) den an die IT herangetragenen Anforderungen, (2) den Auswirkungen dieser Anforderungen auf die mi-

¹⁰⁰ Eine ausführliche Diskussion der theoretischen Implikationen der entwickelten Designprinzipien sowie der IT-Artefakte ist Abschnitt 6.8.1 zu entnehmen.

minimal erforderliche Komplexität der IT-Architektur sowie (3) den Mechanismen, die zu einer Abweichung von dieser minimal erforderlichen Komplexität in Form von unnötiger Komplexität führen. Die qualitativ-empirische Studie hat damit für die Praxis drei Stellschrauben für das Management struktureller IT-Komplexität identifiziert. Insbesondere Ausprägungen von Punkt (3) in Form der Mechanismen, die zu unnötiger Komplexität führen, sollten Unternehmen besondere Aufmerksamkeit widmen und ihre internen Vorgaben, Prozesse und Verantwortlichkeiten bei architekturrelevanten IT-Entscheidungen dahingehend überprüfen.

Die Studie weist zudem darauf hin, dass einmal getroffene Architekturentscheidungen zu einem späteren Zeitpunkt erneut auf deren Angemessenheit überprüft werden sollten. Jede Veränderung der Architektur führt zu einer neuen Ausgangssituation für künftige Architekturentscheidungen. Hierbei besteht die Gefahr, dass einmal erzeugte unnötige Komplexität zu einem späteren Zeitpunkt zu weiteren Kosten, geringerer Flexibilität oder sogar zu weiterer unnötiger Komplexität führt. So kann beispielsweise eine unnötigerweise eingeführte, fachlich redundante Lösung im Laufe der Zeit mit anderen Anwendung der IT-Landschaft so „verwachsen“, dass bei einer Anpassung eine Vielzahl von Abhängigkeiten berücksichtigt werden muss. Es sollte sich demnach in der IT etablieren, die Sinnhaftigkeit von in der Vergangenheit getroffenen Architekturentscheidungen zu hinterfragen und ggf. frühzeitig bestimmte Teilbereiche der Architektur zurückzubauen.

Wie die in Kapitel 5 vorgestellten Anwendungsfälle der eingeführten Maßzahlen zur Quantifizierung struktureller Komplexität in Unternehmens-/IT-Architekturen veranschaulichen, ist die Quantifizierung von Komplexität eine deutliche Hilfestellung für das Management von Komplexität. Die vorgeschlagenen Maßzahlen können von IT-Managern und IT-Architekten auf die für den jeweiligen Unternehmenskontext relevanten Fragestellungen ausgeprägt und dabei für alle vier Ebenen einer Unternehmensarchitektur herangezogen werden. Die Fragestellungen können unterschiedliche Zielsetzungen haben. So zeigt das erste Anwendungsbeispiel eines Unternehmens aus der Finanzbranche, dass die Maßzahlen bei der Identifikation struktureller Komplexität in einem bestimmten Teilbereich der Architektur – hier der eingesetzten Betriebs- und Datenbankmanagementsysteme – sowie der Ursachenanalyse der identifizierten Komplexität unterstützen können. Im zweiten Anwendungsbeispiel eines Unternehmens aus dem Transportsektor wurde der fachliche Zuschnitt der Domänenstruktur überprüft. Hierfür wurde die strukturelle Komplexität der Schnittstellen der je Domäne zugeordneten Anwendungen untersucht. Das dritte Anwendungsbeispiel illustrierte den Einsatz der Maßzahlen im Kontext der Überprüfung einer geplanten IT-Transformation bei einem Unternehmen aus der Versicherungsbranche. Unternehmen können die Maßzahlen demnach in vielfältiger Weise im Rahmen der Weiterentwicklung der Unternehmensarchitektur verwenden, um ein unkontrolliertes Wachstum der Architektur zu verhindern und eine intensive Nutzung der bereits bestehenden Architektur zu erreichen. Für ein Komplexitätsmanagement im Unternehmen sollten die hierfür relevanten Fragestellungen im Vorfeld durch das IT-Management definiert werden. Diese können im Zuge einer Architekturmaßnahme überprüft und die Entwicklung der Maßzahlen im Zeitverlauf beobachtet werden. Eine solche Überprüfung kann dabei als verpflichtender Schritt in allen Regelprozessen, die zu einer Veränderung der IT-Architektur führen, aufgenommen werden.

Aufgrund der einheitlichen Berechnungsvorschrift des Maßes, muss dieses lediglich einmalig verstanden werden. Anschließend ist dessen Verhalten bei Veränderung der Daten des jeweiligen Einsatzgebietes transparent und gut nachvollzieh- und kommunizierbar. Diese Eigenschaften des Maßes erleichtern den Einsatz in der Praxis, da lediglich die jeweilige Fragestellung verdeutlicht werden muss. Voraussetzung für die Nutzung der Maßzahlen ist die Verfügbarkeit der Architekturinformationen, wobei diese häufig aus bereits vorhandenen Architektur-Repositories gewonnen werden können.

Auf Grundlage der in Kapitel 6 entwickelten IT-Artefakte wurden – basierend sowohl auf theoretischen Überlegungen als auch auf den praktisch gesammelten Erfahrungen – abstrakte Designprinzipien für ein Informationssystem abgeleitet, das IT-Manager und IT-Architekten bei dem Management der Komplexität von IT-Architekturen unterstützt. Diese Designprinzipien können als Basis für die Entwicklung von IT-Artefakten der gleichen Problemklasse in weiteren organisationalen Kontexten verwendet werden. Bei der Umsetzung der Designprinzipien wird eine systemtheoretische Perspektive auf die IT-Architektur eingenommen, die es IT-Managern und IT-Architekten ermöglicht, die Anzahl und die Heterogenität der Komponenten und Beziehungen auf den verschiedenen Architekturebenen zu beurteilen und deren Veränderung über die Zeit hinweg zu beobachten. Neben der Bereitstellung von Informationen bzgl. der Anzahl und der Heterogenität der Designelemente erlaubt das IT-Artefakt eine Bewertung der Struktur des Systems aus einer Modularitäts-Perspektive. Hier können IT-Manager und IT-Architekten entscheiden, inwieweit die Organisation der Elemente zu Modulen sowie die Kommunikation innerhalb bzw. zwischen den Modulen angemessen sind.

Ein nach den Designprinzipien entwickeltes IT-Artefakt ermöglicht den Mitarbeitern aus unterschiedlichen Bereichen des Unternehmens, Teilbereiche der IT-Architektur zu untersuchen und den Ursachen eines ggf. zu hohen Grades an Komplexität auf den Grund zu gehen. Geplante Eingriffe in die IT-Architektur – z. B. im Zuge eines Projektes – können getestet und deren Auswirkungen auf die definierten Komplexitätskennzahlen überprüft werden. Diese können als Entscheidungsunterstützung dienen und einen Teil der Projektdokumentation darstellen, um im Nachgang die Sinnhaftigkeit der getroffenen Entscheidung belegen zu können. Für die Nutzung des Informationssystems und zur Erleichterung der internen Kommunikation werden die bereitgestellten Komplexitätsdaten sowohl auf verschiedenen Abstraktionsleveln als auch in unterschiedlichen Formen der Visualisierung zur Verfügung gestellt. Ferner ermöglicht das Informationssystem dem Nutzer, die zu berechnenden Kennzahlen auszuprägen, um spezifische Fragestellung beantworten zu können.

Das Informationssystem kann überdies als Werkzeug für ein Risikoreporting dienen. Hierzu müssen die IT-Manager entscheiden, welche Fragestellungen und somit Ausprägungen der Komplexitätsmaßzahlen die Basis des Risikoreportings darstellen sollen. Ein Beispiel wäre die Komplexität der verwendeten Infrastrukturplattformen, die durch die IT betrieben werden. Je höher die Anzahl der unterschiedlichen Plattformen (Richness), desto mehr qualifiziertes Personal muss vorgehalten werden und desto größer ist der Aufwand für die Dokumentation

(z. B. Wiederanlaufpläne, etc.). Eine zu große Anzahl unterschiedlicher Plattformen erhöht zudem die Anzahl potenzieller Sicherheitsschwachstellen.

