

WI 74

EULVERLAG

Band 74
Reihe: WirtschaftsinformatikHERAUSGEBER
Dietrich Seibt, Hans-Georg Kemper, Georg
Herzwurm, Dirk Stelzer und Detlef Schoder

Martin Mikusz studierte an der Universität Stuttgart technisch-orientierte Betriebswirtschaftslehre. Seit seinem Abschluss im Jahr 2006 ist er akademischer Mitarbeiter am Lehrstuhl für Allgemeine Betriebswirtschaftslehre und Wirtschaftsinformatik II (Unternehmenssoftware) von Prof. Dr. Georg Herzwurm an der Universität Stuttgart. Seine Forschungsschwerpunkte liegen in den Bereichen Organisation und Koordination der Softwareentwicklung sowie Betriebswirtschaftslehre der Softwareunternehmung. Der Autor ist zudem Lehrbeauftragter an der FOM Hochschule für Oekonomie & Management, Stuttgart. Die Promotion zum Dr. rer. pol. erfolgte im Jahr 2011.

Martin Mikusz

Koordination der Standardsoftwareentwicklung

Ein situativer Ansatz

Mikusz

Sowohl empirische Befunde als auch die herrschende Meinung unter Fachleuten deuten darauf hin, dass die Organisationsgestaltung zugleich der mit den größten Risiken als auch mit den größten Verbesserungspotentialen behaftete Gestaltungsbereich der Softwareentwicklung ist. Ein wesentlicher organisatorischer Gestaltungsbereich ist dabei die Koordination. Koordinationsprobleme sind einer der Hauptgründe für die vielfach zitierte Softwarekrise und kommen insbesondere bei umfangreichen Softwareentwicklungsvorhaben zum Tragen. Solche Entwicklungsvorhaben sind im Bereich von Standardsoftware zahlreich vorzufinden. Sie zeichnen sich v. a. durch fehlende Transparenz von Entwicklungszielen, unzureichende Abstimmung der Beteiligten, ungelöste Zielkonflikte und ungeklärte Verantwortlichkeiten aus.

Die vorliegende Forschungsarbeit stellt einen situativ einsetzbaren Koordinationsansatz für umfangreiche Standardsoftware-Entwicklungsvorhaben bereit. Dazu wird zunächst das Konzept der Koordinationseffektivität abgeleitet. Anschließend wird, ausgehend von der Betrachtung des Standardsoftwareentwicklungsprozesses als einem ‚kontinuierlichen und konfliktären Zielbildungsprozess‘, ein Set von konkreten Koordinationsmechanismen entwickelt, um in Abhängigkeit von bestimmten Gegebenheiten stets eine effektive Koordination sicherstellen zu können.

Mit dem Streben nach wissenschaftlicher Qualität („rigor“) und Relevanz für die Praxis („relevance“) richtet sich diese Arbeit sowohl an Wissenschaftler und Studenten mit Interesse an organisatorischen Aspekten der Softwareentwicklung als auch an Verantwortliche in Standardsoftwareunternehmen.

Koordination der Standardsoftwareentwicklung

ISBN 978-3-8441-0073-0



www.eul-verlag.de
€ 59,- (D)



Reihe: Wirtschaftsinformatik · Band 74

Herausgegeben von Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Hans-Georg Kemper, Stuttgart, Prof. Dr. Georg Herzwurm, Stuttgart, Prof. Dr. Dirk Stelzer, Ilmenau, und Prof. Dr. Detlef Schoder, Köln

Dr. Martin Mikusz

Koordination der Standardsoftwareentwicklung

Ein situativer Ansatz

Mit einem Geleitwort von Prof. Dr. Georg Herzwurm,
Universität Stuttgart



Bibliografische Information der Deutschen Nationalbibliothek

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

Dissertation, Universität Stuttgart, 2011

D 93

ISBN 978-3-8441-0073-0

1. Auflage August 2011

© JOSEF EUL VERLAG GmbH, Lohmar – Köln, 2011

Alle Rechte vorbehalten

JOSEF EUL VERLAG GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 90 10 6-6

Fax: 0 22 05 / 90 10 6-88

E-Mail: info@eul-verlag.de

<http://www.eul-verlag.de>

Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.

Geleitwort

Die Entwicklung betrieblicher Informationssysteme zählt zu einem der wichtigsten Forschungsgebiete der Wirtschaftsinformatik. Informationssysteme werden hierbei als sozio-technische Systeme interpretiert, die neben technischen Komponenten wie Hardware und Software, auch Aufgaben bzw. Prozesse, Organisation sowie Menschen beinhalten. Die Wirtschaftsinformatik hat dabei im Laufe der Jahre Instrumente entwickelt, mit welchen bei der Entwicklung und Einführung neuer Anwendungssysteme auch Aspekte der organisatorischen Gestaltung Berücksichtigung finden. Die Softwareentwicklung selbst bedarf jedoch ebenfalls der organisatorischen Gestaltung.

Existierende Forschungsarbeiten fokussieren sich auf die Verbesserung des Softwareentwicklungsprozesses durch neue Paradigmen (z.B. agile Entwicklung), Methoden (z.B. ARIS) und Werkzeuge (z.B. Computer Aided Software Engineering). Fragstellungen der organisatorischen Gestaltung, wie etwa die Koordination der Softwareentwicklung, werden dagegen nur sehr sporadisch untersucht. Darüber hinaus steht bei den meisten Forschungsarbeiten die Entwicklung einer Individualsoftware im Mittelpunkt. Die Entwicklung mehrfach verwendbarer Standardsoftware wird hingegen selten untersucht.

Insofern betritt die Arbeit von Herrn Mikusz gleich in zweifacher Hinsicht Neuland: Zum einen widmet sie sich mit der Standardsoftwareentwicklung einer Domäne, die bislang wissenschaftlich kaum durchdrungen ist, zum anderen untersucht sie mit der organisatorischen Gestaltung, speziell der Koordination von Standardsoftwareentwicklung, einen Bereich, in dem es ebenfalls kaum wissenschaftlich fundierte Literatur gibt.

Insgesamt beschreibt die Arbeit von Herrn Mikusz einen ausgezeichneten Ansatz zur Koordination der Standardsoftwareentwicklung. Die erarbeiteten Kenntnisse gehen weit über den bislang verfügbaren Kenntnisstand zum Themengebiet hinaus. Ich wünsche dieser Arbeit eine entsprechende Verbreitung in Wissenschaft und Praxis.

Stuttgart, im Juli 2011

Univ.-Prof. Dr. Georg Herzworm

Danksagung

Während der Entstehung dieser Arbeit haben mich eine Vielzahl von Personen begleitet und unterstützt. Ihnen möchte ich an dieser Stelle danken.

Mein besonderer Dank gilt meinem Doktorvater, Prof. Dr. Georg Herzwurm. Er hat meine Arbeit immer unterstützt und mir gleichzeitig die notwendigen akademischen Freiheiten gegeben. Auf Seiten meines Praxispartners hat Herr Anton Dillinger meine Forschung schon seit meiner Zeit als Diplomand stets befürwortet und gefördert. Auch ihm bin ich zu besonderem Dank verpflichtet. Ohne die Unterstützung dieser beiden Personen wäre die vorliegende Arbeit in dieser Form nicht möglich gewesen. Bei Prof. Dr. Michael Reiß bedanke ich mich für das Verfassen des Zweitgutachtens. Prof. Dr. Hans-Georg Kemper danke ich für die Übernahme des Prüfungsvorsitzes.

Bei meinen Kolleginnen und Kollegen am Lehrstuhl für Allgemeine Betriebswirtschaftslehre und Wirtschaftsinformatik II an der Universität Stuttgart – Benedikt Krams, Lars Mautsch, Katharina Peine, Norman Pelzl, Sook Ja Schmitz und Sixten Schockert, sowie bei den externen Doktoranden und Ehemaligen möchte ich mich für die wertvollen Diskussionen und die tolle, persönliche Arbeitsatmosphäre bedanken.

Ebenso danke ich meinen ehemaligen Kolleginnen und Kollegen in der Praxis, insbesondere Herrn Markus Gebhard und den zahlreichen Interviewpartnern für Ihr Interesse an meiner Forschung und Ihre Auskunftsbereitschaft. Nicht zuletzt bin ich Herrn Herbert Illgner und Herrn Bernd Leukert zur Danksagung verpflichtet. Sie ermöglichten die interne Finanzierung meiner Arbeit.

Besonderer Dank gebührt schließlich allen Personen in meinem persönlichen Umfeld. Auch sie haben in gewisser Weise die Erstellung dieser Arbeit ermöglicht: meine Frau Evelina, meine Tochter Luisa, meine Eltern Bogumila und Tomasz und mein Bruder Mateusz.

Stuttgart, im Juli 2011

Martin Mikusz

Inhaltsübersicht

Geleitwort	V
Danksagung.....	VII
Inhaltsübersicht.....	IX
Inhaltsverzeichnis	XI
Abbildungsverzeichnis.....	XV
Abkürzungsverzeichnis.....	XIX
Zusammenfassung.....	XXI
Abstract	XXIII
1 Gegenstand und wissenschaftliche Vorgehensweise der Arbeit.....	1
1.1 Motivation.....	1
1.2 Ziel der Arbeit.....	3
1.3 Forschungsstrategie	6
1.4 Wissenschaftstheoretische und -methodologische Basis.....	12
1.5 Stand der Forschung	15
2 Konzeption des situativen Koordinationsansatzes.....	23
2.1 Theoretischer Zugang über den Situativen Ansatz der Organisationsgestaltung.....	23
2.2 Theoretischer Zugang über das entscheidungsorientierte Konzept der Organisationsgestaltung.....	52
2.3 Situativer Koordinationsansatz für die Standardsoftwareentwicklung	73
2.4 Konzeption des situativen Koordinationsansatzes im Überblick	116
3 Umsetzung des situativen Koordinationsansatzes	121
3.1 Fallstudiendesign	121
3.2 Gestaltungsmaßnahmen der Koordination	133
3.3 Bestimmung der übergeordneten Projektzielpräferenzen als Gestaltungsbedingung.....	169
3.4 Organisatorische Umsetzung des situativen Koordinationsansatzes.....	198
3.5 Umsetzung des situativen Koordinationsansatzes im Überblick.....	219
4 Kritische Würdigung der Arbeit	227
4.1 Kritische Würdigung aus Sicht des Situativen Ansatzes der Organisationsgestaltung.....	228
4.2 Kritische Würdigung aus Sicht des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung	234
Literaturverzeichnis	245
Anhang: Beispielhafte Konfigurationen zu den definierten Projekttypen.....	263

Inhaltsverzeichnis

Geleitwort	V
Danksagung.....	VII
Inhaltsübersicht.....	IX
Inhaltsverzeichnis	XI
Abbildungsverzeichnis.....	XV
Abkürzungsverzeichnis.....	XIX
Zusammenfassung.....	XXI
Abstract.....	XXIII
1 Gegenstand und wissenschaftliche Vorgehensweise der Arbeit.....	1
1.1 Motivation.....	1
1.2 Ziel der Arbeit.....	3
1.3 Forschungsstrategie	6
1.4 Wissenschaftstheoretische und -methodologische Basis.....	12
1.5 Stand der Forschung	15
2 Konzeption des situativen Koordinationsansatzes.....	23
2.1 Theoretischer Zugang über den Situativen Ansatz der Organisationsgestaltung.....	23
2.1.1 Der Situative Ansatz der Organisationsgestaltung	23
2.1.2 Entwicklung von Standardsoftware aus Sicht des Situativen Ansatzes	28
2.1.2.1 Gestaltungsziele und -maßnahmen bei der Softwareentwicklung im Allgemeinen.....	28
2.1.2.2 Gestaltungsbedingungen bei der Standardsoftwareentwicklung	32
2.1.3 Softwarespezifische Umsetzungen des Situativen Ansatzes	39
2.1.3.1 Tailoring der Softwareentwicklung	39
2.1.3.2 Risikobasierte Softwareentwicklung und Agile Softwareentwicklung	45
2.2 Theoretischer Zugang über das entscheidungsorientierte Konzept der Organisationsgestaltung.....	52
2.2.1 Der entscheidungsprozessorientierte Ansatz der Organisationsgestaltung .	52
2.2.2 Entwicklung von Standardsoftware aus Sicht des entscheidungsprozessorientierten Ansatzes	57
2.2.2.1 Spezialisierung und strukturelle Eingliederung von Entwicklungsprojekten in die Aufbauorganisation einer Standardsoftwareunternehmung	57
2.2.2.2 Arbeitsteilung und Spezialisierung innerhalb von Entwicklungsprojekten	62
2.2.2.3 Koordination der Softwareentwicklung durch Entwurf der Softwarearchitektur.....	69

2.3	Situativer Koordinationsansatz für die Standardsoftwareentwicklung	73
2.3.1	Koordinations-effektivität als Gestaltungsziel.....	73
2.3.1.1	Das Konzept der Koordinationseffizienz als Ausgangspunkt	73
2.3.1.2	Das Konzept der Koordinationseffektivität	77
2.3.2	Gestaltungsmaßnahmen der Koordination.....	81
2.3.2.1	Der Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess	81
2.3.2.2	Koordinationsphasen und -prozesse in der Standardsoftwareentwicklung.....	89
2.3.2.3	Institutionalisierung von Koordinationsprozessen.....	94
2.3.3	Projektziele als Gestaltungsbedingung	99
2.3.3.1	Das Zielsystem einer Standardsoftwareunternehmung.....	99
2.3.3.2	Übergeordnete Projektzielpräferenzen als Gestaltungsbedingung für die Koordinationseffektivität	104
2.3.4	Organisatorische Umsetzung des situativen Koordinationsansatzes als Koordinations durch Programme.....	110
2.4	Konzeption des situativen Koordinationsansatzes im Überblick	116
3	Umsetzung des situativen Koordinationsansatzes	121
3.1	Fallstudiendesign	121
3.1.1	Vorgehensweise bei der Dokumentenanalyse	123
3.1.2	Vorgehensweise bei den Experteninterviews	127
3.2	Gestaltungsmaßnahmen der Koordination	133
3.2.1	Strukturdimension Artefaktspezifikation.....	134
3.2.1.1	Steuerungsmaßnahmen in der Strukturdimension Artefaktspezifikation	134
3.2.1.2	Umsetzung der Steuerungsmaßnahmen durch Checklisten.....	142
3.2.2	Strukturdimension Kompetenzverteilung.....	147
3.2.2.1	Steuerungsmaßnahmen in der Strukturdimension Kompetenzverteilung.....	147
3.2.2.2	Entscheidungs- und Mitwirkungskompetenzen bei Eskalationsprozessen	153
3.2.3	Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘	157
3.2.3.1	Steuerungsmaßnahmen in der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘	157
3.2.3.2	Personelle und organisatorische Unabhängigkeit des Produktmanagements.....	164

3.3	Bestimmung der übergeordneten Projektzielpräferenzen als Gestaltungsbedingung.....	169
3.3.1	Projektzieldimension Leistungsumfang.....	170
3.3.1.1	Ableitung der Präferenz aus dem Priorisierungskriterium „Schaden“	170
3.3.1.2	Wiederverwendung von Priorisierungsinformationen	176
3.3.2	Projektzieldimension Qualität.....	179
3.3.2.1	Ableitung der Präferenz aus Qualitätsmodellen.....	179
3.3.2.2	Abgrenzung der Projektzieldimensionen Qualität und Leistungsumfang.....	183
3.3.3	Projektzieldimension Zeit	187
3.3.3.1	Ableitung der Präferenz aus Projektabhängigkeiten.....	187
3.3.3.2	Anforderungsabhängigkeiten als Grundlage für die Bestimmung von Projektzielpräferenzen	193
3.4	Organisatorische Umsetzung des situativen Koordinationsansatzes.....	198
3.4.1	Differenzierungsgrad des Klassifikationsschemas	199
3.4.1.1	Anwendung heuristischer Vereinfachungsprinzipien beim Tailoring der Softwareentwicklung	199
3.4.1.2	Bildung des Klassifikationsschemas durch Definition von Projekttypen	203
3.4.2	Detaillierungsgrad der Vorgaben für die definierten Projekttypen	208
3.4.2.1	Projekttypen als Konfigurationen	208
3.4.2.2	Verbindlichkeitsgrad der Vorgaben für den Projektablauf.....	213
3.5	Umsetzung des situativen Koordinationsansatzes im Überblick.....	219
4	Kritische Würdigung der Arbeit	227
4.1	Kritische Würdigung aus Sicht des Situativen Ansatzes der Organisationsgestaltung.....	228
4.2	Kritische Würdigung aus Sicht des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung	234
	Literaturverzeichnis	245
	Anhang: Beispielhafte Konfigurationen zu den definierten Projekttypen.....	263

Abbildungsverzeichnis

Abb. 1: Ziel und Forschungsfragen der Arbeit.....	6
Abb. 2: Übergeordnete Struktur der ‚4+1-Theorie‘ von VBSE	7
Abb. 3: Forschungsstrategie der Arbeit	11
Abb. 4: Wissenschaftstheoretische und -methodologische Basis der Arbeit	15
Abb. 5: Ermittlung von Softwareprojekttypen beim Ansatz von Schramke	17
Abb. 6: Gestaltungsempfehlungen zur Projektablauforganisation beim Ansatz von Schramke	18
Abb. 7: Wettbewerbsstrategische Handlungsalternativen beim Ansatz von Lang.....	19
Abb. 8: Kernfragen der analytischen Variante des Situativen Ansatzes	24
Abb. 9: Das Grundmodell der pragmatischen Variante des Situativen Ansatzes	24
Abb. 10: Stimmigkeitstypen nach der Art der Relation	27
Abb. 11: Perspektiven auf die Gestaltungsziele der Softwareentwicklung.....	29
Abb. 12: Fachlich-technische Prozesse der Softwareentwicklung	30
Abb. 13: Reifegrade des CMMI	41
Abb. 14: Zuordnung von Projektdurchführungsstrategien zu den Projekttypen im V-Modell XT	43
Abb. 15: Grundsätzlicher Ablauf des Spiralmodells	45
Abb. 16: Beispielhafte Vorhabenskategorien beim Ansatz ‚Value-Based Software Quality Achievement Process Framework‘	47
Abb. 17: Gestaltungsbedingungen beim risikobasierten Ansatz von Boehm und Turner	50
Abb. 18: Koordinationsinstrumente als übergeordnete und Motivationsinstrumente als untergeordnete Regeln	56
Abb. 19: Eingliederung von Entwicklungsprojekten in die Aufbauorganisation einer Standardsoftwareunternehmung	61
Abb. 20: Objekt- und verrichtungsorientierte Spezialisierung durch Rollen- und Teilprojektbildung in der Projektorganisation von Standardsoftware- Entwicklungsvorhaben	66
Abb. 21: Objektorientierte Spezialisierung als Parallelisierung des Entwicklungsverlaufes.....	67
Abb. 22: Das Treppenmodell der inkrementellen Softwareentwicklung	68
Abb. 23: Zusammenhang zwischen Autonomie- und Abstimmungskosten.....	75
Abb. 24: Das Konzept der Koordinationseffektivität	81
Abb. 25: Konus der Unsicherheit bei der Softwareentwicklung	83
Abb. 26: Einschränkung des Lösungsraumes entlang des Entwicklungsprozesses	84

Abb. 27: Komponenten des Misserfolgs bei der optimistischen Planung	84
Abb. 28: Zielrichtungen der Beteiligten im ‚Teufelsquadrat der Softwareentwicklung‘	87
Abb. 29: Prozessorientiertes Projektphasenmodell	90
Abb. 30: Zielobjekte und Zielgrößen der Produktentwicklung	99
Abb. 31: Entscheidungstaxonomie des Requirements Engineering	103
Abb. 32: Zielsystem einer Standardsoftwareunternehmung	104
Abb. 33: Zielkonfliktbasierte Kategorien von Projektvorgaben	105
Abb. 34: Zielkonfliktbasierte Behandlung von Risikofaktoren bei Softwareentwicklungsprojekten	106
Abb. 35: Übergeordnete Projektzielpräferenzen als Gestaltungsbedingung für Gestaltungsmaßnahmen der Koordination	110
Abb. 36: Ansätze zur projektspezifischen Adaption eines Softwareentwicklungsprozesses	114
Abb. 37: Konzeption des situativen Koordinationsansatzes im Überblick	116
Abb. 38: Fallstudiendesign beim empirischen Teil der Arbeit	123
Abb. 39: Erfassungsschema für Gestaltungsmaßnahmen bei der Dokumentenanalyse	124
Abb. 40: Erfassungsschema für Gestaltungsbedingungen bei der Dokumentenanalyse	126
Abb. 41: Erfassungsschema für Anhaltspunkte zur organisatorischen Umsetzung des Gesamtansatzes bei der Dokumentenanalyse	127
Abb. 42: Übersicht der durchgeführten Experteninterviews	128
Abb. 43: Interviewleitfaden für die Experteninterviews	130
Abb. 44: Strukturdimensionen der Organisationsgestaltung beim situativen Koordinationsansatz	134
Abb. 45: Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation ...	135
Abb. 46: EVM als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation	136
Abb. 47: Referenzrahmen für situative Projektinformationssystem-Ansätze	138
Abb. 48: Informationsstruktur des Änderungsantrages als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation	139
Abb. 49: Produktreview-Checklisten als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation	140
Abb. 50: Perspektivenbasiertes Lesen als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation	141
Abb. 51: Ausrichtung einer Risikocheckliste auf die übergeordneten Projektzielpräferenzen	145

Abb. 52: Auf Projektzielpräferenzen bezogene Checklistenfragen für die Einflussanalyse beim Änderungsmanagement	146
Abb. 53: Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung ..	148
Abb. 54: Kompetenzverteilung beim Eskalationsprozess für Qualitätszielabweichungen als Steuerungsmaßnahme der Strukturdimension Kompetenzverteilung.....	149
Abb. 55: Kompetenzverteilung beim Änderungsmanagement als Steuerungsmaßnahme der Strukturdimension Kompetenzverteilung	149
Abb. 56: ‚PRINCE2‘-Subprozess ‚Erstellen eines Ausnahmeplans‘	154
Abb. 57: Gestaltungsmaßnahmen in der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘	158
Abb. 58: Projektspezifische Übernahme von ‚Product Standard Expert‘-Rollen als Gestaltungsmaßnahme der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘	159
Abb. 59: Handlungsautonomie einer Rolle.....	161
Abb. 60: Personelle und organisatorische Unabhängigkeit bei der Norm DIN EN 50128	162
Abb. 61: ‚Scrum‘ im Überblick	165
Abb. 62: Priorisierungskriterium ‚Wichtigkeit‘ zur Präferenzbestimmung der Projektzieldimension Leistungsumfang	171
Abb. 63: Einfluss des Erfüllungsgrades von Basis-, Zusatz- und Begeisterungsmerkmalen auf die Kundenzufriedenheit im Kano-Modell..	174
Abb. 64: Kano-Modell und Priorisierungskriterium ‚Schaden‘ zur Präferenzbestimmung der Projektzieldimension Leistungsumfang	175
Abb. 65: Merkmale zur Definition der externen und internen Qualität des Softwareproduktes nach ISO/IEC 9126	180
Abb. 66: Umzusetzende Qualitätsanforderungen als Kriterium für die Präferenzbestimmung der Projektzieldimension Qualität.....	182
Abb. 67: Ausschnitt aus dem Aufbau von Anforderungen nach dem IEEE-Standard 830	183
Abb. 68: Erfüllungsrisiko als Kriterium für die Präferenzbestimmung der Projektzieldimensionen Qualität und Leistungsumfang.....	185
Abb. 69: Zuordnung von Teilmerkmalen der Funktionalität nach ISO/IEC 9126 zu den Projektzieldimensionen Qualität und Leistungsumfang	185
Abb. 70: Interdependenzenarten zwischen organisatorischen Einheiten	189
Abb. 71: Abhängigkeitsbeziehungen auf Projektebene als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit	190
Abb. 72: Abhängigkeitsbeziehungen auf Ebene der Entwicklungsgegenstände als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit.....	190

Abb. 73: Anforderungsabhängigkeiten vom Typ ‚Vorbedingung/Benötigt‘ als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit.....	192
Abb. 74: Klassifizierung von fundamentalen Abhängigkeitstypen.....	194
Abb. 75: Anforderungsnachvollziehbarkeit bei verschiedenen Dimensionen des Requirements Engineering	196
Abb. 76: Tailoringansatz für die Produkt- und Anforderungsdefinition	200
Abb. 77: Tailoringansatz für verschiedene Klassen von Entwicklungsvorhaben	201
Abb. 78: Tailoringansatz für die Implementierung von Szenarien.....	202
Abb. 79: Zwecke der Klassifikation von Projekten in der Softwareentwicklung	203
Abb. 80: Umsetzung des situativen Koordinationsansatzes durch definierte Projekttypen.....	205
Abb. 81: Bei den Experteninterviews genannte Differenzierungsschemata und Projekttypen.....	206
Abb. 82: Detaillierungsgrad der Vorgaben bei den Tailoringansätzen in der Fallstudienunternehmung	209
Abb. 83: Projekttypen des situativen Koordinationsansatzes als Konfigurationen im Sinne des Konfigurationsansatzes der Organisationsgestaltung	211
Abb. 84: Problemkomplexe bei der Umsetzung von Projektkategorisierungsansätzen	214
Abb. 85: Formalisierungsgrad von Projektkategorisierungsansätzen	217
Abb. 86: Projektprogrammstart-Workshop in Verbindung mit Projektstart-Workshops als formaler Rahmen für das Tailoring beim situativen Koordinationsansatz	219
Abb. 87: Umsetzung des situativen Koordinationsansatzes im Überblick.....	220

Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
B2B	Business-to-Business
CAIV	cost as independent variable
CASE	Computer Aided Software Engineering
CMMI	Capability Maturity Model Integration
CSSC	Concept Selection Scorecard
DIN	Deutsche Industrienorm
EN	Europäische Norm
EVM	Earned Value Management
FCM	factor-criteria-metrics-model
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ISR	Information Systems Research
LSRE	Large-Scale Requirements Engineering
MSRE	Medium-Scale Requirements Engineering
RE	Requirements Engineering
SAIV	schedule as independent variable
SCQAIV	schedule/cost/quality as independent variable
SME	Situational Method Engineering
SSRE	Small-Scale Requirements Engineering
TBQ	Taxonomy-Based Questionnaire
VBSE	Value Based Software-Engineering
VLSRE	Very Large-Scale Requirements Engineering
WBS	work breakdown structure
WPD	Work Product Description

Zusammenfassung

Die organisatorische Gestaltung ist der zugleich mit den größten Risiken als auch mit den größten Verbesserungspotentialen behaftete Gestaltungsbereich der Softwareentwicklung. Ein wesentlicher organisatorischer Gestaltungsbereich ist dabei die Koordination, d.h. das Ausrichten von Einzelaktivitäten in einem arbeitsteiligen System auf ein übergeordnetes Gesamtziel. Koordinationsprobleme sind einer der Hauptgründe für die vielfach zitierte Softwarekrise und kommen insbesondere bei umfangreichen Softwareentwicklungsvorhaben zum Tragen. Solche Entwicklungsvorhaben sind im Bereich von Standardsoftware, dem Gegenstandsbereich der vorliegenden Arbeit, zahlreich vorzufinden.

Nach dem Verständnis der vorliegenden Arbeit kann es einen universellen und allseits optimalen Softwareentwicklungsprozess nicht geben. In diesem Sinn ist das Ziel dieser Arbeit die Entwicklung eines situativen Koordinationsansatzes für die Standardsoftwareentwicklung. Dies erfordert die Beantwortung von vier Forschungsfragen, die sich aus dem Situativen Ansatz der Organisationsgestaltung ableiten und mithilfe einer sachlich-analytischen Forschungsstrategie bei der Konzeption, sowie Methoden der qualitativen empirischen Forschung bei der Umsetzung des situativen Koordinationsansatzes angegangen werden:

- (1) Die Frage, welches konkrete Ziel bei der Gestaltung der Koordination der Standardsoftwareentwicklung zu verfolgen ist, wird mit der Entwicklung des Koordinationseffektivität-Konzeptes beantwortet. Die Koordinationseffektivität wird dabei als ein Stimmigkeitskonstrukt definiert.
- (2) Die Frage, durch welche Gestaltungsmaßnahmen das verfolgte Gestaltungsziel herbeigeführt werden kann, wird ausgehend von der Betrachtung des Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess beantwortet. Es werden organisatorische Regelungen zur situativen Ausgestaltung der Informationsstruktur von Arbeitsergebnissen, zur situativen Verteilung von Entscheidungskompetenzen zwischen dem Projekt-, Qualitäts- und Produktmanagement, sowie zur situativen Festlegung der personellen und organisatorischen Unabhängigkeit des Projekt-, Qualitäts- und Produktmanagements in einem Standardsoftware-Entwicklungsvorhaben entwickelt.
- (3) Die Frage, an welchen Gestaltungsbedingungen die Koordination der Standardsoftwareentwicklung ausgerichtet werden muss, führt zur Erkenntnis, dass aufgrund der besonderen Ziel-Mittel-Beziehungen bei der Standardsoftwareentwick-

lung die Projektziele und dabei insbesondere die übergeordneten Projektzielpräferenzen die zentrale Gestaltungsbedingung darstellen. Die Ausgestaltung der Koordinationsprozesse muss dazu führen, dass die einzugehenden Kompromisse immer auf diesen Projektvorgaben basieren. Daher werden an dieser Stelle konkrete Kriterien zur Bestimmung der übergeordneten Projektzielpräferenzen abgeleitet.

- (4) Die Frage, wie die notwendige Stimmigkeit zwischen den Gestaltungsbedingungen und den Gestaltungsmaßnahmen hergestellt werden kann, führt zu einem Vorschlag zur organisatorischen Umsetzung des situativen Koordinationsansatzes in einer Standardsoftwareunternehmung. Die Umsetzung soll in Form der Koordination durch Programme und über acht definierte Projekttypen erfolgen.

Abstract

In software development, organization design is the discipline which at the same time is associated with the highest risks and the greatest room for improvement. A substantial part of organization design is coordination, i.e. the alignment of single activities in a system based on the division of labor with regard to a superordinate overall goal. Coordination problems are one of the main reasons for the often-mentioned software crisis and can have a particularly severe impact in large software development projects. Such projects are often found in the field of standard software—the subject area of the present work.

In the understanding of the present work, there cannot be a universal and by all means optimal software development process. Therefore, the objective of the present work is the development of a contingent coordination approach for the development of standard software. This requires answering four research questions derived from the contingency approach of organization design which are addressed by using an analytical research strategy for conception and qualitative empirical research methods for realization:

- (1) The question which specific goal to pursue when designing the coordination of standard software development is answered by the development of a coordination effectiveness concept. The effectiveness of coordination is here defined as a coherence construct.
- (2) The question by which design measures the anticipated design goal can be achieved is answered—based on the inspection of the standard software development process—as a continuous and conflictual goal formation process. Organizational regulations are developed for the contingent design of the information structure of work results, for the contingent allocation of decision-making authority between project management, quality management, and product management, and for the contingent regulation of personal and organizational independence of project management, quality management, and product management in a standard software development project.
- (3) The question towards which situational requirements the coordination of standard software development should be oriented leads to the insight that due to the specific means-ends relationships in standard software development, the project goals and here in particular the superordinate project goal preferences are the central situational requirement. The design of the coordination processes needs to ensure that

necessary compromises are always based on those project presettings. Therefore, concrete criteria for determining the superordinate project goal preferences are derived.

- (4) The question of how to achieve the necessary coherence between situational requirements and design measures leads to a proposal for the implementation of a contingent coordination approach in a standard software enterprise. The approach is meant to be implemented in the form of coordination through programs and through eight predefined project types.

1 Gegenstand und wissenschaftliche Vorgehensweise der Arbeit

1.1 Motivation

Sowohl empirische Befunde als auch die herrschende Meinung unter Fachleuten deuten darauf hin, dass die organisatorische Gestaltung zugleich der mit den größten Risiken als auch mit den größten Verbesserungspotentialen behaftete Gestaltungsbereich der Softwareentwicklung ist.¹ So führen Maxwell u.a. den größten Teil der Produktivitätsunterschiede bei den untersuchten Softwareentwicklungsprojekten in ihrer empirischen Studie auf Unterschiede in der organisatorischen Gestaltung zurück.² Damit geht einher, dass die fortwährenden Qualitäts-, Zeit- und Kostenprobleme der Softwareentwicklung durch Fortschritt in der Softwaretechnik allein, also bspw. bei Programmiersprachen³, nicht zu bewältigen sind, sondern neue Managementkonzepte erfordern.⁴ Diese Einschätzung findet ihren Niederschlag auch im höheren Stellenwert von Empfehlungen zur Organisationsgestaltung gegenüber den fachlich-technischen Empfehlungen in den Leitfäden des prozessorientierten Softwarequalitätsmanagements.⁵

Ein wesentlicher organisatorischer Gestaltungsbereich ist die Koordination. Sie bezeichnet das Ausrichten von Einzelaktivitäten in einem arbeitsteiligen System auf ein übergeordnetes Gesamtziel.⁶ Koordinationsprobleme sind einer der Hauptgründe für die vielfach zitierte Softwarekrise⁷ und kommen insbesondere bei umfangreichen Softwareentwicklungsvorhaben zu tragen. Denn mit steigender Projektgröße und -komplexität steigen implizit auch der Grad der Arbeitsteilung und das Ausmaß an Interdependenzen überdimensional an.⁸ Hieraus ergeben sich als Koordinationsprobleme v.a. die fehlende Transparenz von Entwicklungszielen, unzureichende Abstimmung der Beteiligten, ungelöste Zielkonflikte der Beteiligten und ungeklärte Verantwortlichkeiten.⁹ Sie führen allesamt zu einer Situation, bei der Organisationsmitglieder auf Grund mangelnder Abstimmung Entscheidungen nicht so treffen, wie es für die Organisation insgesamt optimal wäre.

¹ Vgl. Stelzer (1998), S. 105; vgl. Mellis (2004), S. 69, jeweils mit zahlreichen weiteren Verweisen.

² Vgl. Maxwell u.a. (1996), S. 706ff.; vgl. Stelzer (1998), S. 48

³ Vgl. Maxwell u.a. (1996), S. 706ff.; vgl. Stelzer (1998), S. 48

⁴ Vgl. Wildemann, Kersten (2000), S. 1215

⁵ Vgl. Mellis (2004), S. 70; vgl. Stelzer (1998), S. 106

⁶ Vgl. Albach (1967), zitiert nach Frese (2005), S. 143

⁷ Vgl. Standish Group International (Hrsg., 1994)

⁸ Vgl. Kraut, Streeter (1995), S. 69f.

⁹ Vgl. Herzwurm (2000), S. 173ff. Herzwurm identifiziert im Rahmen einer Sekundäranalyse zahlreiche (weitere) Problemfelder, die der Umsetzung von Kundenorientierung in der Softwareentwicklung entgegen stehen.

Ähnlich wie Maxwell u.a. teilen Kraut und Streeter die bisherigen Lösungsansätze zu der Softwarekrise in drei Gruppen ein:

- Technische Ansätze wie bspw. Programmiersprachen einer höheren Generation, um die Produktivität der Programmier zu erhöhen;
- Modularisierung auf technischer Ebene, bspw. durch Objektorientierung, sowie Modularisierung auf organisatorischer Ebene, bspw. durch klare Abgrenzungen von Zuständigkeiten – beides um das Ausmaß der notwendigen Koordination zu minimieren;
- Formale Prozeduren, sowohl technischer Art wie bspw. Spezifikationssprachen, als auch organisatorischer Art wie bspw. Testplanung.

Die genannten Ansätze haben zwar die Produktivität in einem gewissen Ausmaß erhöht, jedoch die Koordinationsprobleme nicht gelöst.¹⁰ Insbesondere sei erwähnt, dass das Ausmaß der notwendigen Koordination nur begrenzt durch Modularisierung gesenkt werden kann.¹¹ Es ist im Gegenteil zu erwarten, dass v.a. mit der stetig wachsenden Größe von Softwareentwicklungsvorhaben und dem steigenden Trend zur weltweit verteilten Softwareentwicklung die erforderliche Koordinationsintensität und damit potentiell auch Koordinationsprobleme zunehmen werden.¹²

Die bisher eingenommene Sicht auf die Koordination ist dem entscheidungsorientierten Konzept der Organisationsgestaltung zuzuordnen. Es dient dieser Arbeit als einer von zwei gleichberechtigten theoretischen Zugängen zur Problemstellung – neben dem Situativen Ansatz der Organisationsgestaltung: Hier wird davon ausgegangen, dass Organisationsstrukturen bzw. (organisatorische) Gestaltungsmaßnahmen je nach Situation unterschiedlich effizient und effektiv sind und somit so gewählt werden müssen, dass sie der jeweiligen Situation gerecht werden.¹³ Die Situation definiert sich dabei durch Rahmenbedingungen wie bspw. Projektgröße und durch Gestaltungsziele wie bspw. Minimierung der Entwicklungsdauer.¹⁴ Entsprechend wird in dieser Arbeit in Anlehnung an Balzert unter dem Software Engineering die „zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen“¹⁵ verstanden. Zielorientiert bedeutet hierbei, dass in Abhängigkeit

¹⁰ Vgl. Kraut, Streeter (1995), S. 70

¹¹ Vgl. Kap. 2.2.2.3

¹² Vgl. Ramesh u.a. (2006), S. 41; vgl. Royce (2005), S. 43

¹³ Vgl. Kieser, Walgenbach (2003), S. 43f.

¹⁴ Vgl. Kap. 2.1.1

¹⁵ Balzert (2001), S. 36

von den Entwicklungszielen¹⁶ unterschiedliche Prinzipien, Methoden und Werkzeuge geeignet sind.¹⁷

Nach diesem Verständnis liegt für ein Softwareentwicklungsvorhaben die zentrale und grundsätzliche Herausforderung in der Auswahl des problemadäquaten Softwareentwicklungsprozesses, da es weder einen universellen Entwicklungsprozess, noch eine allseits optimale Entwicklungsmethode geben kann.¹⁸ Bereits ein solches situatives Verständnis der Koordination adressiert bestehenden Forschungsbedarf. Denn in der Forschung zu Innovationsprozessen, wozu auch der Softwareentwicklungsprozess gezählt werden muss, wurden situative Faktoren eines Entwicklungsprojektes bisher kaum als Determinante für den Einsatz von Koordinationsinstrumenten betrachtet.¹⁹ Unter diese komplexe Problemstellung ist auch die situationsgerechte organisatorische Gestaltung der Koordination von Entwicklungsvorhaben im Bereich von Standardsoftware einzuordnen, welche im Mittelpunkt der vorliegenden Arbeit steht. Standardsoftware bezeichnet dabei Software, welche zu einem definierten Anwendungsgebiet für den anonymen Markt erstellt wird und über diesen als ein vorgefertigtes Softwareprodukt erworben werden kann.²⁰

1.2 Ziel der Arbeit

In den Realwissenschaften lassen sich Zielsetzungen einer Forschungsarbeit in Erkenntnis- und Gestaltungsziele unterscheiden. Erkenntnisziele heben auf das Verständnis gegebener Sachverhalte ab. Reale Sachverhalte sollen hier zutreffend beschrieben (deskriptives Erkenntnisziel) und erklärt werden (theoretisches Erkenntnisziel). Diese Erkenntnisgewinnung stellt die Grundlage für die Gestaltung dar, die ohne Erklärung prinzipiell nicht möglich ist. Gestaltungsziele betreffen die Veränderung bestehender und Schaffung neuer realer Sachverhalte, um ein bislang unzureichend geklärtes Realweltproblem zu lösen.²¹ Entsprechend können einerseits erkennende, d.h. auf die Beschreibung, Erklärung und Verständnis der Realität gerichtete, sowie andererseits gestaltende, d.h. auf die Prognose, Veränderung oder Schaffung der Realität gerichtete Forschungsarbeiten unterscheiden werden.²²

¹⁶ Bspw. Entwicklungsdauer, Entwicklungskosten, Produktqualität; vgl. hierzu ausführlich in Kap. 2.3.3

¹⁷ Vgl. Balzert (2001), S. 36; vgl. Lang (2004), S. 23f.

¹⁸ Vgl. Liggesmeyer, Rombach (Hrsg., 2005), S. 15; vgl. Wildemann, Kersten (2000), S. 1220; vgl. Xu, Ramesh (2007), S. 294f.

¹⁹ Vgl. Lühring (2006), S. 97

²⁰ Vgl. zur Abgrenzung von Standardsoftware und Standardsoftwareentwicklung im Detail Kap. 2.1.2.2

²¹ Vgl. Zelewski (2008), S. 24; vgl. Heinrich u.a. (2007), S. 18f.

²² Vgl. Müller-Böling (1992), Sp. 1491f.

Eine Forschungsarbeit muss sich stets auf einen Ausschnitt der Realität beschränken. In dieser Arbeit ist das Erkenntnisziel dem Gestaltungsziel untergeordnet. Es liegt in der Verständnisgenerierung für koordinationsrelevante Sachverhalte bei der Standardsoftwareentwicklung. Diese im Hinblick auf das Gestaltungsziel notwendige Vorarbeit erfolgt aus Sicht des Situativen Ansatzes (Kap. 2.1) sowie aus Sicht des entscheidungsorientierten Konzeptes der Organisationsgestaltung (Kap. 2.2).

Das Gestaltungsziel dieser Arbeit ist die Entwicklung eines situativen Koordinationsansatzes für die Standardsoftwareentwicklung.

An (theoretische) Aussagen, wie sie bei der Konzeption des Ansatzes in Kap. 2.3 generiert werden, stellt Grochla drei Anforderungen: Konzeptionelle Präzisierung, entscheidungstechnische Verwendbarkeit und empirische Absicherung der Aussagen.²³ Forschungsarbeiten orientieren sich in der Regel an allen drei Anforderungen, allerdings mit erheblich wechselnden Gewichtungen und oft mit Einseitigkeiten.²⁴ Diese Arbeit strebt bei der Konzeption des Ansatzes vor allem eine hohe konzeptionelle Präzisierung und die direkte entscheidungstechnische Verwendbarkeit der Ergebnisse an:

- Die konzeptionelle Präzisierung erfordert eine begriffliche Ausarbeitung und präzise Aufschlüsselung aller einbezogenen Variablenkategorien und deren Zusammenhänge. Es sind möglichst alle forschungs- und handlungsrelevanten Größen sowie auch Hintergrundfaktoren und allgemeine relevante Zusammenhänge aufzunehmen. Die Basis hierzu wird bereits durch das Erkenntnisziel in den Kap. 2.1 und 2.2 gelegt.
- Die entscheidungstechnische Verwendbarkeit der Konzeption soll die Einsetzbarkeit in der Praxis bei organisatorischen Entscheidungen sicherstellen. Aus den Variablen und ihren Zusammenhängen sollen konkrete Entscheidungshilfen entwickelt werden können. Die entscheidungstechnische Verwendbarkeit steht in Kap. 2.3 im Vordergrund, da hieraus in Kap. 3 konkrete Entscheidungshilfen für die Praxis entwickelt werden.

Der in Kap. 3 umgesetzte situative Koordinationsansatz für die Standardsoftwareentwicklung soll

- in der Praxis einsetzbar sein,
- die bisher unzureichend gewährleistete Koordinationsfähigkeit als identifiziertes Realweltproblem bei der Standardsoftwareentwicklung erhöhen,

²³ Vgl. hier und im Folgenden Grochla (1978), S. 56, 66

²⁴ Vgl. Grochla (1978), S. 68

- stets den Anforderungen der Situation gerecht werden, d.h. situativ im Sinne des Situativen Ansatzes der Organisationsgestaltung sein, sowie
- auf systematisch hergeleiteten und wissenschaftlich begründeten Erkenntnissen und Gestaltungsvorschlägen basieren.

Das Gestaltungsziel der Arbeit erfordert die Beantwortung von vier Forschungsfragen. Sie leiten sich unmittelbar aus dem Situativen Ansatz der Organisationsgestaltung ab,²⁵ der hier den Rahmen für die Konzeptualisierung und Umsetzung stellt. Folglich müssen die Forschungsfragen, ausgenommen Forschungsfrage 1, jeweils zunächst auf konzeptioneller Ebene (Kap. 2.3) sowie anschließend hinsichtlich der Umsetzung in der Praxis (Kap. 3) beantwortet werden (vgl. auch anschließend in Abb. 1):

- 1) Bei einer Forschungsarbeit, die dem Situativen Ansatz folgt, steuern Gestaltungsziele den gesamten Forschungsprozess und nehmen daher eine zentrale und den übrigen Größen vorgelagerte Stellung ein. Es ist zunächst zu beantworten, welches konkrete Ziel bzw. welche angestrebte Wirkung bei der Gestaltung der Koordination der Standardsoftwareentwicklung vorrangig zu verfolgen ist (Kap. 2.3.1).
- 2) Es ist sodann zu erforschen, durch welche Gestaltungsmaßnahmen das verfolgte Gestaltungsziel bzw. die angestrebte Wirkung herbeigeführt werden kann (Kap. 2.3.2 und Kap. 3.2.3).
- 3) Gestaltungsbedingungen (situative Bedingungen) stellen zum einen bestimmte Anforderungen an die Ausprägung von Gestaltungsmaßnahmen dar und besitzen zum anderen Einfluss auf die Wirkung dieser Gestaltungsmaßnahmen. Es ist daher zu fragen, an welchen situativen Bedingungen die Gestaltung der Koordination der Standardsoftwareentwicklung ausgerichtet werden muss, um das Gestaltungsziel zu erreichen (Kap. 2.3.3 und Kap. 3.3).
- 4) Die Ableitung der Gestaltungsmaßnahmen (Forschungsfrage 2) und der Gestaltungsbedingungen (Forschungsfrage 3) beantwortet nicht eine weitere, notwendige Frage, nämlich wie die notwendige Stimmigkeit zwischen der Situation (Gestaltungsbedingungen) und der Struktur (Gestaltungsmaßnahmen) hergestellt werden kann. Hierauf zielt die vierte Forschungsfrage hinsichtlich der organisatorischen Umsetzung des situativen Koordinationsansatzes in einer Standardsoftwareunternehmung ab (Kap. 2.3.4 und Kap. 3.4).

²⁵ Konkreter: Die Forschungsfragen leiten sich aus dem Grundmodell der sog. pragmatischen bzw. handlungsbezogenen Variante des Situativen Ansatzes ab. Vgl. hierzu Kap. 2.1.1

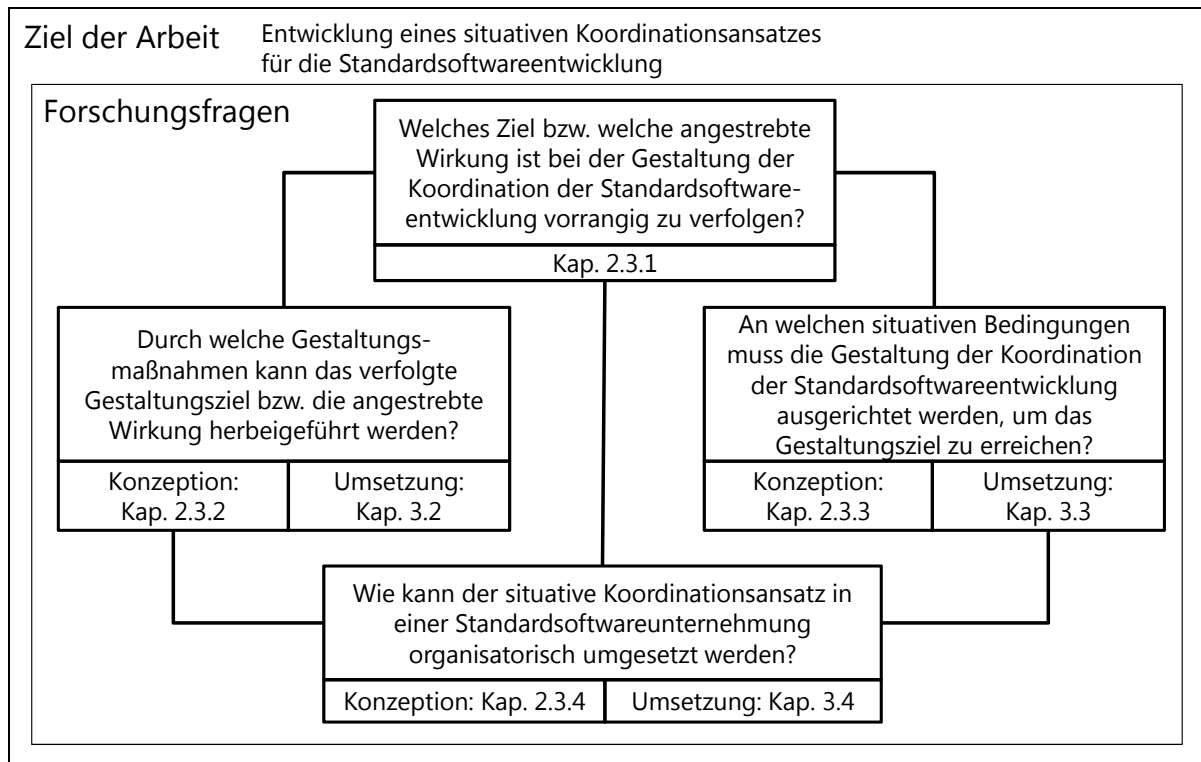


Abb. 1: Ziel und Forschungsfragen der Arbeit

1.3 Forschungsstrategie

Weder in der Wirtschaftsinformatik noch in ihrer Teildisziplin Software Engineering²⁶ haben sich bisher Theorien oder Forschungsansätze herausgebildet, mit denen die vorliegende Zielsetzung angegangen werden könnte. Für die beiden naheliegendsten in Frage kommenden Ansätze Value Based Software-Engineering (VBSE) und Situational Method Engineering (SME) soll dies im Folgenden kurz erläutert werden:

- VBSE befasst sich zwar mit koordinationsrelevanten Ansätzen und ist zudem auch geschichtlich in der Übertragung von betriebswirtschaftlichen Erkenntnissen auf die Softwareentwicklung verwurzelt.²⁷ Auf der anderen Seite ist VBSE jedoch stark Praxis- bzw. empirisch getrieben und kann nicht auf ein stimmiges theoretisches und forschungsstrategisches Grundgerüst zurückgreifen: Kern des Ansatzes sind die ‚4+1-Theorie‘ (vgl. Abb. 2) sowie eine grobe Forschungsagenda.²⁸ VBSE ist damit zumindest zurzeit mehr ein laufendes Forschungsprogramm als eine konzeptionell präzise, entscheidungstechnisch verwendbare und empirisch abgesicherte Theorie.

²⁶ Software Engineering als wissenschaftliche Disziplin beschäftigt sich zwar systematisch mit Fragen der Softwareentwicklung, allerdings stehen dabei v.a. technische Aspekte im Vordergrund. Software Engineering gilt dabei sowohl als Teilgebiet der Informatik als auch der Wirtschaftsinformatik; Vgl. Stelzer (1998), S. 19, mit weiteren Verweisen.

²⁷ Vgl. Boehm (2006), S. 7f.

²⁸ Vgl. Boehm (2006), S. 8f.

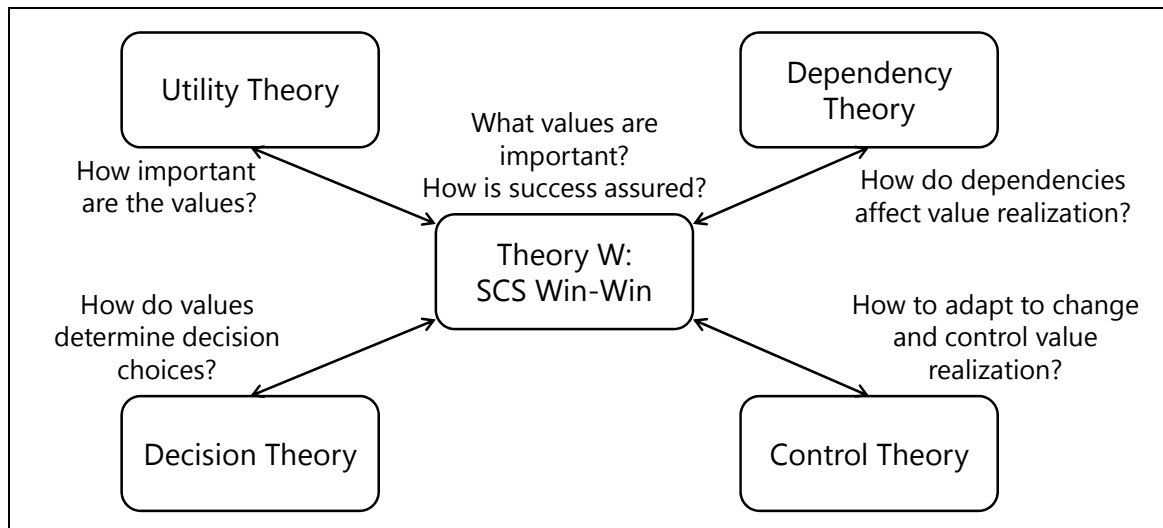


Abb. 2: Übergeordnete Struktur der ‚4+1-Theorie‘ von VBSE²⁹

- SME befasst sich in Analogie zum Situativen Ansatz der Organisationsgestaltung mit dem Prozess der Konstruktion und Adaption von Methoden und Prozessschritten für den Softwareentwicklungsprozess. Sie werden dabei aus einem Prozessrahmenwerk, bestehend aus einem Metamodell, Prozesskomponenten und Verfahrensanweisungen zur Methodenkonstruktion, generiert und müssen auf die situativen Spezifika von Softwareentwicklungsprojekten ausgerichtet bzw. ausrichtbar sein.³⁰ Zur besseren praktischen Anwendbarkeit und Automatisierung wird versucht, die Prozessrahmenwerke als CASE-Tools³¹ umzusetzen.³² Für die Zielsetzung der vorliegenden Arbeit ist der Ansatz SME als zugrundegelegte Theorie nicht geeignet, da er sich mit rein softwaretechnischen Fragestellungen auseinandersetzt. Seine Forschungsschwerpunkte liegen v.a. in der Referenzmodellierung von Prozessen und der Metamodellierung von Methoden. Gestaltungsziele sind v.a. die Entwicklung von wiederverwendbaren Methoden- und Prozessbeschreibungen sowie Beschreibungsschablonen und Sprachen für die Modellierung.³³ Im deutschsprachigen Raum befasst sich bspw. die Arbeit von Münch mit der Entwicklung einer Methodik zur musterbasierten, wiederverwendungsorientierten Erstellung von Vorgehensplänen für die Softwareentwicklung. Sie kapselt Softwareentwicklungsprozesse in Form von wiederverwendbaren Prozessmustern, deren Zusammenstellung und Anpassung durch CASE-Tools unterstützt wird.³⁴

²⁹ Boehm, Jain (2006), S. 18

³⁰ Vgl. Bucher u.a. (2007), S. 33f., mit weiteren Verweisen.

³¹ CASE-Tools sind Softwareprodukte zur Unterstützung einzelner oder mehrerer Entwicklungsschritte im Softwareentwicklungsprozess.

³² Vgl. Nguyen, Henderson-Sellers (2003), S. 7ff.

³³ Vgl. Ralyte u.a. (Hrsg., 2007)

³⁴ Vgl. Münch (2002), S. V, 2f.

Für die Forschungsstrategie dieser Arbeit ergeben sich aus dem Fehlen geeigneter Theorien und Forschungsansätze in der Wirtschaftsinformatik und im Software Engineering drei grundsätzliche Implikationen:

- Die vorliegende Arbeit verfolgt eine Explorationsstrategie. Ob ein Forschungsvorhaben primär zur Erkundung oder zur Überprüfung von Hypothesen durchgeführt wird, richtet sich nach dem Stand der Forschung im zu erforschenden Problemfeld. Vorhandene Theorien, welche die Ableitung von Hypothesen bereits zu Beginn des Forschungsvorhabens zulassen, erfordern eine Hypothesen-prüfende Untersuchung. Fehlen dagegen geeignete Theorien oder Forschungsansätze für die vorliegende Zielsetzung, so sind zunächst Untersuchungen notwendig, welche die Formulierung neuer Hypothesen erleichtern.³⁵ In diesem Sinn ist die explorative Forschungsstrategie bemüht, den Entdeckungszusammenhang stärker unter methodische Regeln zu fassen anstatt beliebige Hypothesen stringent zu prüfen. Schlussendlich soll über das erstmalige Erkennen von Zusammenhängen zu sinnvollen neuen Hypothesen gelangt werden.³⁶
- Der theoretische Zugang zu der Problemstellung erfolgt über Theorien der Organisationslehre. Diese gilt allgemein hin als das Teilgebiet der Betriebswirtschaftslehre mit den meisten Anknüpfungspunkten zur Wirtschaftsinformatik.³⁷ Die Koordination wird im Rahmen der Organisationslehre zwar nicht softwarespezifisch behandelt, für die Gestaltung der Softwareentwicklung liefert sie dennoch relevante Ansatzpunkte. Konkret erfolgt hier der theoretische Zugang zur Problemstellung über den Situativen Ansatz der Organisationsgestaltung (Kap 2.1) und über das entscheidungsorientierte Konzept der Organisationsgestaltung (Kap. 2.2).
- Die Forschungsstrategie richtet sich konsequenter Weise an Methoden der Organisationsforschung aus. Insbesondere bei der Zugrundelegung des Situativen Ansatzes spielt der Entwurf eines sog. Bezugsrahmens eine erhebliche Rolle, weil hier eine größere Zahl an konzeptionellen Größen und Beziehungszusammenhängen in die Untersuchung einbezogen werden muss.³⁸ Der Begriff der Situation ist ein offenes Konzept, das in Abhängigkeit von der jeweiligen Fragestellung und dem jeweiligen Wissen mit konkretem Inhalt gefüllt werden muss.³⁹ Die Bezugsrahmenforschung als Methode der Organisationsforschung kommt in dieser Arbeit wie folgt zum Tragen.

³⁵ Vgl. Bortz Döring (2006), S. 30

³⁶ Vgl. Wollnik (1977), S. 43

³⁷ Vgl. König u.a. (1995)

³⁸ Vgl. Wolf (2008), S. 37

³⁹ Vgl. Kieser, Walgenbach (2003), S. 213

Zunächst wird in Kap. 2 der Arbeit ein Konzeptionsrahmen entwickelt (Kap. 2.4 im Überblick). Dazu werden mit Hinblick auf das Ziel der Arbeit in den Kap. 2.1 und 2.2 die aus Sicht des Situativen Ansatzes und des entscheidungsorientierten Konzeptes der Organisationsgestaltung relevanten Merkmale (konzeptionelle Größen) bzw. Merkmalskomplexe des Gegenstandsbereiches beschrieben. Hier werden v.a. begriffliche und deskriptive Aussagen gemacht. Die Konzeption des Ansatzes erfolgt in Kap. 2.3, indem bestimmte Annahmen über die Beziehungen zwischen den konzeptionellen Größen vorgenommen werden und diese systematisiert werden (Systematisierungsleistungen). Hierdurch ist es möglich, zu explanatorischen Aussagen zu gelangen.⁴⁰

Bei der Entwicklung des Konzeptionsrahmens wird die sachlich-analytische Forschungsstrategie verfolgt. Sie lässt sich dadurch kennzeichnen, dass die Informativität und die entscheidungstechnische Verwendbarkeit, nicht jedoch eine empirische Bestätigung der Aussagen, angestrebt werden.⁴¹ Die Zusammenhänge werden durch Plausibilitätsüberlegungen und eventuelle, empirisch festgestellte Teilzusammenhänge gestützt. Dabei wird keine eigene systematische empirische Überprüfung der Aussagen angestrebt, d.h. der Forscher begründet kein explizites empirisches Prüfinteresse.⁴² „Vielmehr werden bestimmte grundlegende, vermutete oder subjektiv gesehene Zusammenhänge als evident oder aufgrund von Erfahrungen als existent betrachtet und nicht systematisch mit der Realität konfrontiert. Weitere Zusammenhänge werden dann durch Folgerungen erschlossen.“⁴³

In Kap. 3 wird der Konzeptionsrahmen zu einem sog. Entscheidungsrahmen weiterentwickelt, der sich zur Unterstützung realer organisatorischer Problemlösungen heranziehen lässt. „Die theoretischen Aussagen müssen nämlich insbesondere auf praktische Problemstellungen hin ausgerichtet werden, d.h. es besteht die äußerst wichtige und dabei keineswegs leicht zu lösende Aufgabe, begriffliche, beschreibende und erklärende theoretische Aussagen im Hinblick auf ein praktisches Problem auszuwerten und damit ihre vorerst nur implizite praktische Fruchtbarkeit auszuschöpfen.“⁴⁴

Ein Entscheidungsrahmen stellt nach der Entwicklung von begrifflichen, beschreibenden und erklärenden theoretischen Aussagen die vierte Stufe der Theorieentwicklung dar. Hier folgt der Übergang zu sog. praxeologischen Aussagen, d.h. zu solchen Aussagen, die dem Entscheidungsträger unmittelbare Hilfestellungen für seine Problemlösung bieten. Praxeologische Aussagen stellen zugleich instrumentale Aussagen dar

⁴⁰ Vgl. Grochla (1978), S. 62f., 69f.

⁴¹ Vgl. analog Ziel der Arbeit

⁴² Vgl. Grochla (1978), S. 72-74

⁴³ Grochla (1978), S. 72

⁴⁴ Grochla (1978), S. 63

und müssen daher bestimmten inhaltlichen Anforderungen genügen: Sie müssen angeben, unter welcher Zielsetzung (Kap. 2.3.1; Konzeptionsrahmen) welche Gestaltungsmaßnahmen (Kap. 3.2) unter den jeweils herrschenden Bedingungen (Kap. 3.3) bei Berücksichtigung der Wirkungen (Kap. 3.4; organisatorische Umsetzung) ergriffen werden können (Kap. 3.5 im Überblick).⁴⁵

Mit dem Übergang vom Konzeptions- zum Entscheidungsrahmen erfolgt auch der Übergang von der sachlich-analytischen zur empirischen Forschungsstrategie.⁴⁶ Da diese Arbeit eine Explorationsstrategie verfolgt, wird auf Methoden der qualitativen empirischen Forschung zurückgegriffen und nicht auf die für den Situativen Ansatz typischen quantitativen Methoden der empirisch vergleichenden Organisationsforschung.⁴⁷ Die qualitative empirische Forschung⁴⁸ betont die Explorationsfunktion, indem auf eine Hypothesenbildung ex ante verzichtet wird. Sie versteht sich im Gegensatz zur quantitativen empirischen Forschung nicht als ein hypothesenprüfendes, sondern als ein hypothesengenerierendes Verfahren. Im Unterschied zur quantitativen Auffassung ist die Hypothesengenerierung im qualitativen Paradigma also ein konstitutives Element des Forschungsprozesses.⁴⁹

Für den Forschungsprozess ist bei der qualitativen empirischen Forschung „weniger der breit angelegte Vergleich organisationaler Wirkungsmechanismen als vielmehr das Eindringen in die Tiefen des Einzelfalls [das Ziel] Der Blick für das ‚Unbekannte im scheinbar Bekannten‘“⁵⁰ soll freigegeben werden. Daher wird der Entscheidungsrahmen mithilfe einer umfassenden Einzelfallstudie entwickelt. Der Vorteil der Einzelfallstudie liegt darin, dass der Forscher durch die Beschränkung auf ein Untersuchungsobjekt genaueren Einblick in das Zusammenwirken einer Vielzahl von Faktoren erhalten kann. So ist es dem Forscher möglich, zu umfangreicheren und komplexeren Ergebnissen zu kommen.⁵¹

Die Einzelfallstudie gibt an sich noch keine Auskunft darüber, mit welchen konkreten Forschungsmethoden der Forschungsprozess stattfindet. Sie ist keine spezifische und isolierte Forschungsmethode, sondern schon von ihrem Anspruch her multimethodisch

⁴⁵ Vgl. Grochla (1978), S. 55, 63, 70f.

⁴⁶ Vgl. Grochla (1978), S. 78ff.

⁴⁷ V.a. Querschnittsanalysen bzw. Korrelationsanalysen; vgl. Bea, Göbel (2006), S. 41, 116

⁴⁸ Bei den Methoden der qualitativen empirischen Forschung handelt es sich genau genommen um Methoden der qualitativen Sozialforschung. Die Methoden der empirischen Organisationsforschung sind aber grundsätzlich mit den Methoden der empirischen Sozialforschung identisch (vgl. Staehle (1977), S. 105). Daher wird nachfolgend auf den expliziten Hinweis auf die Sozialforschung verzichtet.

⁴⁹ Vgl. Lamnek (2005), S. 21, 26f., 89

⁵⁰ Strodtholz, Kühl (2002), S. 16

⁵¹ Vgl. Lamnek (2005), S. 300f.

anzulegen.⁵² In dieser Arbeit besteht die Einzelfallstudie aus einer umfassenden Dokumentenanalyse und aus mehreren Experteninterviews.⁵³ Unter den vielfältigen Verfahren, die im Bereich der Organisationsanalyse Verwendung finden, kommt dem Experteninterview und der Dokumentenanalyse eine Vorrangstellung zu.⁵⁴

- Mittels Dokumentenanalyse ist es möglich, über Rollen- und Tätigkeitsbeschreibungen, Prozessbeschreibungen, etc. die Strukturen und Prozesse einer Organisation zu erfassen und auf das Untersuchungsziel hin zu verwenden;
- Experteninterviews sind eine geeignete Methode, um neue Einblicke in Forschungsfelder zu gewinnen, ohne konzeptuelle Vorüberlegungen außen vor lassen zu müssen. Indem theoretisches Vorwissen, bspw. wie hier aus der Entwicklung eines Konzeptionsrahmens heraus, offen gelegt und in den Fragekomplexen des Interviews zum Ausdruck gebracht wird, besteht die Möglichkeit, neu gewonnene Erkenntnisse in den Forschungsprozess wieder einzubringen.

Abbildung 3 fasst die Forschungsstrategie dieser Arbeit zusammen.

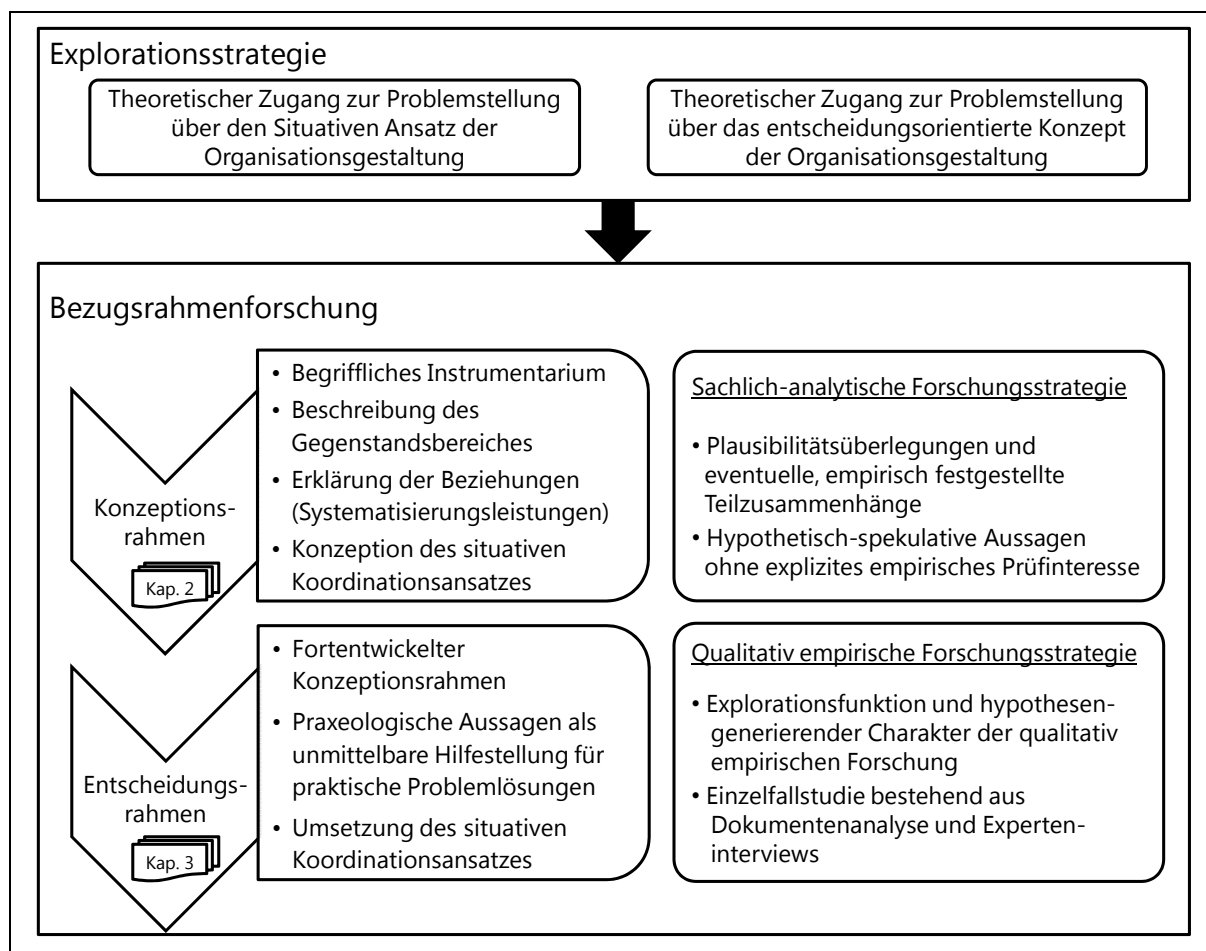


Abb. 3: Forschungsstrategie der Arbeit

⁵² Vgl. Lamnek (2005), S. 298-301

⁵³ Zum Fallstudiendesign im Detail vgl. Kap. 3.1

⁵⁴ Vgl. Strodtholz, Kühl (2002), S. 18f.; vgl. Liebold, Trinczek (2002), S. 66f.

1.4 Wissenschaftstheoretische und -methodologische Basis

Sowohl der qualitativen empirischen Forschung, als auch der Organisations- und insb. der Bezugsrahmenforschung, liegt hermeneutisches Denken und Handeln zugrunde. Bei diesem wird das Vorverständnis des Forschers zu Beginn offen gelegt und am Forschungsgegenstand weiterentwickelt. Dieses iterative Vorgehen ist bekannt als hermeneutischer Zirkel bzw. hermeneutische Spirale.⁵⁵ Diese gemeinsame wissenschaftstheoretische Basis führt auch zu Parallelen in den Forschungsstrategien der beiden Forschungsgebiete.

So ist für die qualitative empirische Forschung der Gebrauch wenig spezifizierter Konzepte zu Beginn der Untersuchung charakteristisch. Sie werden ganz in Analogie zu Bezugsrahmen in der Organisationsforschung als ‚Orienting Concepts‘ oder ‚Sensitizing Concepts‘ bezeichnet.⁵⁶ Auf der anderen Seite schlägt sich die Hermeneutik in der Organisationsforschung an vielen Stellen nieder, auch wenn sie selten explizit erwähnt wird. So u.a.

- im Wissen, dass empirische Daten nicht für sich sprechen, sondern vom Forscher erst interpretiert werden müssen,
- in der Annahme, dass jeder Forscher mit einem Vorverständnis an seinen Untersuchungsgegenstand herangeht, welches auch die weitere Forschung wesentlich prägt,
- und schließlich allgemein in der Anwendung qualitativer Forschungsmethoden.⁵⁷

Insbesondere aber äußert sich hermeneutisches Denken in der Organisationsforschung in der Entwicklung von Bezugsrahmen an sich. Die Bezugsrahmenforschung versteht den Forschungsprozess als einen erfahrungsgestützten Lernprozess mit theoretischen Absichten, der eine iterative Heuristik entfalten soll. Ausgehend von heuristischen Bezugsrahmen wird Erfahrungswissen gewonnen, das zu einer Modifikation dieses Bezugsrahmens und zur Formulierung weiterführender Fragen leitet. Diese erfordern wiederum die Gewinnung von neuem Erfahrungswissen. Der gesamte Forschungsprozess ist dabei durch eine ganze Reihe solcher Zyklen gekennzeichnet.⁵⁸

So sind auch zwischen den verschiedenen Theorieentwicklungsstufen im Verlauf des Forschungsprozesses Iterationen und Interdependenzen vorhanden. Es bestehen zum einen Vorkopplungsbeziehungen: Bereits die Begriffsbildung weist in Richtung bestimmter Deskriptionsleistungen. Die Deskription wiederum muss derart angelegt sein,

⁵⁵ Vgl. Mayring (2002), S. 30

⁵⁶ Vgl. Lamnek (2005), S. 120

⁵⁷ Vgl. Bea, Göbel (2006), S. 41f.

⁵⁸ Vgl. Kubicek (1977), S. 13, 28

dass auf ihrer Grundlage erklärende Aussagen möglich werden. Zum anderen bestehen zwischen den Stufen der Theorieentwicklung auch Rückkopplungsbeziehungen. Bspw. gehen der Begriffsbildung oft deskriptive und explanatorische Aussagen voraus. Eine geeignete Deskription kann häufig nur auf der Grundlage von Beziehungshypothesen als explanatorische Aussagen erfolgen und wird häufig derart vorgenommen, dass eine klare Trennung zwischen Deskription und Erklärung nicht gegeben ist.⁵⁹

Ferner bestehen bei der Weiterentwicklung von Konzeptionsrahmen zu Entscheidungsrahmen Interdependenzen. V.a. die Identifizierung von Gestaltungsmaßnahmen hat Rückwirkungen auf die zu erbringenden theoretischen Vorleistungen. Als Gestaltungsmaßnahmen können nämlich nur solche Größen ausgesucht werden, die bereits im Konzeptionsrahmen berücksichtigt worden sind. „Diese Verzahnung macht es im Grunde erforderlich, dass bei der Entwicklung von Konzeptionsrahmen praktische Problemstellungen, deren Lösungen die Theorie zu unterstützen versucht, den Ausgangspunkt bilden und zumindest teilweise antizipiert werden.“⁶⁰

Neben der Hermeneutik als wissenschaftstheoretische Basis liegt der vorliegenden Arbeit als übergeordnete methodische Basis die Triangulation zugrunde. Triangulation bezeichnet in der qualitativen Forschung die Einnahme unterschiedlicher Perspektiven auf einen untersuchten Gegenstand und bei der Beantwortung von Forschungsfragen. Diese Perspektiven äußern sich in der Anwendung unterschiedlicher Forschungsmethoden und in unterschiedlichen gewählten theoretischen Zugängen, wobei beides miteinander in Zusammenhang steht und verknüpft werden sollte. Die Perspektiven sollten soweit als möglich gleichberechtigt und gleichermaßen konsequent behandelt und umgesetzt werden. So wird prinzipieller Erkenntniszuwachs ermöglicht, der weiter reicht, als es mit einem Zugang möglich wäre.⁶¹

Im Konzeptionsrahmen dieser Arbeit (Kap. 2) steht v.a. die Theorietriangulation im Vordergrund. Die beiden gewählten theoretischen Zugänge sind der Situative Ansatz (Kap. 2.1) und das entscheidungsorientierte Konzept der Organisationsgestaltung (Kap. 2.2). Ihre Verknüpfung zum situativen Koordinationsansatz der Standardsoftwareentwicklung erfolgt in Kap. 2.3. Die Vorrangstellung der Theorietriangulation im Konzeptionsrahmen ergibt sich aus zwei Gründen:

- Ein Bezugsrahmen, der für die Analyse von Zusammenhängen nur eine einzige Theorie zugrundelegt, ist in seinem heuristischen Potential von vornherein begrenzter als ein Bezugsrahmen, dessen Inhalte aus mehreren, sich u. U. sogar wi-

⁵⁹ Vgl. Grochla (1978), S. 97-99

⁶⁰ Vgl. Grochla (1978), S. 63f.

⁶¹ Vgl. Flick (2004), S. 12, 25f.

dersprechenden Theorien, stammen. Daher sollte ein heuristischer Bezugsrahmen pluralistisch sein – er sollte Annahmen, Fragen und Beziehungszusammenhänge aus möglichst unterschiedlichen Richtungen, Ansätzen und Forschungsdisziplinen gegenüberstellen;

- Der Situative Ansatz der Organisationsgestaltung ist nicht als Theorie im konventionellen Sinn aufzufassen, die eine Vielzahl begründeter und miteinander verknüpfter Annahmen umfasst. Vielmehr handelt es sich hierbei um eine Metatheorie, mit der ‚Phänomene‘ konzeptualisiert werden können. Es wird damit ein Rahmen bereitgestellt, der bei der Formulierung und Strukturierung von Forschungsfragen eine handlungsleitende Funktion übernehmen kann.⁶² In diesem Sinn wird der Situative Ansatz auch in der vorliegenden Arbeit verwendet: Die Koordination der Standardsoftwareentwicklung wird in Kap. 2.3 im Rahmen des Situativen Ansatzes der Organisationsgestaltung konzeptualisiert.

Der Entscheidungsrahmen und die empirische Forschungsstrategie in Kap. 3 sind dagegen durch die Methodentriangulation gekennzeichnet.⁶³ Damit ist generell die Verbindung verschiedener Methoden aus unterschiedlichen Forschungsansätzen, aber innerhalb der qualitativen Forschung, gemeint. Die Triangulation verschiedener qualitativer Methoden bringt dann mehr Erkenntnis, wenn die kombinierten methodischen Zugänge unterschiedliche Perspektiven eröffnen und der Erkenntnisgewinn gegenüber der Anwendung einer Einzelmethode systematisch erweitert wird.⁶⁴ Methodentriangulation ist insbesondere in Fallstudien als multimethodischer Ansatz schon alleine von ihrem eigenen Anspruch her obligatorisch. So kann eine Einzelfallstudie unter Einbeziehung unterschiedlicher Datensorten, durch die Verwendung verschiedener Forschungsmethoden, oder auch unter Rückgriff auf unterschiedliche theoretische Ansätze durchgeführt werden.⁶⁵

Wie bereits erläutert wird innerhalb der Fallstudie auf die beiden Methoden Experteninterview und Dokumentenanalyse zurückgegriffen. Diese Methoden liefern jeweils Daten auf unterschiedlichen Ebenen und aus unterschiedlichen Perspektiven und sind erst im Zusammenspiel eine verlässliche Datenquelle. Denn organisatorische Regeln als Ergebnis einer Dokumentenanalyse sind interpretationsnotwendig. Sie stellen lediglich eine Art von Sollvorstellungen dar, über deren Gültigkeit noch nichts ausgemacht ist und von denen das faktische Handeln abweichen kann. Auf der anderen Seite

⁶² Vgl. Jenner (2001), S. 80

⁶³ Wobei bereits die Kombination der sachlich-analytischen mit der empirischen Forschungsstrategie an sich Triangulation darstellt.

⁶⁴ Vgl. Flick (2004), S. 41, 48f.

⁶⁵ Vgl. Flick (2004), S. 88

sind schriftlich dokumentierte Regeln ein greifbarer Ansatzpunkt für die Ermittlung organisatorischer Sachverhalte. Im Umkehrschluss sollte die formale Organisationsstruktur aber auch nicht nur mittels Experteninterviews erforscht werden, ohne dass organisatorische Dokumente hinzugezogen werden. Das Studium organisatorischer Dokumente muss also grundsätzlich um die Analyse ihrer Anwendung im Unternehmen erweitert werden und umgekehrt. Die Erfassung von schriftlichen Organisationsregeln kann danach nur den Anfang einer Organisationsanalyse darstellen und erfordert, wie in der vorliegenden Arbeit im Zusammenspiel mit Experteninterviews, eine gültige Interpretation der Inhalte unter Hinzuziehung von Einsichten in die tatsächlich ablaufenden Prozesse.⁶⁶

Abbildung 4 fasst die wissenschaftstheoretische und -methodologische Basis dieser Arbeit zusammen.

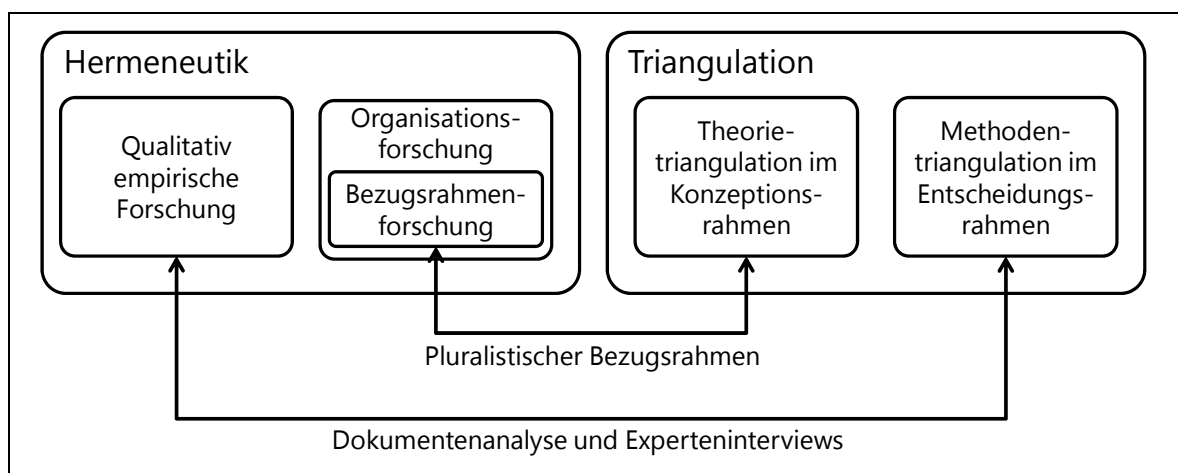


Abb. 4: Wissenschaftstheoretische und -methodologische Basis der Arbeit

1.5 Stand der Forschung

Die übliche organisatorische Form eines Softwareentwicklungsvorhabens ist die Projektorganisation. So erfolgt auch die Entwicklung von Standardsoftware innerhalb oder in Zusammenspiel von mehreren Produktbereichen als ein Projekt.⁶⁷ Dabei ist das Projektmanagement als Disziplin der Organisationslehre relativ jung und unerforscht im Vergleich zu den dauerhaften Organisationsformen. Insbesondere der Situative Ansatz nimmt hier wenig Raum ein. Erst in den letzten Jahren ist zunehmend Forschung zu einzelnen situativen Ansätzen und zu Projekt-Typologien im Projektmanagement zu beobachten.⁶⁸

⁶⁶ Vgl. Kubicek u.a. (1981), S. 103ff.

⁶⁷ Vgl. hierzu im Detail Kap. 2.2.2

⁶⁸ Vgl. Shenhar u.a. (2005), S. 94; vgl. Shenhar (2001a), S. 395

Im Projektmanagement der Softwareentwicklung ergibt sich ein ähnliches Bild. Hier besteht sowohl Handlungsbedarf in der Praxis als auch in der Forschung, obwohl der Situative Ansatz über Fragen des Projektmanagements hinaus schon früh Anwendung in der Softwareentwicklung gefunden hat.⁶⁹ Zwar werden in der Praxis unterschiedliche Projekte durchaus unterschiedlich bzw. projektspezifisch behandelt. Doch geschieht dies in den wenigsten Fällen formal und systematisch, d.h. ausgehend von einem unternehmensweiten Prozessrahmenmodell und einem Klassifikationsschema für die definierten Projekttypen. Eine solche Vorgehensweise wird als Tailoring der Softwareentwicklung bezeichnet.⁷⁰ In der Konsequenz scheitern Softwareentwicklungsprojekte aufgrund von Fehlern bei der Anpassung des Prozessrahmenmodells an die projektspezifischen Gegebenheiten.⁷¹ Hier besteht insbesondere bei der systematischen Anwendung des Projekt-Tailorings Forschungsbedarf.⁷²

Bezogen auf die Organisation der Softwareentwicklung im Allgemeinen sollen an dieser Stelle zwei relativ neuere Forschungsvorhaben aus dem deutschsprachigen Raum vorgestellt werden. Die beiden Forschungsarbeiten von Schramke⁷³ und von Lang⁷⁴ verfolgen, wie auch die vorliegende Arbeit, den Situativen Ansatz der Organisationsgestaltung.

Zielsetzung der Forschungsarbeit von Schramke ist die Erarbeitung einer Gesamtkonzeption zur problemadäquaten organisatorischen Gestaltung eines Softwareprojektes. Der Ansatz soll die Durchlaufzeit von Softwareentwicklungsprojekten verringern. Dabei sollen sich aus der Bestimmung von Einflussfaktoren auf die Softwareentwicklung verschiedene Softwareprojekttypen ergeben. Für diese zu bildenden Projekttypen sind jeweils spezifische Gestaltungsalternativen der Projektorganisation zu formulieren. Die Gestaltungsempfehlungen sollen die Projektaufbau- und Projektablauforganisation umfassen.⁷⁵

In dem Ansatz ergeben sich aus insgesamt 11 produktspezifischen Einflussfaktoren der beiden Dimensionen Komplexität und Funktionalität durch Aggregation zu einer Portfolio-Matrix 3 mögliche Produktschwierigkeiten: einfache, mittelschwere und schwierige Produktanforderungen. Daneben werden analog insgesamt 23 prozessspezifische Indikatoren der beiden Dimensionen Inputfaktoren und Rahmenbedingungen zu den drei Prozessschwierigkeiten einfacher, mittelschwerer und schwieriger Softwareent-

⁶⁹ Vgl. bspw. Arinze (1991); Schonberger (1980); Weill, Olson (1989)

⁷⁰ Vgl. hierzu ausführlich in Kap. 2.1.3.1

⁷¹ Vgl. Shenhar u.a. (2005), S. 8, mit weiteren Verweisen

⁷² Vgl. Xu, Ramesh (2007), S. 296

⁷³ Diss., Technische Universität München; vgl. Schramke (2002)

⁷⁴ Diss., Univ. zu Köln, vgl. Lang (2004)

⁷⁵ Vgl. Schramke (2002), S. 16, 47

wicklungsprozess aggregiert. Anschließend erfolgt aus den beiden zweidimensionalen Portfolio-Matrizen die Ermittlung der Projekttypen hinsichtlich Produkt- und Prozessschwierigkeit, was einen weiteren Aggregationsschritt zu einem 4er-Projektportfolio mit sog. State of the art-Projekten, Komplexen Normalprojekten, Potentialprojekten und Pionierprojekten darstellt (vgl. Abb. 5).⁷⁶

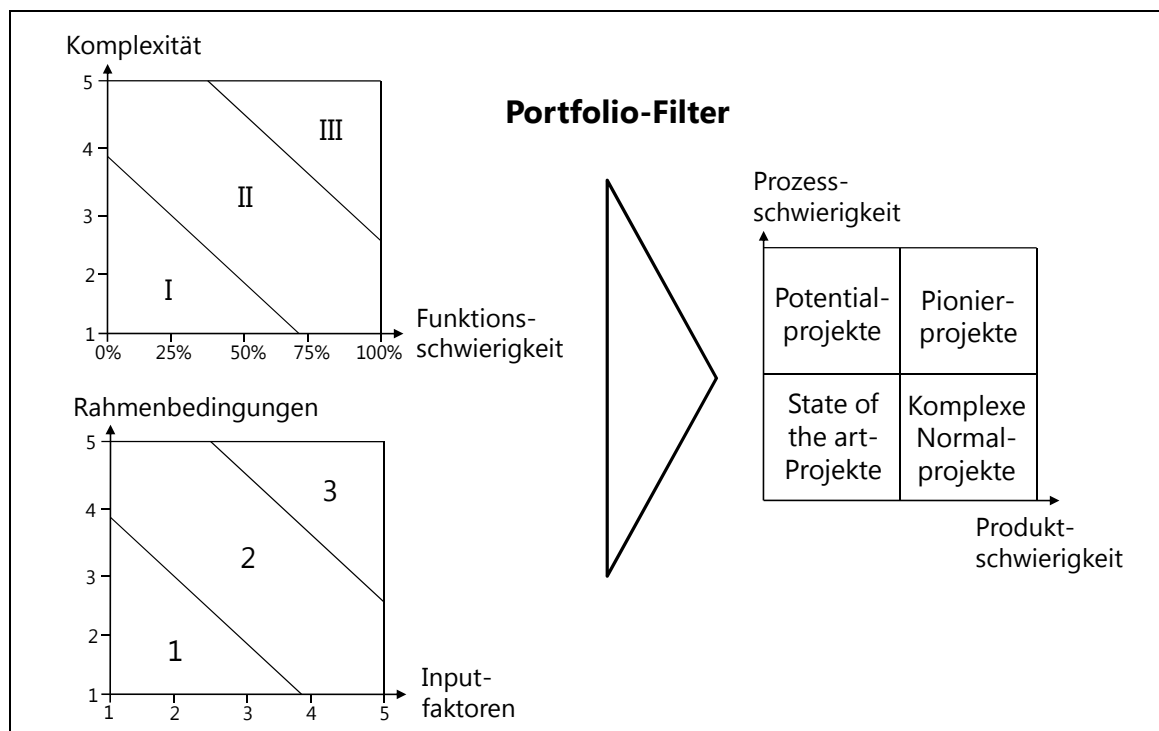


Abb. 5: Ermittlung von Softwareprojekttypen beim Ansatz von Schramke⁷⁷

Der grundlegende Ansatz bei der Gestaltung des organisatorischen Rahmens ist, dass die zunehmende Produkt- und Prozessschwierigkeit durch ein zunehmendes Maß an Dezentralisation abgefangen werden soll. Dezentralisation soll wiederum durch zunehmende Prozessorientierung, Teamorientierung und verstärktes Empowerment der Mitarbeiter sichergestellt werden. Hieraus entsteht dann das Gestaltungsportfolio. Es besteht aus den drei Gestaltungsalternativen Hierarchieansatz - v.a. für State of the art-Projekte, Netzwerkansatz - v.a. für Potential- und Komplexe Normalprojekte und Virtueller Ansatz - v.a. für Pionierprojekte.⁷⁸

Mit dem Hierarchieansatz beschränkt sich die Zunahme an Dezentralisation im Quadranten der State of the art-Projekte nur auf den Bereich einer hierarchischen Organisation. Steigt die Schwierigkeit für die Softwareentwicklung weiter an, so kann dies nicht mehr durch eine alleinstehende Organisation bewältigt werden. Bei Potential- und Komplexen Normalprojekten, die also entweder durch eine hohe Produktschwie-

⁷⁶ Vgl. Schramke (2002), S. 89f., 108ff.