Darüber hinaus kann das Informationssystem dabei unterstützen, Teilbereiche der Architektur zu identifizieren, die sich für eine Auslagerung anbieten. So ist es möglich, hoch standardisierte Komponenten der Architektur an Drittanbieter abzugeben, wie z. B. Aufgaben im Bereich des Betriebes der Infrastruktur.

7.4 Weitere Forschungsmöglichkeiten

Die in der vorliegenden Dissertation erarbeiteten Ergebnisse bieten unterschiedliche Möglichkeiten für weitere Forschung.

In Kapitel 4 wurde im Rahmen der qualitativ-empirischen Studie eine Herangehensweise zur Identifikation der Ursachen und Auswirkungen struktureller Komplexität in Unternehmen eingeführt. Diese erlaubt die Unterscheidung in erforderliche sowie unnötige Komplexität. Weitere Forschung könnte an dieser Stelle die Anwendbarkeit der Vorgehensweise in anderen organisationalen Kontexten untersuchen. Hier wäre interessant, inwiefern die hierdurch identifizierten Ausprägungen der Ursachen und Auswirkungen von Komplexität unternehmens- oder branchenspezifisch sind oder ob diese im Kern mit der Problemklasse des Komplexitätsmanagements zusammenhängen und somit weiter zu dessen Verständnis beitragen können.

Weiterhin wurde in dieser Studie ein Zusammenspiel zwischen der Path Dependence Theory und dem Phänomen struktureller Komplexität von IT-Architekturen angenommen. Diese stellen gemeinsam den Erklärungsansatz für die Abweichung der tatsächlichen von der optimalen strukturellen Komplexität der IT-Architektur dar (vgl. Abbildung 13): Eine zu einem Zeitpunkt getroffene Entscheidung über die Weiterentwicklung der IT-Architektur und der hierdurch entstehende Grad unnötiger Komplexität sind die Basis zukünftiger Entscheidungen. Die Pfadabhängigkeit zeigt sich folglich am Entscheidungsraum kommender Maßnahmen aufgrund der bestehenden Gestalt der IT-Architektur. An dieser Stelle wäre interessant zu untersuchen, inwieweit die Pfadabhängigkeit bei der Weiterentwicklung der IT-Architektur die Entstehung und die Auswirkungen unnötiger Komplexität verstärkt.

Durch die in Kapitel 5 eingeführte Form der Quantifizierung struktureller Komplexität haben IT-Manager und IT-Architekten die Möglichkeit, Kennzahlen für die für sie relevanten Fragestellungen auszuprägen, um die Komplexität verschiedener Bereiche der Unternehmens-/IT-Architektur zu bewerten. Unklar bleibt jedoch, welche Ausprägungen von Komplexitätskennzahlen typischerweise für ein Management der Komplexität von IT-Architekturen zur Verfügung stehen sollten. Die im Rahmen der Dissertation durchgeführten Studien geben einen ersten Hinweis darauf, dass speziell die Komplexität auf Ebene der Anwendungsarchitektur – und somit das Subsystem bestehend aus Anwendungen und deren Schnittstellen – eine wesentliche Rolle für das Architekturmanagement spielt.

Eine Schwäche der entwickelten Kennzahlen ist darin zu sehen, dass Ähnlichkeiten zwischen den betrachteten Charakteristiken durch das eingeführte Vorgehen zur Quantifizierung struktureller Komplexität nicht berücksichtigt werden. Die Heterogenität „verwandter“ Charakte-

ristiken wird demnach genauso behandelt, wie die völlig unterschiedlicher Charakteristiken. Weitere Forschung könnte an dieser Stelle ansetzen und Möglichkeiten zur Modifikation der Kennzahlen untersuchen, die Ähnlichkeiten bzw. unterschiedliche Ähnlichkeitsgrade der Charakteristiken miteinbeziehen.

Das in Kapitel 6 vorgestellte Forschungsprojekt wurde in Zusammenarbeit mit zwei Banken durchgeführt. In regelmäßigen Treffen mit IT-Architekten aus unterschiedlichen Branchen wurde deutlich, dass die in diesem Forschungsprojekt behandelte Problemklasse „Planung und Kontrolle von Komplexität in IT-Architekturen“ nicht ausschließlich eine im Bankenumfeld vorherrschende Problemklasse darstellt, sondern für eine Vielzahl von Branchen und Unternehmen relevant ist. Weitere Forschung könnte untersuchen, inwieweit die in dieser Arbeit vorgestellten Designprinzipien auf andere Branchen und Unternehmen übertragbar sind und diese somit den Kern der Problemklasse bereits erfassen.

Weiterer Forschungsbedarf im Kontext des in Kapitel 6 diskutierten Forschungsprojektes liegt in der Möglichkeit, nicht lediglich die abgeleiteten Designprinzipien, sondern zusätzlich das betrachtete System zu abstrahieren: Es wurden Designprinzipien für ein Informationssystem zur Planung und Kontrolle von Komplexität in IT-Architekturen hergeleitet. Hierbei wurde eine systemtheoretische Perspektive eingenommen und eine IT-Architektur als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht, verstanden. IT-Architekturen sind eingebettet in Unternehmen, die ihrerseits komplexe adaptive Systeme darstellen. Zukünftige Forschung könnte hier untersuchen, inwieweit die aus der vorgestellten Forschung resultierenden Designprinzipien auf Artefakte zur Planung und Kontrolle der Komplexität weiterer künstlicher Systeme, die einen Teil eines komplexen adaptiven Systems darstellen, übertragbar sind. Hiermit könnte ein Beitrag zur Systemforschung im Allgemeinen geleistet werden.

Die IT-Bereiche der Unternehmen müssen sich zunehmend mit kundenorientierten modernen Lösungen und den mit diesen verbundenen Konzepten auseinandersetzen. Ein solches Konzept stellt Cloud-Computing dar, eine spezielle Form des IT-Outsourcings (Buxmann et al. 2008). Dieses Konzept – samt seinen Vorzügen – ist vielen Mitarbeitern durch die private Nutzung diverser Cloud-Dienste wie „Dropbox“ oder Cloud-basierter Office-Lösungen wie z. B. „Google Docs“ bereits bekannt. Cloud-Dienste zeichnen sich dabei für die Nutzer durch einen einfachen On-Demand-Zugriff auf einen Pool konfigurierbarer Rechenressourcen aus, der durch den Anbieter meistens unmittelbar bereitgestellt wird (Mell & Grance 2011). Die Bereitstellung neuer Lösungen durch die IT-Bereiche der Unternehmen nimmt grundsätzlich einen deutlich längeren Zeitraum in Anspruch, wodurch die Erfahrungen mit Cloud-Diensten Begehrlichkeiten bei den Mitarbeitern wecken. Durch die vergleichsweise geringen technischen Hürden des Einsatzes eines Cloud-Dienstes (verglichen mit traditionellen On-Premise-Lösungen) haben die Geschäftsbereiche mit diesem Konzept grundsätzlich die Möglichkeit, auch ohne Zutun des IT-Bereiches IT-Entscheidungen zu treffen und Lösungen „anzuschaffen“. In einer 2012 durch Capgemini durchgeführten Befragung unter 460 Entscheidern wurde festgestellt, dass in ca. 45 % der Fälle die Entscheidung über den Einsatz einer Cloud-

Lösung im Geschäftsbereich getroffen wird (Capgemini 2012). Wie die in Kapitel 4 vorgestellte empirisch-qualitative Studie gezeigt hat, können durch Geschäftsbereiche getroffene Entscheidungen über die Anschaffung von (On-Premise-)Lösungen zu einer Erhöhung der Komplexität führen, da der IT-Bereich oftmals nicht die Möglichkeit hat, bestehende Technologien als alternative Lösungen vorzuschlagen. Weitere Forschung könnte diesen Punkt aufgreifen und untersuchen, welchen Einfluss IT-Outsourcing im Allgemeinen und Cloud-Dienste im Speziellen auf die strukturelle Komplexität der IT-Architektur haben. Interessant sind in diesem Zusammenhang zudem die Fragestellungen, wie die Rolle der IT in einem Unternehmen mit einem hohen Grad an durch Dienstleistern erbrachten IT-Leistungen auszugestalten ist, wie ein Architekturmanagement in einem solchen Umfeld einzuordnen ist und über welche Kompetenzen ein solches verfügen sollte.

Literatur

- Adomavicius, G., Bockstedt, J. C., Gupta, A. und Kauffman, R. J. (2008): Making Sense of Technology Trends in the Information Technology Landscape: A Design Science Approach. *MIS Quarterly*, 32 (4), S. 779-809.
- Ahlemann, F., Stettiner, E., Messerschmidt, M. und Legner, C. (2012): *Strategic Enterprise Architecture Management*. Springer-Verlag, Berlin, Heidelberg.
- Aier, S., Riege, C. und Winter, R. (2008): Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. *Wirtschaftsinformatik*, 50 (4), S. 292-304.
- Aier, S., Winter, R. und Wortmann, F. (2012): Entwicklungsstufen des Unternehmensarchitekturmanagements. *HMD Praxis der Wirtschaftsinformatik*, 49 (2), S. 15-23.
- AIS (2012): MIS Journal Rankings. <http://ais.affiniscap.com/displaycommon.cfm?an=1&subarticlenbr=432> (Abruf am 12.07.2012).
- Alderson, D. L. und Doyle, J. C. (2010): Contrasting Views of Complexity and Their Implications for Network-Centric Infrastructures. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 40 (4), S. 839-852.
- Allen, P. M. und Varga, L. (2006): A Co-Evolutionary Complex Systems Perspective on Information Systems. *Journal of Information Technology*, 21 (4), S. 229-238.
- Amaral, L. A. N. und Uzzi, B. (2007): Complex Systems—A New Paradigm for the Integrative Study of Management, Physical, and Technological Systems. *Management Science*, 53 (7), S. 1033-1035.
- Anderson, P. (1999): Perspective: Complexity Theory and Organization Science. *Organization Science*, 10 (3), S. 216-232.
- Ashby, W. R. (1956): *An Introduction to Cybernetics*. John Wiley and Sons, Oxford England.
- Avison, D. E., Lau, F., Myers, M. D. und Nielsen, P. A. (1999): Action Research. *Communications of the ACM*, 42 (1), S. 94-97.
- Avizienis, A., Laprie, J.-C., Randell, B. und Landwehr, C. (2004): Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1 (1), S. 11-33.
- Bafin (2009): Rundschreiben 3/2009 (VA) - Aufsichtsrechtliche Mindestanforderungen an das Risikomanagement (MaRisk VA). https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/rs_0903_va_marisk.html (Abruf am 03.09.2015).

- Bafin (2012): Rundschreiben 10/2012 (BA) - Mindestanforderungen an das Risikomanagement - MaRisk. [https://www.bafin.de/SharedDocs/Veroeffentlichungen/ DE/ Rundschreiben/rs_1210_marisk_ba](https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/rs_1210_marisk_ba) (Abruf am 03.09.2015).
- Baldwin, C. Y. und Clark, K. B. (2000): *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.
- Banker, R. D., Davis, G. B. und Slaughter, S. A. (1998): Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study. *Management Science*, 44 (4), S. 433-450.
- Banker, R. D. und Slaughter, S. A. (2000): The Moderating Effects of Structure on Volatility and Complexity in Software Enhancement. *Information Systems Research*, 11 (3), S. 219-240.
- Basili, V. R. und Rombach, H. D. (1988): The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*, 14 (6), S. 758-773.
- Belardo, S. und Pazer, H. L. (1985): Scope/Complexity: A Framework for the Classification and Analysis of Information-Decision Systems. *Journal of Management Information Systems*, 2 (2), S. 55-72.
- Benbasat, I., Goldstein, D. K. und Mead, M. (1987): The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, 11 (3), S. 369-386.
- Benbya, H. und Mckelvey, B. (2006): Toward a Complexity Theory of Information Systems Development. *Information Technology & People*, 19 (1), S. 12-34.
- Bertalanffy, L. V. (1948): Zu einer allgemeinen Systemlehre, *Biologia Generalis*. MIT Press/Wiley & Sons, Cambridge/New York, S. 114-129.
- Bertalanffy, L. V. (1968): *General System Theory: Foundations, Development, Applications*. George Braziller, New York.
- Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R. und O'Sullivan, D. (1997): An Overview of Legacy Information System Migration. In *Proceedings of the 4th Asian-Pacific Software Engineering and International Computer Science Conference (APSEC'97, ICSC'97)*, S. 529-530, IEEE.
- Boh, W. F. und Yellin, D. (2006): Using Enterprise Architecture Standards in Managing Information Technology. *Journal of Management Information Systems*, 23 (3), S. 163-207.
- Boisot, M. (2006): Moving to the Edge of Chaos: Bureaucracy, IT and the Challenge of Complexity. *Journal of Information Technology*, 21 (4), S. 239-248.
- Boltzmann, L. (1877): Über die Beziehung zwischen dem zweiten Hauptsatz der mechanischen Wärmetheorie und der Wahrscheinlichkeitsrechnung respektive den Sätzen über das Wärmegleichgewicht. In *Sitzungsber. d. k. Akad. der Wissenschaften zu Wien*, S. 373-435.

- Booch, G. (2008): Measuring Architectural Complexity. *IEEE Software*, 25 (4), S. 14-15.
- Bossert, O., Ip, C. und Laartz, J. (2014): A Two-Speed IT Architecture for the Digital Enterprise. http://www.mckinsey.com/insights/business_technology/a_two_speed_it_architecture_for_the_digital_enterprise (Abruf am 08.10.2015).
- Bradley, R. V., Pratt, R. M. E., Byrd, T. A., Outlay, C. N. und Wynn Jr, D. E. (2012): Enterprise Architecture, IT Effectiveness and the Mediating Role of IT Alignment in US Hospitals. *Information Systems Journal*, 22 (2), S. 97-127.
- Briggs, R. O., Nunamaker Jr, J. F. und Sprague, R. (2011): Applied Science Research in Information Systems: The Last Research Mile. *Journal of Management Information Systems*, 28 (1), S. 13-16.
- Brown, C. V. und Magill, S. L. (1998): Reconceptualizing the Context-Design Issue for the Information Systems Function. *Organization Science*, 9 (2), S. 176-194.
- Browne, T. (1658): *Pseudodoxia Epidemica*. 4. Auflage, Edward Dod, London.
- Bruckmann, G. (1969): Einige Bemerkungen zur statistischen Messung der Konzentration. *Metrika*, 14 (1), S. 183-213.
- Butler, J. K. (1999): Trust Expectations, Information Sharing, Climate of Trust, and Negotiation Effectiveness and Efficiency. *Group & Organization Management*, 24 (2), S. 217-238.
- Buxmann, P. (1996): *Standardisierung betrieblicher Informationssysteme*. Deutscher Universitäts-Verlag, Wiesbaden.
- Buxmann, P., Lehmann, S. und Hess, T. (2008): Software as a Service. *Wirtschaftsinformatik*, 50 (6), S. 500-503.
- Buxmann, P., Weitzel, T., von Westarp, F. und König, W. (1999): The Standardization Problem – An Economic Analysis of Standards in Information Systems. In *Proceedings of the IEEE Conference on Standardisation and Innovation in Information Technology (SIIT)*.
- Byrd, T. A. und Turner, D. E. (2000): Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct. *Journal of Management Information Systems*, 17 (1), S. 167-208.
- Capgemini (2012): Business Cloud: The State of Play Shifts Rapidly. <https://www.de.capgemini.com/cloud-services/business-cloud-the-state-of-play-shifts-rapidly> (Abruf am 12.01.2016).
- Capgemini (2014): Application Landscape Report 2014. <https://www.de.capgemini.com/ressourcen/application-landscape-report-2014> (Abruf am 12.01.2016).
- Cappiello, C., Francalanci, C. und Pernici, B. (2003): Time-Related Factors of Data Quality in Multichannel Information Systems. *Journal of Management Information Systems*, 20 (3), S. 71-92.