⁷⁷ Schramke (2002), S. 111

⁷⁸ Vgl. Schramke (2002), S. 188ff.

rigkeit oder durch eine hohe Prozessschwierigkeit gekennzeichnet sind, soll daher das Prinzip einer Netzwerkgestaltung zur Anwendung kommen. Die Gestaltung eines Pionierprojektes soll demnach durch den Netzwerkansatz oder durch die Ausschreibung von Bedarfen auf Märkten und somit durch die Anwendung des virtuellen Ansatzes erfolgen. Der Netzwerkansatz wie auch der virtuelle Ansatz haben dabei eine weitreichende Dezentralisation der Organisationsstruktur zur Folge. Insgesamt soll bei dieser Einordnung, die jedoch nicht trennscharf ist, ein Optimum zwischen Transaktionskosten und Nutzeffekten entstehen.⁷⁹

In Ergänzung zu den aufbauorganisatorischen Überlegungen macht Schramke auch Gestaltungsempfehlungen zur Projektablauforganisation und greift dabei auf bestehende Vorgehensmodelle der Softwareentwicklung zurück. Deren Zuordnung zu den verschiedenen Schwierigkeitsgraden basiert auf Überlegungen zum Anwendungskontext der Vorgehensmodelle und ergibt insgesamt das folgende Bild für die Ausgestaltung einer hierarchischen Organisation (vgl. Abb. 6).⁸⁰

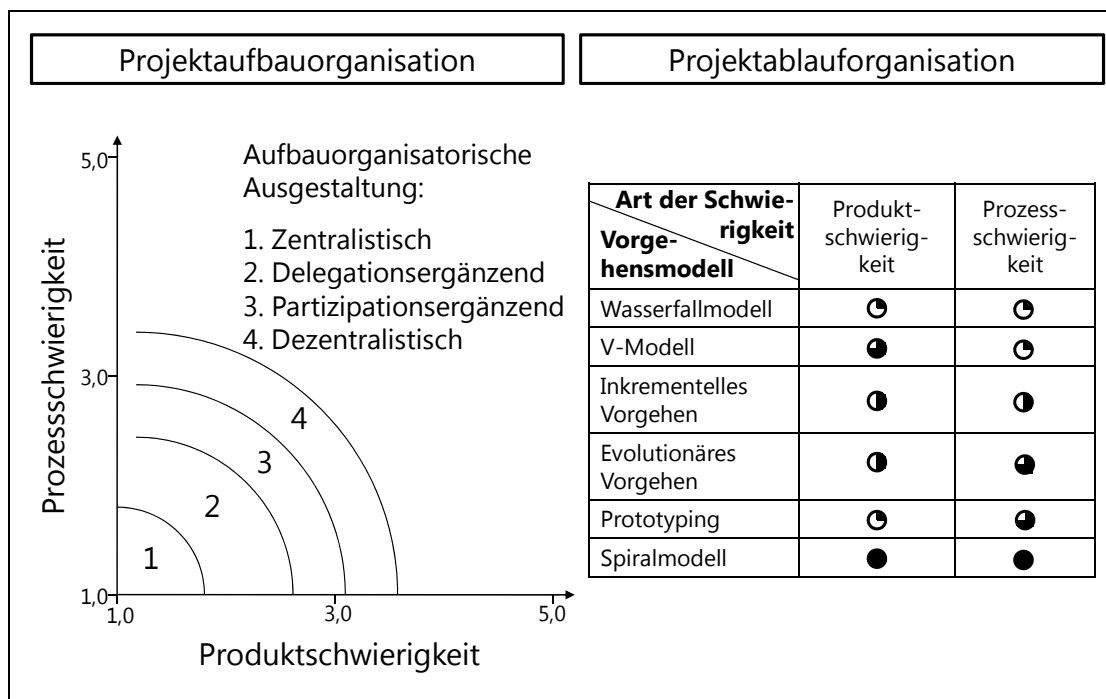


Abb. 6: Gestaltungsempfehlungen zur Projektablauforganisation beim Ansatz von Schramke⁸¹

Gegenstand der Arbeit von Lang ist die Entwicklung von Gestaltungsempfehlungen für bestimmte Gestaltungssituationen, die er abhängig von der Unsicherheit eines Entwicklungsvorhabens definiert. Der Unsicherheitsbegriff wird dabei anhand von den drei Merkmalsvariablen Komplexität, Dynamik und Neuartigkeit konkretisiert:⁸²

⁷⁹ Vgl. Schramke (2002), S. 188ff.

⁸⁰ Vgl. Schramke (2002), S. 192f.

⁸¹ Schramke (2002), S. 193

⁸² Vgl. Lang (2004), S. 255f.

- Die Merkmalsvariable Komplexität wird durch verschiedene Ausprägungen der Marktbedingungen charakterisiert. Es wird vereinfachend davon ausgegangen, dass die Komplexität der Marktbedingungen entweder hoch oder gering ist. Hohe Komplexität liegt insbesondere dann vor, wenn eine Vielzahl von Wettbewerbern vorhanden ist oder wenn die Kundenbedürfnisse sehr heterogen sind;
- Bei der Merkmalsvariable Dynamik wird zwischen hoher und niedriger Marktdynamik, stark dynamischer und mäßig dynamischer Weiterentwicklung von Software- und Hardware-Techniken, sowie hoher oder mäßig häufiger Anforderungsänderungen seitens der Kunden unterschieden;
- Die Merkmalsvariable Neuartigkeit wird hinsichtlich der Entwicklungsaufgabe zu Neuentwicklungen in bekannter oder unbekannter Problemdomäne, Variantenentwicklungen sowie Weiter- und Anpassungsentwicklungen spezifiziert.

Um zu den Gestaltungssituationen zu kommen, für die Gestaltungsempfehlungen genannt werden sollen, kombiniert Lang die verschiedenen Ausprägungen der drei Merkmalsvariablen der Unsicherheit miteinander sowie mit sechs möglichen wettbewerbsstrategischen Handlungsalternativen, die zuvor einführt werden (vgl. Abb. 7). Lang beschränkt sich in seiner weiteren Arbeit auf die drei nachfolgenden Gestaltungssituationen. Sie entsprechen gängigen Gestaltungsbedingungen in der Praxis der Softwareentwicklung und sollen somit idealtypische Situationen darstellen.⁸³

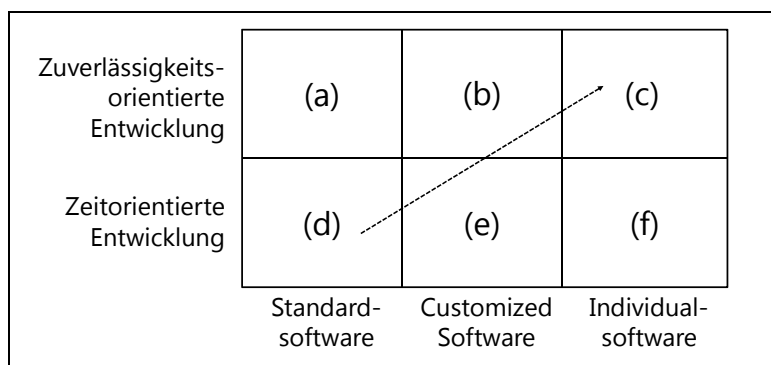


Abb. 7: Wettbewerbsstrategische Handlungsalternativen beim Ansatz von Lang⁸⁴

- (1) Zeitorientierte Standardsoftwareentwicklung als Neuentwicklung in unbekannter Problemdomäne bei hoher marktlicher und technischer Dynamik - als typisch für Erstentwicklungen von Softwareprodukten;
- (2) Zeitorientierte Weiterentwicklung von Standardsoftware bei hoher marktlicher und technischer Dynamik als eine typische Situation für die Entwicklung von Nachfolgeversionen von Softwareprodukten;

⁸³ Vgl. Lang (2004), S. 256; Die Gestaltungssituationen (1) und (2) entsprechen der wettbewerbsstrategischen Handlungsalternative (d); die Gestaltungssituation (3) der wettbewerbsstrategischen Handlungsalternative (c).

⁸⁴ Lang (2004), S. 254

(3) Zuverlässigkeitsorientierte Individualsoftwareentwicklung als Variantenentwicklung bei geringer marktlicher und technischer Dynamik als eine typische Situation, zu der bspw. die Entwicklung von sicherheitskritischer Individualsoftware gerechnet werden kann.

Für diese drei Gestaltungssituationen formuliert Lang Gestaltungsempfehlungen im Hinblick auf die Koordination: Bei den Situationen (a) und (b) wird eine produktorientierte Arbeitsteilung bei den primären Teilprozessen der Softwareentwicklung⁸⁵ als zweckmäßig vorgeschlagen. Für die Gestaltungssituation (c) wird die Empfehlung formuliert, eine prozessorientierte horizontale Arbeitsteilung vorzunehmen. Ferner werden Empfehlungen hinsichtlich des Zentralisationsgrades entwickelt. Im Detail werden dabei drei Klassen von Entscheidungen untersucht: Entscheidungen bez. der Ressourcen- und Zeitplanung, Auswahlentscheidungen bei Entwicklungsmethoden und Architekturentscheidungen. Hinsichtlich der Ressourcen- und Zeitplanung sowie der Wahl von Entwicklungsmethoden wird tendenziell ein hoher Grad an Entscheidungscentralisation vorgeschlagen. Bez. der entscheidungsvorbereitenden Handlungen erweist sich allerdings in den Gestaltungssituationen (a) und mit Einschränkungen (b) ein hoher Grad an Dezentralisation als zweckmäßig. Hinsichtlich der Architekturentscheidungen stellt sich unabhängig von der Gestaltungssituation ein mittlerer Zentralisationsgrad als zweckmäßig heraus.⁸⁶

Neben den beiden vorgestellten Forschungsarbeiten aus dem deutschsprachigen Raum finden sich im Umfeld des Information Systems Research (ISR)⁸⁷ zahlreiche weitere Arbeiten zur situativen Organisationsgestaltung in der Softwareentwicklung.⁸⁸ Sie basieren allesamt auf quantitativen Methoden der empirischen (Organisations-)Forschung, wie sie bei ISR vorherrschend sind – im Gegensatz zu der Situation bei der Wirtschaftsinformatik.⁸⁹ Damit sind diese Arbeiten auch mit den Nachteilen behaftet, die diese Forschungsform mit sich bringt und aufgrund derer ihre Ergebnisse nur in einem sehr beschränkten Umfang für die vorliegende, qualitativ empirisch ausgerichtete Arbeit, verwendet werden konnten.⁹⁰

- Verwendung von einfachen Typologien und Operationalisierungen, um (die notwendigen) großzahligen Untersuchungen zu ermöglichen;

⁸⁵ Vgl. hierzu Kap. 2.1.2.1

⁸⁶ Vgl. Lang (2004), S. 256f., 361f.

⁸⁷ Im englischsprachigen Raum ist die Wirtschaftsinformatik als Information Systems (IS) bzw. Information Systems Research (ISR) bekannt.

⁸⁸ Vgl. Nidumolu (1996); Nidumolu, Subramani (2003); Shenhar (2001b); Barki u.a. (2001); Andres, Zmud (2002)

⁸⁹ Zu Unterschieden zwischen Wirtschaftsinformatik und ISR vgl. bspw. Herzwurm, Stelzer (2008)

⁹⁰ Die folgende Kritik trifft teilweise auch auf die Forschungsarbeiten von Lang und Schramke zu.

- Trivialisierungstendenz durch Prüfung von Hypothesen ohne Überraschungspotential;
- Beschränkter Verständniszuwachs durch Untersuchungen, die auf einem Black Box-Modell basieren;
- Generierung von nicht vergleichbaren und nicht integrierbaren Einzelergebnissen.

Bei großzahligen empirischen Untersuchungen werden Operationalisierungen bevorzugt, die qualitativ gesehen geringe „Eingrenzungen vornehmen und relativ globale Attribute zulassen, um mächtige Eigenschafts- und Relationsräume fassen zu können.“⁹¹ Für qualitativ ausgerichtete Arbeiten mit konkreten Gestaltungszielen braucht man dagegen präzise Operationalisierungen mit weitergehenden inhaltlich-qualitativen Eingrenzungen, wodurch aber die quantitative Bestimmung erschwert wird. Ein einfaches Beispiel hierfür stellt die Operationalisierung der Variable Planungsausmaß dar. Bei einer quantitativ ausgerichteten Arbeit ist es durchaus zweckmäßig, das Planungsausmaß einer Unternehmung durch die Anzahl der in der Unternehmung produzierten (Teil-)Pläne zu ermitteln. Für Gestaltungsüberlegungen ist jedoch eine Aussage, dass bspw. mit zunehmender Unternehmensgröße mehr Pläne erzeugt werden (sollen), dagegen wenig hilfreich. Den Gestalter interessieren vielmehr die Inhalte der Pläne und die Aktivitäten im Planungsprozess. Er ist damit auf eine stärkere qualitative Ausrichtung der Operationalisierung des Planungsausmaßes angewiesen.⁹²

Beim Situativen Ansatz im Projektmanagement äußern sich quantitativ ausgerichtete Operationalisierungen bspw. in einfachen Projekt-Typologien als Ergebnis oder als Grundlage einer Untersuchung. In der Realität ist aber die Variation in Projekten komplexer, als das sie mit einfachen Typologien wie bspw. ‚mechanisch vs. organisch‘ oder ‚radikale vs. inkrementelle Innovation‘ eingefangen werden könnte.⁹³

Als Maßstab für wissenschaftlichen Fortschritt dient unter Bezugnahme auf ein Gestaltungsziel weniger der Zuwachs in der Erkenntnissicherung als vielmehr der Zuwachs im Verständnis und die dadurch besser mögliche Beherrschung der Realität. „Ein derart begriffener wissenschaftlicher Fortschritt kommt .. nicht darin zum Ausdruck, dass bereits im Alltagswissen vorliegende allgemeine Vermutungen über Zusammenhänge durch exaktere Untersuchungen abgesichert werden.“⁹⁴ Solche Untersuchungen weisen eine Trivialisierungstendenz auf, da sie oft nur Hypothesen ohne Überraschungspotential betrachten. Eine Hypothese ist aber v.a. dann interessant, wenn sie ein Überra-

⁹¹ Szyperski, Müller-Böling (1981), S. 180

⁹² Vgl. Szyperski, Müller-Böling (1981), S. 179f.

⁹³ Vgl. Shenhar (2001a), S. 397

⁹⁴ Kubicek (1977), S. 7

schungspotential bietet, d.h. nicht mit begründeten Erwartungen von sachkundigen Rezipienten konform ist.⁹⁵

Die Analyse von organisatorischen Zusammenhängen erfolgt in der quantitativ empirischen (Organisations-)Forschung in erster Linie durch die Berechnung von statistischen Korrelationen. Der analytische Vorteil solcher Korrelationen ist, dass nur so die relative Einflussstärke mehrerer Situationsfaktoren auf einen organisatorischen Sachverhalt ermittelt werden kann. Damit gehen aber auch wesentliche Nachteile einher: Zum einen können durch Korrelationen nur umfangs- oder intensitätsmäßige Zusammenhänge erfasst werden, wodurch aber inhaltliche Aspekte vernachlässigt werden. Zum anderen sagen Korrelationen nur etwas darüber aus, in welchem Ausmaß Merkmalsausprägungen zusammen auftreten. Warum sie das tun und wie es zu diesen Zusammenhängen kommt, ist eine Frage, die über die Aussagefähigkeit statistischer Kennzahlen hinausgeht. Die Beantwortung dieser Frage ist aber zwingend notwendig für das Erreichen von Verständniszuwachs. Denn wissenschaftlicher Fortschritt kommt nicht darin zum Ausdruck, dass auf einem Black-Box-Modell basierende Prognosen genauer werden. Begrifflichkeiten und Konstrukte wie abhängige und unabhängige Variable oder Ursache und Wirkung beruhen auf gedanklichen Annahmen, die der Forscher an seine empirischen Daten heranträgt. Diese Annahmen müssen an sich und im Lichte der Forschungsergebnisse expliziert werden.⁹⁶

Die vielfältigen institutionellen Bedingungen in der Forschungspraxis führen oftmals dazu, dass nur sehr enge Fragestellungen formuliert werden und die Untersuchungen sich damit auf die Prüfung von Einzelhypothesen konzentrieren, da diese leichter und schneller prüfbar erscheinen.⁹⁷ Da die einzelnen Untersuchungen jedoch in unterschiedlichen Kontexten und teilweise mit unterschiedlichen Maßen erfolgen, ist eine nachträgliche Integration zumeist nicht möglich. Ein Verständnis der Realität kann jedoch kaum durch Einzelhypothesen, sondern nur durch integrierte Aussagensysteme als Hypothesensysteme vermittelt werden.⁹⁸ Insofern sieht sich der Forscher hier „vor eine nicht mehr überschaubare Anzahl von empirischen Studien gestellt, in denen Einzelhypothesen oder .. lose miteinander verknüpfte Mengen von Hypothesen unter exakt angegebenen Bedingungen und unter Anwendung anspruchsvoller Analyseverfahren geprüft worden sind. Trotz Beachtung dieser formalen Kriterien hat der Erkenntnisstand im Laufe der Zeit jedoch nicht zugenommen.“⁹⁹

⁹⁵ Vgl. Kubicek (1977), S. 7f.; vgl. Frank (2006), S. 177f.

⁹⁶ Vgl. Kieser, Walgenbach (2003), S. 218, 221; vgl. Kubicek (1977), S. 7f.

⁹⁷ Vgl. Kubicek (1977), S. 11

⁹⁸ Vgl. Kubicek (1977), S. 8

⁹⁹ Kubicek (1977), S. 8

2 Konzeption des situativen Koordinationsansatzes

Im vorliegenden Kap. 2 werden die Forschungsfragen der Arbeit auf konzeptioneller Ebene beantwortet. Hierzu muss zunächst ein theoretischer Zugang zu der Problemstellung geschaffen werden. Dies erfolgt in den Kap. 2.1 und 2.2 über den Situativen Ansatz und das entscheidungsorientierte Konzept der Organisationsgestaltung. Aufbauend auf den hier gewonnenen Erkenntnissen über die Entwicklung von Standardsoftware aus Sicht der beiden Ansätze der Organisationsgestaltung (Erkenntnisziel) erfolgt in Kap. 2.3 die Konzeption des situativen Koordinationsansatzes (Gestaltungsziel). Im Ergebnis entsteht der Konzeptionsrahmen – bestehend aus begrifflichen, deskriptiven und explanatorischen Aussagen (Kap. 2.4 im Überblick).

2.1 Theoretischer Zugang über den Situativen Ansatz der Organisationsgestaltung

Erkenntnisgewinnung stellt die Grundlage für die Gestaltung dar, die ohne eine vorangegangene Erklärung prinzipiell nicht möglich ist. Diese für die Konzeption des situativen Koordinationsansatzes notwendige Vorarbeit erfolgt im vorliegenden Kap. 2.1 zunächst aus Sicht des Situativen Ansatzes der Organisationsgestaltung.

2.1.1 Der Situative Ansatz der Organisationsgestaltung

Den Situativen Ansatz der Organisationsgestaltung kennzeichnen zwei zusammenhängende Grundthesen:¹⁰⁰

- Unterschiedliche Organisationsstrukturen und unterschiedliche Verhaltensweisen der Organisationsmitglieder sind auf Unterschiede in der Situation zurückzuführen, in der sich die Unternehmen befinden;
- Organisationsstrukturen und Verhaltensweisen der Organisationsmitglieder sind je nach Situation unterschiedlich effizient.

Aus diesen beiden Grundthesen sind Beiträge des Situativen Ansatzes in sog. analytischen und pragmatischen Varianten hervorgegangen. Bei den analytischen Varianten geht es um die Verfolgung eines theoretischen Wissenschaftszieles. Die Strukturvariablen der Organisation werden als abhängige Größen und die Situationsvariablen als unabhängige Größen aufgefasst. Relevante Situationsvariablen sind diejenigen Faktoren, die zur Erklärung von Unterschieden in den empirisch untersuchten Organisationsstrukturen beitragen. Die Wirkungsmechanismen werden hierbei nicht direkt untersucht, sondern nachträglich interpretiert. In einer Erweiterung der analytischen Vari-

¹⁰⁰ Vgl. Schulte-Zurhausen (2002), S. 24

ten werden zusätzlich Auswirkungen der Organisationsstruktur auf das Verhalten der Organisationsmitglieder und auf die Zielerreichung bzw. Effizienz erfasst. Als Forschungsmethodik wird die vergleichende empirische Organisationsforschung angewendet: In empirischen Untersuchungen werden diejenigen Ausprägungen der Situationsvariablen identifiziert, die mit bestimmten Ausprägungen der Strukturvariablen korrelieren. Das Programm der vergleichenden Organisationsforschung wird durch die folgenden drei Fragestellungen gekennzeichnet (vgl. Abb. 8).¹⁰¹

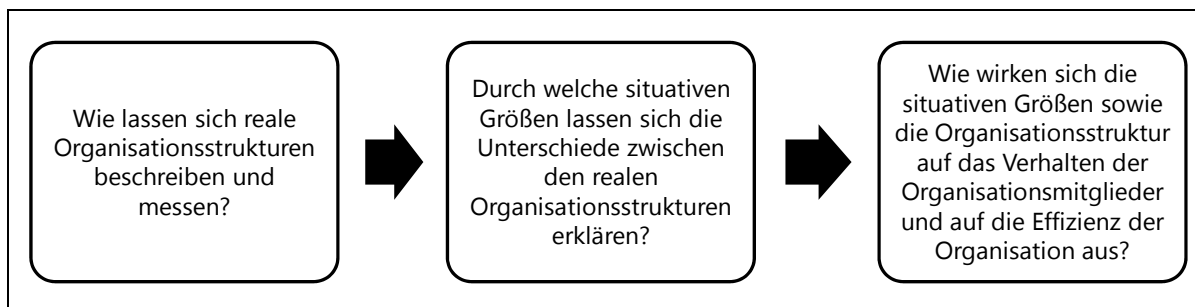


Abb. 8: Kernfragen der analytischen Variante des Situativen Ansatzes¹⁰²

Bei den pragmatischen Varianten des Situativen Ansatzes geht es dagegen um die Formulierung von Gestaltungsempfehlungen und um deren Begründung. Erkenntnisleitend ist v.a. die Frage, wie man Organisationsstrukturen so gestalten kann, dass sie den Anforderungen der Situation gerecht werden. Es wird hierbei von einem Organisationsgestalter ausgegangen, der bestimmte Gestaltungsziele verfolgt und die hierfür am besten geeignete Strukturalternative auswählen möchte. Diese entspricht am besten der Situation des Unternehmens, d.h. sie weist die größtmögliche Stimmigkeit mit der Situation auf.¹⁰³ Abb. 9 verdeutlicht das Grundmodell der pragmatischen Variante des Situativen Ansatzes.

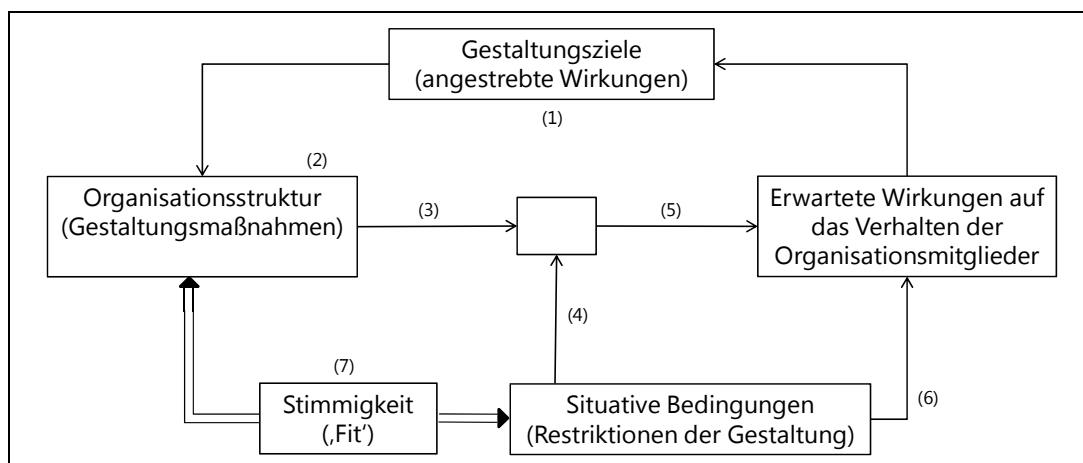


Abb. 9: Das Grundmodell der pragmatischen Variante des Situativen Ansatzes¹⁰⁴

¹⁰¹ Vgl. Schulte-Zurhausen (2002), S. 24f.

¹⁰² Vgl. Kubicek, Kieser (1980), Sp. 1535

¹⁰³ Vgl. Kieser, Kubicek (1992), S. 60f.

¹⁰⁴ Vgl. Kieser, Kubicek (1992), S. 60

- (1) Die pragmatische Variante des Situativen Ansatzes setzt an den Gestaltungszielen an, die auch als angestrebte Wirkungen in Form bestimmter Verhaltensweisen der Organisationsmitglieder verstanden werden können.
- (2) Diese Gestaltungsziele bzw. angestrebten Wirkungen sollen durch eine geeignete Organisationsstruktur herbeigeführt werden.
- (3) Die Organisationsstruktur beinhaltet Vorgaben für die Organisationsmitglieder und schreibt bestimmte Verhaltensweisen vor. Beides zusammen bildet die Aktionsparameter der Gestaltung.
- (4) Die Aufgaben, welche die Organisationsmitglieder zu erfüllen haben, werden jedoch wesentlich durch die vorherrschenden situativen Bedingungen bestimmt. Von diesen situativen Bedingungen gehen demnach Restriktionen für die Gestaltung aus.
- (5) Die erwarteten Wirkungen organisatorischer Regelungen auf das Verhalten der Organisationsmitglieder werden als Kombination von Struktur- und Situationseffekten, also als Kombination von (3) und (4), begriffen.
- (6) Darüber hinaus gehen auch direkte, d.h. von der Organisationsstruktur unabhängige, Verhaltensanforderungen von den situativen Bedingungen aus.
- (7) Wenn in einer konkreten Situation das tatsächliche bzw. das erwartete Verhalten der Organisationsmitglieder von den angestrebten Wirkungen abweicht, so wird davon ausgegangen, dass die Organisationsstruktur nicht situationsgerecht gestaltet ist. In diesem Fall liegt eine Diskrepanz zwischen den organisatorischen Vorgaben und den situativen Bedingungen vor. Der ‚Fit‘ zwischen der Struktur und der Situation ist wieder herzustellen.¹⁰⁵

Das pragmatische Grundmodell des Situativen Ansatzes ist in der Literatur zur organisatorischen Gestaltung weit verbreitet.¹⁰⁶ Es prägt auch im Umfeld der Softwareentwicklung diverse Arbeiten¹⁰⁷ und stellt nicht zuletzt den Rahmen für die Ableitung der Gestaltungsempfehlungen in der vorliegenden Arbeit dar. Wobei erwähnt werden muss, dass sich alle genannten Arbeiten wesentlich voneinander unterscheiden. Dies liegt daran, dass es sich beim Situativen Ansatz, wie bereits in Zusammenhang mit der Triangulation in Kap. 1.4 festgestellt, um einen übergeordneten Rahmen zur Konzeptualisierung handelt. Das pragmatische Grundmodell des Situativen Ansatzes ist in diesem Sinn ein formales Modell, das inhaltlich durch konkrete Ziele, relevante Situationsfaktoren und die damit korrespondierenden Ausprägungen der Strukturvariablen

¹⁰⁵ Vgl. Kieser, Kubicek (1992), S. 55ff.; vgl. Schulte-Zurhausen (2002), S. 24, 26f.

¹⁰⁶ Vgl. Schulte-Zurhausen (2002), S. 27, mit zahlreichen weiteren Verweisen.

¹⁰⁷ Im deutschsprachigen Raum sind es bspw. die Arbeiten von Herzwurm (2000), Mellis (2004), Stelzer (1998), Lang (2004) und Schramke (2002)

ausgefüllt werden muss.¹⁰⁸ So wird es auch in der vorliegenden Arbeit verwendet: Die Koordination der Standardsoftwareentwicklung wird mit Elementen des Situativen Ansatzes in Kap. 2.3 konzeptualisiert.

Beim sog. Konfigurationsansatz wird der Situative Ansatz um das Konzept der inneren Stimmigkeit erweitert. Danach muss nicht nur die Organisationsstruktur mit den situativen Bedingungen stimmig sein, sondern auch die einzelnen Gestaltungsmaßnahmen innerhalb der Organisationsstruktur müssen untereinander konsistent sein.¹⁰⁹ Der Begriff der Konfiguration bezeichnet repräsentative Muster bzw. Kombinationen von solchen stimmigen Ausprägungen der zahlreichen interdependenten Merkmale der Situation und der Organisationsstruktur.¹¹⁰ Das Stimmigkeitskonzept des Konfigurationsansatzes fußt auf drei Hypothesen:¹¹¹

- Die Kongruenzhypothese wird auch schon vom Situativen Ansatz aufgestellt und besagt, dass eine effiziente Organisation einer guten Übereinstimmung zwischen den situativen Bedingungen und den Gestaltungsmaßnahmen der Organisationsstruktur bedarf;
- Die Konfigurationshypothese setzt bei einer effizienten Organisation die interne Konsistenz zwischen den Gestaltungsmaßnahmen der Organisationsstruktur voraus;
- Die erweiterte Konfigurationshypothese vereinigt die beiden erstgenannten Hypothesen. Sie besagt, dass effiziente Organisationen sowohl der Kongruenzhypothese, als auch der Konfigurationshypothese folgen und somit Konfigurationen bilden müssen.

Lindstädt entwickelt eine auf der Kongruenz-Konsistenz-Unterscheidung basierende Verfeinerung des Stimmigkeitsbegriffes. Dazu unterscheidet er unter Einbeziehung des Handlungsstrukturmodells zwischen Zielen, Maßnahmen und Instrumenten, sowie Bedingungen. Hieraus ergeben sich fünf grundsätzlich verschiedene Stimmigkeitstypen nach der Art der Relation zwischen diesen drei Elementen (vgl. Abb. 10).

¹⁰⁸ Vgl. Kieser, Kubicek (1992), S. 62; vgl. Schulte-Zurhausen (2002), S. 27

¹⁰⁹ Vgl. Kieser (2006), S. 244

¹¹⁰ Der Konfigurationsbegriff hat in den letzten Jahren einen Bedeutungswandel erfahren. Ursprünglich wurde die Konfiguration als eine Dimension zur Beschreibung und Messung des Stellengefüges von Organisationsstrukturen benutzt. Vgl. Henselek (1996), S. 49f.

¹¹¹ Vgl. Mintzberg (1992), S. 168f., 206; vgl. Henselek (1996), S. 50f.; vgl. Sauerwald (2007), S. 66

Stimmigkeit von mit Zielen	... mit Maßnahmen und Instrumenten	... mit Bedingungen
Zielen ...	Zielkonsistenz	Instrumentalität	Zielkongruenz
Maßnahmen und Instrumenten ...	Instrumentalität	Maßnahmenkonsistenz	Maßnahmenkongruenz

Abb. 10: Stimmigkeitstypen nach der Art der Relation¹¹²

Die dargestellten Stimmigkeitstypen lassen sich in zwei wesentliche Gruppen unterteilen. Die erste Gruppe fasst die beiden Typen aus der Kategorie Konsistenz zusammen (Maßnahmen- und Zielkonsistenz). Sie werden von Lindstädt als Stimmigkeit im engen Sinn bezeichnet, da sie sich beide auf die Stimmigkeit innerhalb der einzelnen Kategorien beziehen. Die Kongruenzen sowie die Instrumentalität werden ebenfalls zusammengefasst und als Stimmigkeit im weiteren Sinn bezeichnet – hier geht es um Stimmigkeit zwischen den verschiedenen Elementen.¹¹³

Die wechselseitige Angemessenheit bzw. Adäquanz der Gestaltungsvariablen einer Unternehmung spielt in vielen, dem Konfigurations- und dem Situativen Ansatz nahe stehenden und teilweise sehr differenzierten Ansätzen die zentrale Rolle. So sind bspw. bei Krüger 27 Arten der Stimmigkeit zu unterscheiden, 7 davon alleine unter dem Aspekt der Organisationsgestaltung.¹¹⁴ Scholz unterscheidet zwischen 9 Stimmigkeitstypen: Die drei Unternehmungsmerkmale Strategie, System und Kultur müssen zunächst einmal jeweils in sich selbst stimmig sein (Intra-Strategie-Fit, Intra-System-Fit und Intra-Kultur-Fit). Darüber hinaus müssen sie paarweise untereinander und jeweils mit der Unternehmungsumwelt Stimmigkeit ausweisen (Strategie-System-Fit, Kultur-Strategie-Fit, Kultur-System-Fit, Strategie-Umwelt-Fit, Kultur-Umwelt-Fit, System-Umwelt-Fit).¹¹⁵

Stimmigkeit kann aber auch in einer weniger differenzierten, auf der anderen Seite aber auch gleichzeitig weniger abstrakten und mehr praxisnahen Form, die wesentliche Grundlage eines Ansatzes bilden. So stellen Herzwurm u.a. als zentrale Forderung bei der Realisierung einer kundenorientierten Softwareentwicklung, den Softwareentwicklungsprozess kohärent, d.h. durchgängig an der Erreichung des höchsten Kundennutzens, auszurichten. Konkret wird dabei die Stimmigkeit zwischen der relativen Bedeutung einer Kundenanforderung an das zu erstellende Softwareprodukt und den ‚An-

¹¹² Lindstädt (2005), S. 26

¹¹³ Vgl. Lindstädt (2005), S. 26f.

¹¹⁴ Vgl. Krüger (1992), Sp. 228 ff.

¹¹⁵ Vgl. Scholz (1992), Sp. 543f.

strengungen', die der Erfüllung dieser Kundenanforderung im Entwicklungsprozess zukommen (bspw. Ressourceneinsatz), gefordert.¹¹⁶

In einem ähnlichen Verständnis kommt der Stimmigkeit eine zentrale Bedeutung in dieser Arbeit zu. Die situative Koordination der Standardsoftwareentwicklung wird in Kap. 3.4 als ein Konfigurationsansatz umgesetzt. So soll die Stimmigkeit zwischen den einzelnen Gestaltungsmaßnahmen der Koordination (Kap. 3.3) untereinander und zu den übergeordneten Projektzielen (Kap. 3.2) sichergestellt werden.

2.1.2 Entwicklung von Standardsoftware aus Sicht des Situativen Ansatzes

2.1.2.1 Gestaltungsziele und -maßnahmen bei der Softwareentwicklung im Allgemeinen

Gestaltungsziele der Softwareentwicklung dürfen nicht mit den Sach- und Formalzielen eines Softwareentwicklungsvorhabens verwechselt werden. Letztere beschreiben immer Anforderungen an ein konkretes Entwicklungsvorhaben, d.h. Anforderungen an die zu erstellende Software und an das zugrunde liegende Entwicklungsprojekt. Sach- und Formalziele eines Softwareentwicklungsvorhabens lassen sich in Ziele bez. Produktqualität, Leistungsumfang der Software, Entwicklungskosten und Entwicklungszeit unterteilen.¹¹⁷

Gestaltungsziele der Softwareentwicklung beziehen sich nicht auf ein einzelnes, sondern auf eine Abstraktion von verschiedenen Entwicklungsvorhaben im Softwareunternehmen. Sie adressieren den Softwareentwicklungsprozess im Softwareunternehmen und werden daher als Prozessziele bezeichnet. Der Begriff des Softwareentwicklungsprozesses stellt eine Abstraktion bzw. eine modellhafte Abbildung der Durchführung verschiedener Softwareentwicklungsvorhaben dar.¹¹⁸ Trotz der beiden verschiedenen Abstraktionsebenen besteht ein enger kausaler Zusammenhang zwischen den Gestaltungszielen der Softwareentwicklung und den Sach- und Formalzielen eines konkreten Softwareentwicklungsvorhabens: Die Softwareentwicklung muss so gestaltet sein, dass die Sach- und Formalziele der einzelnen Entwicklungsvorhaben bestmöglich erreicht werden können. Gestaltungsziele der Softwareentwicklung lassen sich in fachlich-technische, umfeldbezogene und organisatorische Ziele unterteilen (vgl. Abb. 11).¹¹⁹

¹¹⁶ Vgl. Herzwurm u.a. (1997), S. 5, 11f.

¹¹⁷ Vgl. Kap. 2.3.3.2 im Detail

¹¹⁸ Vgl. Mellis (2004), S. 63; vgl. Lang (2004), S. 27

¹¹⁹ Vgl. Stelzer (1998), S. 83, 91; Zu einer anderen Systematik von Gestaltungszielen der Softwareentwicklung (auf Projektebene) mit u.a. Planungstreue, Stabilität, Einhaltung von Vorgehensmodellen, Wiederholbarkeit, Produktivität als Ziele; vgl. Herzwurm (2000), S. 70

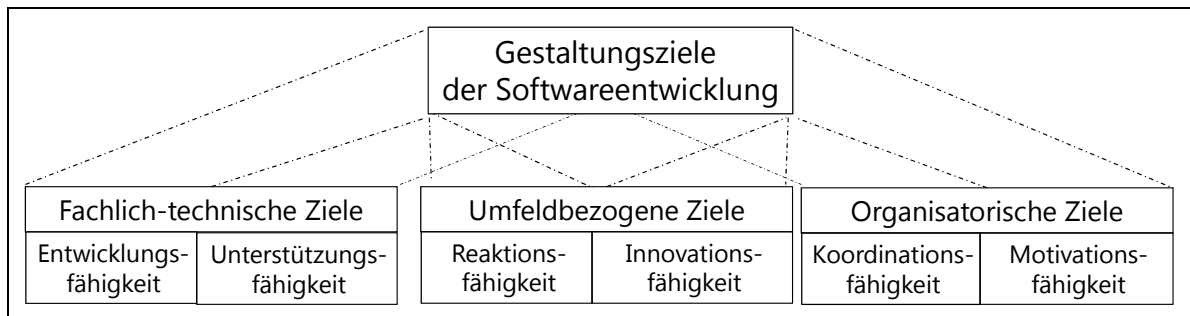


Abb. 11: Perspektiven auf die Gestaltungsziele der Softwareentwicklung¹²⁰

Fachlich-technische Gestaltungsziele der Softwareentwicklung liegen in der Erhöhung der Entwicklungs- und Unterstützungsfähigkeit. Entwicklungsfähigkeit ist die allgemeine, entwicklungsprojektübergreifende Fähigkeit einer Softwareunternehmung, die fachlich-technischen Aufgaben in den Entwicklungsvorhaben so durchzuführen, dass die Produkt- und Projektziele erreicht werden können. Insofern ist Entwicklungsfähigkeit eng mit den Produkt- und Projektzielen verbunden, lediglich die Perspektive ist eine andere. Unterstützungsfähigkeit bezeichnet das Ausmaß, in dem eine Softwareunternehmung in der Lage ist, die für die Entwicklung von Software notwendigen Unterstützungsaufgaben entwicklungsprojektübergreifend zu erbringen.¹²¹

Die fachlich-technischen Gestaltungsziele der Softwareentwicklung müssen durch geeignete Gestaltungsmaßnahmen herbeigeführt werden. Als solche Gestaltungsmaßnahmen werden aus der fachlich-technischen Perspektive die einzelnen Teilaufgaben des Softwareentwicklungsprozesses angesehen und anhand von Vorgehensmodellen der Softwareentwicklung beschrieben. Vorgehensmodelle beschreiben, welche Teilaufgaben in welcher Reihenfolge ausgeführt werden müssen. Sie definieren zudem, welche Teilaufgaben zu Entwicklungsphasen bzw. Teilprozessen zusammengefasst werden und zu welchen (Zwischen-)Ergebnissen die einzelnen Teilaufgaben und Teilprozesse führen.¹²²

Die fachlich-technischen Prozesse der Softwareentwicklung können weiterhin analog zu den Gestaltungszielen in diesem Bereich in primäre und unterstützende Teilprozesse bzw. in Entwicklungsaufgaben und Unterstützungsaufgaben untergliedert werden. Da die verschiedenen Teilprozesse schwerpunktmäßig bestimmten Phasen im Entwicklungsprozess zugeordnet werden können, entspricht diese Einteilung auch weitgehend der Unterteilung in phasenbezogene und phasenübergreifende Prozesse.¹²³

¹²⁰ Stelzer (1998), S. 97

¹²¹ Vgl. Stelzer (1998), S. 91f.

¹²² Vgl. Lang (2004), S. 33

¹²³ Vgl. Mellis (2004), S. 63f.

- Entwicklungsaufgaben dienen unmittelbar der Erfüllung des Sachziels, d.h. der Entwicklung und Bereitstellung von Software. Zu den Entwicklungsaufgaben zählen die phasenbezogenen Prozesse Konzeption, Analyse, Architekturentwurf (Grob- und Feinentwurf), Implementierung, sowie Abnahme (und Einführung).
- Als Unterstützungsaufgaben werden diejenigen Teilaufgaben zusammengefasst, die nicht eindeutig einem bestimmten primären Teilprozess zugeordnet werden können. Etwas simplifiziert ausgedrückt helfen Unterstützungsaufgaben den Trägern von Entwicklungsaufgaben bei der Erfüllung ihrer Tätigkeiten und erbringen somit Leistungen für mehrere oder alle primären Teilprozesse. Zu den Unterstützungsaufgaben werden die Teilprozesse Qualitäts-, Risiko-, Risiko-, sowie Konfigurations- und Änderungsmanagement gezählt.

Abbildung 12 fasst die fachlich-technischen Prozesse der Softwareentwicklung zusammen.

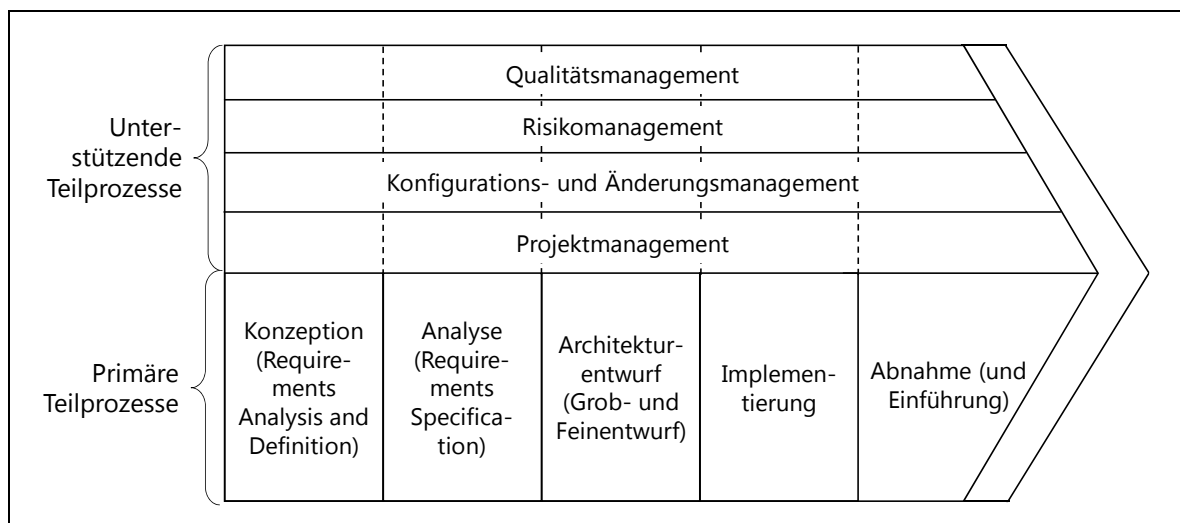


Abb. 12: Fachlich-technische Prozesse der Softwareentwicklung¹²⁴

Die umfeldbezogenen Gestaltungsziele der Softwareentwicklung werden in Reaktions- und Innovationsfähigkeit unterteilt. Reaktionsfähigkeit beschreibt die Fähigkeit, auf Veränderungen der Rahmenbedingungen mit einer veränderten Gestaltung der Entwicklungsprozesse oder der Softwareprodukte reagieren zu können. U.a. sich häufig ändernde Kundenanforderungen im Verlauf des Entwicklungsprozesses oder die Notwendigkeit, als Reaktion auf neue Softwareprodukte von Wettbewerbern reaktions-schnell eigene entsprechende Softwareprodukte ausliefern zu können, bedingen eine relativ hohe Reaktionsfähigkeit einer Softwareunternehmung. Innovationsfähigkeit bezeichnet die Fähigkeit der Softwareunternehmung, Neuerungen zu schaffen und für sich nutzbar zu machen. Neuerungen können dabei Merkmale des Produktes, aber auch Veränderungen von Methoden und Techniken des Entwicklungsprozesses betref-

¹²⁴ Zu den einzelnen Teilprozessen der Softwareentwicklung vgl. Mellis (2004), S. 63ff.

fen. Insgesamt ist die Softwareindustrie u.a. aufgrund der kurzen Innovationszyklen bei den zugrundeliegenden Technologien von einer hohen Innovationsrate gekennzeichnet.¹²⁵ Wie auch die fachlich-technischen werden die umfeldbezogenen Gestaltungsziele durch die zielführende Gestaltung der primären und unterstützenden Teilprozesse der Softwareentwicklung umgesetzt.

Die bisherige Betrachtung der fachlich-technischen Prozesse abstrahiert davon, dass v.a. bei umfangreichen Softwareentwicklungsvorhaben die Entwicklungs- und die Unterstützungsaufgaben in Arbeitsteilung erbracht werden. Dementsprechend werden aus der organisatorischen Perspektive alle Aspekte der Gestaltung von Prozessen der Softwareentwicklung betrachtet, die sich durch die arbeitsteilige Erledigung der Softwareentwicklung ergeben. Über die Aufteilung der Gesamtaufgabe und die Zuordnung der Teilaufgaben an verschiedene organisatorische Einheiten hinaus müssen eine Reihe weiterer Regelungen getroffen werden. Für die Gestaltung dieser Regelungen gibt es verschiedene Alternativen. Sie werden bei der Betrachtung aus der fachlich-technischen Perspektive, mit Ausnahme der impliziten Festlegung der Ablauforganisation durch die Phaseneinteilung in den Vorgehensmodellen, ausgeblendet.¹²⁶

Die organisatorischen Gestaltungsziele der Softwareentwicklung werden in die beiden grundlegenden Teilprobleme des Organisationsproblems, nämlich Koordinations- und Motivationsfähigkeit, getrennt. Sie bezeichnen zusammen das Ausmaß, in dem die Softwareunternehmung fähig ist, den Softwareentwicklungsprozess aus der organisatorischen Perspektive zielführend zu gestalten.¹²⁷ Da sich die vorliegende Arbeit mit der Koordination der Softwareentwicklung befasst, besteht hier das Gestaltungsziel in der Erhöhung der Koordinationsfähigkeit – speziell bei Entwicklungsvorhaben von Standardsoftware, bei welchen dies eine besondere Herausforderung darstellt.¹²⁸

Die organisatorische Gestaltung der Softwareentwicklung kann in Anlehnung an die gängigen Trennungsansätze in der Organisationslehre in die folgenden Gestaltungsbereiche aufgespalten werden:¹²⁹

- Spezialisierung als die Art und Weise einer Zerlegung der Gesamtaufgabe in abgegrenzte Teilaufgaben und ihre Zuordnung zu den verschiedenen organisatorischen Teilbereichen der Softwareunternehmung;

¹²⁵ Vgl. Stelzer (1998), S. 92-95

¹²⁶ Vgl. Mellis (2004), S. 63, 69

¹²⁷ Vgl. Stelzer (1998), S. 95f.; vgl. im Detail Kap. 2.2.1

¹²⁸ Vgl. im Detail nachfolgendes Kap. 2.1.2.2

¹²⁹ Vgl. Mellis (2004), S. 70ff.; vgl. Stelzer (1998), S. 104ff.

- Koordination als die Gesamtheit von Maßnahmen zur Ausrichtung der abgegrenzten, jedoch in der Regel voneinander abhängigen Teilaufgaben in einem arbeitsteiligen System, auf ein übergeordnetes Gesamtziel;
- Konfiguration als die Gesamtheit von Regelungen, welche Weisungsbefugnisse zwischen den organisatorischen Teilbereichen festlegen;
- Entscheidungsdelegation als die Verteilung von Entscheidungskompetenzen zwischen den organisatorischen Teilbereichen;
- Motivationsmaßnahmen als alle Möglichkeiten, die Einstellungen und das individuelle Verhalten von Mitarbeitern durch organisatorische Regelungen auf das gemeinsame Unternehmensziel auszurichten;
- Kommunikationsmaßnahmen als Regelungen zum Austausch von Informationen zwischen den verschiedenen organisatorischen Teilbereichen der Softwareunternehmung untereinander sowie mit der externen Unternehmensumwelt.

Bei einer Arbeit, die sich mit der Koordination der (Standard-)Softwareentwicklung befasst, werden die Prozesse der Softwareentwicklung naturgemäß aus der organisatorischen Perspektive heraus betrachtet. Die empirisch abgeleiteten Strukturdimensionen und Gestaltungsmaßnahmen der Koordination in Kap. 3.2 weichen dabei etwas von der hier vorgestellten (gedanklichen) Trennung der organisatorischen Gestaltungsbereiche ab. Im Wesentlichen werden Koordinationsmaßnahmen in dieser Arbeit weiter gefasst und umfassen u.a. die Konfiguration und Entscheidungsdelegation im obigen Sinn. Motivation und Kommunikation werden dagegen nicht als eigenständige Gestaltungsbereiche behandelt.¹³⁰

2.1.2.2 Gestaltungsbedingungen bei der Standardsoftwareentwicklung

Softwareentwicklungsprojekte unterscheiden sich wesentlich von Projekten auf anderen Gebieten. So ist Software ein immaterielles Produkt, dessen Fertigstellungsgrad und Qualität nur sehr schwer zu messen sind und visuell praktisch nicht wahrgenommen werden können. Insbesondere die Zwischenergebnisse im Entwicklungsprozess sind oft nicht nachvollziehbar. Hinzu kommt eine sehr hohe Dynamik, d.h. Geschwindigkeit und Ausmaß von Veränderungen im Verhalten der Auftraggeber, der technischen Weiterentwicklungen und der Veränderung der Wettbewerbssituation. Desweiteren muss bei der Softwareentwicklung auch von einer hohen Unsicherheit ausgegangen werden, da die Verfügbarkeit von Informationen über die zuvor genannten Bereiche im Verlauf des Entwicklungsprozesses gering ist.¹³¹

¹³⁰ So auch bei Mellis (2004)

¹³¹ Vgl. Henrich (2002), S. 12-16; vgl. Gerhardt (1992); vgl. Stelzer (1998), S. 56

Im Rahmen der eben genannten Besonderheiten von Softwareprojekten unterliegt die Entwicklung von Standardsoftware weiteren, spezifischen Gegebenheiten. Sie werden im Folgenden im Detail abgegrenzt. Die Merkmale des Entwicklungsgegenstandes an sich, hier Standardsoftware, führen zu drei Betrachtungsebenen im Hinblick auf eine aussagefähige Abgrenzung: Betrachtung der Produktstruktur, des Innovationsgrades sowie der externen Produktionsfaktoren.¹³²

Unter der Produktstruktur werden alle Eigenschaften der Software subsumiert, die einen Einfluss auf die Interaktion zwischen den einzelnen Produktkomponenten sowie zwischen den Kunden bzw. Auftraggebern und der Softwareunternehmung ausüben. Die Eigenschaften des Produkts werden weiter in die zwei Betrachtungsebenen Produktarchitektur und Produktkomplexität unterteilt:

- Produktarchitektur im Hinblick auf die Zusammensetzung und Interaktion der einzelnen Produktkomponenten: Es wird zwischen reinen Softwareprodukten, kombinierten Software-Hardware- oder Software-Service-Produkten, sowie Software-Hardware-Service-Produktbündeln unterschieden. Dabei ist nicht ausgeschlossen, dass die einzelnen Produktbestandteile in nachfolgenden bzw. nachgelagerten Wertschöpfungsschritten oder auch von Partnern erbracht werden. Sowohl mit der zunehmenden Einbettung von Software in Hardwareprodukte, als auch durch die zunehmende Ergänzung von Software um Service-Elemente steigen die Komplexität der Produktentwicklungsaufgabe und damit die Notwendigkeit der interdisziplinären Zusammenarbeit und Abstimmung zwischen den Beteiligten.

Standardsoftware findet als reine Softwareentwicklung statt. Erst im Vermarktungsprozess erfolgt eine Bündelung mit Servicekomponenten, bspw. durch Kopplung der Standardsoftware mit Wartungsverträgen. Eine enge Kopplung mit Hardware, wie es bei sog. Embedded Systems der Fall ist,¹³³ findet weder im Entwicklungsprozess, noch im Vermarktungsprozess des Standardsoftwareherstellers statt. Insgesamt beschränkt sich somit die Interaktion der Produktkomponenten im Entwicklungsprozess auf die einzelnen Softwarekomponenten der Standardsoftware untereinander. Die Interaktion mit Hardware- und Service-Komponenten¹³⁴ kann hier vernachlässigt werden.

¹³² Vgl. zu den im Folgenden dargestellten allgemeinen Merkmalen des Entwicklungsgegenstandes Herzwurm (2000), S. 71-76, 81-85. Zu anderen Merkmalssystematiken vgl. Stelzer (1998), S. 74-80 oder Mellis (2004), S. 41-47

¹³³ Vgl. hierzu detaillierter im Folgenden bei der Abgrenzung der Standardsoftwareentwicklung von der Entwicklung von Embedded Systems im vorliegenden Kapitel.

¹³⁴ Zur Integration mit Service-Komponenten vgl. im Folgenden bei der Abgrenzung der Standardsoftware- von der Individualsoftwareentwicklung im vorliegenden Kapitel.

- Produktarchitektur im Kontext des Gesamtsystems: Wesentlichen Einfluss auf die Interaktion zwischen den einzelnen Produktkomponenten hat die Frage, ob es sich bei dem Entwicklungsvorhaben um die Entwicklung eines einzelnen Moduls, eines Subsystems oder eines kompletten Softwaresystems handelt. Mit der Stellung des Entwicklungsvorhabens im Kontext des Gesamtsystems steigt die Interdependenzproblematik bei der Planung der Eigenschaften der Produktkomponenten bzw. Module und desto höher und kritischer wird die Notwendigkeit der aufgabenbezogenen und technischen Integration der Module.

Auch ein reines Softwareprodukt kann aus mehreren zu integrierenden Produktkomponenten bestehen. Standardsoftwareprodukte sind in der Regel solche integrierten Softwarepakete. Die aufgabenbezogene und technische Integration der einzelnen Module und damit die Koordination der einzelnen Teilprojekte zur Entwicklung der Produktbestandteile ist höchst erfolgskritisch für das Gesamtsystem.¹³⁵

- Produktkomplexität: Die Produktkomplexität steigt mit dem Vorhandensein von bestimmten Produkteigenschaften, von welchen auch wesentlich das Ausmaß der Interaktion zwischen Kunden bzw. internen Auftraggebern und der Softwareunternehmung bzw. Entwicklungsabteilung abhängt. Solche Produkteigenschaften können auf die technische Komplexität (Sicht der Softwareentwicklung) oder auf die anwenderbezogene Produktkomplexität zurückgeführt werden.

V.a. aufgrund der anwenderbezogenen Produktkomplexität durch die Vielzahl von Geschäftsprozessen, die Standardsoftware üblicherweise unterstützen muss, kann hier insgesamt von einer hohen Produktkomplexität ausgegangen werden.

Der Innovationsgrad des zu entwickelnden Produktes bestimmt den Neuheitsgrad und somit die Ungewissheit der Aufgaben bzw. der Lösungswege in der Produktentwicklung. Mit steigendem Innovationsgrad des Produktes geht somit auch eine zunehmende Komplexität, Neuheit und Variabilität des Entwicklungsprozesses einher. Im Rahmen der Produktentwicklung lassen sich aus Sicht einer Softwareunternehmung vier Innovationsgrade unterscheiden.¹³⁶

- Vorfeldentwicklung als die Entwicklung und labormäßige Erprobung von neuen Technologien, Lösungsprinzipien, Schlüsselkomponenten etc.;
- Neuentwicklung als die Entwicklung von neuen Produkten mit neuen Lösungs- bzw. Funktionsprinzipien bei gleichen, veränderten oder neuen Aufgabenstellungen;

¹³⁵ Vgl. Kap. 2.2.2

¹³⁶ Vgl. Schmelzer (1991), S. 8ff.; Im Folgenden absteigend im Innovationsgrad aufgeführt.

- Angleichung bestehender Produkte an veränderte Anforderungen im Rahmen einer Anpassungsentwicklung;
- Variantenentwicklung als die Modifikation von Baugruppen bzw. Modulen bei gleichen bzw. bereits bekannten Lösungsprinzipien.

Standardsoftwareentwicklung findet nicht als Vorfeldentwicklung statt. Entwicklungsvorhaben, die auf neuen Technologien oder Lösungsprinzipien aufsetzen und somit besonders riskant sind, werden, wie es der Name schon bezeichnet, der eigentlichen Produktentwicklung vorgezogen und als Pilotprojekte bzw. Produktprototypen bewältigt. Aufgrund des Umfangs und der relativ langen Lebensdauer von Standardsoftware finden zwischen einer Neuentwicklung mehrere Anpassungs- oder Variantenentwicklungen in Form von neuen und/oder branchenspezifischen Produktversionen statt.

Der Anteil der benötigten externen Produktbestandteile ist bei Standardsoftware relativ gering, sodass der externe Faktor Lieferanten eine untergeordnete Rolle in der Abgrenzung spielt. Der externe Faktor Kunde ist dagegen von zentraler Bedeutung bei der Charakterisierung der Entwicklung von Standardsoftware.

Standardsoftware stellt aus Sicht des Kunden fremdbezogene Software dar, die im Gegensatz zur innerbetrieblich erstellten Software oder fremdbezogener Software nach Kundenspezifikation (Individualsoftware) einen relativ geringen Grad an Kundenindividualität aufweist. Diese wird lediglich über Standardsoftware mit Varianten oder über kundenspezifischen Anpassungen erreicht. Ansonsten handelt es sich bei Standardsoftware um Software, die für den anonymen Markt entwickelt wird und hier daher die prognostizierten Bedürfnisse einer größeren Anzahl von potentiellen Kunden zugrunde gelegt werden. Auftraggeber und damit interner Kunde ist im Fall von Standardsoftware die Marketingabteilung oder das Produktmanagement der Softwareunternehmung.¹³⁷

Neben Standardsoftware bilden zwei weitere Arten von Softwareprodukten den gesamten Softwaremarkt:¹³⁸

- Individualsoftware bzw. individuelle Anwendungsentwicklung, welche an die Bedürfnisse des Kunden angepasst und singular eingesetzt wird, sowie
- eingebettete Software, die in ein technisches System integriert ist, um es zu steuern, zu regeln oder zu überwachen.

¹³⁷ Vgl. Ludewig, Lichter (2006), S. 91

¹³⁸ Vgl. Kremer (2010); vgl. Buxmann u.a. (2008), S. 33. Spielesoftware bzw. ‚interaktive Unterhaltungssoftware‘ wird wegen der Größe und den Eigenarten des Marktes, der zwischen der Software- und der Medienbranche angesiedelt ist, nicht dem Softwaremarkt zugerechnet.

Die bisherige Diskussion der Gestaltungsbedingungen bei der Standardsoftwareentwicklung bezieht sich nicht explizit auf die Abgrenzung zur Entwicklung von Individualsoftware und eingebetteter Software, auch wenn Unterschiede zur Individualsoftwareentwicklung bei den externen Produktbestandteilen sowie bei der Produktarchitektur kurz angesprochen wurden. Bei Letzterem ist auch ein Gegensatz zur Entwicklung von eingebetteter Software deutlich geworden. Diese Abgrenzung soll im Folgenden erfolgen, wobei sich die Ausführungen auf die koordinationsrelevanten Aspekte beschränken.

Viele Methoden, Techniken, organisatorische Ansätze etc., die bei der Softwareentwicklung und auch der Standardsoftwareentwicklung im Besonderen eingesetzt werden, können nahezu in gleicher Weise auch im Kontext eingebetteter Software adaptiert werden. Lediglich ein gewisser Anpassungsbedarf kann sich dadurch ergeben, dass ihre Bedeutung für die Entwicklung eingebetteter Software größer sein kann, als es bei der Entwicklung von nicht-eingebetteter Software der Fall ist. Dies gilt bspw. für die Anwendung von formalen Spezifikations- und Verifikationstechniken. Der Grund hierfür sind die hohen Sicherheitsrisiken, die aus Restfehlern bei eingebetteter Software resultieren können und damit die einhergehende Notwendigkeit nach einem verlässlichen Korrektheitsnachweis.¹³⁹ Darüber hinaus treten jedoch wesentliche Aspekte, sowohl technischer als auch organisatorischer Natur, hinzu, die ausschließlich im Kontext eingebetteter Software eine Rolle spielen. Hierzu zählen

- die Kapselung von Software in die Hardwareumgebung,
- der interdisziplinäre Charakter der Gesamtaufgabe der Systementwicklung,
- sowie eine Reihe weiterer Besonderheiten wie bspw. die hohe Bedeutung des Software-Produktlinienansatzes.

Die Kapselung von Software in die Hardwareumgebung hat erheblichen Einfluss auf die Gesamtaufgabe der Systementwicklung. Moderne Ansätze der Entwicklung von eingebetteten Systemen folgen dabei dem sog. Hardware/Software Codesign-Ansatz. Bei diesem gemeinsamen Entwurf von Hardware- und Softwarekomponenten eines Systems wird erst zu einem späten Zeitpunkt im Entwicklungsprozess entschieden, welche Teilbereiche des Gesamtsystems als Hardware und welche als Software realisiert werden sollen. Bis zu diesem Zeitpunkt werden beide Teilbereiche gemeinsam und gleichzeitig entwickelt, was erheblichen Koordinationsbedarf mit sich bringt.^{140 141}

¹³⁹ Vgl. Liggesmeyer, Rombach (2005), S. 8f.

¹⁴⁰ Dieser Zusatzaufwand wird jedoch durch bestimmte softwaretechnische Vorteile wie bspw. eine bessere Berücksichtigung von diversen Anforderungen an die Echtzeitsteuerung überkompensiert.

¹⁴¹ Vgl. Scholz (2005), S. 25, 155f.

Der Entwicklungsprozess eingebetteter Software ist einer von mehreren Subprozessen der Gesamtsystementwicklung. An dem gesamten interdisziplinären Entwicklungsvorhaben sind unterschiedliche Disziplinen beteiligt - bspw. mechanisches Engineering, elektrisches Engineering, Software Engineering, Optik etc. Die abgestimmte Zusammenarbeit dieser unterschiedlichen Disziplinen bzw. Subprozesse bedarf über das Hardware/Software Codesign hinaus nochmals weiteren Koordinationsaufwands. Dies muss durch einen zusätzlichen Prozess mit entsprechenden Prozessschnittstellen und Kommunikationskanälen erfolgen.¹⁴²

Es gibt eine Reihe weiterer Charakteristika der Entwicklung eingebetteter Systeme, die koordinationsrelevanten Einfluss auf den Subprozess der Softwareentwicklung haben. Erwähnt sei bspw. die zunehmende Notwendigkeit, gleichzeitig mehrere Varianten eines eingebetteten Systems zu entwickeln.¹⁴³ Für den Subprozess Softwareentwicklung eignet sich hierzu insbesondere der Software-Produktlinienansatz.¹⁴⁴ Auch die Tatsache, dass bei eingebetteten Systemen die klassische Produktion zu beachten ist, hat Auswirkungen auf die Koordination der Softwareentwicklung. Denn hier muss der gesamte Entwicklungsprozess letztlich noch auf den Produktionsplanungs- und Produktionsprozess abgestimmt werden.¹⁴⁵ Ein weiterer koordinationsrelevanter Prozess kommt bei der Entwicklung sicherheitskritischer eingebetteter Systeme hinzu, die wegen dieser Kritikalität eine Zulassung benötigen. Dieser Prozess dient primär der Realisierung und dem Nachweis der Sicherheit und ist mit dem eigentlichen Entwicklungsprozess eng verbunden.¹⁴⁶

Neben der Kundennähe bzw. der Individualität der Anwendungsentwicklung an sich stellen bestimmte Gegebenheiten des Individualsoftwaremarktes koordinationsrelevante Charakteristika dar. Sie lassen sich allesamt auf das wesentliche Unterscheidungsmerkmal zwischen Standard- und Individualsoftware zurückführen: Das Produkt Standardsoftware, ggf. auch mit Anpassungsmöglichkeiten, durchläuft als ein Konsum- oder Investitionsgut einen Produktentwicklungsprozess. Dagegen stellt Individualsoftware zwar auch ein (Software-)Produkt dar, doch dessen Entwicklung hat mehr den Charakter einer investiven Dienstleistung. Diese wird oftmals um weitere, funktionelle Dienstleistungen wie Organisationsberatung, Schulung, Softwarewartung und weitere softwarebezogene Leistungen ergänzt.¹⁴⁷ Wertschöpfungsprozesse bei Dienst-

¹⁴² Vgl. Liggesmeyer, Rombach (2005), S. 26f.

¹⁴³ Z.B. Motorsteuerungen für unterschiedliche Fahrzeugtypen

¹⁴⁴ Vgl. Liggesmeyer, Rombach (2005), S. 30

¹⁴⁵ Vgl. Liggesmeyer, Rombach (2005), S. 28, 30

¹⁴⁶ Vgl. Liggesmeyer, Rombach (2005), S. 8

¹⁴⁷ Vgl. Hauer (1996), S. 94f., mit weiteren Verweisen zu den einzelnen Typisierungsansätzen mit derselben Auffassung.

leistungen unterscheiden sich dabei grundlegend von einem Produktentwicklungsprozess – auch hinsichtlich koordinationsrelevanter Aspekte.¹⁴⁸

Aus diesem Dienstleistungscharakter der Individualsoftwareentwicklung ergeben sich auch die üblichen Marktpositionierungen von Individualsoftwareanbietern als Systemhäuser, IT-Beratungsunternehmungen, Systemintegratoren oder IT-Service-Dienstleister, deren Angebote nicht nur die Softwareentwicklung an sich, sondern auch die o.g. ergänzenden softwarenahen Dienstleistungen umfassen. Die Softwareentwicklung beschränkt sich dabei oftmals nur auf Einführungsprojekte, die eine Parametrisierung bzw. ein Customizing sowie gegebenenfalls eine Erweiterung von bestehenden Standardsoftware-Komponenten umfassen.¹⁴⁹

Betrachtet man das gesamte Leistungsspektrum eines Individualsoftwareanbieters mit der eben beschriebenen Marktpositionierung, so kann diese Unternehmung nicht mehr zwingend als ein Softwareanbieter im engen Sinn bezeichnet werden, da ihre primäre Aufgabe nicht eindeutig in der Entwicklung von Software liegt.¹⁵⁰ Ebenso handelt es sich bei Unternehmungen aus den sog. Software-Sekundärbranchen nicht um Softwareanbieter im engen Sinn. Hierzu zählen alle softwareentwickelnden Unternehmungen, deren primärer Unternehmungszweck jedoch nicht in der Entwicklung von Individual- oder Standardsoftware liegt. Zu den Software-Sekundärbranchen zählen praktisch alle Branchen der Wirtschaft und Verwaltung.¹⁵¹

Die Unterschiede zwischen der Software-Primärbranche und den Software-Sekundärbranchen liegen insbesondere in der Art der Softwareentwicklung. Unternehmen der Software-Primärbranche sind überwiegend mit der Neu- und Weiterentwicklung von Software beschäftigt. In den Software-Sekundärbranchen liegt der Fokus dagegen eindeutig auf der Individualisierung und Weiterentwicklung extern bezogener Standardsoftware. Diese wird wiederum nahezu ausschließlich in der Software-Primärbranche entwickelt, wobei insbesondere Softwareanbieter im engen Sinn mit 200 und mehr Mitarbeitern Standardsoftware entwickeln.¹⁵²

¹⁴⁸ Vgl. bspw. Bruhn, Strauss (Hrsg., 2007)

¹⁴⁹ Vgl. Buxmann, Hahn (2010), vgl. Buxmann u.a. (2008), S. 6

¹⁵⁰ Zu der Typisierung von Softwareanbietern im engeren und im weiteren Sinn vgl. Buxmann u.a. (2008), S. 3, 7; wobei hier bei der Einordnung von Individualsoftwareanbietern eine andere Auffassung vertreten wird.

¹⁵¹ Vgl. GfK Marktforschung u.a. (Hrsg., 2000), S. 33f.

¹⁵² Vgl. GfK Marktforschung u.a. (Hrsg., 2000), S. 72, 90

2.1.3 Softwarespezifische Umsetzungen des Situativen Ansatzes

2.1.3.1 Tailoring der Softwareentwicklung

Wie bereits kurz erläutert werden die einzelnen Teilaufgaben des Softwareentwicklungsprozesses anhand von Vorgehensmodellen beschrieben. Diese definieren, welche Teilaufgaben in welcher Reihenfolge ausgeführt und zu Entwicklungsphasen bzw. Teilprozessen zusammengefasst werden, sowie zu welchen (Zwischen-)Ergebnissen die einzelnen Teilaufgaben und Teilprozesse führen. Die situationsgerechte Anpassung von Vorgehensmodellen an die projekt- bzw. unternehmensspezifischen Gegebenheiten im Sinne des Situativen Ansatzes wird in der Softwareentwicklung, auch im deutschsprachigen Raum, als Tailoring bezeichnet.¹⁵³

Tailoring wird grundsätzlich durch Weglassen, Hinzufügen, Ändern oder Konkretisieren von Prozesselementen¹⁵⁴ vollzogen. Da dies auf verschiedenen Ebenen erfolgen kann, wird zwischen unternehmensweitem und projektbezogenem Tailoring unterschieden. Dabei kann das Tailoring systematisch, d.h. anhand von Regeln und/oder Richtlinien und somit als ein formaler Prozess, oder gänzlich informal und ad hoc erfolgen:¹⁵⁵

- Unternehmensweites Tailoring findet in Zusammenhang mit der Adaption eines industrieweiten Prozessrahmenwerks oder eines Vorgehensmodells statt. Ziel ist dessen Anpassung an die unternehmensspezifischen Gegebenheiten. Das Ergebnis ist ein unternehmensweit gültiger und verbindlicher Softwareentwicklungsprozess.¹⁵⁶ Da sich die von einer Softwareunternehmung durchgeführten Entwicklungsprojekte wesentlich voneinander unterscheiden können, ist weiterhin Tailoring auf der Projektebene notwendig. V.a. für Softwareunternehmungen mit vielen parallel stattfindenden und zudem umfangreichen Entwicklungsprojekten, was besonders bei großen Anbietern von Standardsoftware den Normalfall darstellt, muss der unternehmensweit gültige Softwareentwicklungsprozess auf die jeweiligen Projektgegebenheiten zugeschnitten werden.
- Informale Tailoringansätze finden sich v.a. in kleinen Softwareunternehmungen, wo die Vorteile von einem pragmatischen und einfachen Tailoringprozess dessen Nachteile wie die Abhängigkeit von der Erfahrung und den Präferenzen der Beteiligten am Tailoringprozess überwiegen.¹⁵⁷ Der großen Mehrzahl von Tailoringan-

¹⁵³ Vgl. Bund (Hrsg., 2008)

¹⁵⁴ Schritte, Meilensteile, Rollen etc.

¹⁵⁵ Vgl. Pedreira u.a. (2007), S. 3f.

¹⁵⁶ Vgl. Nanda (2001)

¹⁵⁷ Vgl. bspw. Hanssen u.a. (2005)

sätzen liegt aber ein mehr oder weniger formaler Prozess zugrunde.¹⁵⁸ Das Tailoring auf der Projektebene kann dabei in jedem Einzelfall oder über den Zwischenschritt der Definition von Projekttypen erfolgen. Letzteres ist sinnvoll, sofern die definierten Projektklassen häufig genug als tatsächliche Projekte im Softwareunternehmen durchgeführt werden. Die Definition von Projektklassen ist dabei bereits an sich schon eine Formalisierung und Systematisierung des Tailoring. Zwangsläufig handelt es sich dabei also um systematisches Tailoring.

Für die vorliegende Arbeit ist das systematische Tailoring auf Projektebene von Relevanz.¹⁵⁹ Die beiden wohl bekanntesten und in der Praxis sehr verbreiteten industrieweiten Prozessstandards, das Prozessrahmenwerk Capability Maturity Model Integration (CMMI)¹⁶⁰ und das Vorgehensmodell V-Modell XT¹⁶¹, geben auch zum Projekt-Tailoring Empfehlungen ab, die im Folgenden kurz skizziert werden.

Das CMMI ist ein sogenanntes Reifegradmodell. Reife bezieht sich dabei auf das Ausmaß, in dem der Softwareentwicklungsprozess definiert und beschrieben ist und in dem er geplant, gesteuert und kontrolliert werden kann. Das wesentliche Strukturierungsmerkmal des CMMI sind seine fünf Reifegrade ‚Initial‘, ‚Managed‘, ‚Defined‘, ‚Quantitatively Managed‘ und ‚Optimizing‘. Sie werden wiederum u.a. durch bestimmte Prozessgebiete strukturiert. Ein Prozessgebiet des CMMI entspricht dabei weitgehend einem fachlich-technischen oder organisatorischen Prozess der Softwareentwicklung. Das CMMI besteht aus insgesamt 22 Prozessgebieten, die sich auf die verschiedenen Reifegrade verteilen. D.h. die Reifegrade setzen sich aus verschiedenen Prozessgebieten zusammen, die ein Softwareunternehmen beherrschen muss, um ebendiesen Reifegrad seines Softwareentwicklungsprozesses erreichen zu können (vgl. Abb. 13).¹⁶²

Das Prozessgebiet ‚Organizational Process Definition‘, das dem Reifegrad 3 zugeordnet ist, fordert die Etablierung und Pflege eines zielgerichteten unternehmensweiten Softwareentwicklungsprozesses (unternehmensweites Tailoring). Dieser setzt sich aus den (Beschreibungen der) einzelnen Teilprozesse der Softwareentwicklung zusammen. Ein weiterer wesentlicher Bestandteil sind Regeln und Kriterien zum Tailoring des

¹⁵⁸ Bei 67% der von Pedreira u.a. (2007) analysierten Veröffentlichungen zum Tailoring.

¹⁵⁹ Vgl. Kap. 3.4

¹⁶⁰ Vgl. SEI (Hrsg., 2006); Alleine zwischen 2002 und 2009 haben sich 4468 öffentliche und nicht-öffentliche softwareentwickelnde Organisationen einer Begutachtung unterzogen (sog. Appraisal) und das Begutachtungsergebnis SEI zur Veröffentlichung zurückgemeldet. Hinzu kommt eine große Anzahl an Begutachtungen, deren Ergebnisse dem SEI nicht vorliegen. Vgl. SEI (Hrsg., 2010)

¹⁶¹ Vgl. Bund (Hrsg., 2008). Das V-Modell XT ist generell bei allen zivilen und militärischen IT-Vorhaben des Bundes einzusetzen. Eine eventuelle Möglichkeit davon abzuweichen besteht nur mit Begründung. Vgl. Bund (Hrsg., 2009), URL siehe Literaturverzeichnis

¹⁶² Vgl. Herzwurm, Pietsch (2009), S. 255f.

Softwareentwicklungsprozesses auf die einzelnen Entwicklungsprojekte im Unternehmen (projektbezogenes Tailoring). Sie definieren die Vorgehensweise, anhand welcher Projekte ihren projektspezifischen Entwicklungsprozess über die Übernahme und Anpassung der einzelnen Teilprozesse bzw. Prozesselemente ableiten. Dazu gehören u.a. auch bestimmte Anforderungen an die Dokumentation des Tailoring-Prozesses an sich, sowie an die Dokumentation des definierten projektspezifischen Softwareentwicklungsprozesses. Insgesamt also lässt sich festhalten: Fehlen Regeln und Kriterien zum Tailoring des Softwareentwicklungsprozesses, so kann das Softwareunternehmen höchstens den Reifegrad 2 des CMMI erreichen.¹⁶³

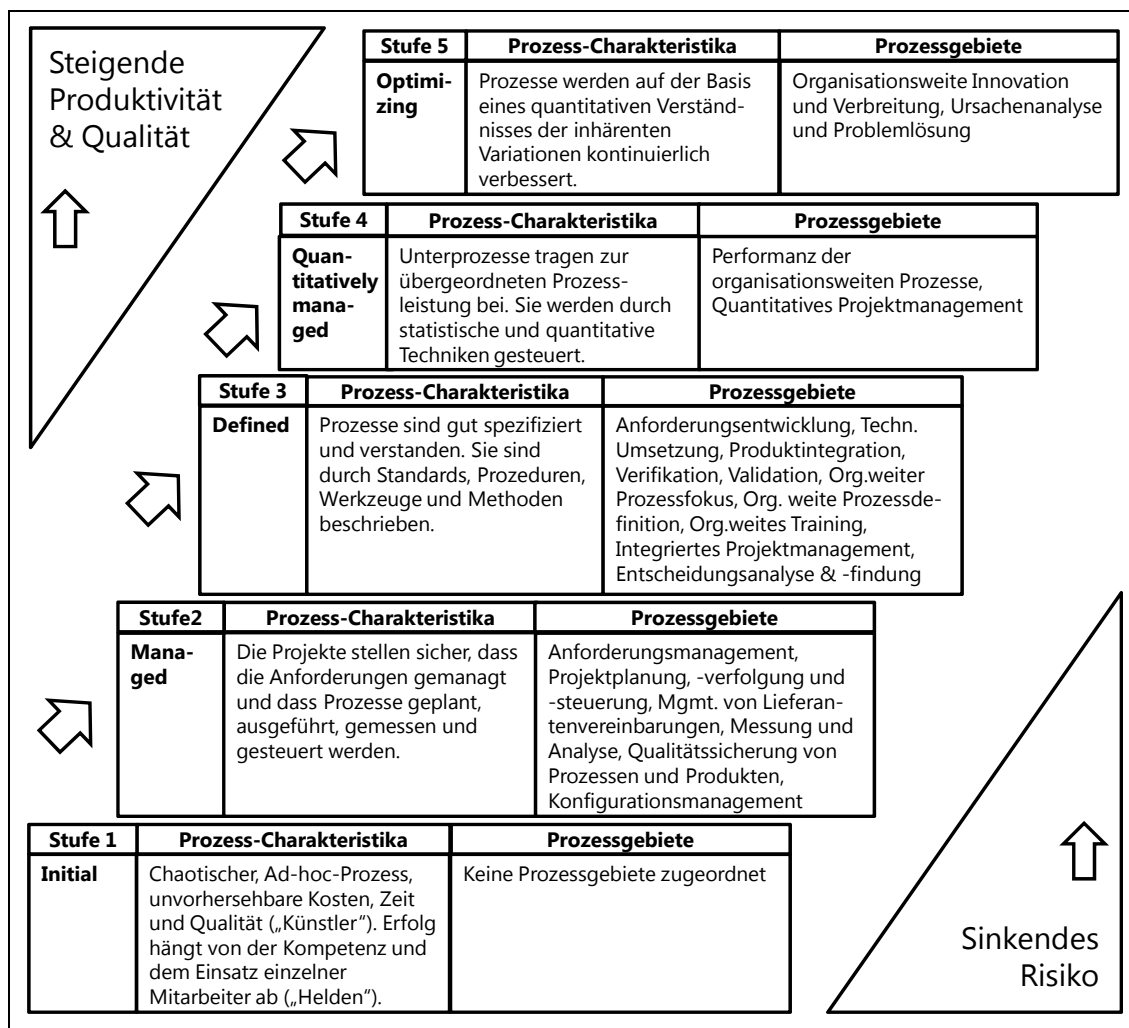


Abb. 13: Reifegrade des CMMI¹⁶⁴

Das CMMI beschreibt, welche Prozessgebiete Softwareunternehmungen auf den verschiedenen Reifegraden beherrschen müssen. Es empfiehlt jedoch keine spezifischen Methoden oder Werkzeuge – d.h. vereinfacht gesprochen beschreibt das CMMI nicht, wie diese Prozessgebiete im Einzelnen ausgeführt werden sollen.¹⁶⁵ Konkrete Empfeh-

¹⁶³ Vgl. SEI (Hrsg., 2006), S. 34, 37, 53, 224ff.

¹⁶⁴ Vgl. Balzert (2008), S. 572; vgl. SEI (Hrsg., 2006)

¹⁶⁵ Vgl. Herzwurm, Pietsch (2009), S. 257

lungen in diesem Sinn können aber Vorgehensmodellen der Softwareentwicklung, wie bspw. dem V-Modell XT, entnommen werden. Das Prozessgebiet ‚Organizational Process Definition‘ wird dabei vollständig vom V-Modell XT abgedeckt¹⁶⁶ und wie folgt umgesetzt.

Die Anpassung des V-Modell XT an ein konkretes Projekt erfolgt über einen dreistufigen Prozess: Die Festlegung des Projekttyps als erster Schritt bildet die Basis für die Auswahl der sog. Projektdurchführungsstrategie in zweiten Schritt. Anschließend erfolgt im letzten Schritt die Festlegung der konkreten Vorgehensbausteine.¹⁶⁷ Dabei wird im V-Modell XT zwischen vier grundlegenden Projekttypen unterschieden:¹⁶⁸

- Systementwicklungsprojekt aus Sicht eines Auftraggebers;
- Systementwicklungsprojekt aus Sicht eines Auftragnehmers;
- Systementwicklungsprojekt eines Auftragnehmers mit Auftragnehmer in der gleichen Organisation;
- Einführung und Pflege eines organisationsspezifischen Vorgehensmodells.

Für jeden der vier Projekttypen definiert das V-Modell XT mindestens eine anwendbare Projektdurchführungsstrategie. Sie umschreibt den grundlegenden Rahmen für eine geordnete und nachvollziehbare Durchführung des Projektes. Dazu gehört die Festlegung des inhaltlichen und zeitlichen Ablaufes eines Projektes, um so eine zuverlässige Planung und Steuerung des Projektes zu ermöglichen. Es wird eine Reihenfolge festgelegt, in der die Vorgehensbausteine und die darin enthaltenen Aktivitäten und Produkte erbracht, sowie die für das Projekt relevanten Entscheidungspunkte durchlaufen werden. Welche Projektdurchführungsstrategie für welchen Projekttyp geeignet ist, wird anhand von Projektmerkmalen bestimmt.¹⁶⁹

- Für den Projekttyp ‚Einführung und Pflege eines organisationsspezifischen Vorgehensmodells‘ existiert nur eine geeignete, gleichnamige Projektdurchführungsstrategie, sodass hier keine Projektmerkmale notwendig sind;
- Für den Projekttyp ‚Systementwicklungsprojekt eines Auftraggebers‘ erfolgt die Wahl der Projektdurchführungsstrategie anhand des Projektmerkmals Projekttrolle. Je nachdem, ob der Auftraggeber mit einem oder mit mehreren Auftragnehmern gleichzeitig zusammenarbeitet, ergibt sich die entsprechende Projektdurchführungsstrategie;

¹⁶⁶ Vgl. Bund (Hrsg., 2008), Teil 7, S. 22

¹⁶⁷ Vgl. Bund (Hrsg., 2008), Teil 8, S. 38, Teil 1, S. 15

¹⁶⁸ Vgl. Bund (Hrsg., 2008), Teil 8, S. 35

¹⁶⁹ Vgl. Bund (Hrsg., 2008), Teil 8, S. 34, Teil 1, S. 18

- Für die Ermittlung der geeigneten Projektdurchführungsstrategie innerhalb der beiden Projekttypen ‚Systementwicklungsprojekt eines Auftragnehmers‘ und ‚Systementwicklungsprojekt eines Auftraggebers mit Auftragnehmer in der gleichen Organisation‘ leiten sich die Projektmerkmale
 - aus dem Systemlebenszyklusausschnitt, der mit dem Projekt abgedeckt wird,
 - aus der Frage, ob projektextern bezogene Systemelemente (Software- oder Hardwaremodule) bei der Entwicklung berücksichtigt werden müssen, so wie
 - aus der Höhe der Realisierungsrisiken ab.

Abb. 14 zeigt die Zuordnung von Projektdurchführungsstrategien zu den Projekttypen.

Projekttyp +	Systementwicklungsprojekt (AG)	Systementwicklungsprojekt (AN)	Systementwicklungsprojekt (AG/AN)	Einführung und Pflege eines organisations-spezifischen Vorgehensmodells
	weitere Projektmerkmale ↓	Projekttrolle	Systemlebenszyklusausschnitt	
Hohe Realisierungsrisiken			Hohe Realisierungsrisiken	
Fertigprodukte			Fertigprodukte	
Projektdurchführungsstrategie	Vergabe und Durchführung eines Systementwicklungsprojektes (AG)	Inkrementelle Systementwicklung (AN)	Inkrementelle Systementwicklung (AG/AN)	Einführung und Pflege eines organisations-spezifischen Vorgehensmodells
	Vergabe und Durchführung mehrerer Systementwicklungsprojekte (AG)	Komponentenbasierte Systementwicklung (AN)	Komponentenbasierte Systementwicklung (AG/AN)	
		Agile Systementwicklung (AN)	Agile Systementwicklung (AG/AN)	
		Wartung und Pflege von Systemen (AN)	Wartung und Pflege von Systemen (AG/AN)	

Abb. 14: Zuordnung von Projektdurchführungsstrategien zu den Projekttypen im V-Modell XT¹⁷⁰

Die konkrete Festlegung der Vorgehensbausteine im dritten Schritt ist der zentrale Punkt beim Tailoring des V-Modell XT. Ein Vorgehensbaustein kapselt alle diejenigen Produkte, Aktivitäten und Rollen, die für die Erfüllung einer bestimmten Aufgabenstellung im Projekt relevant sind und damit inhaltlich zusammengehören, wie bspw. Inhalte des Projektmanagements. Beim Tailoring werden die für ein konkretes Projekt benötigten Vorgehensbausteine aus den insgesamt 21 Vorgehensbausteinen des V-Modells XT entsprechend den Vorgaben des Projekttyps bez. verpflichtender

¹⁷⁰ Bund (Hrsg., 2008), Teil 1, S. 17

und optionaler Vorgehensbausteine ausgewählt. Anpassungen innerhalb von Vorgehensbausteinen stellen den Ausnahmefall dar.¹⁷¹

Neben den Vorgaben zum Tailoring in industrieweiten Prozessrahmenwerken und Vorgehensmodellen existieren zahlreiche, in den Softwareunternehmungen direkt oder in Zusammenarbeit mit einer Forschungseinrichtung, entstandenen Ansätze.¹⁷² Die überwiegende Mehrzahl der Ansätze baut dabei auf der Definition von Projektklassen auf, wenngleich sie von völlig verschiedenen Gestaltungsbedingungen bzw. Tailoring-Kriterien ausgehen und damit auch verschiedene Gestaltungsempfehlungen machen.

Der Ansatz der Softwareunternehmung IBM ist insofern erwähnenswert, als dass hier die Definition von Projektklassen abgelehnt wird. Es wird stattdessen argumentiert, dass sich die potentielle Vielfalt von Gestaltungsbedingungen bei der Softwareentwicklung und den notwendigen Maßnahmen als Reaktion darauf nicht in einige wenige Kategorien fassen lässt.¹⁷³ Für die Softwareentwicklung sei stattdessen ein flexiblerer und modularisierbarer Tailoringansatz notwendig. Ein solcher Tailoringansatz wird bei IBM umgesetzt, in dem die (Zwischen-)Produkte¹⁷⁴ als greifbare Ergebnisse von Teilprozessen bzw. Aktivitäten der Softwareentwicklung das primäre Element des Tailoring darstellen.¹⁷⁵ Insgesamt über 300 sog. Work Product Descriptions (WPDs), also Beschreibungsschablonen für (Zwischen-)Produkte über den gesamten Softwareentwicklungsprozess hinweg, definieren aus dieser Sicht den Softwareentwicklungsprozess von IBM. Die WPDs beschreiben jeweils

- Ziel und Inhalt des zu erbringenden (Zwischen-)Produktes,
- diverse detaillierte Anweisungen zur Umsetzung und Validierung des zu erbringenden (Zwischen-)Produktes,
- die Auswirkungen durch das Weglassen des zu erbringenden (Zwischen-)Produktes, sowie
- die möglichen Gründe für das Weglassen des zu erbringenden (Zwischen-)Produktes.¹⁷⁶

¹⁷¹ Vgl. Bund (Hrsg., 2008), Teil 8, S. 38ff., Teil 1, S. 13ff.

¹⁷² Vgl. bspw. Shenhar u.a. (2005), Fitzgerald u.a. (2003), Cameron (2002); Zu einer tabellarischen Zusammenstellung von zahlreichen Tailoringansätzen vgl. Xu, Ramesh (2007), S. 298f.

¹⁷³ In diesem Zusammenhang vgl. Kap. 3.4.1.2 zur Begründung der Umsetzung des Tailoring in dieser Arbeit mithilfe von Projekttypen.

¹⁷⁴ Bspw. Lastenheft, Pflichtenheft etc. Zu einer vollständigen alphabetischen Auflistung aller (Zwischen-)Produkte bspw. im V-Modell XT vgl. Bund (Hrsg., 2008), Teil 5, S. 7

¹⁷⁵ Vgl. Cameron (2002), S. 72

¹⁷⁶ Vgl. Cameron (2002), S. 73

Die letzten beiden Punkte unterstützen unmittelbar das Tailoring. Es findet auf der Projektebene zu Beginn der Projektplanungsphase statt. Die zentrale Fragestellung ist es, ob das (Zwischen-)Produkt in dem konkreten Projekt notwendig ist oder nicht. Die Tailoring-Kriterien ergeben sich aus diversen Informationen, die während der Planungsphase auch bei anderen Entscheidungen eine zentrale Rolle spielen, bspw. Projektgröße, Erfahrung des Projektteams, Projektrisiken etc. Eine tabellarische Zusammenstellung aller WPDs als Ausgangspunkt bietet dabei weitere Entscheidungs- und Dokumentationsunterstützung.¹⁷⁷

2.1.3.2 Risikobasierte Softwareentwicklung und Agile Softwareentwicklung

Die Grundidee der risikobasierten Softwareentwicklung liegt darin, zuerst die Aufgaben detailliert auszuarbeiten, welche mit dem größten Risiko behaftet sind. So wird die Ressourcenbindung für nicht erfolgreich durchgeführte Aufgaben und somit das Projektfehlschlagrisiko insgesamt erheblich vermindert. Der bekannteste Ansatz der risikobasierten Softwareentwicklung ist das sog. Spiralmodell. Es ist ein Ansatz zur risikobasierten Auswahl eines Vorgehensmodells der Softwareentwicklung und baut somit gewissermaßen einen Rahmen für die Anwendung und Nutzung anderer Vorgehensmodelle auf. Das Spiralmodell ermöglicht eine flexible Anpassung an unterschiedliche Ausgangsbedingungen im Sinne des Situativen Ansatzes.¹⁷⁸ Abbildung 15 zeigt den grundsätzlichen Ablauf des Spiralmodells.