- Capra, E. und Merlo, F. (2009): How to Select Measures for Decision Support Systems - An Optimization Approach Integrating Informational and Economic Objectives. In *Proceedings of the 17th European Conference on Information Systems*, Verona, Italy.
- Charmaz, K. (2014): *Constructing Grounded Theory*. 2. Auflage, Sage, London.
- Choi, T. Y., Dooley, K. J. und Rungtusanatham, M. (2001): Supply Networks and Complex Adaptive Systems: Control Versus Emergence. *Journal of Operations Management*, 19 (3), S. 351-366.
- Clausius, R. (1865): Ueber verschiedene für die Anwendung bequeme Formen der Hauptgleichungen der mechanischen Wärmetheorie. *Annalen der Physik*, 201 (7), S. 353-400.
- Cooper, H. M. (1988): Organizing Knowledge Syntheses: A Taxonomy of Literature Reviews. *Knowledge in Society*, 1 (1), S. 104-126.
- Cross, J., Earl, M. J. und Sampler, J. L. (1997): Transformation of the IT Function at British Petroleum. *MIS Quarterly*, 21 (4), S. 401-423.
- Davis, J. S. und LeBlanc, R. J. (1988): A Study of the Applicability of Complexity Measures. *IEEE Transactions on Software Engineering*, 14 (9), S. 1366-1372.
- Davison, R. M., Martinsons, M. G. und Kock, N. (2004): Principles of Canonical Action Research. *Information Systems Journal*, 14 (1), S. 65-86.
- De Vries, H. J. (2006): IT Standards Typology. In *Advanced Topics in Information Technology Standards and Standardization Research* (Jakobs, K., Hrsg.), S. 1-26, Idea Group Publishing, London.
- Dern, G. (2011): *Integrationsmanagement in der Unternehmens-IT: Systemtheoretisch fundierte Empfehlungen zur Gestaltung von IT-Landschaft und IT-Organisation*. Vieweg+ Teubner Verlag, Wiesbaden.
- Dey, I. (1993): *Qualitative Data Analysis: A User-Friendly Guide for Social Scientists*. Routledge, London.
- Drucker, P. F. (1986): *Management: Tasks, Responsibilities, Practices*. 1. Auflage, Truman Talley Books, New York.
- Dubé, L. und Paré, G. (2003): Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations. *MIS Quarterly*, 27 (4), S. 597-635.
- Dyer, W. G. und Wilkins, A. L. (1991): Better Stories, Not Better Constructs, to Generate Better Theory: A Rejoinder to Eisenhardt. *Academy of Management Review*, 16 (3), S. 613-619.
- Eisenhardt, K. M. (1989): Building Theories from Case Study Research. *Academy of Management Review* 14 (4), S. 532-550.

- Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J. P., Voß, M. und Willkomm, J. (2008): *Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten*. Dpunkt-Verlag.
- Espinosa, J. A., Boh, W. F. und DeLone, W. (2011): The Organizational Impact of Enterprise Architecture: A Research Framework. In *Proceedings of the 44th Hawaii International Conference on System Sciences*, Kauai, Hawaii, USA.
- Ethiraj, S. K. und Levinthal, D. (2004): Modularity and Innovation in Complex Systems. *Management Science*, 50 (2), S. 159-173.
- Fenton, N. E. und Pfleeger, S. L. (1998): *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Co.
- Ferreira, K. N., Iverson, T. M., Maghlaoui, K., Barber, J. und Iwata, S. (2004): Architecture of the Photosynthetic Oxygen-Evolving Center. *Science*, 303 (5665), S. 1831-1838.
- Fischer, C., Winter, R. und Aier, S. (2010): What is an Enterprise Architecture Design Principle? Towards a Consolidated Definition. In *Computer and Information Science 2010*, S. 193-205, Springer-Verlag, Berlin, Heidelberg.
- Flood, R. L. und Carson, E. R. (1993): *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science. 2. Auflage*, Plenum Press, New York.
- Francalanci, C. und Piuri, V. (1999): Designing Information Technology Architectures: A Cost-Oriented Methodology. *Journal of Information Technology*, 14 (2), S. 181-192.
- Frank, U., Schauer, C. und Wigand, R. T. (2008): Different Paths of Development of Two Information Systems Communities: A Comparative Study Based on Peer Interviews. *Communications of the Association for Information Systems*, 22 (1), S. 391-412.
- Fürstenau, D. und Rothe, H. (2014): Shadow IT Systems: Discerning the Good and the Evil. In *Proceedings of the 22nd European Conference on Information Systems*, Tel Aviv, Israel.
- Garrison, C. B. und Paulson, A. S. (1973): An Entropy Measure of the Geographic Concentration of Economic Activity. *Economic Geography*, 49 (4), S. 319-324.
- Gleick, J. (1988): *Chaos-die Ordnung des Universums: Vorstoß in Grenzbereiche der modernen Physik*. Droemer Knaur.
- Gollop, F. M. und Monahan, J. L. (1991): A Generalized Index of Diversification: Trends in U.S. Manufacturing. *The Review of Economics and Statistics*, 73 (2), S. 318-330.
- Greefhorst, D. und Proper, E. (2011): *Architecture Principles - The Cornerstones of Enterprise Architecture*. Springer-Verlag, Berlin, Heidelberg.
- Gregor, S. (2002): Design Theory in Information Systems. *Australian Journal of Information Systems*, Special Issue, S. 14-22.
- Gregor, S. und Hevner, A. R. (2013): Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37 (2), S. 337-355.

- Gregor, S. und Jones, D. (2007): The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8 (5), S. 313-335.
- Guillemette, M. G. und Paré, G. (2012): Toward a New Theory of the Contribution of the IT Function in Organizations. *MIS Quarterly*, 36 (2), S. 529-551.
- Haki, M. K. und Legner, C. (2013): Enterprise Architecture Principles in Research and Practice: Insights from an Exploratory Analysis. In *Proceedings of the 21st European Conference on Information Systems*, Utrecht, Netherlands.
- Hall, A. D. und Fagen, R. E. (1969): *Definition of System*. Wiley, New York.
- Hall, M. und Tideman, N. (1967): Measures of Concentration. *Journal of the American Statistical Association*, 62 (317), S. 162-168.
- Halstead, M. H. (1977): *Elements of Software Science*. Elsevier, New York.
- Häni, P. K. (1987): *Die Messung der Unternehmenskonzentration: Eine theoretische und empirische Evaluation von Konzentrationsmaßen*. Rüegger.
- Hanseth, O., Jacucci, E., Grisot, M. und Aanestad, M. (2006): Reflexive Standardization: Side Effects and Complexity in Standard Making. *MIS Quarterly*, 30, Special Issue, S. 563-581.
- Hanseth, O. und Lyytinen, K. (2010): Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet. *Journal of Information Technology*, 25 (1), S. 1-19.
- Hart, P. E. (1971): Entropy and Other Measures of Concentration. *Journal of the Royal Statistical Society, Series A (General)*, 134 (1), S. 73-85.
- Henningsson, S. und Hanseth, O. (2011): The Essential Dynamics of Information Infrastructures. In *Proceedings of the 32nd International Conference on Information Systems*, Shanghai, China.
- Herfindahl, O. C. (1950): Concentration in the U.S. Steel Industry. Columbia University, New York.
- Hevner, A. R. (2007): The Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19 (2), S. 87-92.
- Hevner, A. R., March, S. T., Park, J. und Ram, S. (2004): Design Science in Information Systems Research. *MIS Quarterly*, 28 (1), S. 75-105.
- Hexter, J. L. und Snow, J. W. (1970): An Entropy Measure of Relative Aggregate Concentration. *Southern Economic Journal*, 36 (3), S. 239-243.
- Hill, M. O. (1973): Diversity and Evenness: A Unifying Notation and Its Consequences. *Ecology*, 54 (2), S. 427-432.
- Hirschman, A. O. (1945): *National Power and the Structure of Foreign Trade*. The University of California Press, Berkeley, Los Angeles.

- Hirschman, A. O. (1964): The Paternity of an Index. *The American Economic Review*, 54 (5), S. 761-762.
- Holland, J. H. (1995): *Hidden Order: How Adaptation Builds Complexity*. Addison Wesley, Reading, MA.
- Holland, J. H. (1998): *Emergence: From Chaos to Order*. Addison Wesley, Reading, MA.
- Horvath, J. (1970): Suggestion for a Comprehensive Measure of Concentration. *Southern Economic Journal*, 36 (4), S. 446-452.
- Hsieh, J. J. P. A. und Wang, W. (2007): Explaining Employees' Extended Use of Complex Information Systems. *European Journal of Information Systems*, 16 (3), S. 216-227.
- IEEE Architecture Working Group (2000): IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE Std 1471-2000*, S. i-23.
- ISO/IEC (2008): Systems and Software Engineering — System Life Cycle Processes (ISO/IEC 15288:2008).
- ISO/IEC/IEEE (2011): Systems and Software Engineering — Architecture Description (ISO/IEC/IEEE 42010:2011).
- Iyer, B. und Gottlieb, R. (2004): The Four-Domain Architecture: An Approach to Support Enterprise Architecture Design. *IBM Systems Journal*, 43 (3), S. 587-597.
- Jacquemin, A. P. und Berry, C. H. (1979): Entropy Measure of Diversification and Corporate Growth. *The Journal of Industrial Economics*, 27 (4), S. 359-369.
- Jiang, Z. J. und Benbasat, I. (2007): The Effects of Presentation Formats and Task Complexity on Online Consumers' Product Understanding. *MIS Quarterly*, 31 (3), S. 475-500.
- Jonkers, H., Lankhorst, M. M., Ter Doest, H. W. L., Arbab, F., Bosma, H. und Wieringa, R. J. (2006): Enterprise Architecture: Management Tool and Blueprint for the Organisation. *Information Systems Frontiers*, 8 (2), S. 63-66.
- Kearney, J. P., Sedlmeyer, R. L., Thompson, W. B., Gray, M. A. und Adler, M. A. (1986): Software Complexity Measurement. *Communications of the ACM*, 29 (11), S. 1044-1050.
- Khoshgoftaar, T. M., Munson, J. C., Bhattacharya, B. B. und Richardson, G. D. (1992): Predictive Modeling Techniques of Software Quality from Software Measures. *IEEE Transactions on Software Engineering*, 18 (11), S. 979-987.
- Kim, R. M. und Kaplan, S. M. (2006): The Co-Evolutionary Dynamics of IS Engagement. In *Proceedings of the 14th European Conference on Information Systems*, Göteborg, Sweden.
- Klir, G. J. (2001): *Facets of Systems Science*. 2. Auflage, Springer Science & Business Media.
- Klompé, R. und Pothuizen, L. (1997): The SIMA: A Practical Approach to Information Technology Management. *Journal of Information Technology*, 12 (2), S. 121-129.

- Kohlweyer, D. (2011): Quantifizierung von Heterogenität in IT-Landschaften. *FG Wirtschaftsinformatik, Software Business & Information Management*, Technische Universität Darmstadt, Bachelorthesis.
- Kokert, J. und Held, M. (2013): IT-Sicherheit: Erwartungen der Bankenaufsicht. https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Fachartikel/2013/fa_bj_2013_11_it_sicherheit.html (Abruf am 17.12.2015).
- Krcmar, H. (1990): Bedeutung und Ziele von Informationssystem-Architekturen. *Wirtschaftsinformatik*, 32 (5), S. 395-402.
- Krcmar, H. (2009): *Informationsmanagement. 5. Auflage*, Springer-Verlag, Berlin, Heidelberg.
- Kruchten, P., Obbink, H. und Stafford, J. (2006): The Past, Present, and Future for Software Architecture. *IEEE Software*, 23 (2), S. 22-30.
- Kruchten, P. B. (1995): Architectural Blueprints—The "4+1" View Model of Software Architecture. *IEEE Software*, 12 (6), S. 42-50.
- Kwoka Jr, J. E. (1985): The Herfindahl Index in Theory and Practice. *Antitrust Bull.*, 30, S. 915-947.
- Ladner, R. E. (1977): The Computational Complexity of Provability in Systems of Modal Propositional Logic. *SIAM Journal on Computing*, 6 (3), S. 467-480.
- Lagerström, R., Baldwin, C. Y., Maccormack, A. D. und Aier, S. (2014): Visualizing and Measuring Enterprise Application Architecture: An Exploratory Telecom Case. *Harvard Business School Working Paper, No. 13-103*.
- Landwehr, C. E. (2001): Computer Security. *International Journal of Information Security*, 1 (1), S. 3-13.
- Langlois, R. N. (2002): Modularity in Technology and Organization. *Journal of Economic Behavior & Organization*, 49 (1), S. 19-37.
- Langlois, R. N. und Robertson, P. L. (1992): Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries. *Research Policy*, 21 (4), S. 297-313.
- Lee, A. S. und Baskerville, R. L. (2003): Generalizing Generalizability in Information Systems Research. *Information Systems Research* 14 (3), S. 221-243.
- Lee, G. und Xia, W. (2002): Development of a Measure to Assess the Complexity of Information Systems Development Projects. In *Proceedings of the 23th International Conference on Information Systems*, Barcelona, Spain.
- Li, X. und Chen, Y. (2012): Corporate IT Standardization: Product Compatibility, Exclusive Purchase Commitment, and Competition Effects. *Information Systems Research*, 23 (4), S. 1158-1174.