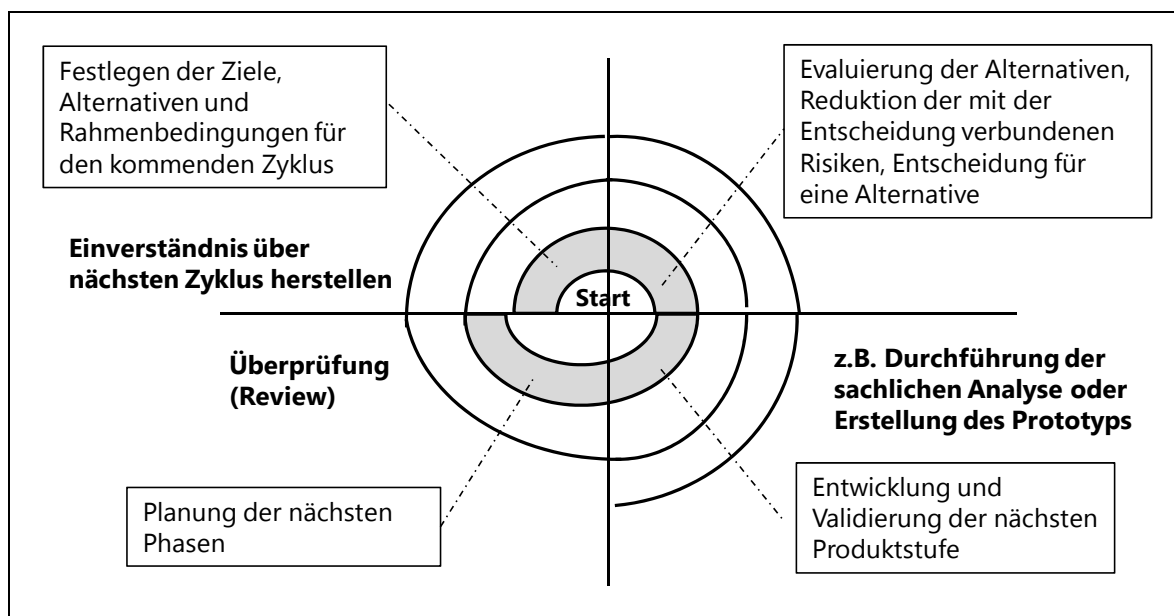


Abb. 15: Grundsätzlicher Ablauf des Spiralmodells¹⁷⁹

¹⁷⁷ Vgl. Cameron (2002), S. 74f.

¹⁷⁸ Vgl. Pietsch (1992), S. 95f.; zur Erstveröffentlichung zum Spiralmodell im Original vgl. Boehm (1986)

¹⁷⁹ Vgl. Henrich (2002), S. 52

Ein vollständiger Umlauf wird im Spiralmodell in vier Teilphasen unterteilt, die jeweils einer Phase bzw. einem primären Teilprozess der Softwareentwicklung entsprechen. So könnte der in der Abbildung grau hinterlegte Bereich z.B. die Phase Feinentwurf, Grobentwurf oder Implementierung darstellen. Die vier Teilphasen des Spiralmodells können wie folgt beschrieben werden:¹⁸⁰

- In der ersten Phase werden die inhaltlichen Vorgaben bzw. Ziele, alternative Vorgehensweisen und die Rahmenbedingungen für den kommenden Zyklus, bzw. mit anderen Worten für den nächsten Arbeitsschritt, festgelegt. So könnten bspw. bei der Phase Grobentwurf unterschiedliche Analyse- und Entwurfstechniken aufgelistet sowie Einschränkungen hinsichtlich Zeit, Personal und Kosten, die für die Entscheidung zu beachten sind, ermittelt werden;
- In der zweiten Phase erfolgen die Evaluierung der Alternativen und die Auswahl der Vorgehensweise, welche die mit dieser Entscheidung verbundenen Risiken so gering wie möglich hält;
- In der dritten Phase erfolgt die Realisierung der ausgewählten Alternative und damit die Realisierung und Überprüfung des Zwischenproduktes. Im Beispiel könnte das die Durchführung der fachlichen Analyse und ggf. die Erstellung eines Prototypen zur Bewältigung der größten Risiken sein;
- Die vierte Phase dient zur Überprüfung der Ergebnisse des aktuellen und zur Planung des nächsten Zyklus.

Das Spiralmodell dient seinem Autor Boehm als konzeptionelle Grundlage für die Entwicklung von weiteren risikobasierten Ansätzen der Softwareentwicklung. Die Arbeiten in diesem Umfeld finden mittlerweile im Zuge des Forschungsprogramms VBSE¹⁸¹ statt. So stellen die sog. ‚SCQAIV‘¹⁸²-Prozessmodelle eine direkte Weiterentwicklung des Spiralmodells dar. Boehm geht hier bei seinen Überlegungen davon aus, dass Softwareentwicklungsprojekte üblicherweise den geplanten Leistungsumfang der Software als absolut fix betrachten und einen entsprechend darauf ausgerichteten Entwicklungsprozess einsetzen. Er fordert daher ein Umdenken hin zum Situativen Ansatz, bei dem in Abhängigkeit von den jeweiligen Risiken entweder der Fertigstellungstermin, die Kosten oder die Qualität als fix oder als ‚Stellschrauben‘ im Verlauf des Entwicklungsprozesses betrachtet werden (‚SCQAIV‘-Prozessmodell).¹⁸³

¹⁸⁰ Vgl. Mellis (2004), S. 90f.; vgl. Henrich (2002), S. 51-54

¹⁸¹ Vgl. Kap. 1.3

¹⁸² SCQAIV: schedule/cost/quality as independent variable

¹⁸³ Vgl. Boehm u.a. (2002), S. 1f.; vgl. Boehm u.a. (2000)

Die immer noch eher konzeptionellen Vorschläge der ‚SCQAIV‘-Prozessmodelle werden in weiteren Arbeiten zu zwei Situativen Ansätzen konkretisiert, dem ‚Risk Driven Model-Based Architecture and Software Engineering Spiral Approach‘ und in einer etwas abgewandelten Darstellung zum ‚Value-Based Software Quality Achievement Process Framework‘.¹⁸⁴ Als Gestaltungsbedingung dient bei beiden Ansätzen die Zuordnung des Entwicklungsprozesses zu einer der Kategorien ‚CAIV‘ und ‚SAIV‘.¹⁸⁵ Dies erfolgt in Abhängigkeit von den jeweiligen Risiken und im Wesentlichen über ein Risiko-Assessment, wie im Risikomanagement allgemein üblich. Die Risikoanalyse und die Kategorisierung der Risiken als termin-, kosten- oder qualitätsbezogenes Risiko sind dabei eng verknüpft mit dem Geschäftsszenario („Business-Case“) des Entwicklungsvorhabens. Dominieren bspw. qualitätsbezogene Risiken, so wird von einem ‚Produkt-getriebenen‘ Geschäftsszenario („Product-Driven Business Case“) ausgegangen. Stehen dagegen terminbezogene Risiken im Vordergrund, so handelt es sich um einen ‚Schedule-Driven Business Case‘.¹⁸⁶ Abbildung 16 stellt die beiden genannten sowie ein weiteres Geschäftsszenario („Market Trend-Driven Business Case“) beispielhaft dar.

Business Case	Schedule-Driven	Market Trend-Driven	Product-Driven
Primary Objective	Rapid value by adding small extra functionalities	Rapid Market Share Occupation	Version upgrade with Q-attribute achievement: reliability, availability, performance, evolvability, etc.
Quality Risks	Low	Medium	High; major business losses
Schedule Risks	High; major business losses	High; market share loss	Low
Stakeholders	Single collocated representatives	Many success-critical stakeholders	Multiple success-critical stakeholders with various Q-attribute requirements
Requirements	1) A few specific and stable requirements; 2) Mostly functional	1) Goals generally known (e.g., platform changes); 2) Detailed requirements often vague, volatile and emergent; 3) Functional and non-functional	1) Critical and conflicting Q-attribute requirements from various stakeholders; 2) Most requirements relatively stable; others volatile, emergent 3) Functional and nonfunctional
Architecture	1) Extend from existing system architecture 2) Little architecting effort 3) Stakeholder high confidence	1) Brand new architecture; 2) Most architecting effort; 3) Stakeholder low confidence	1) Evolve based on existing product-line architecture 2) High confidence in some parts; low confidence in others
Refactoring	Inexpensive with skilled people	More expensive with mix of people skills	Very expensive, with mix of people skills

Abb. 16: Beispielhafte Vorhabenskategorien beim Ansatz ‚Value-Based Software Quality Achievement Process Framework‘¹⁸⁷

¹⁸⁴ Auf die Diskussion der Unklarheiten und Inkonsistenzen zwischen den beiden Ansätzen wird an dieser Stelle verzichtet.

¹⁸⁵ CAIV: cost as independent variable; SAIV: schedule as independent variable

¹⁸⁶ Vgl. Huang u.a. (2006), S. 58

¹⁸⁷ Huang u.a. (2006), S. 59

Die Gestaltungsmaßnahmen sind mit dem Verweis auf eine notwendige risikobezogene Projektstrategie incl. entsprechender Migrationspläne für die entweder termin-, kosten- oder qualitätsbezogenen Risiken¹⁸⁸ sehr allgemein gehalten. Lediglich für das ‚SAIV‘-Prozessmodell, bei dem terminbezogene Risiken dominieren, werden konkrete und empirisch entwickelte Gestaltungsmaßnahmen vorgeschlagen. Sie zielen im Prinzip darauf ab, dass der Softwareentwicklungsprozess stets die geplanten Meilensteine termingerecht erreicht:¹⁸⁹

- Die Erwartungen aller Projektbeteiligten müssen durch ein gemeinsames Verständnis über die erreichbaren Projektziele definiert werden;
- Die einzelnen Bestandteile des geplanten Leistungsumfangs müssen priorisiert werden, wobei Kernfunktionalitäten, die unbedingt zu erbringen sind, definiert werden;
- Die Priorisierung des Leistungsumfangs, die Erwartungen der Projektbeteiligten, sowie der wahrscheinlich machbare Leistungsumfang müssen kontinuierlich aktualisiert werden;
- Optionale Bestandteile des geplanten Leistungsumfangs müssen definiert werden;
- Die Softwarearchitektur muss so konzipiert werden, dass die Hinzunahme optionaler Funktionalität und die Reduzierung des Leistungsumfangs flexibel und ohne Gefährdung der Kernfunktionalität möglich sind;¹⁹⁰
- Die Software muss inkrementell, beginnend mit der Kernfunktionalität und den hoch priorisierten Bestandteilen, entwickelt werden;¹⁹¹
- Ein Änderungsprozess und ein Projektcontrolling müssen definiert sein und auch stattfinden.

Ein weiterer risikobasierter Ansatz der Softwareentwicklung, der konzeptionell im Spiralmodell verwurzelt ist, beschäftigt sich mit der Fragestellung, welcher Aufwand für die Projektplanung und die Entwicklung der Softwarearchitektur für ein konkretes Entwicklungsvorhaben optimal ist. Für eine mehr bzw. weniger ausgeprägte Softwarearchitektur und Projektplanung stehen dabei als alternative Gestaltungsmaßnahmen

- die ‚klassische‘, d.h. auf diversen Plänen wie bspw. dem Projektplan und der Softwarearchitektur basierende Softwareentwicklung, und auf der anderen Seite
- die sog. Agile Softwareentwicklung.

¹⁸⁸ Vgl. Huang u.a. (2006), S. 58

¹⁸⁹ Vgl. Boehm u.a. (2002), S. 1f.

¹⁹⁰ Vgl. hierzu auch Kap. 2.2.2.3

¹⁹¹ Vgl. hierzu auch Kap. 2.2.2.2

Das Ziel der Agilen Softwareentwicklung ist es, den Softwareentwicklungsprozess flexibler und schlanker zu gestalten, als das nach der Argumentation der Verfechter der Agilen Softwareentwicklung bei der ‚klassischen‘ Softwareentwicklung der Fall ist. Des Weiteren soll eine Fokussierung auf die Sachziele sowie auf die technischen und sozialen Probleme der Softwareentwicklung erreicht werden. Alle konkreten Ansatzpunkte, mit welchen die genannten (Gestaltungs-)Ziele der Agilen Softwareentwicklung erreicht werden sollen, leiten sich aus dem sog. Agilen Manifest ab:

- “Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan;
- That is, while there is value in the items on the right, we value the items on the left more.”¹⁹²

Das Agile Manifest wird von insgesamt 12 Prinzipien konkretisiert. Sie können als die grundsätzlichen Gestaltungsmaßnahmen der Agilen Softwareentwicklung aufgefasst werden. Dabei sind aus der organisatorischen Perspektive auf die Softwareentwicklung v.a. zwei Gestaltungsmaßnahmen relevant:¹⁹³

- Bevorzugte Kommunikations- und Koordinationsform innerhalb und über die Grenzen eines Entwicklungsprojektes hinaus (bspw. zu Kunden) ist der informale persönliche Informationsaustausch;
- Die Sachziele der Softwareentwicklung werden am besten durch sich selbst organisierende Teams erreicht.

Als Gestaltungsbedingungen für die beiden alternativen Gestaltungsmaßnahmen Agile und klassische Softwareentwicklung¹⁹⁴ dienen bei diesem risikobasierten Ansatz fünf kritische Faktoren, von welchen die Eignung der beiden Ansätze der Softwareentwicklung abhängig gemacht wird (vgl. Abb. 17). Sie werden jedoch in Abweichung zum Situativen Ansatz der Organisationsgestaltung nicht als Gestaltungsbedingungen, sondern als ‘Planungs-getriebene‘ und ‚Agile Risiken‘ bezeichnet und interpretiert. In diesem Sinn vollzieht sich auch die Wahl zwischen den beiden alternativen Ansätzen der Softwareentwicklung über ein Risiko-Assessment.

¹⁹² Agile Alliance (Hrsg., 2001a), URL siehe Literaturverzeichnis

¹⁹³ Vgl. Agile Alliance (Hrsg., 2001b), URL siehe Literaturverzeichnis

¹⁹⁴ Die Abgrenzung der Agilen und ‚klassischen‘ Softwareentwicklung orientiert sich hier im Wesentlichen am Agilen Manifest; im Detail hierzu vgl. Boehm, Turner (2004), S. 51f.

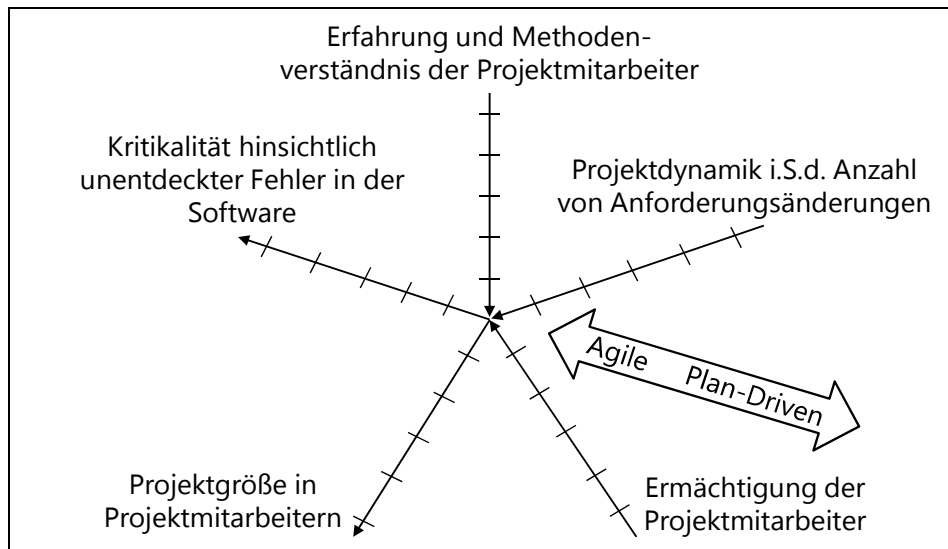


Abb. 17: Gestaltungsbedingungen beim risikobasierten Ansatz von Boehm und Turner¹⁹⁵

Die Agile Softwareentwicklung wurde bisher als eine gewissermaßen entgegengesetzte Gestaltungsmaßnahme zur klassischen Softwareentwicklung innerhalb eines situativen Ansatzes beschrieben. Darüber hinaus weist die Agile Softwareentwicklung jedoch auch selber Tendenzen hin zu einem situativen Ansatz auf – etwa in der Zielsetzung, den Softwareentwicklungsprozess flexibler und schlanker zu gestalten, sowie in der Aussage des Agilen Manifest „responding to change over following a plan“¹⁹⁶.

Die Frage, in wie weit die Agile Softwareentwicklung tatsächlich einen Situativen Ansatz darstellt, bildet einen Aspekt bei einer Untersuchung von 9 einzelnen Methoden, die sich im Umfeld der Agilen Softwareentwicklung herausgebildet haben. Dabei werden die agilen Methoden daraufhin untersucht, ob und in wie weit sie Gestaltungsbedingungen angeben, die für ihren zielführenden Einsatz erfüllt sein müssen, und in wie weit sie der jeweiligen Situation anpassbar sind.¹⁹⁷ Als Ergebnis wird festgehalten, dass nicht situative, universelle agile Methoden dominieren. Die wenigen situativen agilen Methoden, hier allen voran ‚Crystal‘, bieten zudem kaum Hilfestellung für ihr Tailoring im Hinblick auf die jeweilige Situation an. Beide Erkenntnisse werden als eine Forschungslücke herausgehoben, die durch die (Weiter-)Entwicklung von situativen agilen Methoden geschlossen werden sollte.¹⁹⁸

Die agile Methode ‚Crystal‘ hebt sich als ausgesprochen situativ von den anderen agilen Methoden ab. ‚Crystal‘ besteht aus mehreren Prozessmodellen - welches davon im Einzelfall Anwendung finden soll, hängt von drei Gestaltungsbedingungen ab:¹⁹⁹

¹⁹⁵ Vgl. Boehm, Turner (2004), S. 56

¹⁹⁶ Agile Alliance (Hrsg., 2001a), URL siehe Literaturverzeichnis

¹⁹⁷ Vgl. Abrahamsson u.a. (2003), S. 247

¹⁹⁸ Vgl. Abrahamsson u.a. (2003), S. 249-251

¹⁹⁹ Vgl. Cockburn (2005), S. 244-246; vgl. Cockburn (2000); vgl. Ludwig, Lichter (2006), S. 215-217

- Projektgröße gemessen an der Zahl der Mitarbeiter;
- Risiken, die durch Softwarefehler entstehen - hier wird zwischen vier Risikoklassen nach der Kritikalität von Softwarefehlern unterschieden;
- Projektpriorität - so können Projektauftraggeber die Projektpriorität auf Produktivität, Fehlerfreiheit oder Sichtbarkeit des Entwicklungsprozesses legen.

Aus der Kombination der drei Gestaltungsbedingungen ergeben sich mehrere mögliche Projektkonstellationen. In diesem konzeptionellen Rahmen werden jedoch nicht alle davon mit konkreten Gestaltungsmaßnahmen unterlegt. Stattdessen konzentriert sich der Ansatz auf Gestaltungsempfehlungen für die kleineren und unkritischen Projekte, welche v.a. auf das Projektmanagement, die Kommunikation sowie die Prozessverbesserung abzielen. Zu den konkreten vorgeschlagenen Gestaltungsempfehlungen gehören

- die inkrementelle Entwicklung und Produktauslieferung in regelmäßigen Schritten,
- die direkte Beteiligung des Benutzers und
- ein Workshop zu Beginn und in der Mitte von jedem Auslieferungszyklus, um Produkt- und Entwicklungsdetails festzulegen.

Neben der agilen Methode ‚Crystal‘ ist aus Sicht des Situativen Ansatzes der Organisationsgestaltung die Forschung im Umfeld der sog. ‚agile patterns‘ hervorzuheben. ‚Agile patterns‘ sollen der Agilen Softwareentwicklung insgesamt zu einer größeren situativen Ausrichtung verhelfen, wobei die Idee, wiederverwendbare und konfigurierbare Prozessmuster als modulare Hauptelemente eines Prozessrahmenwerks zu nehmen, nicht erst im Zuge der Agilen Softwareentwicklung entstanden ist. Prozessmuster fanden erstmals große Beachtung in Zusammenhang mit dem Objektorientierten Paradigma der Softwareentwicklung.

Ein Muster ist allgemein eine generelle Lösung eines Problems, die zur Ableitung einer konkreten Lösung herangezogen werden kann. Prozessmuster sind in diesem Sinn abstrakte, modulare Prozesselemente, welche sich in einer bestimmten Situation bewährt haben und Referenzcharakter besitzen. Mit einem Rahmenwerk aus konfigurierbaren Prozessmustern bzw. -elementen kann der Softwareentwicklungsprozess wiederkehrend und jeweils situationsspezifisch zusammengestellt werden. Ein solches Prozessmuster-Rahmenwerk wurde mit dem generischen ‚Object-Oriented Software Process‘ für die objektorientierte Softwareentwicklung vorgestellt.²⁰⁰ In Anlehnung daran und nicht zuletzt aufgrund der immer größer werdenden Zahl an (konkurrieren-

²⁰⁰ Vgl. Ambler (1998)

den) agilen Methoden finden Prozessmuster zunehmend Anwendung in der Agilen Softwareentwicklung.²⁰¹

2.2 Theoretischer Zugang über das entscheidungsorientierte Konzept der Organisationsgestaltung

Erkenntnisziele heben auf das Verständnis gegebener Sachverhalte ab. Bei der vorliegenden Arbeit ist die Verständnisgenerierung für koordinationsrelevante Sachverhalte bei der Standardsoftwareentwicklung eine zentrale Vorarbeit für die Konzeption des situativen Koordinationsansatzes. Daher erfolgt zusätzlich zum Situativen Ansatz im vorliegenden Kap. 2.2 ein zweiter theoretischer Zugang zur Problemstellung über das entscheidungsorientierte Konzept der Organisationsgestaltung.

2.2.1 Der entscheidungsprozessorientierte Ansatz der Organisationsgestaltung

Die entscheidungsorientierte Organisationstheorie vermittelt methodische Grundlagen für Entscheidungen über Organisationsstrukturen. Organisationsgestalter sollen damit in die Lage versetzt werden, die Auswahlentscheidung zwischen den verfügbaren Gestaltungsinstrumenten so vorzunehmen, dass die Unternehmungsziele in einem möglichst hohen Maß realisiert werden können.²⁰² Die Ansätze zur entscheidungsorientierten Organisationstheorie lassen sich in zwei Gruppen einteilen. Sie unterscheiden sich in der Grundsatzannahme, ob Organisationsentscheidungen optimierbar sind (entscheidungslogischer Ansatz), oder ob das nicht möglich ist und vielmehr von der sog. begrenzten Rationalität der Entscheider ausgegangen werden muss (entscheidungsprozessorientierter Ansatz):²⁰³

- Die entscheidungslogischen Ansätze versuchen, organisatorische Gestaltungsprobleme mithilfe von mathematischen Algorithmen oder in Form von verbalen Entscheidungsmodellen zu lösen. Es wird angestrebt, unter Anwendung von Methoden des Operations Research Organisationsstrukturen zu optimieren;
- Die Vertreter des entscheidungsprozessorientierten Ansatzes sehen Organisationen als Systeme, in welchen Entscheidungen getroffen und koordiniert werden müssen. Das Entscheidungsverhalten der Organisationsmitglieder wird dabei wesentlich durch die Organisationsstruktur beeinflusst. D.h. Organisationsmitglieder können durch eine zweckmäßige Gestaltung der Organisationsstruktur in die Lage versetzt werden, Entscheidungen rationaler und zielorientierter zu treffen. Dabei wird

²⁰¹ Vgl. bspw. Tasharofi, Ramsin (2007), S. 222ff. oder Bozheva, Gallo (2007), S. 4ff.

²⁰² Vgl. Frese (2005), S. 3

²⁰³ Vgl. Bea, Göbel (2006), S. 126; vgl. Schulte-Zurhausen (2002), S. 22f.

grundsätzlich ein zufriedenstellendes Anspruchsniveau angestrebt. Satisfizierung bedeutet jedoch keinen generellen Verzicht auf eine Optimierung von Organisationsentscheidungen. Auch ein begrenzt rationaler Entscheidungsträger würde die optimale Lösung dem befriedigenden Anspruchsniveau vorziehen, sofern diese ohne einen Zusatzaufwand an Zeit- und Ressourcen ermittelt werden könnte.

Die vorliegende Arbeit ordnet sich in die Gruppe der entscheidungsprozessorientierten Ansätze ein. Es wird hier nicht das Ziel verfolgt, einen Ansatz für die situative Koordination der Standardsoftwareentwicklung zu entwickeln, der auf quantitativen Entscheidungsmodellen oder Methoden des Operations Research basiert. Dafür sind zum gegenwärtigen Zeitpunkt die Prozesse, Entscheidungssituationen und Möglichkeiten zum Messen in der Softwareentwicklung nicht ausreichend verstanden und erforscht. Zudem sind die Prozesse der Softwareentwicklung sehr unbeständig. Grundlegende Prozessschwankungen aufgrund des hohen Anteils menschlicher kognitiver Leistungen müssen als gegeben hingenommen werden.²⁰⁴ Daher werden in dieser Arbeit ganz im Sinn eines entscheidungsprozessorientierten Ansatzes die Entscheidungsprozesse in der Standardsoftwareunternehmung in den Mittelpunkt der Betrachtung gestellt und es wird angenommen, dass diese unter der begrenzten Rationalität der Entscheider stattfinden.

Das Konzept der begrenzten Rationalität von Simon²⁰⁵ bildet den Ausgangspunkt der gesamten verhaltenswissenschaftlichen Entscheidungstheorie. Es besagt, dass Individuen zwar intentional rational handeln, jedoch nur begrenzt in der Lage, Entscheidungen rational zu treffen. Dies liegt daran, dass sie nur begrenzte quantitative und qualitative Kapazitäten haben, die der Bewältigung von Entscheidungsaufgaben Grenzen setzen. Das gilt auch für gut strukturierte Entscheidungsprobleme, wenn die zu berücksichtigende Informationsmenge eine kritische Grenze überschreitet. Die begrenzten quantitativen und qualitativen Kapazitäten der Entscheidungsträger sind im Einzelnen darin begründet, dass Individuen

- nur ein unvollständiges Wissen über die Konsequenzen aus der Wahl der möglichen Entscheidungsalternativen haben,
- nur begrenzt in der Lage sind, zukünftige Ereignisse ex ante zu bewerten oder zu antizipieren, sowie
- aufgrund der begrenzten Kapazität und der Geschwindigkeit der Informationsverarbeitung nicht fähig sind, alle Entscheidungsalternativen in Betracht zu ziehen.²⁰⁶

²⁰⁴ Vgl. Mikusz, Herzwurm (2008)

²⁰⁵ Engl.: bounded rationality; vgl. Simon (1955)

²⁰⁶ Vgl. Kieser, Walgenbach (2003), S. 40f.; vgl. Frese (2005), S. 73f.

Bei den entscheidungsprozessorientierten Ansätzen geht es folglich um die Frage, wie unter den o.g. Umständen die Rationalität der in der Organisation gefällten Entscheidungen gewährleistet werden kann. Als Antwort wird neben anderen Instrumenten v.a. die formale Organisation als ein Mechanismus betrachtet, mit dem die Komplexität und Unsicherheit für den einzelnen Entscheider reduziert und damit die Rationalität seiner Entscheidung sichergestellt werden kann.²⁰⁷ Die formale Organisation wird dabei als ein System von Entscheidungen interpretiert.²⁰⁸

- Objektentscheidungen sind die eigentlichen Entscheidungen wie bspw. die Entscheidung für oder gegen die Verschiebung eines Projektmeilensteines;
- Kommunikationsentscheidungen sind Entscheidungen über die Weitergabe von Informationen - bspw. über die Weitergabe der Information über die Verschiebung eines Projektmeilensteines an die Unternehmensleitung;
- Organisationsentscheidungen sind Entscheidungen über die Wahl einer Steuerungsmaßnahme, um die nachfolgende Objektentscheidung zu beeinflussen - wie bspw. die Entscheidung über den Prozess bei Anforderungsänderungen mit potentieller Auswirkung auf Projektmeilensteine.

Zwar geht es bei allen organisatorischen Gestaltungsmaßnahmen letztendlich darum, Objektentscheidungen zu beeinflussen. Dies geschieht aber nicht nur unmittelbar, sondern auch mittelbar über die Steuerung von Organisations- und Kommunikationsentscheidungen.²⁰⁹ Eine Entscheidung über die Ausgestaltung der Koordination als ein Gebilde von Steuerungsmaßnahmen ist immer eine Organisationsentscheidung. Sie soll sicher stellen, dass die nachfolgenden eigentlichen Objektentscheidungen möglichst rational und nicht weitestgehend begrenzt rational gefällt werden, d.h.

- nicht nur subjektiv aus Sicht des Entscheiders rational (Irrtum), sondern auch objektiv rational;
- nicht nur persönlich rational aber organisational falsch (Eigennutz), sondern rational aus Sicht der Organisation;
- nicht nur lokal rational (Identifikation mit Teilbereichen), sondern in einem größeren Zusammenhang rational;
- nicht nur kurzfristig rational aber langfristig falsch (begrenzte Voraussicht), sondern langfristig rational.²¹⁰

²⁰⁷ Vgl. Kieser, Walgenbach (2003), S. 40f.

²⁰⁸ Vgl. Laux, Liermann (2003), S. 13f., S. 23

²⁰⁹ Vgl. Laux, Liermann (2003), S. 23

²¹⁰ Vgl. Bea, Göbel (2006), S. 135

Die enge Verbindung zwischen der Organisationstheorie und der Entscheidungstheorie ergibt sich beim entscheidungsprozessorientierten Ansatz neben der Prämisse, dass alle organisatorischen Maßnahmen dazu dienen, Entscheidungen zu steuern, aus einem weiteren Grund. Die Auswahl von geeigneten Organisationsmaßnahmen ist selbst Gegenstand von mehr oder weniger komplexen Entscheidungsproblemen.²¹¹ Auch hier wird beim entscheidungsprozessorientierten Ansatz, und damit ebenso in der vorliegenden Arbeit, von der begrenzten Rationalität der Entscheider als Hauptproblem ausgegangen.²¹²

Koordination bezeichnet das Ausrichten von Einzelaktivitäten in einem arbeitsteiligen System auf ein übergeordnetes Gesamtziel.²¹³ Die Notwendigkeit der Koordination ergibt sich durch das grundlegende Dilemma arbeitsteiliger Systeme: Die begrenzte qualitative und quantitative Kapazität von Organisationseinheiten bedingen auf der einen Seite die Aufteilung eines komplexen Gesamtproblems und die bis zu einem gewissen Grad isolierte Entwicklung von Teillösungen. Das Streben nach einer möglichst weitgehenden Lösung des Gesamtproblems erfordert auf der anderen Seite die integrierende Abstimmung aller Teilaktivitäten und Teillösungen. Koordinationsmaßnahmen werden daher dem Entscheidungssystem zugeordnet – Koordination bezweckt die Abstimmung von Entscheidungen.²¹⁴

Koordinationsmaßnahmen sind aufgabenorientierte Maßnahmen, bei welchen das sog. Motivationsproblem ausgeblendet wird. D.h. es wird unterstellt, dass alle Mitarbeiter das übergeordnete Gesamtziel nicht nur verfolgen können, sondern auch verfolgen wollen. Da jedoch der einzelne Mitarbeiter mit seiner Teilnahme an der Organisation auch immer seine eigenen Ziele verfolgt, ist nicht von der automatischen Befolgung der ihm übertragenen Aufgaben auszugehen. Neben dem Koordinationsproblem stellt sich also der Organisation auch ein Motivationsproblem und erst der Einsatz von Koordinations- und Motivationsmaßnahmen bestimmt die Lösung des gesamten Organisationsproblems.²¹⁵ Prägnant ausgedrückt: „Das Koordinationsproblem beschreibt die Überwindung des Nichtwissens der Akteure. Beim Motivationsproblem geht es dagegen zusätzlich um die Überwindung des Nichtwollens.“²¹⁶

²¹¹ Vgl. Laux, Liermann (2003), S. 23

²¹² Begrenzte Rationalität wird in der vorliegenden Arbeit angenommen, aber nicht in Gänze analysiert, da nur das Koordinationsproblem und nicht das Motivationsproblem betrachtet wird. Begrenzte Rationalität tangiert jedoch in vielen Punkten auch das Motivationsproblem, bspw. beim Eigennutz.

²¹³ Vgl. Albach (1967), zitiert nach Frese (2005), S. 143

²¹⁴ Vgl. Frese (2005), S. 143f., vgl. auch Kap. 2.2.2 zu Arbeitsteilung und Spezialisierung

²¹⁵ Vgl. Frese (2005), S. 4f., vgl. Jost (2009), S. 24

²¹⁶ Picot u.a. (2005), S. 7

Der entscheidungsprozessorientierte Ansatz löst das komplexe organisatorische Gestaltungsproblem zwar iterativ, beginnt dabei aber mit der koordinationsbezogenen Teillösung. Die Koordinationsperspektive dominiert die Entwicklung einer Organisationsstruktur und stellt mit den Koordinationsmaßnahmen die übergeordneten organisatorischen Regeln. Motivationsmaßnahmen werden nur flankierend eingesetzt und stellen die untergeordneten Regeln.²¹⁷ Die (heuristische) Trennung des gesamten Organisationsproblems in ein Koordinations- und ein Motivationsproblem bedeutet jedoch nicht, dass es keine Wechselwirkungen zwischen diesen Teilproblemen gibt (vgl. Abb. 18). So kann bspw. ein Mitarbeiter aufgrund seines zu geringen Entscheidungsspielraums (Koordinationsdimension) unzufrieden sein (Motivationsdimension).²¹⁸

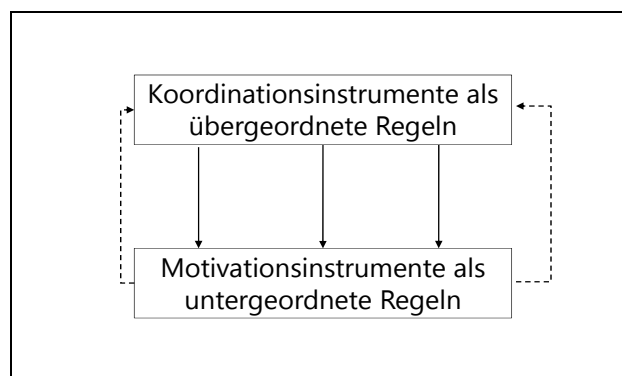


Abb. 18: Koordinationsinstrumente als übergeordnete und Motivationsinstrumente als untergeordnete Regeln²¹⁹

In der vorliegenden Arbeit wird das Koordinationsproblem thematisiert. Motivationsaspekte werden nur dort betrachtet, wo Wechselwirkungen zwischen den beiden Dimensionen zentral für das Ziel der Arbeit sind. Auch in der vorliegenden Arbeit wird also das Motivationsproblem als das untergeordnete Problem angesehen, das im Nachrang an die koordinationsbezogene Teillösung des gesamten Organisationsproblems flankierend, d.h. im Sinne der übergeordneten Koordinationsperspektive, zu lösen ist. Diese Lösung des Motivationsproblems ist allerdings nicht Gegenstand der vorliegenden Arbeit. Denn eine gleichzeitige tiefgehende Thematisierung des Koordinations- und des Motivationsproblems in einer Forschungsarbeit ist mit einem wesentlichen Nachteil aus Sicht der Koordinationsperspektive behaftet. Eine solche angestrebte Lösung des gesamten Organisationsproblems würde nämlich die Berücksichtigung zahlreicher Faktoren erfordern, die sich negativ auf den Zugang zum Koordinationsproblem auswirken:

²¹⁷ Vgl. Frese (2005), S. 8; vgl. Jost (2008), S. 460f.

²¹⁸ Vgl. Jost (2009), S. 26f.

²¹⁹ Jost (2008), S. 460

- Motivationsmaßnahmen sind durch Personenorientierung gekennzeichnet. Um das Motivationsproblem lösen zu können, müssten daher die individuellen Einstellungen und Verhaltensweisen der Mitarbeiter betrachtet und berücksichtigt werden;²²⁰
- Sollen die individuellen Merkmale konkreter Personen in die Organisationsgestaltung einbezogen werden, so ist die Berücksichtigung von Erkenntnissen aus Disziplinen der empirischen Verhaltenswissenschaften notwendig;²²¹
- Die Lösung des Motivationsproblems vollzieht sich vorrangig im Rahmen eines Steuerungskonzeptes, d.h. über die Vorgabe von Ergebniszielen und ihren Vergleich mit den realisierten Ergebniswerten. Daher wäre das Steuerungssystem der Unternehmung in die Lösung mit einzubeziehen;²²²
- Die Lösung des Motivationsproblems besteht in der Gestaltung einer Anreizstruktur. Die Gesamtheit der in einer Organisation eingesetzten Motivationsinstrumente bestimmt die Anreizstruktur der Organisation und wäre ebenso von der vorliegenden Arbeit zu erbringen.²²³

2.2.2 Entwicklung von Standardsoftware aus Sicht des entscheidungsprozessorientierten Ansatzes

2.2.2.1 Spezialisierung und strukturelle Eingliederung von Entwicklungsprojekten in die Aufbauorganisation einer Standardsoftwareunternehmung

Arbeitsteilung und Spezialisierung resultieren aus der sog. Aufgabenanalyse, d.h. der Zerlegung einer Aufgabe in abgegrenzte Teilaufgaben. Die Aufgabenanalyse kann nach verschiedenen Gesichtspunkten erfolgen, was zu der jeweiligen Art der Spezialisierung führt. Das Ergebnis der Zerlegung wird auch als ‚work breakdown structure‘ (WBS) bezeichnet. In der Softwareentwicklung im Allgemeinen ist die Zerlegung nach der Verrichtung und nach dem Objekt von besonderer Bedeutung.²²⁴

- Die verrichtungs- bzw. prozessorientierte Zerlegung teilt die Aufgabe in Teilaufgaben unterschiedlicher Art. Die Zerlegung der Softwareentwicklung in die Teilprozesse Analyse, Entwurf, Implementierung, Qualitätssicherung, etc. ist eine verrichtungsorientierte Zerlegung. D.h. alle Teilaufgaben werden von jeweils einer organisatorischen Einheit erbracht, wodurch die Spezialisierung auf Verrichtungen entsteht.

²²⁰ Vgl. Frese (2005), S. 4f.

²²¹ Vgl. Frese (2005), S. 153

²²² Vgl. Frese (2005), S. 5f.; Zur Unterscheidung zwischen dem Kompetenz- und dem Steuerungssystem als zwei Gestaltungsebenen vgl. Frese (2005), S. 5, 167ff.

²²³ Vgl. Jost (2009), S. 24

²²⁴ Vgl. Mellis (2004), S. 80f.; vgl. Kieser, Walgenbach (2003), S. 87

- Werden hingegen alle Tätigkeiten, die sich auf einzelne Produktteile oder Produkte beziehen, jeweils zu einer organisatorischen Einheit zusammengefasst, so liegt eine Spezialisierung auf Objekte vor. Die objektorientierte Zerlegung teilt also die Aufgabe nach ihren Gegenstandsmerkmalen in Teilaufgaben auf.

Bei beiden Arten der Spezialisierung handelt es sich um eine horizontale Arbeitsteilung, während Spezialisierungsformen nach dem Rang der vertikalen Arbeitsteilung zugeordnet werden.²²⁵ In einem wesentlichen Punkt unterscheiden sich aber die beiden Spezialisierungsarten: Die Spezialisierung auf Objekte ist vorrangig eine Mengenteilung. Denn hier werden verschiedenen Organisationseinheiten von der Verrichtung her gleichartige (Teil-)Aufgaben übertragen, die an unterschiedlichen Objekten durchgeführt werden. D.h. die Menge der Objekte ist zu groß und muss auf mehrere Organisationseinheiten aufgeteilt werden. Spezialisierungsvorteile entstehen hier nur, wenn sich die Objekte wesentlich voneinander unterscheiden und dies auch Auswirkungen auf die Verrichtung von Aufgaben hat. Bei der Spezialisierung auf Verrichtungen werden dagegen unterschiedlichen Organisationseinheiten von der Funktion her unterschiedliche Aufgaben zugeordnet, womit in jedem Fall eine Spezialisierung dieser Einheiten einhergeht.²²⁶

In der Praxis werden bei der Zerlegung der Gesamtaufgabe Softwareentwicklung in Teilaufgaben häufig die Spezialisierung auf Verrichtungen und die Spezialisierung auf Objekte kombiniert.²²⁷ Auf der obersten Ebene dominiert dabei üblicherweise die organisatorische Objektorientierung als Spezialisierung nach Produkt-, Leistungs-, oder Kundengruppen und/oder nach räumlichen Kriterien. Dabei wird eine hohe Selbständigkeit der organisatorischen Einheiten angestrebt.²²⁸

Bei Standardsoftwareunternehmen, die verschiedene Standardprodukte entwickeln, ist von einer solchen nach Produktbereichen oder Produktlinien gegliederten Organisationsstruktur auszugehen. Denn die Entwicklung von Standardsoftware stellt eine längerfristige und kontinuierliche Daueraufgabe dar. Der Produktlebenszyklus beträgt in der Regel mehrere Jahre. Während dieser Zeit wird das Produkt kontinuierlich weiterentwickelt und in einer Reihe von sog. Releases²²⁹ dem Markt zur Verfügung gestellt.²³⁰ Die (Weiter-)Entwicklung der Standardsoftware erfolgt dabei innerhalb oder

²²⁵ Vgl. Bea, Göbel (2006), S. 301

²²⁶ Vgl. Alewell (2004), Sp. 39

²²⁷ Vgl. Mellis (2004), S. 81

²²⁸ Vgl. Stelzer (1998), S. 57; vgl. Gerhardt (1992)

²²⁹ Ein Release ist eine neue, mehr oder weniger stark veränderte Version des Softwareproduktes. Der Umfang der Veränderung gegenüber der Vorversion kann von kleinen Korrekturreleases bis zu umfangreichen Funktionserweiterungen reichen.

²³⁰ Vgl. Mellis (2004), S. 171

im Zusammenspiel von mehreren Produktbereichen als ein Projekt. Das führt zu der weiten Verbreitung von mehrdimensionalen Organisationsformen bei Standardsoftwareunternehmen und dabei insbesondere zur Matrixorganisation in Verbindung mit Projektstrukturen (Matrix-Projektorganisation).²³¹

Der Begriff der Matrix-Projektorganisation bezeichnet eine von drei idealtypischen Arten der strukturellen Eingliederung von Projekten in die (dauerhafte) Organisationsstruktur. Sie unterscheiden sich v.a. in der Ausstattung des Projektleiters mit Kompetenzen, werden jedoch meistens in Abwandlungen oder Mischformen realisiert.²³² Die Matrix-Projektorganisation liegt dabei zwischen den beiden Extremformen, der reinen Projektorganisation und der Einfluss-Projektorganisation.

Bei der Einfluss-Projektorganisation erfolgt die Projektarbeit in den verschiedenen am Projekt beteiligten Bereichen und wird durch einen Projektleiter, der häufig in Stabsfunktion agiert, koordiniert und gelenkt. Damit erhält der Projektleiter keine eigenen Entscheidungs- und Weisungsbefugnisse, sondern muss projektbezogene Entscheidungen übergeordneten Instanzen, denen er gegenüber in der Regel berichtspflichtig ist, zur Entscheidung vorlegen. Desweiteren unterliegen die dem Projektleiter für die Projektaufgabe zugeordneten Mitarbeiter auch während der Projektlaufzeit den bereits vor dem Projekt bestehenden Weisungsbefugnissen. Insgesamt sind die Ressourcenautonomie und Verselbstständigung des Projektes schwach ausgeprägt. Die Aufgabe des Projektleiters beschränkt sich auf die Ausarbeitung und Abstimmung von Plänen, Überwachung und Dokumentation des Projektfortschritts und ggf. Entwicklung von Korrekturvorschlägen. Die dazu notwendigen Informationen beschafft sich der Projektleiter auf der Basis umfassender Informations- und Beratungsrechte. Er ist jedoch darauf angewiesen, dass seine Vorschläge von den übergeordneten Instanzen akzeptiert und in entsprechende Entscheidungen und Weisungen umgesetzt werden.²³³

Die Einfluss-Projektorganisation bedient sich der bestehenden Organisation, sodass zur Durchführung eines Projektes nur geringe organisatorische Veränderungen notwendig sind. Diesem Vorteil stehen jedoch gewichtige Nachteile gegenüber: Die betroffenen organisatorischen Einheiten werden zusätzlich zu ihren bestehenden Aufgaben mit Entscheidungs- und Weisungsaufgaben für das Projekt belastet.²³⁴

Das gegenteilige Extrem zur Einfluss-Projektorganisation ist die reine Projektorganisation als die zeitlich befristete Schaffung eigener Organisationseinheiten ausschließlich

²³¹ Vgl. Stelzer (1998), S. 57; vgl. Gerhardt (1992)

²³² Vgl. Marr, Steiner (2004), Sp. 1205; vgl. Mellis (2004), S. 167

²³³ Vgl. Marr, Steiner (2004), Sp. 1205; vgl. Mellis (2004), S. 167f.