- Liebowitz, S. J. und Margolis, S. E. (1995): Path Dependence, Lock-in, and History. *Journal of Law, Economics, and Organization*, 11 (1), S. 205-226.
- Light, B., Holland, C. P., Kelly, S. und Willis, K. (2000): Best of Breed IT Strategy: An Alternative to Enterprise Resource Planning Systems. In *Proceedings of the 8th European Conference on Information Systems*, Vienna, Austria.
- Lindgren, R., Henfridsson, O. und Schultze, U. (2004): Design Principles for Competence Management Systems: A Synthesis of an Action Research Study. *MIS Quarterly*, 28 (3), S. 435-472.
- Lloyd, M. und Ghelardi, R. J. (1964): A Table for Calculating the 'Equitability' Component of Species Diversity. *The Journal of Animal Ecology*, 33 (2), S. 217-225.
- Long, J. G. und Denning, D. E. (1995): Ultra-Structure: A Design Theory for Complex Systems and Processes. *Communications of the ACM*, 38 (1), S. 103-120.
- Lorenz, E. (2000): The Butterfly Effect. In *The Chaos Avant-Garde: Memories of the Early Days of Chaos Theory* (Abraham, R. H. und Ueda, Y., Hrsg.), S. 91-94, World Scientific Publishing Co. Pte. Ltd., Singapore.
- March, J. G. und Simon, H. A. (1958): *Organizations*. John Wiley, New York.
- March, S. T. und Smith, G. F. (1995): Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15 (4), S. 251-266.
- Marfels, C. (1972a): The Consistency of Concentration Measures: A Mathematical Evaluation. *Zeitschrift für die gesamte Staatswissenschaft / Journal of Institutional and Theoretical Economics*, S. 196-215.
- Marfels, C. (1972b): On Testing Concentration Measures. *Journal of Economics*, 32 (4), S. 461-486.
- Markus, M. L., Majchrzak, A. und Gasser, L. (2002): A Design Theory for Systems that Support Emergent Knowledge Processes. *MIS Quarterly*, 26 (3), S. 179-212.
- Marshall, T. E. und Byrd, T. A. (1998): Perceived Task Complexity as a Criterion for Information Support. *Information & Management*, 34 (5), S. 251-263.
- Mattern, F., Schönwälder, S. und Stein, W. (2003): Fighting Complexity in IT. *McKinsey Quarterly*, 1, S. 57-65.
- Matthes, D. (2011): Enterprise Architecture Frameworks Kompendium. *Über 50 Rahmenwerke für das IT-Management*, Springer-Verlag, Berlin Heidelberg.
- May, R. M. (1976): Simple Mathematical Models with Very Complicated Dynamics. *Nature*, 261 (5560), S. 459-467.
- McCabe, T. J. (1976): A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2 (4), S. 308-320.
- McIntosh, R. P. (1967): An Index of Diversity and the Relation of Certain Concepts to Diversity. *Ecology*, 48 (3), S. 392-404.

- Mell, P. und Grance, T. (2011): The NIST Definition of Cloud Computing (Special Publication 800-145). Gaithersburg, USA.
- Merali, Y. (2006): Complexity and Information Systems: The Emergent Domain. *Journal of Information Technology*, 21 (4), S. 216-228.
- Merali, Y. und Mckelvey, B. (2006): Using Complexity Science to Effect a Paradigm Shift in Information Systems for the 21st Century. *Journal of Information Technology*, 21 (4), S. 211-215.
- Mesarovic, M. D. (1964): *Views on General Systems Theory*. John Wiley & Sons, New York.
- Mesarovic, M. D. und Macko, D. (1969): Foundations for a Scientific Theory of Hierarchical Systems. In *Hierarchical Structures* (White, L. L., Wilson, A. G. und Wilson, G. E., Hrsg.), S. 29-50, New York.
- Midha, V. (2008): Does Complexity Matter? The Impact of Change in Structural Complexity on Software Maintenance and New Developers' Contributions in Open Source Software. In *Proceedings of the 29th International Conference on Information Systems*, Paris, France.
- Midha, V. und Slaughter, S. (2011): Mitigating the Effects of Structural Complexity on Open Source Software Maintenance through Accountability. In *Proceedings of the 32nd International Conference on Information Systems*, Shanghai, China.
- Miles, M. B. und Huberman, A. M. (1994): *Qualitative Data Analysis: An Expanded Sourcebook*. 2. Auflage, Sage, London.
- Möller, D., Legner, C. und Heck, A. (2011): Understanding IT Transformation – an Explorative Study. In *Proceedings of the 19th European Conference on Information Systems*, Helsinki, Finnland.
- Muchnik, A. A. und Vereshchagin, N. K. (2001): Logical Operations and Kolmogorov Complexity II. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*.
- Müller, R., Gerdali, J. und Turner, J. R. (2012): Relationships between Leadership and Success in Different Types of Project Complexities. *IEEE Transactions on Engineering Management*, 59 (1), S. 77-90.
- Myers, M. D. und Newman, M. (2007): The Qualitative Interview in IS Research: Examining the Craft. *Information and Organization*, 17 (1), S. 2-26.
- Namba, Y. und Iljima, J. (2004): City Planning Approach for Enterprise Information Systems. In *Proceedings of the 8th Pacific Asia Conference on Information System*, Shanghai, China.
- Network World (2011): Getting a Handle on Complexity. <http://www.networkworld.com/article/2182908/data-center/getting-a-handle-on-complexity.html> (Abruf am 12.10.2015).

- Niemann, K. D. (2005): *Von der Unternehmensarchitektur zur IT-Governance. 1. Auflage*, Springer Fachmedien, Wiesbaden.
- Olovsson, T. (1992): A Structured Approach to Computer Security. *Technical Report No 122*, Chalmers University of Technology, Göteborg, Schweden.
- Orlikowski, W. J. und Iacono, C. S. (2001): Research Commentary: Desperately Seeking the 'IT' in IT Research—A Call to Theorizing the IT Artifact. *Information Systems Research*, 12 (2), S. 121-134.
- Parnas, D., Clements, P. und Weiss, D. (1985): The Modular Structure of Complex Systems. *IEEE Transactions on Software Engineering*, 11 (3), S. 259-266.
- Parnas, D. L. (1972): On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15 (12), S. 1053-1058.
- Pauling, L. (1946): Molecular Architecture and Biological Reactions. *Chemical and Engineering News*, 24 (10), S. 1375-1377.
- Peet, R. K. (1974): The Measurement of Species Diversity. *Annual Review of Ecology and Systematics*, 5, S. 285-307.
- Pfeifer, W. (1997): *Etymologisches Wörterbuch des Deutschen*. Deutscher Taschenbuch Verlag, München.
- Pielou, E. C. (1966): The Measurement of Diversity in Different Types of Biological Collections. *Journal of Theoretical Biology*, 13, S. 131-144.
- Piesch, W. (1975): *Statistische Konzentrationsmaße: Formale Eigenschaften und verteilungstheoretische Zusammenhänge*. J. C. B. Mohr (Paul Siebeck), Tübingen.
- Priem, R. L., Butler, J. E. und Li, S. (2013): Toward Reimagining Strategy Research: Retrospection and Prospection on the 2011 AMR Decade Award Article. *Academy of Management Review*, 38 (4), S. 471-489.
- Pries-Heje, J. und Baskerville, R. (2008): The Design Theory Nexus. *MIS Quarterly*, 32 (4), S. 731-755.
- Pries-Heje, J., Baskerville, R. und Venable, J. (2008): Strategies for Design Science Research Evaluation. In *Proceedings of the 16th European Conference on Information Systems*, Galway, Ireland.
- Pulkkinen, M. (2006): Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, Kauai, Hawaii, USA.
- Pulkkinen, M. und Hirvonen, A. (2005): EA Planning, Development and Management Process for Agile Enterprise Development. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA.

- Purao, S. (2002): Design Research in the Technology of Information Systems: Truth or Dare. Working Paper, Department of Computer Information Systems, Georgia State University.
- Richardson, G. L., Jackson, B. M. und Dickson, G. W. (1990): A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. *MIS Quarterly*, 14 (4), S. 385-403.
- Roberts, T. L., Cheney, P. H., Sweeney, P. D. und Hightower, R. T. (2004): The Effects of Information Technology Project Complexity on Group Interaction. *Journal of Management Information Systems*, 21 (3), S. 223-247.
- Ropohl, G. (2009): *Allgemeine Technologie: Eine Systemtheorie der Technik*. 3. Auflage, Universitätsverlag Karlsruhe, Karlsruhe.
- Ross, J. W. (2003): Creating a Strategic IT Architecture Competency: Learning in Stages. *MIS Quarterly Executive*, 2 (1), S. 31-43.
- Ross, J. W., Weill, P. und Robertson, D. (2006): *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. Harvard Business School Press, Boston, MA.
- Ross, J. W. und Westerman, G. (2004): Preparing for Utility Computing: The Role of IT Architecture and Relationship Management. *IBM Systems Journal*, 43 (1), S. 5-19.
- Rudie, K. und Willems, J. C. (1995): The Computational Complexity of Decentralized Discrete-Event Control Problems. *IEEE Transactions on Automatic Control*, 40 (7), S. 1313-1318.
- Ruskin, J. (1851): *The Stones of Venice: Volume I—The Foundations*. Smith, Elder & Co., London.
- Ruskin, J. (1865): *The Seven Lamps of Architecture*. Wiley, New York.
- Saat, J., Aier, S. und Gleichauf, B. (2009): Assessing the Complexity of Dynamics in Enterprise Architecture Planning—Lessons from Chaos Theory. In *Proceedings of 15th Americas Conference on Information Systems*, San Francisco, USA.
- Saldaña, J. (2013): *The Coding Manual for Qualitative Researchers*. 2. Auflage, Sage, London.
- Sanchez, R. und Mahoney, J. T. (1996): Modularity, Flexibility, and Knowledge Management in Product and Organization Design. *Strategic Management Journal*, 17, Special Issue, S. 63-76.
- Sangwan, R. S., Vercellone-Smith, P. und Laplante, P. A. (2008): Structural Epochs in the Complexity of Software over Time. *IEEE Software*, 25 (4), S. 66-73.
- Sargut, G. und McGrath, R. G. (2011): Learning to Live with Complexity. *Harvard Business Review*, 89 (9), S. 68-76.

- Scheer, A.-W. (1991): *Architektur integrierter Informationssysteme: Grundlagen der Unternehmensmodellierung*. Springer-Verlag, Berlin, Heidelberg.
- Schilling, M. A. (2000): Toward A General Modular Systems Theory and Its Application to Interfirm Product Modularity. *Academy of Management Review*, 25 (2), S. 312-334.
- Schmidt, C. (2009): *Management komplexer IT-Architekturen: Empirische Analyse am Beispiel der internationalen Finanzindustrie. 1. Auflage*, Gabler, Wiesbaden.
- Schmidt, C. und Buxmann, P. (2011): Outcomes and Success Factors of Enterprise IT Architecture Management: Empirical Insight from the International Financial Services Industry. *European Journal of Information Systems*, 20 (2), S. 168-185.
- Schmidt, C., Widjaja, T. und Schütz, A. (2013): Messung der Komplexität von IT-Landschaften auf der Basis von Architektur-Metamodellen: Ein generischer Ansatz und dessen Anwendung im Rahmen der Architektur-Transformation. In *INFORMATIK 2013 – Beiträge der 43. Jahrestagung der Gesellschaft für Informatik*, Koblenz.
- Schneberger, S. L. und McLean, E. R. (2003): The Complexity Cross—Implications for Practice. *Communications of the ACM*, 46 (9), S. 216-225.
- Schneider, A., Reschenhofer, T., Schütz, A. und Matthes, F. (2015): Empirical Results for Application Landscape Complexity. In *Proceedings of the 48th Hawaii International Conference on System Science*, Kauai, Hawaii, USA.
- Schreiner, M., Hess, T. und Benlian, A. (2015): Gestaltungsorientierter Kern oder Tendenz zur Empirie? Zur neueren methodischen Entwicklung der Wirtschaftsinformatik. *Arbeitsbericht des Instituts für Wirtschaftsinformatik und Neue Medien (1/15)*, LMU München.
- Schütz, A., Widjaja, T. und Gregory, R. W. (2013a): Escape from Winchester Mansion—Toward a Set of Design Principles to Master Complexity in IT Architectures. In *Proceedings of the 34th International Conference on Information Systems*, Milan, Italy.
- Schütz, A., Widjaja, T. und Kaiser, J. (2013b): Complexity in Enterprise Architectures—Conceptualization and Introduction of a Measure from a System Theoretic Perspective. In *Proceedings of the 21st European Conference on Information Systems*, Utrecht, Netherlands.
- Schwaber, K. und Beedle, M. (2002): *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River.
- Seebold, E. und Kluge, F. (2011): *Kluge - Etymologisches Wörterbuch der deutschen Sprache. 25., durchges. und erw. Auflage*, De Gruyter, Berlin; Boston, MA.
- Sein, M., Henfridsson, O., Purao, S., Rossi, M. und Lindgren, R. (2011): Action Design Research. *MIS Quarterly*, 35 (1), S. 37-56.
- Shafiei-Monfared, S. und Jenab, K. (2012): Fuzzy Complexity Model for Enterprise Maintenance Projects. *IEEE Transactions on Engineering Management*, 59 (2), S. 293-298.

- Shannon, C. E. (1948): A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, S. 379-423 und 623-656.
- Sherif, K. und Xing, B. (2006): Adaptive Processes for Knowledge Creation in Complex Systems: The Case of a Global IT Consulting Firm. *Information & Management*, 43 (4), S. 530-540.
- Sherman, A. J. (2010): *Mergers and Acquisitions from A to Z*. 3. Auflage, AMACOM, New York.
- Shewchuk, J. P. und Moodie, C. L. (1998): Definition and Classification of Manufacturing Flexibility Types and Measures. *International Journal of Flexible Manufacturing Systems*, 10 (4), S. 325-349.
- Simon, H. A. (1962): The Architecture of Complexity. *Proceedings of the American philosophical society*, 106 (6), S. 467-482.
- Simon, H. A. (1996): *The Sciences of the Artificial*. 3. Auflage, MIT Press, Cambridge, MA.
- Simpson, E. H. (1949): Measurment of Diversity. *Nature*, 163, S. 688.
- Singh, K. (1997): The Impact of Technological Complexity and Interfirm Cooperation on Business Survival. *Academy of Management Journal*, 40 (2), S. 339-367.
- Sneppen, K., Bak, P., Flyvbjerg, H. und Jensen, M. H. (1995): Evolution as a Self-Organized Critical Phenomenon. *Proceedings of the National Academy of Sciences*, 92 (11), S. 5209-5213.
- Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., McDermid, J. und Paige, R. (2012): Large-Scale Complex IT Systems. *Communications of the ACM*, 55 (7), S. 71-77.
- Sowa, J. F. und Zachman, J. A. (1992): Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, 31 (3), S. 590-616.
- Spewak, S. und Hill, S. C. (1993): *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology*. John Wiley & Sons, New York.
- Stachowiak, H. (1973): *Allgemeine Modelltheorie*. Springer-Verlag, Wien.
- Sullivan, T. (2011): Embracing Complexity. *Harvard Business Review*, 89 (9), S. 89-92.
- Susman, G. I. und Evered, R. D. (1978): An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly*, 23 (4), S. 582-603.
- Tallon, P. P. und Kraemer, K. L. (2003): Using Flexibility to Enhance the Alignment between Information Systems and Business Strategy: Implications for IT Business Value. *Center for Research on Information Technology and Organizations (CRITO)*, University of California, Irvine.
- Tamm, T., Seddon, P. B., Shanks, G. und Reynolds, P. (2011): How Does Enterprise Architecture Add Value to Organisations? *Communications of the Association for Information Systems*, 28 (1), S. 141-168.