²³⁴ Vgl. Mellis (2004), S. 168

für die Projektbearbeitung. Die Projektorganisation verfügt dabei über eigene personelle und sachliche Ressourcen und kann ggf. weiter horizontal und vertikal untergliedert werden. Der Projektleiter trägt die Gesamtverantwortung und erhält für die Dauer des Projektes die volle Entscheidungs- und Weisungsbefugnis wie ein Linienvorgesetzter – d.h. alle Projektmitarbeiter werden ihm für die Dauer des Projektes auch disziplinarisch unterstellt. Daher können die Projektziele ungehindert von anderen laufenden Aufgaben erreicht werden. Auf der anderen Seite werden aber dadurch Ressourcen oft nicht effizient eingesetzt – eben um Ressourcenabhängigkeiten innerhalb der ständigen Organisation zu vermeiden. So entsteht ineffizienter Ressourceneinsatz dadurch, dass einzelne Teilaufgaben nicht an vorhandene Spezialisten in der ständigen Organisation zugewiesen werden, sondern im Projekt Spezialisten eingesetzt werden, die dort aber nicht vollständig ausgelastet sind.²³⁵

Die Matrix-Projektorganisation verbindet den Vorteil der reinen Projektorganisation, organisatorisch unabhängig die Projektziele verfolgen zu können, mit dem Vorteil des effizienten Ressourceneinsatzes bei der Einfluss-Projektorganisation. Nicht zuletzt deswegen ist es, wie bereits angesprochen, die dominierende Form der Eingliederung von Entwicklungsvorhaben in die Aufbauorganisation einer Standardsoftwareunternehmung. Der Projektleiter verfügt hier, wie bei der reinen Projektorganisation, über projektbezogene Kompetenzen und die Gesamtverantwortung für das Projekt, sowie insbesondere über die Vorgehensverantwortung incl. Projektplanung, -überwachung und -steuerung. Zur Vermeidung der Gefahr eines ineffizienten Ressourceneinsatzes als Nachteil der reinen Projektorganisation werden ihm aber nur die Ressourcen fest zugeordnet, die während der Projektdauer vollständig ausgelastet werden. In der Standardsoftwareentwicklung sind es v.a. die Softwareentwickler. Für Projektaufgaben, die nur vorübergehend anfallen oder einen entsprechenden Spezialisten nicht vollständig auslasten, muss der Projektleiter auf Spezialisten aus den Fachabteilungen der ständigen Organisation zurückgreifen. Diese Fachabteilungen, oftmals als sog. Kompetenzzentren organisiert, existieren hauptsächlich zur Unterstützung von Projekten.

Aus Projektsicht entsteht durch die temporäre Hinzuziehung von Mitarbeitern aus den Fachabteilungen eine laufende Veränderung des Projektteams, oft einhergehend mit neuen Projektphasen oder Arbeitspaketen. Daher werden moderne Ausgestaltungen einer Matrix-Projektorganisation auch als virtuelle Projektorganisation oder Pool-Projektorganisation bezeichnet.²³⁶ Funktionsbezogene Anweisungen und disziplinarische Weisungsbefugnisse verbleiben dabei gänzlich oder zumindest zum Teil bei den

²³⁵ Vgl. Marr, Steiner (2004), Sp. 1205; vgl. Mellis (2004), S. 168f.

²³⁶ Vgl. Jenny (1995), S. 112; vgl. Patzak, Rattay (2004), S. 120f.

Linienvorgesetzten der Fachabteilungen, aus welchen die Projektmitarbeiter jeweils stammen. Durch diese Überschneidung der projekt- und fachbereichsbezogenen Kompetenzen zwischen dem Projektleiter und den Linienvorgesetzten entsteht eine zweidimensionale Matrix. Die meisten Projektmitarbeiter sind nun zwei Führungskräften zugeordnet. Durch den Zwang zum Konsens entstehen zwar ein inhärentes Potential für Kompetenzkonflikte und ein recht hoher Kommunikationsaufwand. Auf der anderen Seite soll aber insgesamt eine bessere Lösung entstehen, weil projektbezogene und fachlich-funktionale Faktoren gleichberechtigt in die Entscheidungen einfließen. Insgesamt gilt die Matrix-Projektorganisation als die ideale Organisationsform für projektorientierte Unternehmen.²³⁷

Die objekt- und die verrichtungsorientierte Spezialisierung können bezogen auf ein Vorhaben oder übergreifend für mehrere Projektvorhaben erfolgen. Eine übergreifende Spezialisierung liegt bspw. vor, wenn eine organisatorische Einheit in verschiedenen Projekten immer wieder die Entwicklung der Benutzeroberfläche übernimmt oder wenn eine Person in allen Projekten, an welchen sie beteiligt ist, die Qualitätssicherung übernimmt.²³⁸ Bei einer Matrix-Projektorganisation entsteht gewissermaßen automatisch eine dauerhafte Spezialisierung. Abb. 19 stellt den für die Eingliederung von Entwicklungsprojekten relevanten Ausschnitt der Aufbauorganisation einer Standardsoftwareunternehmung dar.²³⁹

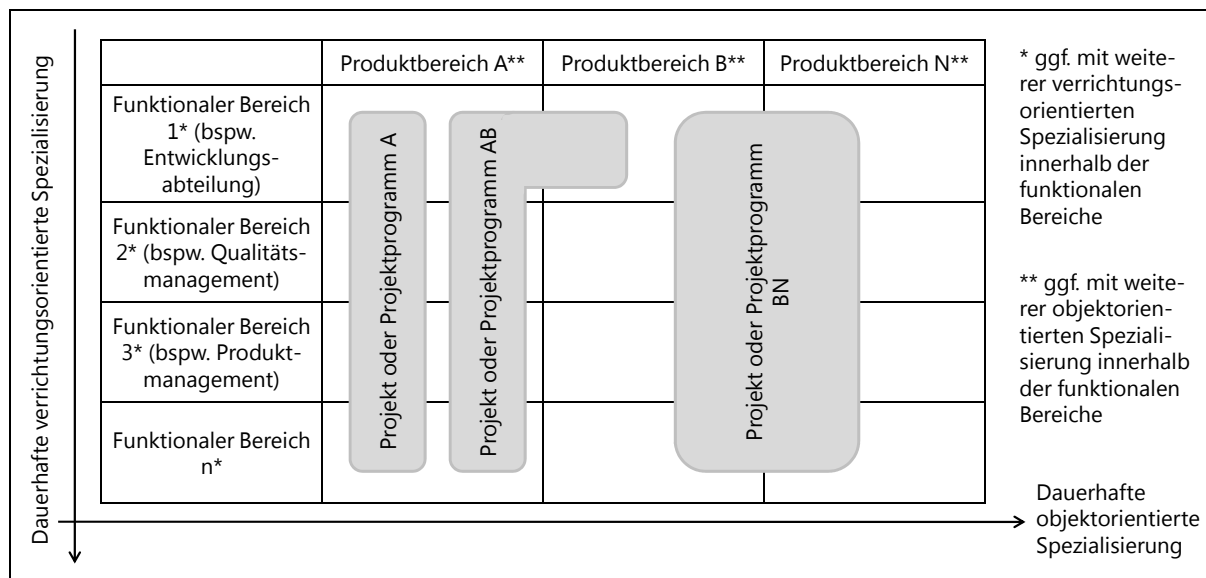


Abb. 19: Eingliederung von Entwicklungsprojekten in die Aufbauorganisation einer Standardsoftwareunternehmung

²³⁷ Vgl. Marr, Steiner (2004), Sp. 1205; vgl. Mellis (2004), S. 169; vgl. Henrich (2002), S. 128, 130; vgl. Kerzner (2003), S. 87, 91f.

²³⁸ Vgl. Mellis (2004), S. 114

²³⁹ In der Praxis ähnlich bspw. bei Microsoft; vgl. Cusumano, Selby (1996), S. 48

2.2.2.2 Arbeitsteilung und Spezialisierung innerhalb von Entwicklungsprojekten

Die Kombination der verrichtungs- und objektorientierten Spezialisierung überträgt sich über die Matrix-Projektorganisation auch auf die Entwicklungsprojekte selber (dauerhafte Spezialisierung). Eine verrichtungsorientierte Spezialisierung auf die Entwicklungsaufgaben Analyse, Entwurf, Implementierung, etc. und Unterstützungsaufgaben wie bspw. Qualitätsmanagement ist nämlich nicht ausreichend, um hinreichend kleine Aufgaben für ein Projekt abzugrenzen. Daher werden insbesondere die Implementierung und u.U. auch der Entwurf einer Standardsoftware objektorientiert in die Teilaufgaben (Entwurf und) Implementierung der verschiedenen Komponenten oder Features zerlegt (objektorientierte Spezialisierung). Es entstehen so mehrere Projekte, die zu einem Entwicklungsvorhaben beitragen. Sie sind alle jeweils verrichtungsorientiert spezialisiert, durchlaufen prinzipiell dieselben Projektphasen und müssen dieselben Entwicklungs- und Unterstützungsaufgaben erbringen - jedoch um jeweils eine andere Komponente bzw. ein anderes Feature zu entwickeln.²⁴⁰

Die Übertragung der verrichtungsorientierten Spezialisierung von der primären Organisation auf die Projektorganisation der Entwicklungsvorhaben erfolgt über das Konzept der Rolle. Die objektorientierte Spezialisierung bzw. die Mengenteilung wird über die Bildung von mehreren Teilprojekten innerhalb eines Entwicklungsvorhabens in Verbindung mit der sog. Inkrementellen Softwareentwicklung übertragen. Diese drei Konzepte sind daher zentral in Zusammenhang mit der Arbeitsteilung und Spezialisierung in der Projektorganisation von Entwicklungsvorhaben und werden daher im Folgenden in ihrem Zusammenwirken erläutert.

Eine Rolle beschreibt eine bestimmte Menge von Aufgaben und Verantwortlichkeiten im Rahmen eines Projektes. Durch die Festlegung von Rollen wird die Unabhängigkeit der verrichtungsorientierten Spezialisierung von organisatorischen und projektspezifischen Rahmenbedingungen erreicht. Die Zuordnung von Personen zu den Rollen erfolgt zu Beginn eines Projektes. Dabei kann eine Person mehrere Rollen besetzen, aber auch umgekehrt kann eine Rolle durch mehrere Personen besetzt werden.²⁴¹ Das Konzept der Rolle ist im V-Modell XT neben konkreten Vorgehensweisen und den zugehörigen (Zwischen-)Ergebnissen eine von drei Abstraktionsebenen für die Vorgaben. Zusätzlich trifft das V-Modell XT über das Konzept der Rolle beschränkt Aussagen über die Organisation des Projektumfelds. V.a. über die zum Vorgehensbaustein Projektmanagement zugeordneten Rollen zeigt sich das Zusammenspiel zwischen dem Projekt und dem Projektumfeld in der Organisation der Softwareunternehmung.²⁴²

²⁴⁰ Vgl. ähnlich bei Mellis (2004), S. 119

²⁴¹ Vgl. Bund (Hrsg., 2008), Teil 8, S. 36

²⁴² Vgl. Friedrich u.a. (2008), S. 47, 52ff.

Das V-Modell XT definiert insgesamt 31 verschiedene Projektrollen.²⁴³ Eine empirische Untersuchung von Jones bestätigt, dass eine derart stark ausgeprägte Spezialisierung auch tatsächlich in der Realität vorkommt. Der Untersuchung liegen Daten aus etwa 12.000 Softwareentwicklungsprojekten zugrunde. Die Daten werden nach Projektgröße und Projekttyp klassifiziert. Als Maßstab für die Projektgröße dient das abstrakte Maß ‚Function Points‘. Es ist das zentrale Konstrukt der Function Point-Methode zur Aufwandschätzung bei Softwareentwicklungsprojekten.²⁴⁴ Jones klassifiziert Projekte simpel in die Größenklassen 1, 10, 100, 1.000, 10.000 und 100.000 Function Points. Umfangreiche Entwicklungsprojekte im Bereich Standardsoftware zählen dabei zu den beiden Größenklassen mit 10.000 bis 100.000 Function Points. Das entspricht nach gängigen Umrechnungsheuristiken von Function Points in reale Aufwandsgrößen in etwa Projekten²⁴⁵ mit einem Umfang von 66-660 Projektbeteiligten.²⁴⁶ Als Projekttypen fungieren ‚End User-, Management Information System-, Outsourced, Commercial-, Systems-, Military‘-Entwicklung, wobei nach Jones´ Definition der einzelnen Projekttypen Entwicklungsprojekte im Bereich Standardsoftware zur ‚Commercial‘-Entwicklung gezählt werden. Bezieht man sich hier wiederum auf die Projektgröße 10.000 bis 100.000 Function Points, so kommt die Untersuchung zu dem Ergebnis, dass eine sehr ausgeprägte Spezialisierung in derartigen Projekten vorherrscht. Jones ermittelt hier die Beteiligung von 21 bis 29 notwendigen Rollen.²⁴⁷

Entwicklungsvorhaben mit über 10.000 Function Points Gesamtaufwand erfordern eine weitere Unterteilung innerhalb der Projektorganisation in Teilprojekte. Projekte, an welchen zwischen 60 und 250 Mitarbeiter beteiligt sind, werden bei Oestereich und Weiss als Megaprojekte bezeichnet. Ihre Größe, gemessen an der hohen Personenzahl, führt organisatorisch zwangsläufig zu einer Untergliederung in Teilprojekte, Teams, etc., die ihrerseits auch über eine eigene vollständige Projektorganisation verfügen. Neben dieser objektorientierten Spezialisierung als Mengenteilung haben spezielle Rollen und ihre Aufgaben als verrichtungsorientierte Spezialisierung weiteren Einfluss auf die Projektorganisation. Ebenso ergeben sich aus der Auftraggeber-Auftragnehmer-Situation und der Organisationsstruktur im Projektumfeld (v.a. Matrix-Projektorganisation) Anforderungen und Einflüsse auf die interne und externe Projektorganisation.²⁴⁸

²⁴³ Vgl. Bund (Hrsg., 2008), Teil 4, S. 3ff.

²⁴⁴ ‚Function Points‘ ist ein Maß zur Bestimmung des fachlich-funktionalen Umfangs einer Software.

²⁴⁵ Bei einer derartigen Projektgröße muss hier korrekter von sog. Projektprogrammen gesprochen werden; siehe hierzu auch im Folgenden.

²⁴⁶ Vgl. Jones (1996)

²⁴⁷ Vgl. Jones (2003), S. 22ff.

²⁴⁸ Vgl. Oestereich, Weiss (2007), S. 34f.

Die einzelnen Teilprojekte eines Entwicklungsvorhabens erbringen v.a. die notwendigen Features und Komponenten, aus welchen sich anschließend das Gesamtprodukt Standardsoftware zusammensetzt:

- Komponenten-Teilprojekte übernehmen die Entwicklung von Subsystemen, die sich im Wesentlichen aus den verschiedenen Schichten einer Softwarearchitektur ergeben. Die Struktur eines Entwicklungsvorhabens, das ausschließlich aus Komponententeams bestehen würde, wäre somit kongruent zur Softwarearchitektur.²⁴⁹ In einem solchen Fall würden bspw. Teilprojekte A und B die Benutzeroberflächenschicht verantworten, Teilprojekte C und D die Geschäftsschicht entwickeln und Teilprojekte E und F die Infrastrukturschicht erstellen.²⁵⁰
- Feature-Teilprojekte sind nicht nach der Softwarearchitektur, sondern nach den Endanwender-Anforderungen an die Software organisiert. Jedes Teilprojekt setzt Endanwenderanforderungen um und ist somit auf die Wertschöpfung aus Kundensicht fokussiert. Feature-Teilprojekte sind immer dann sinnvoll, wenn eine zusammengehörende Menge von zu entwickelnden Funktionen aufwendig und zeitintensiv ist und zudem spezielle fachliche Qualifikationen erfordert.²⁵¹

Kein Komponententeam kann alleine Endanwenderanforderungen in ein Endprodukt umsetzen. Hierzu ist die Integration der Arbeitsergebnisse von beteiligten Feature-Teilprojekten notwendig – v.a. in der Geschäftsschicht.²⁵² Neben Komponenten- und Feature-Teilprojekten bestehen Entwicklungsvorhaben aus zahlreichen weiteren organisatorischen Einheiten (weitere Teilprojekte, Stabsstellen, Lenkungskreise etc.). Sie nehmen allesamt Querschnitts-, Koordinations- oder Auftraggeber-Funktionen wahr. Herausragende Bedeutung hat dabei für die vorliegende Arbeit das Projektprogramm-Management als eine solche zusätzliche organisatorische Einheit mit Koordinationsfunktion bei großen Entwicklungsprojekten.

Die aus Sicht der Organisationslehre korrekte Bezeichnung für den umgangssprachlichen Begriff ‚Megaprojekt‘ ist Projektprogramm. In einem Projektprogramm wird eine Gruppe von Projekten zusammen koordiniert. So sollen Potentiale erschlossen werden, die durch eine separate Steuerung auf Projektebene nicht zugänglich wären.²⁵³ Die Koordination erfolgt hier auf der Projektprogramm-Ebene. Mithilfe eines systematischen Projektprogramm-Managements soll die Vielzahl an inhaltlichen und zeitlichen

²⁴⁹ Vgl. hierzu im Detail nachfolgendes Kap. 2.2.2.3

²⁵⁰ Vgl. Pichler (2008a), S. 136f.

²⁵¹ Vgl. Pichler (2008a), S. 138f.; vgl. Oestereich, Weiss (2007), S. 37

²⁵² Vgl. Pichler (2008a), S. 140f.

²⁵³ Vgl. Project Management Institute (Hrsg., 2006), S. 4f.

Abhängigkeiten in Projektprogrammen sichtbar gemacht und effizient gesteuert werden.²⁵⁴

Derartige Abhängigkeiten lassen sich nach Laux auf vier mögliche Verbundbeziehungen zurückführen: Restriktionsverbund, Erfolgsverbund, Risikoverbund und Bewertungsverbund.²⁵⁵ Zwar greifen Projekte innerhalb eines Projektprogrammes auf gemeinsame Ressourcen zu, bspw. über diverse Querschnittsfunktionen, Stabsstellen, Ressourcen aus den Fachabteilungen bei der Matrix-Projektorganisation etc., und stehen damit in einem Restriktionsverbund. Doch aus Sicht der Problemstellung dieser Arbeit ist das Projektprogramm-Management v.a. in Zusammenhang mit dem Erfolgsverbund der Projekte innerhalb eines Projektprogramms zu sehen.

Ein Erfolgsverbund (auch Ergebnisverbund) zwischen zwei Projekten als Entscheidungsbereiche liegt vor, wenn zumindest für eines der beiden Projekte gilt: Wie weit bei der Durchführung bestimmter Aktionen im Projekt der Gesamterfolg des Projektprogramms steigt oder fällt, hängt davon ab, welche Maßnahmen im anderen Projekt realisiert werden.²⁵⁶ Der Gesamterfolg des Projektprogramms setzt sich also nicht additiv aus den Erfolgen der Einzelmaßnahmen zusammen, sondern wird von der Gesamtheit der Aktionen in den beiden Projekten bestimmt. Damit kann auch ein einzelner Bereich isoliert für sich nicht beurteilen, welche seiner Maßnahmen aus der übergeordneten Projektprogramm-Sicht erfolgreich sind. Die Aktionen in den einzelnen Projekten müssen aufeinander abgestimmt werden.²⁵⁷

Diese Abstimmung der Projekte in einem Projektprogramm aufeinander ist die Aufgabe des Projektprogramm-Managements. Das Project Management Institute definiert drei erfolgskritische Aufgabenbereiche des Projektprogramm-Managements: ‚Benefits Management‘, ‚Program Stakeholder Management‘ und ‚Program Governance‘. Letzteres bezeichnet den Prozess der Entwicklung, Implementierung, Kommunikation und Überwachung von Grundsätzen, organisatorischen Strukturen und Verfahrensweisen, die in einem Projektprogramm Geltung haben sollen. ‚Program Governance‘ resultiert in einem übergeordneten Rahmen, der eine effektive und effiziente Entscheidungsfindung in Projektprogrammen sicherstellen soll.²⁵⁸

²⁵⁴ Vgl. Patzak, Rattay (2004), S. 404

²⁵⁵ Vgl. Laux, Liermann (2003), S. 191ff.

²⁵⁶ Bei Laux, Liermann (2003) bez. Entscheidungsbereichen im Allgemeinen und dem Gesamterfolg der Organisation.

²⁵⁷ Vgl. Laux, Liermann (2003), S. 192

²⁵⁸ Vgl. Project Management Institute (Hrsg., 2006), S. 9f., 12

Abb. 20 verdeutlicht zusammenfassend die objekt- und verrichtungsorientierte Spezialisierung durch Rollen- und Teilprojektbildung in der Projektorganisation von Standardsoftware-Entwicklungsvorhaben.

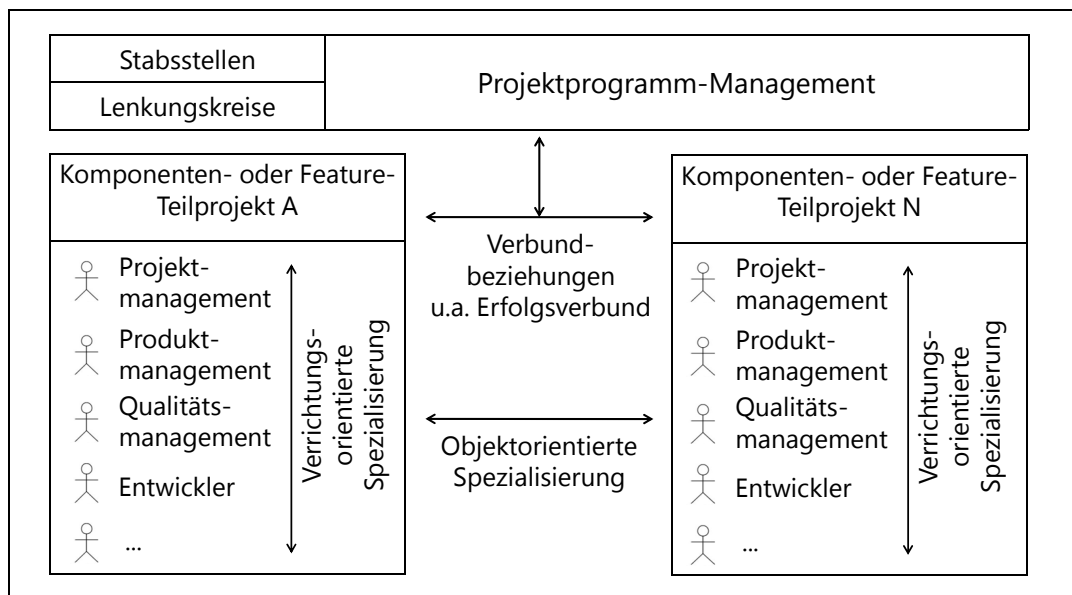


Abb. 20: Objekt- und verrichtungsorientierte Spezialisierung durch Rollen- und Teilprojektbildung in der Projektorganisation von Standardsoftware-Entwicklungsvorhaben

Der objektorientierten Spezialisierung als Mengenteilung eines Projektprogramms in mehrere Komponenten- und Feature-Teilprojekte sind Grenzen gesetzt. Denn diese Aufteilung ist nichts anderes als eine Parallelisierung der prozessorientierten Ablauforganisation. Zum einen kann Parallelisierung aber nur insoweit stattfinden, als dass die notwendigen Ressourcen vorhanden sind. Bspw. könnten in einer Organisation nur Ressourcen für maximal 10 parallel laufende Feature-Projekte vorhanden sein. Zum anderen bestehen zahlreiche Interdependenzen zwischen den Komponenten und Features. So könnte die Implementierung von einem Feature nicht nur die Kenntnis über den Entwurf einer bestimmten Basisfunktion, sondern auch deren erfolgte Implementierung erfordern. Und nicht zuletzt führt die zunehmende Parallelisierung der prozessorientierten Ablauforganisation zu überproportional höheren Koordinationskosten.

Im Beispiel der folgenden Abb. 21 findet nach der Grobanalyse und dem Grobentwurf des zu entwickelnden Gesamtsystems die Parallelisierung des weiteren Entwicklungsverlaufes statt.²⁵⁹ Die Komponenten ‚Datenhaltung‘, ‚Basisfunktionen‘ und ‚Benutzerschnittstelle‘ sowie die Features ‚Funktionen F_1, \dots, F_n ‘ werden nicht sequenziell entwickelt, sondern parallel von mehreren gebildeten Feature- und Komponenten-Projekten.

²⁵⁹ Die objektorientierte Spezialisierung der Entwicklungsaufgabe kann erst konkretisiert werden, wenn eine hinreichend detaillierte Architektur vorliegt. Diese ist in der Regel das Ergebnis der beiden (verrichtungsorientiert abgegrenzten) Teilaufgaben Grobanalyse und Grobentwurf (auch als Architekturentwurf bezeichnet); vgl. Mellis (2004), S. 81

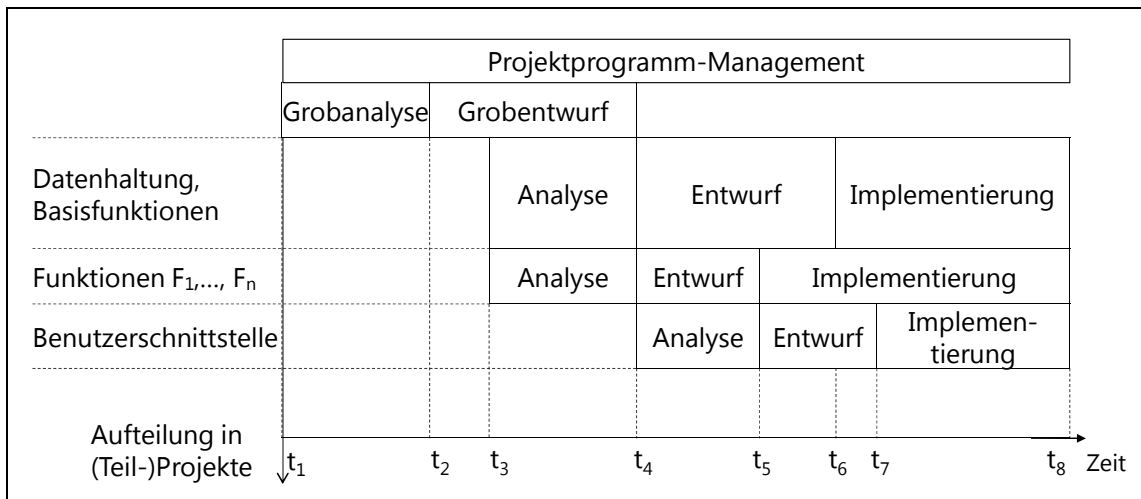


Abb. 21: Objektorientierte Spezialisierung als Parallelisierung des Entwicklungsverlaufes²⁶⁰

Über die Aufteilung eines Projektprogramms in parallel laufende Komponenten- und Feature-Projekte hinaus muss daher bei der Entwicklung von Standardsoftware eine weitere Mengenteilung stattfinden. Diese erfolgt über eine inkrementelle Softwareentwicklung, d.h. die Software wird nicht in einem Zug, sondern in einer Reihe von aufeinander aufbauenden, jeweils funktional erweiterten Ausbaustufen (Inkrementen), entwickelt.²⁶¹ Die erste Stufe ist das Kernsystem. Jedes weitere Inkrement wird sequentiell und jeweils innerhalb einer vollständigen Projektprogrammorganisation erstellt und erweitert das vorhandene Softwaresystem v.a. um weitere Features. In der Praxis wird Software nahezu ausschließlich inkrementell entwickelt.²⁶² Dabei haben sich im Laufe der Zeit zahlreiche Verfeinerungen des Ansatzes herausgebildet:

- Da die Reihenfolge der Inkremente bzw. der zu entwickelnden Features Spielraum zulässt und nur zum Teil, bspw. durch Interdependenzen, vorgegeben ist, wird diese so gewählt, dass schnell Teilergebnisse zur Verfügung gestellt werden können.²⁶³ In diesem Zusammenhang wird die inkrementelle Softwareentwicklung mit der risikobasierten Softwareentwicklung kombiniert: Die mit einem hohen Risiko behafteten Komponenten und/oder hoch priorisierten Features werden vorgezogen und den ersten Inkrementen zugeordnet. Auf diese Weise wird das Projektfehlschlagrisiko erheblich gemindert. Denn im Fall eines Projektverzuges ist so sichergestellt, dass die hoch priorisierten Features und Komponenten bereits implementiert sind und lediglich die Implementierung niedrig priorisierter Features gegen eine mögliche Verschiebung des Fertigstellungstermins abgewogen werden muss.²⁶⁴

²⁶⁰ Vgl. Mellis (2004), S. 94

²⁶¹ Vgl. Ludewig, Lichter (2006), S. 164

²⁶² Vgl. Mellis (2004), S. 86

²⁶³ Vgl. Mellis (2004), S. 86

²⁶⁴ Inkrementell-zeitorientierte Softwareentwicklung bzw. sog. ‚Design-to-schedule-model‘; vgl. McConnell (1996), S. 149f.

- Mit der Bereitstellung von nachfolgenden Inkrementen wird in aller Regel auch eine Verbesserung der bisherigen Teilprodukte verbunden. Z.B. werden bei der Entwicklung nach dem ‚Synch-and-Stabilize-Model‘ von Microsoft Standardsoftware-Komponenten als Inkremente in vielen kleinen Schritten entwickelt und begleitend zur Entwicklung sehr häufig zu einem lauffähigen Softwaresystem integriert. Dabei wird nach jeder Integration das verbesserte Gesamtverhalten des integrierten Softwaresystems getestet.²⁶⁵
- In Zusammenhang mit dem vorstehenden Punkt steht die Verbindung der inkrementellen Softwareentwicklung mit der iterativen Softwareentwicklung. Bei letzterer wird Software in mehreren geplanten und kontrolliert durchgeführten Iterationsschritten entwickelt. Beginnend mit der zweiten Iteration wird dabei bei jedem Iterationsschritt das vorhandene Softwaresystem auf Basis der erkannten Mängel korrigiert und verbessert, jedoch in Abgrenzung zur inkrementellen Softwareentwicklung nicht zwingend erweitert. Bei jedem Iterationsschritt werden die charakteristischen Phasen der Softwareentwicklung (Analyse, Entwurf, Implementierung, Test etc.) durchgeführt.²⁶⁶ Bei der Kombination der beiden Ansätze werden die Inkremente iterativ entwickelt.²⁶⁷
- Im Treppenmodell wird die inkrementelle Softwareentwicklung teilweise wieder parallelisiert. Die überlappende Entwicklung der Inkremente ist v.a. bei großen Entwicklungsvorgaben ohne Weiteres möglich, da hier die Phasen der Softwareentwicklung durch verschiedene organisatorische Einheiten durchgeführt werden. Somit entstehen durch die Überlappungen keine Engpässe bei den Projektressourcen (vgl. Abb. 22).²⁶⁸

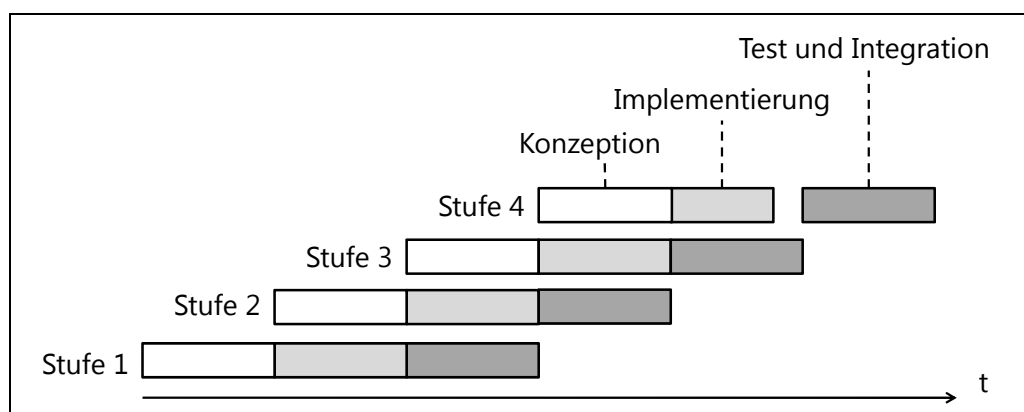


Abb. 22: Das Treppenmodell der inkrementellen Softwareentwicklung²⁶⁹

²⁶⁵ Vgl. Mellis (2004), S. 137f.; vgl. Cusumano, Selby (1996)

²⁶⁶ Vgl. Ludewig, Lichter (2006), S. 163

²⁶⁷ Vgl. Cockburn (2008), S. 28f.

²⁶⁸ Vgl. Ludewig, Lichter (2006), S. 166f.

²⁶⁹ Ludewig, Lichter (2006), S. 167

2.2.2.3 Koordination der Softwareentwicklung durch Entwurf der Softwarearchitektur

Der Entwurf einer Softwarearchitektur in der Phase Grobkonzept beschreibt die Strukturen und Schnittstellen der einzelnen Softwarekomponenten und macht so Vorgaben für ihre Entwicklung durch die unterschiedlichen Organisationseinheiten bzw. Projekte innerhalb eines Projektprogramms. Eine wesentliche Funktion der Softwarearchitektur, die im Folgenden im Mittelpunkt stehen wird, besteht in der Koordination der nachfolgenden Phasen der Softwareentwicklung. Insgesamt dienen Softwarearchitekturen jedoch zahlreichen Zwecken, nämlich

- allgemein dem Verstehen des Entwicklungsvorhabens,
- der Kommunikation zwischen den Einflussgruppen der Softwareentwicklung,
- der Förderung von Wiederverwendung von Softwareprodukten,
- der Realisierung von nicht-funktionalen Anforderungen an die Software²⁷⁰, sowie
- eben der Koordination der Softwareentwicklung.²⁷¹

Umfangreiche Projektprogramme weisen sehr hohe Abstimmungskosten auf. Durch die Zerlegung des Gesamtsystems in möglichst unabhängige Komponenten und damit möglichst unabhängige Projekte können diese Abstimmungskosten gering gehalten werden. Im Idealfall können sich erforderliche Abstimmungsmaßnahmen auf die Spezifikation der Schnittstellen zwischen den Komponenten beschränken. Werden die Vorgaben der Softwarearchitektur eingehalten, können Komponenten und Features problemlos zum gesamten Softwaresystem integriert werden. Zudem kann die Implementationsstruktur für die Zeitplanung und die Allokation der Projektressourcen herangezogen werden. Insofern dient die Softwarearchitektur insgesamt in wesentlichem Ausmaß der Koordination der Softwareentwicklung.²⁷²

Bereits 1968 hat Conway eine sehr weitgehende Vermutung zum Zusammenhang zwischen der Softwarearchitektur und der Organisation der Softwareentwicklung aufgestellt: Jedes Softwaresystem spiegelt demnach die Struktur der Organisation wieder, innerhalb welcher es entwickelt wurde. Nicht also die Softwarearchitektur determiniert die Organisation, sondern umgekehrt – auch wenn das nicht beabsichtigt wird, da eine solche Softwarearchitektur nicht optimal ist. V.a. wird das der Fall sein, wenn keine ausreichend spezifizierte Softwarearchitektur zu Beginn des Entwicklungsvorhabens entwickelt wurde.²⁷³

²⁷⁰ Vgl. Kap. 2.3.3.2

²⁷¹ Vgl. Lang (2004), S. 51-53

²⁷² Vgl. Mellis (2004), S. 156; vgl. Lang (2004), S. 52

²⁷³ Vgl. Conway (1968), zitiert nach Pichler (2008a), S. 130 und Oestereich, Weiss (2007), S. 32

Auch vor dem Hintergrund der Vermutung von Conway ist es kein Zufall, dass die Prinzipien des Softwarearchitekturentwurfs weitgehende Analogien zu den Prinzipien der Organisationsgestaltung aufweisen. Die koordinationsminimierenden Wirkungen der Softwarearchitektur treten auch nur auf, wenn die entsprechenden Prinzipien eingehalten werden: Das Prinzip der Modularisierung und das Prinzip der Lokalität, wobei beide schlussendlich durch weitere Entwurfsprinzipien umgesetzt werden.

Ist das Prinzip der Modularisierung umgesetzt, so kommunizieren die Teilkomponenten eines Softwaresystems ausschließlich über definierte Schnittstellen miteinander und sind ansonsten unabhängig. Das Prinzip der Modularisierung wird in der modernen Softwareentwicklung durch das Geheimnisprinzip umgesetzt: Danach muss jedes Modul über einen sog. Rumpf und eine Schnittstelle verfügen. Die Schnittstelle kommuniziert nach außen, welche Dienste das Modul seiner Umgebung zur Verfügung stellt. Die Realisierung dieser Dienste beinhaltet der Rumpf. Das Geheimnisprinzip verlangt an dieser Stelle, dass alle Informationen, die die Umgebung eines Moduls benötigt, in der Schnittstelle spezifiziert sind. Die jeweilige Realisierung der Dienste, d.h. der Rumpf, soll gegenüber der Umgebung geheim gehalten werden. Hierzu muss die Schnittstelle so spezifiziert werden, dass sie so wenig wie möglich und so viel wie nötig über die Realisierung der Dienste kommuniziert. Durch das Geheimnisprinzip wird gewährleistet, dass jedes Modul nur insoweit auf Dienste anderer Module zugreift, wie dies in den Schnittstellen spezifiziert ist. Somit kann die Koordination zwischen Entwicklern der verschiedenen Module auf die Spezifikation der Schnittstelle beschränkt werden. Änderungen im Rumpf eines Moduls können dagegen, sofern sie sich nicht auf die Schnittstellenspezifikation auswirken, ohne weitere Abstimmung vorgenommen werden.²⁷⁴

Das Prinzip der Lokalität besagt, dass alle für die Lösung einer bestimmten Teilaufgabe erforderlichen Informationen an einer einzigen Stelle zu bündeln sind. Es wird in der modernen Softwareentwicklung durch die Prinzipien der Kohäsion und der losen Kopplung umgesetzt. Beide sind wiederum eine Ergänzung des Geheimnisprinzips:²⁷⁵

- Nach dem Prinzip der Kohäsion sollte die Intensität der Beziehungen zwischen den einzelnen Elementen eines Moduls (Grad der Kohäsion) möglichst hoch sein. Dies wird erreicht, wenn der Rumpf genau die Elemente enthält, die zur Realisierung einer bestimmten Funktion erforderlich sind;

²⁷⁴ Vgl. Lang (2004), S. 25, 58-60, mit weiteren Verweisen; vgl. Balzert (2001), S. 821, 1027f.

²⁷⁵ Vgl. Lang (2004), S. 25, 60, mit zahlreichen weiteren Verweisen.

- Nach dem Prinzip der losen Kopplung sollte die Intensität der Beziehungen, die zwischen verschiedenen Modulen bestehen, möglichst gering sein. Diese Intensität der Beziehungen wird dabei durch softwaretechnische Gegebenheiten wie bspw. Art der ausgetauschten Datentypen oder Anzahl der Kommunikationsparameter bestimmt.

Welche Interdependenzen zwischen der Entwicklung der verschiedenen Bestandteile des Softwaresystems bestehen, hängt also entscheidend von der Softwarearchitektur ab. Werden bei ihrer Entwicklung die beschriebenen Entwurfsprinzipien eingehalten, so entstehen die geforderten modularen und lokalen Softwarearchitekturen. Durch die Modularisierung sollen sich eventuelle Änderungen auf ein Modul oder Subsystem beschränken. Die Modularität der Softwarearchitektur überträgt sich dann auch auf die Organisation der Softwareentwicklung. Die Kommunikation vollzieht sich in einem solchen Fall weitgehend innerhalb und nicht zwischen den organisatorischen Einheiten. Entscheidungen innerhalb eines Projektes sind dann nur in geringem Maße von Entscheidungen bei der Entwicklung anderer Komponenten und Features in den anderen Projekten des Entwicklungsvorhabens bzw. Projektprogramms abhängig. Eventuelle Abhängigkeiten sind zudem gut bekannt, da sie an explizit definierten Schnittstellen abgelesen werden können. Der koordinationsorientierten organisatorischen Gestaltung der Softwareentwicklung und der Entwicklung von modularen und lokalen Softwarearchitekturen ist somit die Minimierung der Intensität der zwischen den einzelnen Subsystemen bestehenden Schnittstellen gemein.²⁷⁶

An dieser Stelle stellt sich die Frage, in wie weit die Entwicklung der Softwarearchitektur als eine Gestaltungsmaßnahme der Organisation und insb. der Koordination der Softwareentwicklung angesehen werden kann. Aus organisationstheoretischer Sicht lässt sich diese Frage nicht eindeutig beantworten. Denn es geht dabei um nicht weniger als um die generelle Fragestellung, ob Koordinationsbedarf als Folge der Arbeitsteilung angesehen werden soll oder ob Arbeitsteilung und Spezialisierung an sich bereits als Koordination angesehen werden sollen.²⁷⁷

- Die häufige und klassische These besagt, dass Spezialisierung die Ursache der Koordination darstellt. Weil Aufgaben in viele Teilaufgaben zerlegt und an Organisationseinheiten und Mitarbeiter verteilt werden, muss eine zielgerichtete Zusammenarbeit herbeigeführt werden. Spezialisierung macht also Koordination erforderlich, womit Koordinationsbedarf als Folge der Arbeitsteilung und Spezialisierung angesehen wird.

²⁷⁶ Vgl. Mellis (2004), S. 100; vgl. Lang (2004), S. 260f.

²⁷⁷ Vgl. Beck (1996), S. 34f., mit zahlreichen weiteren Verweisen; vgl. Jost (2009), S. 24

- Die These des zweiteiligen Koordinationsproblems bezieht Koordination dagegen nicht nur auf die Zusammenarbeit und Abstimmung zwischen Organisationseinheiten. Vielmehr bestimmt der vorgelagerte Schritt der Arbeitsteilung, welche Interdependenzen zwischen den Organisationseinheiten überhaupt bestehen. Daher ist die Frage nach der reibungslosen Kommunikation und Abstimmung nicht zu trennen von der Frage nach der produktivsten Spezialisierung. Das Anordnen von Teilen eines Ganzen ist nach dieser These in sich bereits ein Abstimmen. Koordination in diesem (umfassenden) Sinn erstreckt sich somit über die laufende Abstimmung und Kommunikation zwischen Organisationseinheiten hinaus auch auf die Gestaltung der Arbeitsteilung und Spezialisierung.

Obwohl die vorliegende Arbeit das Koordinationsproblem grundsätzlich als das zweiteilige Problem ansieht, wird hier der Entwurf der Softwarearchitektur nicht als eine Gestaltungsmaßnahme der Organisation und insb. der Koordination der Standardsoftwareentwicklung angesehen. Der Grund hierfür ist, dass es sich beim Grob- und Feinentwurf der Softwarearchitektur um phasenspezifische Entwicklungsaufgaben handelt, welche unmittelbar der Erfüllung des Sachziels, d.h. der Entwicklung und Bereitstellung von Software, dienen. Ausgenommen der Koordination der Softwareentwicklung sind auch die eingangs genannten Zwecke des Softwarearchitekturentwurfs eindeutig der fachlich-technischen und nicht der organisatorischen Perspektive auf die Softwareentwicklung zuzuordnen. Damit geht einher, dass sich Entscheidungen beim Architekturentwurf nicht vordergründig am Ziel der Koordinationsfähigkeit ausrichten, sondern diverse fachlich-technische Ziele verfolgen. Zudem unterliegen sie auch vielen fachlich-technischen Gestaltungsbedingungen, auf die im Rahmen der vorliegenden Arbeit nicht eingegangen werden kann, da es sich dabei um rein softwaretechnische Fragestellungen und um Fragestellungen der Informatik handelt.²⁷⁸

Abschließend soll auf eine wesentliche Erkenntnis in Zusammenhang mit der Koordination durch Entwurf der Softwarearchitektur eingegangen werden: Das Koordinationsproblem kann durch den Entwurf einer Softwarearchitektur nicht gelöst bzw. der Koordinationsbedarf eines Entwicklungsvorhabens nicht eliminiert werden. Das Koordinationsproblem bzw. der Koordinationsbedarf können lediglich auf ein beherrschbares Maß reduziert werden. Der verbliebene Koordinationsbedarf muss mit geeigneten Koordinationsmechanismen abgedeckt werden.²⁷⁹ Die Gründe dafür liegen zum großen Teil darin, dass die Stabilität des Architekturentwurfs begrenzt ist.

²⁷⁸ Vgl. Kap. 2.1.2.1 zu Gestaltungsmaßnahmen in der Softwareentwicklung; vgl. zur Architekturentwicklung als eine softwaretechnische Aufgabe Bass u.a. (1998)

²⁷⁹ Vgl. Mellis (2004), S. 141

In der Praxis der Softwareentwicklung lassen sich Anforderungen an die zu entwickelnde Software trotz hohen Aufwandes kaum vollständig erheben und ändern sich zudem im Laufe der Entwicklungszeit. Die Anforderungsänderungen und die mangelnde Stabilität der Anforderungen können dabei soweit gehen, dass sich dadurch auch Änderungen in der Softwarearchitektur ergeben. Ferner ist der Entwurf einer Softwarearchitektur mit optimierter Komponentenstruktur eine äußerst schwierige Aufgabe, bei der sich erst im Laufe der Entwicklung der Komponenten zeigen kann, dass bspw. Veränderungen an den Schnittstellen oder eine andere Verteilung der Anforderungen auf die Komponenten vorteilhafter ist als zunächst angenommen.²⁸⁰ Hierbei kann es zu einer veränderten Abgrenzung der Module oder zu neuen Schnittstellenspezifikationen kommen. Dementsprechend können Änderungen in der Softwarearchitektur wesentliche Auswirkungen auf die Koordination und sogar auf die Art der horizontalen Arbeitsteilung haben.²⁸¹

2.3 Situativer Koordinationsansatz für die Standardsoftwareentwicklung

Das Gestaltungsziel dieser Arbeit ist die Entwicklung eines situativen Koordinationsansatzes für die Standardsoftwareentwicklung. Um dieses Ziel zu erreichen müssen die Forschungsfragen zunächst auf konzeptioneller Ebene beantwortet werden. Dies erfolgt durch die Kombination der beiden theoretischen Zugänge zur Problemstellung im vorliegenden Kap. 2.3. Hieraus werden in Kap. 3 konkrete Entscheidungshilfen für die Praxis entwickelt.

2.3.1 Koordinationseffektivität als Gestaltungsziel

2.3.1.1 Das Konzept der Koordinationseffizienz als Ausgangspunkt

Koordination bezeichnet das Ausrichten von Einzelaktivitäten in einem arbeitsteiligen System auf das übergeordnete Gesamtziel.²⁸² Die Notwendigkeit der Koordination ergibt sich durch das grundlegende Dilemma arbeitsteiliger Systeme. Auf der einen Seite erfordert die begrenzte qualitative und quantitative Kapazität der Organisationseinheiten die Aufteilung eines komplexen Gesamtproblems in Teilprobleme und die bis zu einem gewissen Grad isolierte Entwicklung von Teillösungen dafür. Die Erreichung des übergeordneten Gesamtziels erfordert auf der anderen Seite die integrierende Abstimmung aller Teilaktivitäten. Der Realisierungsgrad des Gesamtziels hängt in einem solchen arbeitsteiligen System von der Abstimmung von Entscheidungen ab, die durch

²⁸⁰ Vgl. Mellis (2004), S. 141

²⁸¹ Vgl. Lang (2004), S. 68f.

²⁸² Vgl. Albach (1967), zitiert nach Frese (2005), S. 143

Koordination bezweckt wird. Koordinationsmaßnahmen werden deshalb dem Entscheidungssystem zugeordnet.²⁸³

Da sich der Verzicht auf Koordination in einer Gewährung von Entscheidungsautonomie äußert, lässt sich die Gestaltung der Koordination auf die Beantwortung der Frage reduzieren, in welchem Ausmaß den Organisationseinheiten eines arbeitsteiligen Entscheidungssystems Entscheidungsautonomie eingeräumt werden soll. Einen Zugang zur Analyse dieser Problematik eröffnet die Unterscheidung zwischen Autonomie- und Abstimmungskosten.²⁸⁴

Die Abkehr von dem Versuch einer vollständigen Berücksichtigung bestehender Interdependenzen zwischen den Teillösungen des Gesamtproblems bedeutet zwangsläufig, dass auf die Verwirklichung der theoretisch denkbaren Optimallösung, wie sie der simultanen Planung entspricht, verzichtet wird. Die Differenz zwischen dem theoretisch möglichen Optimum und dem tatsächlich realisierten Ergebnis lässt sich durch Kosten der Autonomie ausdrücken. Zur Beurteilung der Autonomiekosten ist also stets von den Zielen des zu koordinierenden Systems auszugehen: Autonomiekosten stellen einen globalen Ausdruck für alle Nachteile dar, die eine unvollkommene Abstimmung der Entscheidungen der einzelnen autonomen Einheiten für das Gesamtsystem bzw. das Gesamtproblem aufweist. Die Vermeidung von Autonomiekosten und die Verbesserung der Entscheidungsqualität resultieren dabei aus der Verbesserung der Informationsbasis der Entscheider und der Verbesserung der Methodenbasis bei der Informationsverarbeitung.²⁸⁵

Auf der anderen Seite erfordert der Abbau von Autonomiekosten Abstimmungen, die entsprechende Abstimmungskosten verursachen. Es sind ganz allgemein Kosten des Einsatzes von Ressourcen und Zeit bei der Koordination der verschiedenen Entscheidungseinheiten – so bspw. Kosten der Ausarbeitung von Projektprogrammzielen oder auch Kommunikationskosten. Die ökonomische Gestaltung der Koordination muss deshalb einen Ausgleich zwischen Abstimmungs- und Autonomiekosten anstreben.²⁸⁶ Abb. 23 stellt den Zusammenhang zwischen Autonomie- und Abstimmungskosten graphisch dar.

²⁸³ Vgl. Frese (2005), S. 143f.

²⁸⁴ Vgl. Frese (2005), S. 144

²⁸⁵ Vgl. Frese (2005), S. 145, 323f.; vgl. Fieten (1977), S. 68

²⁸⁶ Vgl. Frese (2005), S. 145, 323f.; vgl. Fieten (1977), S. 70

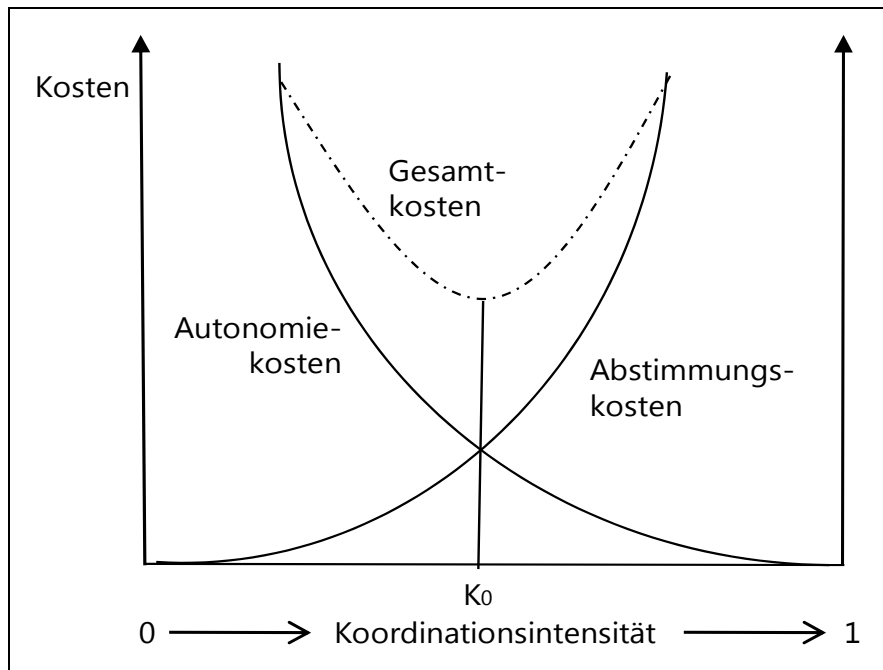


Abb. 23: Zusammenhang zwischen Autonomie- und Abstimmungskosten²⁸⁷

Mit zunehmender Koordinationsintensität sinken die Autonomiekosten, während die Abstimmungskosten steigen. Die Skala der Koordinationsintensität ist durch die beiden Grenzfälle der vollkommenen Entscheidungsautonomie (Koordinationsintensität = 0) und der vollkommenen Koordination (Koordinationsintensität = 1) begrenzt. Die optimale Koordinationsintensität ‚ K_0 ‘ wird durch das Minimum der Gesamtkostenkurve bestimmt, die sich wiederum aus der Addition der Autonomie- mit der Abstimmungskostenkurve ergibt.²⁸⁸

In der Softwareentwicklung im Allgemeinen ist Koordinationseffizienz tendenziell bei einer sehr hohen Koordinationsintensität gegeben. Die Koordination kann dabei als Vorauskoordination, also ganz oder teilweise vor Beginn der Aufgabenerfüllung, oder als Feedbackkoordination i.S. der Reaktion auf Störungen, erfolgen. Die Vorauskoordination basiert auf unsicheren Prognosen bei der Einschätzung der zukünftigen Koordinationsbedürfnisse, sodass ihre Wirksamkeit mit dem Risiko dieser Prognose behaftet ist. Kritisch für die Effizienz der Feedbackkoordination ist dagegen, dass Projektmitarbeiter über geeignete Instrumente zur ständigen Identifizierung von Koordinationsbedarfen verfügen.

Man kann davon ausgehen, dass Softwareunternehmungen sowohl die Voraus-, als auch die Feedbackkoordination einsetzen, wobei aber der Schwerpunkt auf der Feedbackkoordination liegt. Die Möglichkeiten der Vorauskoordination werden, bedingt durch die typische hohe Dynamik und Unbestimmtheit der Kundenanforderungen an

²⁸⁷ Frese (2005), S. 146; nach Emery (1969), S. 31

²⁸⁸ Vgl. Frese (2005), S. 146

das zu entwickelnde Softwaresystem sowie durch die hohe technologische Unsicherheit, als eingeschränkt angesehen. Gleichzeitig werden Störungen im Entwicklungsverlauf sogar für unvermeidbar gehalten. Der Entwicklungsverlauf wird daher bewusst so gestaltet, dass, in Zusammenspiel mit der Vorkoordination, einer effektiven und effizienten Feedbackkoordination eine entscheidende Bedeutung zukommt.²⁸⁹

Auch wenn dies nicht explizit formuliert wird, spielt die Koordinationseffizienz bei mehreren Ansätzen im Umfeld der Softwareentwicklung eine wesentliche Rolle. So bedient sich VBSE zwar einer eigenen Terminologie, doch das Problem, das hier thematisiert wird, ist die Erhöhung der Koordinationseffizienz bzw. die Minimierung von Autonomiekosten. Ebenso spielt die Koordinationseffizienz bei der Argumentation zur Vorteilhaftigkeit der Agilen Softwareentwicklung²⁹⁰ die entscheidende Rolle. Die Koordinationseffizienz ist in beiden Fällen ein zentrales Gestaltungsziel.

VBSE wird als Gegenbild zur ‚wertneutralen Softwareentwicklung‘ charakterisiert, bei der jedes (Teil-)Projekt, jede Kundenanforderung, jeder potenzielle Fehler etc. als gleichwertig betrachtet werden. Neben dieser ‚Wertneutralität‘ prangert VBSE die ‚implizite Wertorientierung‘ an: Im Laufe eines Softwareentwicklungsprojektes stehen Entscheidungsträger vor einer Vielzahl von Alternativen bei der Abwägung von Kundenanforderungen gegen Entwicklungskosten und -zeit: Termine könnten durch Abstriche beim Leistungsumfang oder durch eine geringere Testabdeckung doch noch eingehalten werden usw. Wenn aber solche Betrachtungen implizit bleiben und jeder beteiligte Akteur damit nur aus seiner Sicht entscheidet, werden Entscheidungen nicht im Sinn der dem Projekt übergeordneten Produkt- und Unternehmensziele getroffen.²⁹¹ ‚Implizite Wertorientierung‘ und ‚Wertneutralität‘ bei VBSE führen offensichtlich zu Autonomiekosten und damit zur mangelnden Koordinationseffizienz.

In der Terminologie des Koordinationseffizienz-Ansatzes wird bei der Agilen Softwareentwicklung implizit angenommen, dass

- der informale persönliche Informationsaustausch als bevorzugte Kommunikations- bzw. Koordinationsform innerhalb und über die Grenzen eines Entwicklungsprojektes hinaus koordinationseffizient ist, sowie dass
- die Sachziele der Softwareentwicklung im Punkt der Koordinationseffizienz durch sich selbst organisierende Teams erreicht werden.²⁹²

²⁸⁹ Vgl. Mellis (2004), S. 136-139

²⁹⁰ Vgl. Kap. 2.1.3.2 und 2.1.3.3

²⁹¹ Vgl. Biffel u.a. (2006), S. IX; vgl. Boehm (2006), S. 3, 7

²⁹² Vgl. hierzu auch Kap. 2.1.3.2

Beide Annahmen gehen offensichtlich davon aus, dass die zusätzlichen Autonomiekosten durch eine höhere Autonomie der Einheiten niedriger sind als die zusätzlichen Koordinationskosten, die durch die Definition und Einhaltung von (Koordinations-) Prozessen und die Ausformulierung und Einhaltung von Plänen entstehen würden. Bei dem ebenfalls in Kap. 2.1.3.2 vorgestellten risikobasierten Ansatz von Boehm und Turner wird diese Ansicht relativiert, indem fünf Gestaltungsbedingungen definiert werden, von welchen die Eignung der beiden Ansätze abhängig gemacht wird. Sie sind aus Sicht der Koordinationseffizienz als Gestaltungsbedingungen zu interpretieren, die sich positiv oder negativ auf den Verlauf der Autonomie- und Koordinationskostenkurven auswirken und damit die optimale Koordinationsintensität im Punkt der Koordinationseffizienz beeinflussen.

2.3.1.2 Das Konzept der Koordinationseffektivität

Die bisherigen Ausführungen lassen zwei wesentliche Fragen offen, zu deren Beantwortung das Koordinationseffizienz-Konzept keine Hilfestellung bietet.²⁹³

- Es ist zwar unstrittig, dass die optimale Koordinationsintensität nur über die Betrachtung der Koordinations- und der Autonomiekostenkurve exakt bestimmt werden kann. Es muss jedoch auch darauf hingewiesen werden, dass die praktische Ermittlung der beiden Kostenkurven schwierig sein dürfte. Denn es ist nicht anzunehmen, dass die Autonomie- wie auch die Koordinationskosten sämtlicher möglicher Koordinationsintensitäten ex ante bekannt sind. In der Praxis können sie bestenfalls ex post, d.h. nachdem die Gestaltung der Koordination erfolgt ist, grob abgeschätzt werden. Dies hat wiederum zur Folge, dass die optimale Koordinationsintensität in existenten Entscheidungssystemen nur annähernd genau bestimmt werden kann;
- Mit der Feststellung der optimalen Koordinationsintensität ist das Koordinationsproblem in seiner Gesamtheit keineswegs gelöst, denn es wird damit nicht angegeben, welche konkreten Koordinationsmaßnahmen zu ergreifen sind. Die optimale Zusammensetzung der verschiedenen zur Verfügung stehenden Koordinationsmaßnahmen ist aus der Kenntnis der optimalen Koordinationsintensität nicht ableitbar.

Die vorliegende Arbeit setzt vor allem an dem zweiten Kritikpunkt an und erarbeitet konkrete Gestaltungsmaßnahmen für die Koordination (Kap. 2.3.2 und Kap. 3.2), die über die geeignete organisatorische Umsetzung (Kap. 2.3.4 und Kap. 3.4) das Koordinationsproblem situationsspezifisch (Gestaltungsbedingungen; Kap. 2.3.3 und Kap.

²⁹³ Vgl. Fieten (1977), S. 71f.

3.3) lösen. Daher soll in der vorliegenden Arbeit beim Gestaltungsziel des situativen Koordinationsansatzes von der Koordinationseffektivität gesprochen werden. Zwar kann beim Koordinationseffizienz-Ansatz die Kurve der (Vermeidung von) Autonomiekosten über die Verbesserung der Entscheidungsqualität als Effektivität der Koordination angesehen werden.²⁹⁴ Trotzdem soll im Folgenden Koordinationseffektivität ausgehend von dem Begriffspaar Effektivität und Effizienz als ein eigenständiges Konstrukt definiert werden. Der Grund dafür ist, dass hier die grundsätzliche Wirksamkeit (Effektivität) der Koordination und das Entscheidungssystem im Vordergrund stehen und nicht der Mitteleinsatz in Relation zur Zielerreichung (Effizienz) bei der Koordination.

Für die beiden Begriffe Effektivität und Effizienz konnte sich in der Literatur bislang keine einheitliche und in der Forschung als allgemein anerkannt angesehene Globaldefinition etablieren. Im Gegenteil, in der Literatur zur Effektivitätsforschung werden die beiden Begriffe Effektivität und Effizienz inkonsistent und in einem unterschiedlichen Verhältnis zueinander verwendet. Teilweise werden gar die Termini Effektivität und Effizienz ohne eine adäquate Beantwortung der Frage, was sich dahinter konkret verbirgt, verwendet. Zudem erfolgt die Effektivitäts-/Effizienzdiskussion häufig nicht bezogen auf einen speziellen Untersuchungsgegenstand, sondern in allgemeiner Form und insb. ohne eine intensive Auseinandersetzung mit der Effektivität und Effizienz von Entscheidungssystemen. Offen ist neben einer Globaldefinition und -abgrenzung der beiden Begriffsverwendungen folglich ebenfalls die Beziehung der Dimensionen zueinander – insbesondere bei dem speziellen Untersuchungsgegenstand des organisationalen Entscheidungssystems.²⁹⁵

Trotz der im Detail liegenden inhaltlichen Divergenzen bei den diversen Abgrenzungen von Effektivität und Effizienz kristallisieren sich, zumindest grob betrachtet, wesentliche inhaltliche Gemeinsamkeiten heraus. Im Kern stellen die folgenden beiden Punkte das sachliche Kontinuum über alle Abgrenzungsbemühungen dar:²⁹⁶

- Effektivität befasst sich immer mit Aspekten der finalen Zielerreichung im Sinne eines Ergebnisses der Leistungserstellung. Im Kern der Überlegungen steht stets die grundsätzliche Wirksamkeit von Maßnahmen zur Zielerreichung. Effektivität bezieht sich also darauf, ob in einer Organisation die richtigen Ziele angestrebt und auch erreicht werden (plakativ und vereinfacht: ‚doing the right things‘). Effektiv

²⁹⁴ Vgl. Billing (2003), S. 50

²⁹⁵ Vgl. Sauerwald (2007), S. 5, 34, 38; vgl. Bunting (1995), S. 73; vgl. Scholz (1992), Sp. 533; vgl. Ahn, Dyckhoff (1997), S. 2f.

²⁹⁶ Vgl. Sauerwald (2007), S. 37; vgl. Bunting (1955), S. 74; vgl. Lindstädt (2005), S. 32f.; vgl. Schulte-Zurhausen (2002), S. 5

ist eine Maßnahme demnach dann, wenn sie grundsätzlich zur Erreichung eines definierten Zieles geeignet ist;

- Während in Zusammenhang mit der Effektivität das Ziel selbst thematisiert wird, setzt die Effizienz Mitteleinsatz und Zielerreichung zueinander in Beziehung. Sie befasst sich immer mit Aspekten des Transformationsprozesses hin zu einem angestrebten Ergebnis, wobei meist eine Fokussierung auf den Faktorverbrauch erfolgt. Effizienz wird dabei als Relation zwischen Input- und Outputgrößen ermittelt (plakativ und vereinfacht: ‚doing things right‘).

Neben alternativen Abgrenzungen von Effektivität und Effizienz an sich existieren auch im Hinblick auf die hierarchische Stellung der beiden Konstrukte zueinander verschiedene Ansichten. Theoretisch möglich ist dabei eine Gleichstellung oder eine Einordnung des einen über oder unter das andere Konstrukt. Daraus ergeben sich drei Muster, bei welchen die beiden Konstrukte jeweils eine unterschiedliche Gewichtung erfahren können. Nicht zuletzt wegen der bislang fehlenden Klarheit und allgemein anerkannten Interpretation des Begriffspaares Effektivität und Effizienz sowie dem fehlenden theoretischen Unterbau konnte sich auch hier kein Konsens herausbilden. Stattdessen sind die drei grundsätzlichen Ansätze bezüglich Gleich-, Über- oder Unterordnung von Effektivität und Effizienz zueinander gleichermaßen in der Literatur zu finden.²⁹⁷

- Effektivität wird als das der Effizienz übergeordnete Konstrukt betrachtet, womit der Effizienz kommt nur eine untergeordnete Bedeutung zukommt;
- Effizienz wird als das der Effektivität übergeordnete Konstrukt betrachtet, womit wiederum der Effektivität nur eine untergeordnete Bedeutung zukommt;
- Effektivität und Effizienz werden als gleichrangige Konstrukte betrachtet.

Bei der Gleich-, Über- oder Unterordnung von Effektivität und Effizienz zueinander kann es sich sowohl um eine Gewichtung im streng hierarchischen Sinn, als auch um eine Gewichtung im Sinn einer Priorisierung ohne einen hierarchischen Bezug, handeln. In beiden Fällen besteht ein rein sachlicher Zusammenhang zwischen Effektivität und Effizienz. Darüber hinaus besteht aber auch ein zweiter, zeitlicher Zusammenhang der beiden Konstrukte zueinander. Die Realisierung der Effektivität als die generelle Möglichkeit, mit einer Maßnahme das definierte Ziel zu erreichen, erfolgt stets vor der Realisierung der Effizienz. Im Rahmen des organisationalen Handelns muss chronologisch also erst die Effektivität und dann die Effizienz im Fokus stehen. Diese zeitliche

²⁹⁷ Vgl. Ahn, Dyckhoff (1997), S. 2f.; vgl. Sauerwald (2007), S. 34-41, mit zahlreichen weiteren Verweisen.

Abfolge ist ein möglicher Erklärungsansatz für die in der Literatur oft anzutreffende Unterordnung der Effizienz.²⁹⁸

Unabhängig von der Abgrenzung von Effektivität und Effizienz im Detail und von der hierarchischen Stellung der beiden Konstrukte zueinander zeichnet die Effektivität eine wesentliche Tatsache aus: Effektivität kann im Gegensatz zur Effizienz nicht als eine Einzelgröße bzw. als ein Kriterium per se betrachtet werden, sondern stellt vielmehr ein weitreichendes Konstrukt dar. Daher erschließt sich das Effektivitätskonstrukt erst über die Konstituierung eines Modells der Effektivität.²⁹⁹ „Organizational effectiveness must be located in some kind of theoretical context for it to make any sense.“³⁰⁰

Die Koordinationseffektivität wird als ein Stimmigkeitskonstrukt definiert. Das Stimmigkeitsparadigma ist ein sinnvolles Konzept zur Erklärung von organisatorischer Effektivität,³⁰¹ insbesondere wenn eine Arbeit den Situativen Ansatz der Organisationsgestaltung verfolgt. Stimmigkeit drückt die Kompatibilität von mindestens zwei Variablen hinsichtlich konkreter Ziele aus. Es gibt danach keine absolute Stimmigkeit, sondern nur situative Stimmigkeiten. Hinsichtlich der abzustimmenden Variablen existieren diverse Forschungsarbeiten, die auch im Zusammenhang mit dem Situativen Ansatz diskutiert werden.³⁰²

Die abzustimmenden Variablen beim Stimmigkeitskonstrukt Koordinationseffektivität sind, wie in den folgenden Kapiteln noch näher erläutert und konkretisiert wird, Projektzielkritikalität und Koordinationsprozesse, wobei letzteres einen Variablenkomplex darstellt. Koordinationseffektivität ist demnach bei der Stimmigkeit der einzelnen Gestaltungsmaßnahmen innerhalb der Koordinationsprozesse zueinander und zur Kritikalität der (untereinander konkurrierenden) Projektziele³⁰³ gegeben. Damit gibt das Konzept der Koordinationseffektivität im Gegensatz zur Koordinationseffizienz konkret Gestaltungsmaßnahmen der Koordination vor. Analog zur Effektivität im Allgemeinen steht hier die grundsätzliche Wirksamkeit der Koordinationsmaßnahmen im Kern der Überlegungen. Im Punkt der Koordinationseffektivität erfolgen die nachfolgenden, eigentlichen Objektentscheidungen möglichst rational und nicht weitgehend begrenzt rational. Koordinationseffektivität stellt damit nicht wie die Koordinationseffizienz primär Mitteleinsatz und Zielerreichung (Autonomie- und Koordinationskosten) miteinander in Beziehung (vgl. Abb. 24).

²⁹⁸ Vgl. Sauerwald (2007), S. 34, 45

²⁹⁹ Vgl. Bünting (1955), S. 74, 76, mit zahlreichen weiteren Verweisen.

³⁰⁰ Campbell (1977), S. 14

³⁰¹ Stimmigkeitskriterien eignen sich insbesondere zur Beurteilung der Effektivität, da in Zusammenhang mit Effektivität Ziele selbst Gegenstand der Überlegungen sind. Vgl. Lindstädt (2005), S. 33

³⁰² Vgl. Scholz (1992), Sp. 544

³⁰³ Vgl. Kap. 3.2 und Kap. 3.3

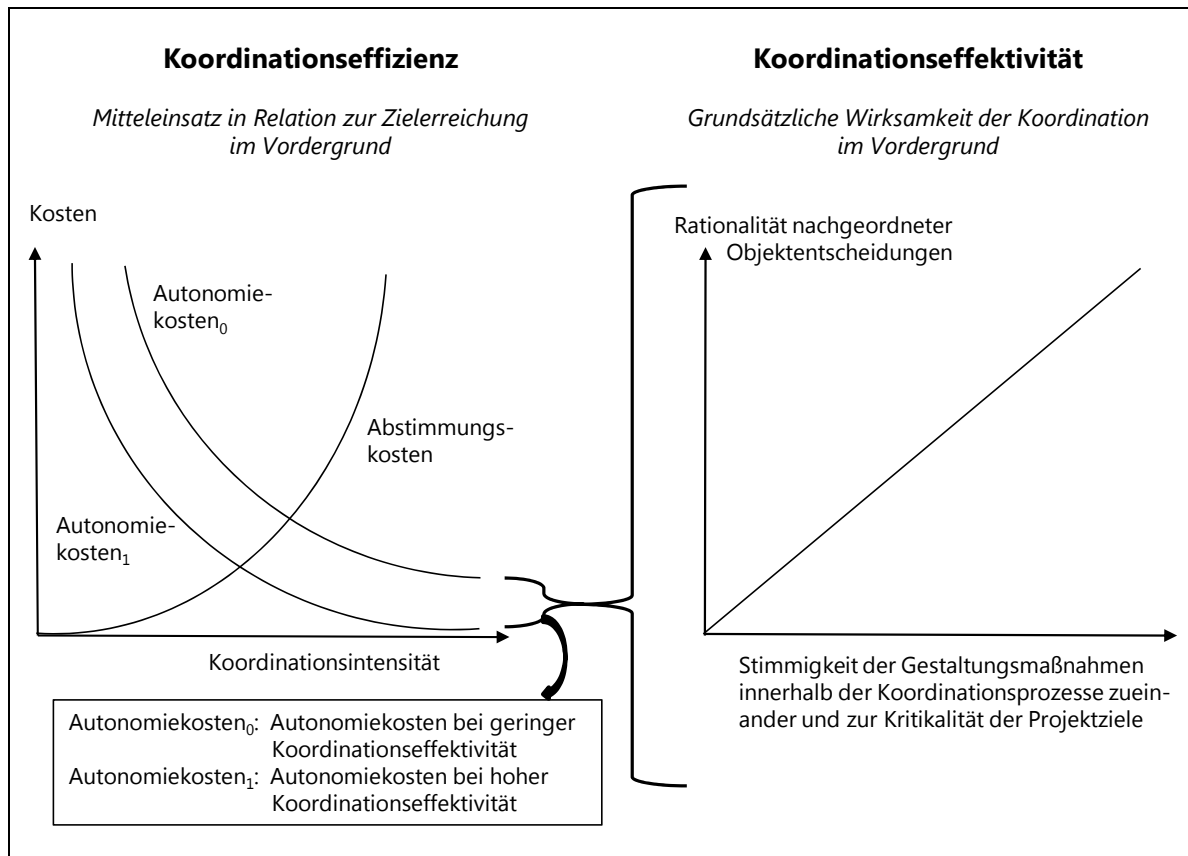


Abb. 24: Das Konzept der Koordinationseffektivität

2.3.2 Gestaltungsmaßnahmen der Koordination

2.3.2.1 Der Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess

Empirische Untersuchungen von arbeitsteiligen Entscheidungsprozessen bei hoch innovativen Vorhaben zeigen, dass Ziele vielfach in multipersonalen und konfliktgeladenen Prozessen herausgearbeitet werden. Zudem verläuft der Zielbildungsprozess zeitlich parallel zum Problemlösungsprozess und endet in Extremfällen erst mit dem Ende des Vorhabens. Bei diesem Zielbildungsprozess handelt es sich sowohl um Lernen als auch um Abstimmungs- und Verhandlungsprozesse. Die sich daraus ergebenden Ziele sind nicht konstant, sondern variieren im Zeitablauf – insbesondere dann, wenn Problemlösungsaktivitäten den Zielbildungsprozess erneut anstoßen. Desweiteren sind die formulierten Ziele häufig nur unscharf und vage formuliert. Diese Unklarheiten resultieren aus der hohen Ungewissheit und Komplexität der Entscheidungssituationen bei innovativen Vorhaben. Unter diesem Aspekt kann es nach Hauschildt in der Entscheidungsforschung sogar vertreten werden, Ziele unklar zu formulieren. Allerdings sollten sie nicht willkürlich und unüberlegt unklar formuliert werden, sondern nach dem Prinzip einer bewusst gestalteten und kontrollierten Zielunklarheit.³⁰⁴

³⁰⁴ Vgl. Hauschildt (1977), S. 98ff.; vgl. Schrader, Göpfert (2001), S. 77-79

Das beschriebene Bild der Entscheidungsprozesse bei hoch innovativen Vorhaben bestätigt sich auch bei der Betrachtung der Softwareentwicklung im Allgemeinen. So erweist sich in der Praxis bspw. die vollständige Formulierung von Kundenanforderungen an das Softwareprodukt in der Definitions- und Analysephase als schwierig. Vielmehr werden im Verlauf der nachfolgenden Entwicklungsphasen Entwurf und Implementierung weitere Anforderungen sichtbar und (stillschweigende) Annahmen hinsichtlich festgelegter Anforderungen als unzutreffend erkannt.³⁰⁵

„Die Softwareentwicklung besteht darin, im wahrsten Sinne des Wortes tausende von Entscheidungen ... zu fällen.“³⁰⁶ Die Ungewissheit bei der Softwareentwicklung resultiert aus der Ungewissheit darüber, wie diese Entscheidungen getroffen werden. Sie reduziert sich kontinuierlich mit dem Treffen von Entscheidungen im Verlauf eines Entwicklungsvorhabens. Das Ausmaß dieser Zielunklarheiten in der Softwareentwicklung und ihre Reduktion zeigen sich sehr anschaulich beim sog. Konus der Unsicherheit: Empirische Untersuchungen ergeben hier, dass Projektaufwandschätzungen als Folge des kontinuierlichen Zielbildungsprozesses im Laufe der Zeit verschiedenen Graden von Ungewissheit unterliegen. Der Konus der Unsicherheit zeigt, dass die Schätzgenauigkeit mit dem zeitlichen Projektverlauf zunimmt (vgl. Abb. 25).³⁰⁷ Insgesamt aber sind die Ungewissheit und die Entscheidungsspielräume in der Praxis der Softwareentwicklung derart groß, dass hier nicht von einer kontinuierlich zunehmenden Konkretisierung, sondern eher von einer Reduzierung von Unbestimmtheit gesprochen werden muss.³⁰⁸

Die empirisch begründeten Aussagen des Konus der Unsicherheit werden bei Pohl konzeptionell untermauert. Er geht auch von einer zunehmenden Einschränkung des Lösungsraumes bzw. der Entscheidungsspielräume im Verlauf des Entwicklungsprozesses aus und trennt dabei zwischen dem ‚Wie‘ und dem ‚Was‘ als Problem- und Lösungsdimension. Wie im Folgenden erläutert wird, verringert sich durch die ständige Wechselwirkung zwischen der Problem- und der Lösungsdimension die Anzahl der möglichen Lösungen als Lösungsraum bzw. Entscheidungsspielraum in jeder nachfolgenden Phase des Entwicklungsprozesses sukzessive.

³⁰⁵ Vgl. Mellis (2004), S. 102

³⁰⁶ McConnell (2006), S. 67

³⁰⁷ Vgl. McConnell (2006), S. 67ff.

³⁰⁸ Vgl. Stelzer (1998), S. 40f.; vgl. Weltz, Ortmann (1992), S. 34, 141

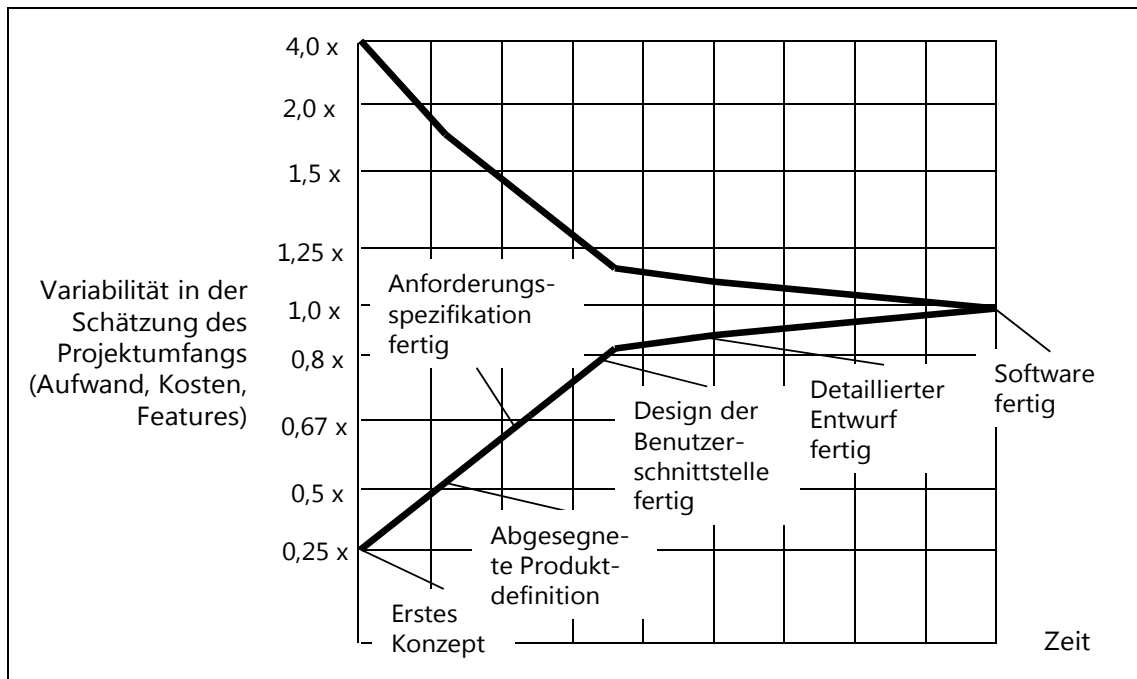


Abb. 25: Konus der Unsicherheit bei der Softwareentwicklung³⁰⁹

Die Frage, was die Problemdefinition als das ‚Was‘ und was die Lösungsbeschreibung als das ‚Wie‘ darstellt, ist von der Betrachtungsperspektive abhängig. Aus der Sicht des Anforderungsingenieurs beschreiben die Kundenanforderungen das Problem. Die Anforderungsspezifikation stellt somit die Problemdefinition dar, für die die Softwarearchitektur die zugehörige Lösung beschreibt. Kundenanforderungen dokumentieren somit aus Sicht des Anforderungsingenieurs das ‚Was‘ und die Softwarearchitektur das ‚Wie‘. Eine solche Differenzierung zwischen der Problemdefinition und der Lösungsbeschreibung in Abhängigkeit von der Sichtweise lässt sich auf den gesamten Softwareentwicklungsprozess übertragen. Wie in Abbildung 26 dargestellt, alterniert der Entwicklungsprozess zwischen der Problemdefinition und der Lösungsbeschreibung. Die Lösungsbeschreibung eines höheren Abstraktionsniveaus stellt dabei die Problemdefinition für das jeweils niedrigere Abstraktionsniveau dar. Basierend auf einer zu Beginn vorgegebenen Vision werden für einen bestimmten Kontext zunächst die Kundenanforderungen an die Software spezifiziert. Diese Anforderungen definieren das Problem für den Architekturf Entwurf. Dieser beschreibt eine mögliche Architektur eines Softwaresystems, die den Anforderungen genügt. Gleichzeitig stellt der Architekturf Entwurf die Problembeschreibung für die Anforderungen an die Komponenten dar.³¹⁰

³⁰⁹ McConnell (2006), S. 69

³¹⁰ Vgl. Pohl (2007), S. 20ff.

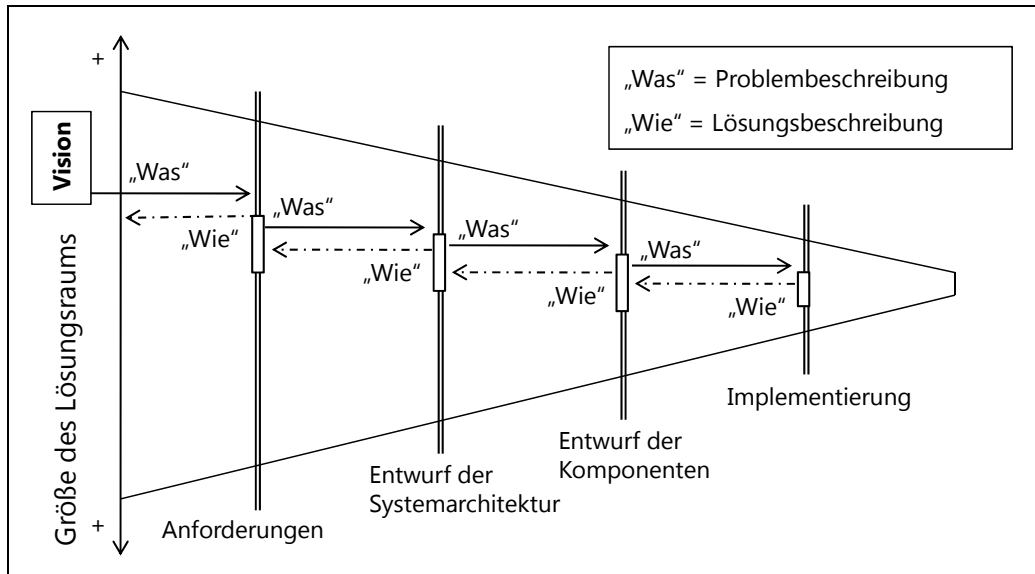


Abb. 26: Einschränkung des Lösungsraumes entlang des Entwicklungsprozesses³¹¹

Der beschriebene kontinuierliche Zielbildungsprozess bei der Softwareentwicklung birgt ein hohes Konfliktpotenzial. Dieses resultiert nicht zuletzt daraus, dass die Unsicherheit zu Beginn eines Entwicklungsvorhabens naturgemäß zu Planungsfehlern in Form einer zu optimistischen Planung führt.³¹² Ein Planungsfehler ergibt sich dabei als Unterschied zwischen dem Geplanten und dem Erreichbaren. Die zweite Art von Fehlern sind tatsächliche Fehler als Folge einer mangelhaften Leistung. Ein tatsächlicher Fehler ergibt sich als Unterschied zwischen dem Erreichbaren und dem, was tatsächlich erreicht wurde. Der wahrgenommene Fehler setzt sich bei der optimistischen Planung aus dem tatsächlichen Fehler und dem Planungsfehler zusammen.³¹³ Wie aus Abb. 27 zu entnehmen ist, wird bei der optimistischen Planung ein Fehler selbst dann wahrgenommen (wahrgenommener Fehler), wenn tatsächlich keine Fehler im Entwicklungsverlauf entstanden sind und das Erreichbare auch erreicht worden ist.

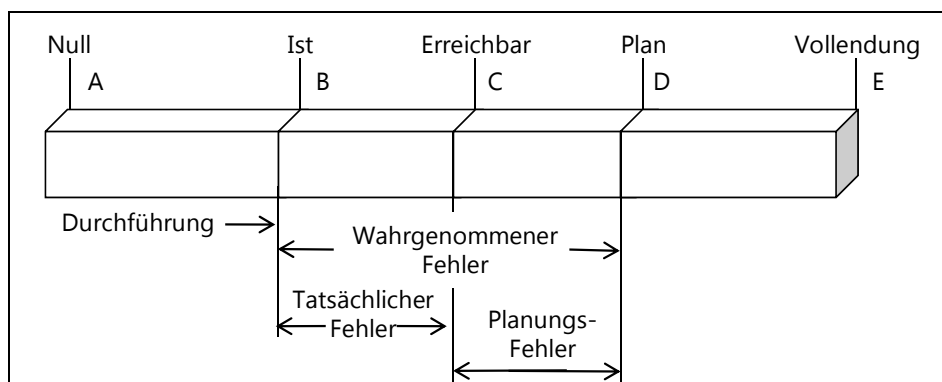


Abb. 27: Komponenten des Misserfolgs bei der optimistischen Planung³¹⁴

³¹¹ Pohl (2007), S. 21

³¹² Vgl. McConnell (1996), S. 170; Theoretisch zwar denkbar, aber in der Praxis kaum von Belang, sind Planungsfehler i.S. einer zu pessimistischen Planung.

³¹³ Vgl. Kerzner (2003), S. 56

³¹⁴ Kerzner (2003), S. 57

Unabhängig von einem eventuell wahrgenommenen Fehler muss das Missverhältnis zwischen den verfügbaren Ressourcen und den Projektzielen, also der Planungsfehler (optimistische Planung) im Projektverlauf nicht zuletzt im Zuge der Koordination ausgeräumt werden. Konflikte sind in einem solchen Umfeld unausweichlich und treten auf, wenn sich mindestens zwei Ausgangspositionen des Verhaltens von Beteiligten an einer Entscheidungssituation widersprechen. Allgemein können dabei die folgenden Konfliktarten unterschieden werden:³¹⁵

- Wissens- und Wahrnehmungskonflikte durch widersprüchliche Wissensstände, Definitionen oder Abgrenzungen sowie durch deren individuelle Erkenntnis, Filterung und Verzerrung;
- Motiv- und Zielkonflikte durch widersprüchliche Vorstellungen von dem anzustrebenden zukünftigen Zustand;
- Ressort- und Rollenkonflikte als widersprüchliche Interessenstandpunkte, die entweder institutionell vorgegeben sind, oder aus den Rollenerwartungen der Beteiligten am Entscheidungsprozess erwachsen;
- Verteilungs- oder Ressourcenkonflikte durch widersprüchliche Forderungen der Interaktionspartner bei der Zuteilung von knappen Ressourcen;
- Machtkonflikte aus widersprüchlichen Machtansprüchen der Interaktionspartner – mit anderen Worten Konflikte bei der Forderung nach Zuteilung von Potentialen zur jeweils vorteilhaften Lösung zukünftiger Konflikte.

Die Konflikttypologie verdeutlicht auch, dass die Koordination einer Innovationstätigkeit als Konfliktmanagement aufzufassen ist und ein höchst differenziertes Instrumentarium benötigt. Jede der genannten Konfliktarten kann negative Auswirkungen auf die Erreichung von aus Sicht der Interaktionspartner übergeordneten Ziele haben. Die Koordination der Innovationstätigkeit als Einflussnahme auf die Interaktion und Kommunikation hat daher die Aufgabe, „die Beteiligten und Betroffenen einer Innovation auf ein gemeinsames Ziel auszurichten und die dabei auftretenden Konflikte vorausschauend oder nachträglich zu regulieren.“³¹⁶

Wie beim Aspekt der kontinuierlichen Zielbildung findet sich auch das bei innovativen Vorhaben zutreffende Bild der Koordination als Konfliktmanagement in der Softwareentwicklung im Allgemeinen wieder. Konflikte spielen hier eine bedeutende Rolle, obgleich sie von den existierenden Softwareentwicklungsmethoden im Umfeld des Projekt- und Anforderungsmanagements schlecht gehandhabt werden. Konfliktbehandlung und -auflösung werden gar bei den meisten Methoden nicht explizit adressiert.

³¹⁵ Vgl. Hauschildt (2004), S. 127

³¹⁶ Hauschildt (2004), S. 127

Softwareentwicklung ist aber ein kollaborativer Prozess und die Identifizierung und Behandlungsweise von gemeinsamen und entgegengesetzten Zielsetzungen der Projektbeteiligten ist Voraussetzung für den Projekterfolg.³¹⁷ Das Abstimmen und Treffen von Entscheidungen muss stets mit Bezug zu den widersprüchlichen Zielen der Projektbeteiligten stattfinden. Das Verständnis und der Einsatz von Techniken zur Konflikt-handhabung und Entscheidungsfindung unter Mehrfachzielsetzung sind zwingend notwendig.³¹⁸

Insbesondere müssen die Koordinationsmechanismen der Softwareentwicklung die Abstimmung der widersprüchlichen Ziele und Interessenslagen der Projektbeteiligten sicherstellen. Sie betrachten das Projekt aus ihrer jeweiligen Perspektive, was zwangsläufig zu Konflikten führt.³¹⁹ Bspw. haben die verschiedenen Projektbeteiligten wie Projektleiter, Auftraggeber, Qualitätsmanagement, Entwicklungsabteilung, etc. ein gänzlich anderes Verständnis vom Begriff der Qualität.³²⁰ Eine besondere Stellung kommt demnach bei der Koordination der Softwareentwicklung den Ressort- und Rollenkonflikten als widersprüchliche Interessenstandpunkte zu. Aus diesen ergeben sich wiederum v.a. Verteilungs- und Ressourcenkonflikte durch widersprüchliche Forderungen der Interaktionspartner bei der Zuteilung von knappen Ressourcen, was das sog. Teufelsquadrat der Softwareentwicklung gut verdeutlicht.