- Tegarden, D. P., Sheetz, S. D. und Monarchi, D. E. (1995): A Software Complexity Model of Object-Oriented Systems. *Decision Support Systems*, 13 (3), S. 241-262.
- The Chief Information Officers Council (1999): Federal Enterprise Architecture Framework V. 1.1.
- The Open Group (2011): TOGAF (The Open Group Architecture Framework) V. 9.1. <http://pubs.opengroup.org/architecture/togaf9-doc/arch/> (Abruf am 08.12.2015).
- Theil, H. (1967): *Economics and Information Theory*. North-Holland, Amsterdam.
- Tully, S. und Welsh, T. (1993): The Modular Corporation. *Fortune*, 127 (3), S. 106-111.
- U.S. Department of Justice and the Federal Trade Commission (2010): Horizontal Merger Guidelines. <http://www.justice.gov/atr/public/guidelines/hmg-2010.html> (Abruf am 20.08.2015).
- United Nations (1992): Convention on Biological Diversity. *Treaty Series: Treaties and International Agreements Registered or Filed and Recorded with the Secretariat of the United Nations*, 1760, S. 143-307.
- Van Aardt, A. (2004): Open Source Software Development as a Complex Adaptive System: Survival of the Fittest? In *Proceedings of the 17th Annual Conference of the National Advisory Committee on Computing Qualifications*, Christchurch, New Zealand.
- Van de Ven, A. H. (2007): *Engaged Scholarship: A Guide for Organizational and Social Research*. Oxford University Press Inc., New York.
- Van Gigch, J. P. (1991): *System Design Modeling and Metamodeling*. Plenum Press, New York.
- Van Solingen, R. und Berghout, E. (1999): *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill Inc., London.
- Venkatesh, V. und Davis, F. D. (2000): A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46 (2), S. 186-204.
- Verhulst, P.-F. (1838): Notice sur la loi que la population suit dans son accroissement. In *Correspondance Mathématique et Physique*, S. 113-121.
- Vitruvius Pollio, M. (2004): *De Architectura Libri Decem: Zehn Bücher über Architektur*. Marix, Wiesbaden.
- Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R. und Cleven, A. (2009): Reconstructing The Giant: On the Importance of Rigour in Documenting the Literature Search Process. In *Proceedings of the 17th European Conference on Information Systems*, Verona, Italy.
- Walls, J. G., Widmeyer, G. R. und El Sawy, O. A. (1992): Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, 3 (1), S. 36-59.

- Webster, J. und Watson, R. T. (2002): Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26 (2), S. xiii-xxiii.
- Weill, P. und Ross, J. W. (2004): *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Press, Boston, MA.
- Weitzel, T., Beimborn, D. und König, W. (2006): A Unified Economic Model of Standard Diffusion: The Impact of Standardization Cost, Network Effects, and Network Topology. *MIS Quarterly*, 30, Special Issue, S. 489-514.
- Widjaja, T. (2010): *Standardisierungsentscheidungen in mehrschichtigen Systemen: Untersuchung am Beispiel serviceorientierter Architekturen*. Springer, Wiesbaden.
- Widjaja, T. und Gregory, R. W. (2012): Design Principles for Heterogeneity Decisions in Enterprise Architecture Management. In *Proceedings of the 33rd International Conference on Information Systems*, Orlando, USA.
- Widjaja, T., Kaiser, J., Tepel, D. und Buxmann, P. (2012): Heterogeneity in IT Landscapes and Monopoly Power of Firms: A Model to Quantify Heterogeneity. In *Proceedings of the 33rd International Conference on Information Systems*, Orlando, USA.
- Wiener, N. (1961): *Cybernetics or Control and Communication in the Animal and the Machine*. 2. Auflage, MIT Press, Cambridge, MA.
- Winchester Mystery House (2015): Winchester Mystery House. <http://www.winchestermysteryhouse.com/thehouse.cfm> (Abruf am 08.12.2015).
- Winter, R. und Aier, S. (2011): How Are Enterprise Architecture Design Principles Used? In *IEEE International EDOC Conference Workshops, Trends in Enterprise Architecture Research (TEAR)*, Helsinki, Finland.
- Woese, C., Winker, S. und Gutell, R. (1990): Architecture of Ribosomal Rna: Constraints on the Sequence of "Tetra-Loops". *Proceedings of the National Academy of Sciences*, 87 (21), S. 8467-8471.
- Woolsey, J. P. (1994): 777. *Air Transport World*, 31 (4), S. 22-31.
- Wout, J., Waage, M., Hartman, H., Stahlecker, M. und Hofman, A. (2010): *The Integrated Architecture Framework Explained: Why, What, How*. Springer-Verlag, Berlin, Heidelberg.
- Xia, W. und Lee, G. (2005): Complexity of Information Systems Development Projects: Conceptualization and Measurement Development. *Journal of Management Information Systems*, 22 (1), S. 45-83.
- Xue, L., Ray, G. und Sambamurthy, V. (2012): Efficiency or Innovation: How Do Industry Environments Moderate the Effects of Firms' IT Asset Portfolios? *MIS Quarterly*, 36 (2), S. 509-528.
- Yin, R. K. (2014): *Case Study Research: Design and Methods*. 5. Auflage, Sage Publications, Inc., Thousand Oaks, CA.

Zachman, J. A. (1987): A Framework for Information Systems Architecture. *IBM Systems Journal*, 26 (3), S. 276-292.

Anhang

Interviewleitfaden zur Untersuchung der Ursachen und Auswirkungen struktureller Komplexität (vgl. Kapitel 4)

(1) Projekte und deren Verlauf

- Was waren die letzten Projekte an denen Sie beteiligt waren und wie sind diese verlaufen?
- Welche Projekte waren Ihrer Ansicht nach problematisch und warum?
 - Sind während der Projekte aufgetretene Probleme ggf. auf die Ausgestaltung der Anwendungs-/Infrastrukturarchitektur zurückzuführen?
 - Können Sie die aufgetretenen Probleme bitte näher beschreiben?
- Inwieweit hatten die Projekte aus Ihrer heutigen Sicht einen bedeutenden Einfluss (positiv/negativ) auf die Anwendungs-/Infrastrukturarchitektur?
- Aus welchen Gründen wurden negative Einflüsse in Kauf genommen?

(2) Durchführung von Änderungen an der IT-Architektur

- Wie ist in Ihrem Bereich der Prozess zur Durchführung von Änderungen an bestehenden Systemen oder der Einführung neuer Systeme ausgestaltet?
 - Wie kommt es zu neuen Anforderungen?
 - Welchem Prozess oder Vorgehen folgt die Bearbeitung neuer Anforderungen?
 - Wie arbeiten IT und Fachbereiche an dieser Stelle zusammen?

(3) Komplexität der Anwendungs-/Infrastrukturarchitektur

- Was verstehen Sie im Kontext der IT-Architektur unter Komplexität?
- Wie würden Sie die strukturelle Komplexität der aktuellen IT-Architektur beurteilen?¹⁰¹
- Wie äußert sich die strukturelle Komplexität der IT-Architektur?
- Was sind die Ursachen der aktuellen Struktur der IT-Architektur?

¹⁰¹ Bezug genommen wurde an dieser Stelle auf die in Unterkapitel 3.2 eingeführte systemtheoretische Konzeptualisierung struktureller Komplexität in Unternehmens-/IT-Architekturen.

- Welche Fehler wurden nach Ihrer Einschätzung bei der Änderung bestehender oder der Einführung neuer Systeme begangen? (Anknüpfend an (2))

(4) Eigenschaften der aktuellen Anwendungs-/Infrastrukturarchitektur

- Welche Eigenschaften würden Sie der bestehenden IT-Architektur zuordnen?
- Wie wirkt sich die bestehende Komplexität der IT-Landschaft auf diese Eigenschaften aus?
- Wie wirkt sich die bestehende strukturelle Komplexität der IT-Architektur auf Prozesse Ihrer Abteilung aus?

(5) Von Komplexität besonders betroffene Bereiche und Mittel für das Management dieser Komplexität

- In welchen Prozessen und Bereichen liegt nach Ihrer Einschätzung ein besonders hoher Grad an IT-Komplexität vor und wie kam es hierzu?
 - Welche Eigenheiten/Besonderheiten dieser Prozesse und Bereiche führen dort zu einer besonders hohen Komplexität?
 - Woran zeigt sich diese Komplexität?
- Welche Strategien werden heute bereits angewandt, um mit der Komplexität und den mit dieser verbundenen „negativen“ Eigenschaften umzugehen?
- Welche Änderungen und Maßnahmen würden Ihrer Ansicht nach am schnellsten zu einer Verringerung struktureller Komplexität führen? („quick-wins“)
- Was wären Ihrer Ansicht nach hingegen nachhaltige Maßnahmen?