Die an den Ecken des Vierecks in Abb. 28, dem sog. Teufelsquadrat der Softwareentwicklung, angetragenen Projektziele konkurrieren um die auf kurzfristige Sicht konstante verfügbare Produktivität resp. um die endlichen Ressourcen eines Softwareentwicklungsprojektes. Das bedeutet, dass in einem Projekt bspw. die Entwicklungskosten nicht vermindert werden können, ohne dass sich daraus ein geringerer Zielerfüllungsgrad bei der Qualität, beim Leistungsumfang oder in der Entwicklungszeit ergibt und umgekehrt.³²¹ Bei der ‚Navigation im Teufelsquadrat‘ werden die Projektbeteiligten immer eine ihren Interessenstandpunkten entsprechende Entscheidung anstreben.³²² Konflikte ergeben sich somit nicht nur zu Beginn des Entwicklungsvorhabens, sondern auch im Projektverlauf bei der immer wiederkehrenden Frage, ‚in welche

³¹⁷ Vgl. Grünbacher, Seyff (2005), S. 143

³¹⁸ Vgl. Boehm (1981), S. 205

³¹⁹ Die Konflikte ergeben sich zum großen Teil auch aus den organisatorischen Besonderheiten der Matrix-Projektorganisation, die sich hinsichtlich der Interessenslagen und Weisungsbefugnisse auf die Organisation innerhalb der Projekte übertragen. Die Matrix-Projektorganisation hat ein sehr hohes inhärentes Potential für Kompetenzkonflikte, sodass das Gleichgewicht von Zeit, Kosten und Leistungsumfang hier besonders überwacht werden muss. Vgl. Kerzner (2003), S. 91f.; vgl. Henrich (2002), S. 130

³²⁰ Vgl. Boehm, Ross (1989); vgl. Wong (2005)

³²¹ Vgl. Balzert (2001), S. 76-78

³²² Vgl. Herzwurm, Pietsch (2009), S. 238

Richtung im Teufelsquadrat³²³ der Planungsfehler (optimistische Planung) korrigiert werden soll. Hierunter ist der kontinuierliche Zielbildungsprozess bei der Softwareentwicklung zu verstehen.

Beim kontinuierlichen und konfliktären Zielbildungsprozess in der Standardsoftwareentwicklung lassen sich v.a. drei am Projekt beteiligte zentrale Rollen mit konfliktären Interessenstandpunkten festmachen: Das Projektmanagement und ‚seine‘ Entwicklungsabteilung als Kern des Projektteams, das Qualitätsmanagement sowie das Produktmanagement (vgl. Abb. 28).

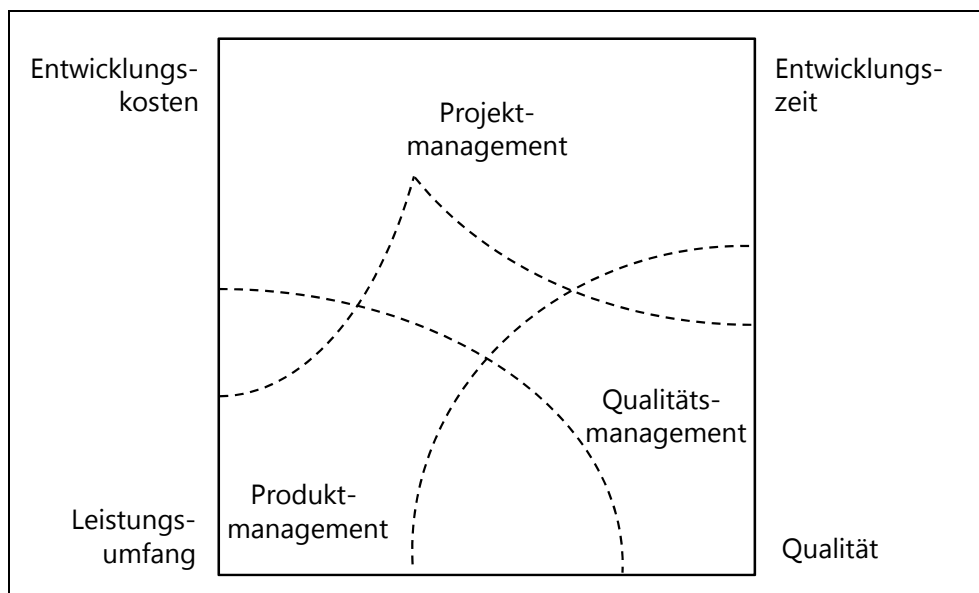


Abb. 28: Zielrichtungen der Beteiligten im ‚Teufelsquadrat der Softwareentwicklung‘³²³

Bei der Matrix-Projektorganisation als übliche Eingliederungsform von Projekten bei der Standardsoftwareentwicklung werden dem Projektleiter bzw. dem Projektmanagement nicht alle notwendigen Ressourcen fest zugewiesen, sondern nur solche, die während der Projektdauer vollständig ausgelastet sind. Zur Erfüllung der Projektaufgabe muss der Projektleiter jedoch in Abstimmung mit den zuständigen Leitungsstellen auf weitere Ressourcen aus der ständigen Organisation zurückgreifen.³²⁴ Zu den fest zugeordneten Ressourcen zählt die Entwicklungsabteilung, welcher der Projektleiter gewöhnlich auch selber zugeordnet ist und oft vorsteht, jedoch nicht das Qualitätsmanagement und das Produktmanagement, letzteres als interner Auftraggeber. Entsprechend neigen Projektleiter dazu, Kosten- und Terminziele am stärksten zu priorisieren, v.a. stärker als Qualitätsziele.³²⁵

Empirische Untersuchungen belegen, dass in einem Entwicklungsprojekt, das den geplanten Fertigstellungstermin nicht einhalten kann, sog. ‚Abkürzungen‘ genommen

³²³ In Anlehnung an Herzwurm, Pietsch (2009), S. 238

³²⁴ Vgl. Mellis (2004), S. 169

³²⁵ Vgl. Friedrich u.a. (2008), S. 89f.

werden. ‚Abkürzungen‘ sind Entscheidungen, die von den Entwicklern und/oder dem Projektleiter ohne Abstimmung mit der vorgesetzten Rolle (Projektleiter bzw. Projektprogrammleiter) getroffen werden, um einen vorgegebenen Termin zu halten. Dabei sind diese Entscheidungen für das übergeordnete Ziel (Projektziele im Zusammenspiel und Projektprogrammziele) nicht vorteilhaft. Zu den ‚Abkürzungen‘ gehört allen voran die Erreichung von Terminvorgaben auf Kosten der Nicht-Erreichung von Qualitätszielen. Die (Nicht-)Erreichung von Qualitätszielen ist nämlich im Vergleich zu der (Nicht-)Termineinhaltung wesentlich schwieriger zu messen und auf konkrete Projektmanagement-Fehler und/oder Fehler der Entwicklungsabteilung zurück zu verfolgen.³²⁶ Daher kann das Qualitätsmanagement bzw. ein Qualitätsverantwortlicher im Projekt als der ‚Gegenspieler‘ des Projektleiters bezeichnet werden. Dieser ist für die Erfüllung der Qualitätsziele verantwortlich. Im Zusammenspiel mit dem Projektleiter werden so beide Projektziele mit einem Zwang zum Konsens und ausgeglichener Priorität verfolgt. In diesem Zusammenhang ist auch die Forderung des V-Modell XT verständlich, die eine personelle Trennung der Rollen des Projektleiters und des Qualitätsverantwortlichen im Projekt vorschreibt.³²⁷

Das Produktmanagement wird in einer funktional ausgerichteten Unternehmung meistens in das Marketing integriert, da dieses nahe am Markt und dem Kunden ausgerichtet ist und Komplementaritäten mit den Zielen des Produktmanagements hat.³²⁸ Die Spannbreite der Tätigkeiten des Produktmanagements ist bei Standardsoftwareunternehmungen insgesamt enorm.³²⁹ An dieser Stelle sind jedoch nur die Tätigkeiten des Produktmanagements im Rahmen seiner Produktverantwortung bei Entwicklungsprojekten relevant.

Dem Produktmanagement wird die interne Auftraggeber-Rolle für die Entwicklung der Standardsoftware übertragen. Er vertritt in einem Entwicklungsprojekt ‚sein Produkt‘, das neu erstellt oder verbessert werden soll. Im Unterschied zu den Interessen des Projektleiters erstreckt sich der Betrachtungshorizont des Produktmanagers auf den mittel- bis langfristigen Erfolg des Softwareproduktes, also über die Dauer eines Entwicklungsprojektes hinaus. Die Verantwortung des Projektleiters begrenzt sich dagegen nur auf die Dauer des Entwicklungsprojektes.³³⁰ Je genauer ein Softwareprodukt den Anforderungen entspricht, die der Markt stellt, desto größer ist das Produktpotential. Aus dieser mittel- bis langfristigen Interessenslage heraus verfolgt der Produkt-

³²⁶ Vgl. Austin (2001), S. 195

³²⁷ Vgl. Friedrich u.a. (2008), S. 89f.

³²⁸ Vgl. Herzwurm, Pietsch (2009), S. 69f.; vgl. Kittlaus u.a. (2004), S. 150

³²⁹ Vgl. Kittlaus u.a. (2004), S. 148, 153f.; vgl. Herzwurm, Pietsch (2009), S. 45ff.

³³⁰ Vgl. Kittlaus u.a. (2004), S. 132, 139f.

manager die Erbringung der definierten Entwicklungsinhalte als Menge der zu entwickelnden Funktionen und Features im Sinne der Kundenorientierung. Dabei können sich Anforderungen, die im Vorfeld definiert wurden, aufgrund von neuen oder geänderten Kundenwünschen und Marktgegebenheiten ändern.³³¹

2.3.2.2 Koordinationsphasen und -prozesse in der Standardsoftwareentwicklung

Wie in Kap. 2.1.2.1 ausgeführt zeichnet sich der Softwareentwicklungsprozess durch mehrere fachlich-technische Phasen aus. Die Phaseneinteilung wird in das Projektmanagement übernommen, sodass sich schlussendlich auch Softwareentwicklungsprojekte in Projektphasen einteilen lassen. Ein prozessorientiertes Phasenmodell ist besonders für komplexe und umfangreiche Projekte typisch, auch über die Softwareentwicklung hinaus. Im prozessorientierten Projektphasenmodell wechseln sich Ausführungsphasen und Koordinationsphasen ab. Projektkoordinationsphasen sind dabei meistens mit dem Start oder mit dem Ende einer inhaltlichen Ausführungsphase verknüpft.

Sowohl in Abb. 25 zum Konus der Unsicherheit, als auch in Abb. 26 zur Einschränkung des Lösungsraumes ist eine Phasenorientierung zu erkennen. Insbesondere die Projektkoordinationsphasen kennzeichnen jeweils die Übergänge zwischen den beiden Abstraktionsebenen Problemdefinition und Lösungsbeschreibung. Sie sind damit im zuvor beschriebenen Verständnis des Softwareentwicklungsprozesses als ein kontinuierlicher und konfliktärer Zielbildungsprozess ein zentraler Ansatzpunkt für Gestaltungsmaßnahmen der Koordination: Im Laufe einer Koordinationsphase wird eine inhaltliche Ausführungsphase abgeschlossen, indem

- die Zwischenergebnisse der Ausführungsphase zusammengeführt und überprüft werden,
- Steuerungsmaßnahmen gegen Abweichungen und Änderungen vom Plan eingeleitet werden, und schließlich
- die erzielten Ergebnisse in die nächste inhaltliche Phase transferiert werden, incl. der Formulierung von Zielen und Strategien für die nächste Phase.³³²

Abb. 29 stellt ein aus Sicht der Standardsoftwareentwicklung vereinfachtes Abbild des prozessorientierten Projektphasenmodells dar. Denn im Unterschied dazu überlappen

³³¹ Vgl. Herzwurm, Pietsch (2009), S. 50, 58f., 71f.; Die Aufgabenbereiche des IT-Produktmanagers können im Detail von Unternehmen zu Unternehmen verschieden sein - je nachdem, wie er organisatorisch eingebunden ist und welche Aufgabenbereiche ihm zugeordnet werden. Herzwurm und Pietsch definieren in diesem Zusammenhang vier Produktmanagement-Typen und verdeutlichen anhand einer ‚IT-Produktmanagement-Pyramide‘ deren teilweise unterschiedliche Wirkungs- und Interessensbereiche sowie die unterschiedlichen Ebenen, auf welchen sie tätig sein können. Vgl. Herzwurm, Pietsch (2009), S. 60

³³² Vgl. Patzak, Rattay (2004), S. 25f., 354f.

sich bei Entwicklungsvorhaben im Bereich Standardsoftware Ausführungsphasen sowohl untereinander, als auch vollständig mit Koordinationsphasen. Letztere finden neben festen Zeitpunkten auch situativ je nach Projektverlauf statt. Desweiteren ist nicht von einem, sondern von mehreren Projekten im Zusammenspiel (Projektprogramm) auszugehen. Koordinationsphasen beziehen sich daher sowohl auf die Koordination innerhalb von Projekten, als auch auf die Koordination zwischen den Projekten im Projektprogramm.³³³

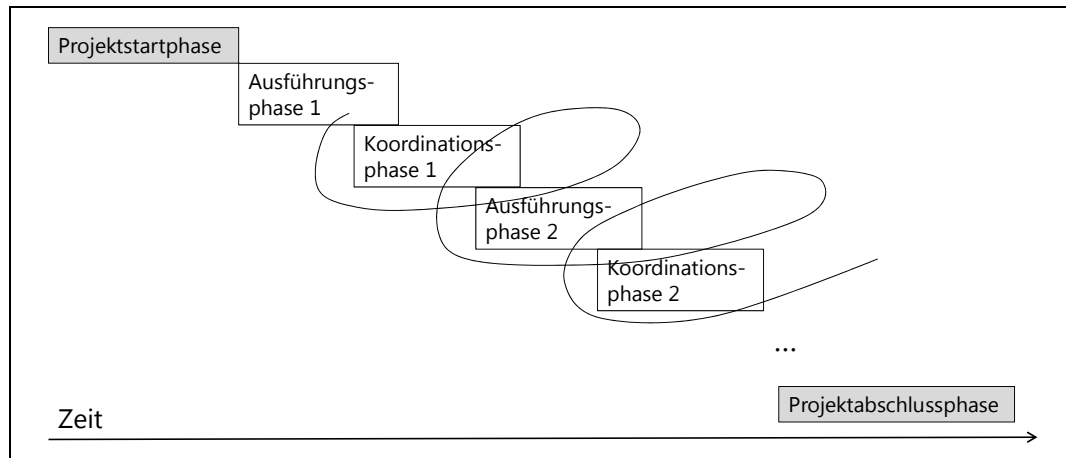


Abb. 29: Prozessorientiertes Projektphasenmodell³³⁴

Im prozessorientierten Projektphasenmodell wird allgemein von Aufgaben und Sitzungen in Koordinationsphasen gesprochen.³³⁵ Das Modell geht nicht näher darauf ein, innerhalb welcher primären oder unterstützenden Teilprozesse diese stattfinden. Bei der Klärung dieser Frage muss man allen voran feststellen, dass es nicht den einen bestimmten Koordinationsprozess in der Standardsoftwareentwicklung gibt. Zum (abstrakten) Koordinationsprozess müssen vielmehr alle primären und unterstützenden Teilprozesse gezählt werden, die eng mit der Abstimmung interdependenter, abweichungsinduzierter Entscheidungen auf ein übergeordnetes Ziel hin verknüpft sind. Bei der Standardsoftwareentwicklung gehören hierzu die unterstützenden phasenübergreifenden Teilprozesse Projektsteuerung sowie Risiko-, Änderungs- und Qualitätsmanagement.

Ziel der Projektsteuerung ist es, stets einen Überblick über den Projektfortschritt zu haben, um gegebenenfalls geeignete Korrekturmaßnahmen einleiten zu können. Im Zuge der Projektumsetzung ergeben sich Ist-Daten zum Projektfortschritt, die den geplanten Soll-Werten gegenübergestellt werden. Auf dieser Grundlage werden auch Prognosen erstellt, um abschätzen zu können, ob die ursprünglichen Soll-Werte auch zum Projektende noch erreichbar erscheinen. Ergeben sich bei diesen Vergleichen

³³³ Vgl. Lühring (2006), S. 77f.

³³⁴ Patzak, Rattay (2004), S. 25

³³⁵ Vgl. Patzak, Rattay (2004), S 354f.

Abweichungen, so gibt es je nach Art und Stärke der Abweichung drei generelle Möglichkeiten, wie Entscheidungen zugunsten korrigierender Steuerungsmaßnahmen und deren Durchführung ergriffen werden können.³³⁶

- Sind die Abweichungen gering, werden üblicherweise mit Hilfe einer ‚Offene-Punkte-Liste‘ zusätzliche Aktivitäten spezifiziert und überwacht;
- Sind die Abweichungen größer, müssen die Projektplanung wieder aufgenommen werden und Korrekturmaßnahmen auf der Einzelprojektebene und evtl. auch auf der Projektprogrammebene erfolgen;
- Bei sehr gravierenden Abweichungen muss evtl. eine Änderung der Gesamtplanung oder im Extremfall der Projektabbruch erfolgen.

An dieser Stelle wird deutlich, wie stark die Projektumsetzung mit der Projektplanung und -kontrolle verknüpft ist. Letzteres ist immer im Sinne der Auslösung von Anpassungsmaßnahmen zu verstehen. Die angesprochenen Soll-Ist- und Soll-Wird-Vergleiche beziehen sich dabei auf alle vier Projektzieldimensionen. Diese stehen stellvertretend für die Gesamtheit aller operativen Projektziele, die den jeweiligen Dimensionen zuzuordnen sind.³³⁷ Anpassungsmaßnahmen resultieren also stets aus abweichungsinduzierten Entscheidungen mit Auswirkungen auf die Zielerreichung in den vier Zieldimensionen.

Als Koordinationsinstrumente stehen bei der Projektsteuerung vor allem Projektberichte und projektspezifische Kennzahlen sowie auf dieser Grundlage stattfindende Projektreviews zur Verfügung. Projektberichte und projektspezifische Kennzahlen sind Arbeitsergebnisse (Artefakte) des Projektinformationsmanagements, welches der Bereitstellung und dem Austausch von projektrelevanten Daten dient, sowohl innerhalb eines Projektes als auch darüber hinaus. Somit schafft das Projektinformationsmanagement die Grundlage für die Koordination der Projektaufgaben.³³⁸

In einem Projektreview (auch Statussituation) wird der aktuelle Projektstatus in Bezug zum geplanten Status gesetzt. Projektreviews finden entweder regelmäßig, bspw. wöchentlich oder monatlich, oder zu bestimmten Ereignissen wie bspw. dem Erreichen eines Meilensteines statt. Üblich ist des Weiteren neben dem regulären Turnus die Einberufung von Statussituationen bei besonderen Anlässen wie bspw. Eskalationen. Das Ergebnis einer Statussituation sind ggf. Entscheidungen zugunsten korrigierender Steuerungsmaßnahmen.³³⁹ V.a. bei Projekten, die Teil eines Projektprogrammes sind

³³⁶ Vgl. Jacobs (2010); vgl. Bea u.a. (2008), S. 249f.

³³⁷ In Anlehnung an Bea u.a. (2008), S. 279

³³⁸ Vgl. Bea u.a. (2008), S. 249-252

³³⁹ Vgl. Schelle u.a. (2005), S. 284

und damit zahlreiche Interdependenzen zu den anderen Projekten aufweisen, finden die Projektsteuerung und damit Statussitungen auf mehreren Ebenen statt. In diesem Zusammenhang sind Statussitungen bzw. Projektreviews eng mit der Bildung von sog. Kommissionen verknüpft.

Bei einem Projekt werden zumindest ein Lenkungsausschuss sowie ggf. Beratungskommissionen gebildet. Der Lenkungsausschuss, auch Koordinationsgremium oder ‚steering committee‘ genannt, ist eine Entscheidungskommission und dient der Projektkoordination und Konfliktregulierung. Bei innovativen Prozessen spielen neben dem Lenkungsausschuss Beratungskommissionen eine zentrale Rolle. In jedem Innovationsprozess gibt es unterhalb des Lenkungsausschusses eine bestimmte Anzahl von Beratungskommissionen. Ihre Aufgabe besteht ganz allgemein darin, ein Problem zu analysieren und einen Lösungsvorschlag zu erarbeiten. Die Beratungskommission hat jedoch keine Kompetenz, ihre Vorschläge bindend festzulegen. Sie hat sie vielmehr einem anderen Organ, meistens dem Lenkungsausschuss, zur Entscheidung vorzulegen.³⁴⁰

Der Kern des Änderungsmanagements liegt in der Festlegung und Einhaltung einer systematischen Vorgehensweise für die Freigabe, Überwachung und Steuerung von Änderungen, die durch zahlreiche Anlässe während des Projektverlaufes ausgelöst werden können. Auslöser für notwendige Änderungen können dabei bspw. vom Markt ausgehen, da sich die Anforderungen des Marktes im Laufe der Entwicklung mehrmals ändern können. Pohl systematisiert die möglichen Arten von Änderungen bei Anforderungen und kommt dabei zu den fünf Änderungsarten ‚Integration einer neuen Anforderung‘ sowie ‚Löschen; Ergänzen; Reduktion; Veränderung einer bestehenden Anforderung‘. Dabei kann es sich jeweils um korrektive Änderungen aufgrund eines Fehlverhaltens des Softwaresystems oder um adaptive Änderungen aufgrund von Veränderungen im Kontext des Softwaresystems handeln.³⁴¹

Anforderungsänderungen müssen identifiziert, beschrieben, klassifiziert, bewertet, genehmigt, durchgeführt und verifiziert werden. Die genannten Prozessschritte verlangen ein systematisches, kontrolliertes und dokumentiertes Änderungsverfahren, bei dem alle notwendigen Informationen erfasst werden. Dessen Grundlage bildet ein Änderungsantrag, der die wichtigsten Daten zur jeweiligen Änderung beinhaltet und so eine genaue Betrachtung der Auswirkungen der Änderung ermöglicht. Ein Änderungsantrag enthält konkret

³⁴⁰ Vgl. Hauschildt (2004), S. 136-140

³⁴¹ Vgl. Pohl (2007), S. 552-554

- eine Kurzbeschreibung der Änderung mit der Darstellung aller möglichen Alternativen und einer Empfehlung für die Auswahl der aus Sicht des Beantragenden sinnvollsten Alternative, sowie
- eine Analyse der Auswirkungen, die sich durch die Umsetzung der Anforderungsänderung auf die vier Projektzieldimensionen sowie auf die Schnittstellen zu anderen Arbeitspaketen und Projekten ergeben.³⁴²

Das Qualitätsmanagement wird teilweise als ein begleitender Prozess des Projektmanagements angesehen. Aber auch die umgekehrte Ansicht wird vertreten: „Definiert man Leistung, Termine und Kosten als die Qualitätsparameter in der Projektarbeit, dann lässt sich Projektmanagement als ein Qualitätsmanagement-System für die Projektarbeit bezeichnen.“³⁴³ In diesem Sinn können die bei der Projektsteuerung angesprochenen Projektreviews (Statusitzungen) dem Qualitätsmanagement zugeordnet werden.³⁴⁴ Unabhängig von dieser Zuordnung müssen Projektreviews von Produktreviews unterschieden werden. Zweiteres stellt die zentralen Koordinationsphasen innerhalb der Qualitätslenkung dar. Bei Produktreviews wird ein Softwaredokument³⁴⁵ zeitnah zu seiner Erstellung in einer Gruppensitzung mit definierten Rollen auf Fehler hin untersucht. Im Ergebnis lösen Produktreviews ggf. korrigierende Steuerungsmaßnahmen aus.

Über Produktreviews hinaus setzt die Qualitätslenkung ganz allgemein die Qualitätsplanung um. D.h. im Rahmen der Qualitätslenkung wird der Entwicklungsprozess gesteuert, überwacht und korrigiert. Ziel ist es sicherzustellen, dass die vorgegebenen Anforderungen an das Softwareprodukt und an den Softwareentwicklungsprozess erfüllt werden. Ist das nicht der Fall, müssen korrektive Maßnahmen ergriffen werden. An dieser Stelle sind die Aufgaben der Qualitätslenkung eng mit dem Projektmanagement verknüpft. Denn um eine frühestmögliche Fehlerentdeckung und -behebung realisieren zu können und jederzeit Sichtbarkeit über das Produkt- und Prozessqualitätsniveau zu haben, ist eine durchgängig entwicklungsbegleitende und in das Projekt- und Risikomanagement integrierte Qualitätslenkung notwendig. Nach diesem Verständnis sind auch die traditionell getrennt behandelten Aktivitäten der Qualitätslenkung und -prüfung nicht mehr sinnvoll trennbar.³⁴⁶

³⁴² Vgl. Bea u.a. (2008), S. 266f.

³⁴³ Schelle u.a. (2005), S. 245

³⁴⁴ Vgl. Schelle u.a. (2005), S. 261

³⁴⁵ Bspw. Anforderungen, Konzepte, Szenarien, Design, Code

³⁴⁶ Vgl. Herzwurm, Mikusz (2010b); vgl. Balzert (2008), S. 480, 488

„Das Risikomanagement legt fest, wie laufend mit Risiken als potenziellen Störungen der Zielerreichung umgegangen werden soll, um ihren negativen Einfluss auf das übergeordnete Ziel zu minimieren.“³⁴⁷ Mithilfe des Risikomanagements werden die Unsicherheiten im Projektverlauf identifiziert, um sie zu bewerten und umzugehen, oder um bei ihrem Eintreten auf sie reagieren zu können. Bei Projekten kann das Risikomanagement als eine integrierte Teildisziplin des Projektmanagements angesehen werden. Das Risikomanagement ist dabei kein einmaliger Prozess, sondern muss kontinuierlich über den gesamten Projektlebenszyklus hinweg durchgeführt werden.³⁴⁸ Um nochmals auf das ‚Teufelsquadrat der Softwareentwicklung‘ zurückzukommen: Wegen der immer bestehenden Unsicherheit für die Zielgrößen der Projektsteuerung, verbunden mit den potenziellen Schäden, sind grundsätzlich alle vier Projektzieldimensionen in ein Risikomanagement einzubetten.³⁴⁹

Beim Risikomanagementprozess werden die Phasen Risikoidentifikation, -analyse, -bewertung, -steuerung und -überwachung unterschieden. Institutionell ist das Risikomanagement größtenteils in das Projektmanagement eingebunden. Es finden jedoch auch wesentliche Aktivitäten, allen voran Risiko-Workshops (als Koordinationsphasen) in der Startphase sowie auch in späteren Projektabschnitten, außerhalb des Projektmanagements statt.³⁵⁰ Zahlreiche domänenspezifische Beispiele von Risikomanagementprozessen in der Softwareentwicklung finden sich bei Wallmüller.³⁵¹

2.3.2.3 Institutionalisation von Koordinationsprozessen

Die bisherige Charakterisierung des Standardsoftwareentwicklungsprozesses als ein kontinuierlicher und hoch konfliktärer Zielbildungsprozess verdeutlicht auch nochmal, warum hier die optimale Koordinationseffizienz tendenziell bei einer sehr hohen Koordinationsintensität mit Schwerpunkt auf der Feedbackkoordination liegt.³⁵² Darüber hinaus stellt sich aber die Frage, wie sichergestellt werden kann, dass die im Projektverlauf laufend stattfindenden Entscheidungen in den Koordinationsprozessen Projektsteuerung sowie Risiko-, Änderungs- und Qualitätsmanagement im Sinne der dem Projekt übergeordneten Produkt- und Projektprogrammziele getroffen werden. Die potentiellen Lösungsansätze zu dieser Fragestellung müssen die zwei folgenden Bedingungen erfüllen, da ansonsten dem Charakter der Standardsoftwareentwicklung als ein kontinuierlicher und konfliktärer Zielbildungsprozess nicht entsprochen wird:

³⁴⁷ Schelle u.a. (2005), S. 120

³⁴⁸ Vgl. Wallmüller (2004), S. 52

³⁴⁹ In Anlehnung an Patzak, Rattay (2004), S. 149

³⁵⁰ Vgl. Schelle u.a. (2005), S. 152ff.

³⁵¹ Vgl. Wallmüller (2004), S. 119-125

³⁵² Vgl. Kap. 2.3.1.1

- Die Entscheidungen im Projektverlauf dürfen nicht vorweggenommen werden, d.h. aus der Feedbackkoordination im Sinne der Reaktion auf Störungen darf nicht eine ausschließliche Vorkoordination vor dem Beginn der Aufgabenerfüllung entstehen;
- Die Selbstabstimmung der Projektbeteiligten untereinander darf nicht grundsätzlich durch eine hierarchische Entscheidungsfindung bzw. persönliche Weisung durch hierarchisch übergeordnete Einheiten ersetzt werden.

Die beiden Bedingungen lassen sich in etwa mit den Vor- und Nachteilen der Koordination durch Selbstabstimmung (als Feedbackkoordination) gleichsetzen. Somit geht es anders ausgedrückt um die Erhaltung der Vorteile bei gleichzeitiger Eliminierung der Nachteile der Koordination durch Selbstabstimmung:³⁵³

- Der wesentliche Vorteil der Koordination durch Selbstabstimmung ist, dass sie eine unkomplizierte und flexible Reaktion auf unvorhergesehene Ereignisse erlaubt. Denn hier werden einerseits nur die unmittelbar vom Koordinationsbedarf Betroffenen involviert und daher unnötige Abstimmungsgesichtspunkte vermieden, sowie andererseits die Detailkenntnisse der Beteiligten unkomplizierter und direkter angewendet. Dadurch werden auch hierarchisch höher stehende Einheiten von der Aufgabe der Koordination entlastet und die vertikale Kommunikation reduziert. Wegen des großen Umfangs der Feedbackkoordination und der häufig für die koordinierenden Entscheidungen notwendigen detaillierten technologischen Kenntnisse wäre hier eine zentrale koordinierende Stelle ohnehin überlastet, was verzögerte Entscheidungen und eine geringe Entscheidungsqualität zur Folge hätte;
- Koordination durch Selbstabstimmung setzt allerdings voraus, dass die Organisationsmitglieder in der Lage sind, den Koordinationsbedarf selbständig zu erkennen und angemessen darauf zu reagieren. Der zweite wesentliche Nachteil der Koordination durch Selbstabstimmung ist, dass sie v.a. auf Verhandlungen unter Gleichgestellten basiert und daher in der Regel mit einem hohen Zeitbedarf verbunden ist.

Die Selbstabstimmung als Koordinationsinstrument ist von einem unverbindlichen Informationsaustausch zu unterscheiden, welcher schon allein zur Sicherung eines guten persönlichen Verhältnisses zwischen den Beteiligten praktisch immer, auch bei den anderen Koordinationsmechanismen, zu beobachten ist. Von einer Koordination durch Selbstabstimmung wird erst dann gesprochen, wenn sie offiziell vorgesehen ist und die Entscheidungen für alle Beteiligten auch verbindlich sind. In diesem Rahmen kann die Feedbackkoordination durch Selbstabstimmung der Eigeninitiative der betroffenen

³⁵³ Vgl. Mellis (2004), S. 146f.

Rollen überlassen bleiben oder nach expliziten, strukturellen Regelungen, die die Organisation zur Unterstützung der Selbstkoordination definiert, erfolgen.³⁵⁴

Strukturelle Regelungen können sich auf die Einrichtung von Kommunikations- und Informationskanälen, Bildung von Koordinationsorganen und ihre Ausstattung mit Entscheidungskompetenzen, Vorgabe von Anlässen für Koordinationsmaßnahmen bzw. Spezifikation von abstimmungsbedürftigen Fragen etc. erstrecken. In der Festlegung solcher struktureller Regelungen liegt ein erster Ansatzpunkt für die vorliegende Fragestellung. Unter Berücksichtigung der zahlreichen Möglichkeiten hierzu lassen sich die folgenden drei Arten der Selbstabstimmung unterscheiden.³⁵⁵

- Fallweise Interaktion nach eigenem Ermessen: Hier sind keine spezifizierten strukturellen Regelungen für die Selbstabstimmung getroffen. Die Selbstabstimmung bleibt dem eigenen Ermessen der Betroffenen überlassen;
- Themenspezifische Interaktion: Durch Regeln wird festgelegt, zu welchen abstimmungsbedürftigen Fragen die (Feedbackkoordination durch) Selbstabstimmung erfolgen muss. Somit wird in bestimmten Fällen die Abstimmung zur Pflicht. Ob ein solcher Fall vorliegt oder nicht, hängt also nicht mehr vom Ermessen der einzelnen Organisationsmitglieder ab, sondern wird durch generelle Regelungen definiert. So wird das Risiko reduziert, dass notwendige Abstimmungen unterbleiben;
- Da sowohl die themenspezifische Interaktion als auch die fallweise Interaktion nach eigenem Ermessen die Abstimmung den Beteiligten überlassen, müssen in einer weiteren Auslegung auch Regelungen zur Herbeiführung von Entscheidungen und zur Konfliktlösung eingerichtet werden. Damit weist die Koordination nochmals einen höheren Strukturierungsgrad auf und wird als institutionalisierte Interaktion bezeichnet.

Mit dem Übergang von der fallweisen Interaktion nach eigenem Ermessen in die themenspezifische und weiter in die institutionalisierte Selbstabstimmung steigt auch der Grad an struktureller Regulierung der Koordination und damit die Menge der potentiellen Möglichkeiten, Entscheidungen im Projektverlauf im Sinne der dem Projekt übergeordneten Produkt- und Projektprogrammziele zu steuern. Offensichtlich werden dabei die Entscheidungen an sich nicht vorweggenommen. Es erfolgt also keine Vorauskoordination, gewissermaßen jedoch eine Vorauskoordination der Feedbackkoordination. Hierfür werden konkrete Anknüpfungspunkte in Kap. 3.3 entwickelt. Als konzeptionelle Grundlage dient dabei das im Folgenden dargestellte Strukturierungskonzept der entscheidungsorientierten Organisationslehre nach Laux.

³⁵⁴ Vgl. Mellis (2004), S. 145; vgl. Kieser, Walgenbach (2003), S. 112

³⁵⁵ In Anlehnung an Mellis (2004), S. 145; in Anlehnung an Kieser, Walgenbach (2003), S. 112-114

Entscheidungen, die der eigentlichen Entscheidung, also der Wahl einer Handlungsalternative, vorgelagert sind, werden als Meta-Entscheidungen bezeichnet. Meta-Entscheidungen bestimmen letztlich die eigentliche Entscheidung. Daher können Entscheidungen anderer Organisationsmitglieder gerade dadurch beeinflusst und gesteuert werden, indem auf die Meta-Entscheidungen der Entscheidungsträger Einfluss genommen wird.³⁵⁶ Solche Meta-Entscheidungen als Steuerungsmaßnahmen können sich auf das Verhalten eines Entscheidungsträgers nur unter der Voraussetzung auswirken, dass mindestens eine der Determinanten seiner Entscheidung verändert wird. D.h. die Beeinflussung des

- Typs des Entscheidungsmodells, der
- Menge der erwogenen Handlungsalternativen, der
- Informationsstruktur, der
- Prognosefunktion, der
- Zielfunktion sowie der
- Ergebnisse der Alternativen

sind als die Zielkomponenten der Steuerung von Entscheidungen nachgeordneter Mitarbeiter zu sehen.³⁵⁷

Durch die Beeinflussung der Wahl eines Entscheidungsmodells kann Einfluss darauf genommen werden, durch welches Modell der Entscheider die eigentliche Entscheidungssituation abbildet. So kann hier durch generelle Regelungen wie bspw. Investitionshandbücher festgelegt werden, welche Entscheidungsmodelltypen (bspw. die Kapitalwertmethode) anzuwenden sind. Zudem kann der anzustrebende Komplexitätsgrad von Entscheidungsmodellen vorgegeben werden.

Bei der Beeinflussung der Menge der erwogenen Handlungsalternativen sind zwei Arten von Maßnahmen zu unterscheiden. Zum einen kann durch eine direkte Begrenzung der Verfügungsmöglichkeiten über Ressourcen die Menge jener Handlungsalternativen bestimmt werden, die die Entscheidungsträger realisieren können. Zum anderen kann durch die Vorgabe von Verhaltensnormen und durch weitere ergänzende Maßnahmen die Menge der Handlungsalternativen beeinflusst werden, die von den nachgeordneten Mitarbeitern in ihren jeweiligen Entscheidungskalkülen erwogen werden. Hier wird also Einfluss auf die Chancen von Handlungsalternativen genommen, gewählt und realisiert zu werden. Die Vorgabe von Verhaltensnormen kann wiederum

³⁵⁶ Vgl. Laux, Liermann (2003), S. 36

³⁵⁷ Vgl. auch im Folgenden Laux, Liermann (2003), S. 132-144, wobei hier auf eine separate Betrachtung von Objekt-, Steuerungs- und Kommunikationsentscheidungen als die zu steuernden Entscheidungen verzichtet wird.

auf zwei Wirkungsweisen abzielen: Verhaltensnormen können dazu dienen, die Menge um die Handlungsalternativen zu erweitern, die vom Standpunkt der Organisation zwar vorteilhaft sind, aber von dem Entscheidungsträger nicht in seinem Kalkül berücksichtigt würden, weil sie ihm bspw. nicht bekannt sind. Analog können Verhaltensnormen dazu dienen, die Menge der erwogenen Handlungsalternativen um solche Alternativen zu reduzieren, die vom Standpunkt der Organisation nachteilig sind, aber möglicherweise trotzdem aufgrund von beschränkter Rationalität von einem Entscheidungsträger gewählt werden könnten.

Die Beeinflussung der Informationsstruktur von Entscheidungsträgern erfolgt, damit diese die Konsequenzen der Wahl von relevanten Alternativen besser beurteilen können. Dies erfolgt insgesamt über die Manipulation der Möglichkeiten und Grenzen der Informationsbeschaffung durch die Entscheidungsträger. Hier können bspw. generelle Regelungen vorgegeben werden, welche Informationen bei welchen Entscheidungssituationen einzuholen sind, welche Informationen in welchen Fällen an andere Entscheidungsträger weiterzugeben sind (ggf. in verdichteter Form), welche zusätzlichen Informationen zur Prognose der Konsequenzen von Entscheidungen beschafft werden müssen, etc.

Die Beeinflussung der Prognosefunktion wirkt sich auf die Abbildung der ungewissen Konsequenzen bei der Wahl von Handlungsalternativen im Kalkül des Entscheidungsträgers aus. Die Prognosefunktion bestimmt u.a. die subjektiven Wahrscheinlichkeiten bei der Entscheidung unter Risiko und ist damit nur relevant, wenn eben Entscheidungen bei Risiko zu treffen sind.

Die Beeinflussung der Zielfunktion zielt auf die Weise ab, wie ein Entscheidungsträger seine Zielsetzung im Entscheidungsmodell konkretisiert. Die Zielsetzung ist dabei nicht eine Größe, sondern sie wird durch eine Zielfunktion gekennzeichnet – bestehend aus der Menge der Zielgrößen, an welchen sich der Entscheidungsträger bei seiner Entscheidung orientiert, der Gestalt seiner Präferenzfunktion, sowie seinem Optimierungskriterium³⁵⁸. In der Präferenzfunktion kommt v.a. zum Ausdruck, welches Gewicht der Entscheidungsträger den einzelnen Zielgrößen seiner Zielfunktion beimisst. Damit ist die Präferenzfunktion ein wesentlicher Ansatzpunkt für Meta-Entscheidungen bei der Standardsoftwareentwicklung im Sinne eines kontinuierlichen und konfliktären Zielbildungsprozess. Sie kann u.a. beeinflusst werden, indem dem Entscheidungsträger (zusätzliche) Informationen über die Bedeutung der einzelnen Zielgrößen für das übergeordnete Ziel gegeben werden. Wird der Entscheidungsträger bspw. darüber informiert, welche Konsequenzen sich für das Projektprogramm erge-

³⁵⁸ Bspw. Maximierung oder ein bestimmtes Anspruchsniveau

ben können, wenn die Meilensteintermine des Projektes nicht eingehalten werden, so wird er dieser Zielgröße u.U. ein höheres Gewicht einräumen, als er es ohne diese Information getan hätte.

Die Steuerung von nachgeordneten Entscheidungen kann schließlich über die Beeinflussung der Ergebniswerte erfolgen. Bei einer gegebenen Zielfunktion des Entscheidungsträgers hängt dessen Entscheidung nämlich wesentlich davon ab, zu welchen Ergebniswerten die erwogenen Alternativen führen. Von besonderer Bedeutung sind hierbei vor allem jene Ergebniskomponenten, die den Entscheidungsträger persönlich betreffen.³⁵⁹ Da das Motivationsproblem bei der vorliegenden Arbeit nicht im Vordergrund steht, wird auch die Beeinflussung der Ergebniswerte im Weiteren keine Rolle spielen.

2.3.3 Projektziele als Gestaltungsbedingung

2.3.3.1 Das Zielsystem einer Standardsoftwareunternehmung

Als eine erste Annäherung an das Zielsystem einer Standardsoftwareunternehmung sollen die Zielobjekte und Zielgrößen der Produktentwicklung nach Schmelzer³⁶⁰ dienen (vgl. Abb. 30).

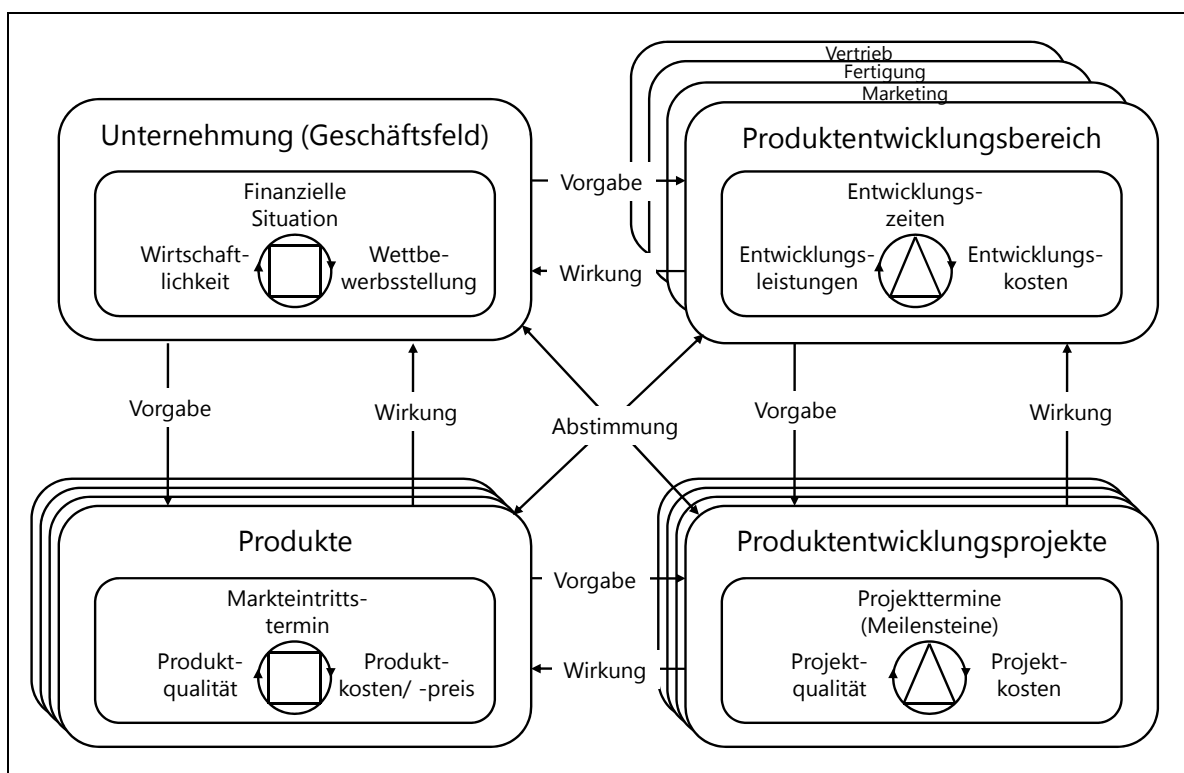


Abb. 30: Zielobjekte und Zielgrößen der Produktentwicklung³⁶¹

³⁵⁹ Bspw. im Fall von Gewinnbeteiligungen o.ä.

³⁶⁰ Vgl. Schmelzer (1991)

³⁶¹ Vgl. Schmelzer (1991), S. 86; vgl. Herzwurm (2000), S. 57

Der Ansatz verdeutlicht anschaulich, jedoch nur konzeptionell und auch nicht softwarespezifisch, dass zwischen den Zielobjekten und Zielgrößen der Produktentwicklung eine Instrumentalbeziehung besteht. Eine solche Beziehung wird durch ein Ziel-Mittel-Verhältnis zwischen den Zielen begründet. Unterziele stellen hier die Mittel zur Erreichung von Oberzielen dar, wobei ein Element in Abhängigkeit vom Blickwinkel des Betrachters sowohl ein Mittel für ein übergeordnetes Ziel, als auch ein durch ein untergeordnetes Mittel zu erreichendes Ziel darstellen kann. Jede Form von Ziel-Mittel-Beziehung besitzt daher einen relativen Charakter und ist durch ein kaskadenartiges Verhältnis zueinander geprägt.³⁶²

Der Produktentwicklungsbereich hat die Aufgabe, die erforderlichen Rahmenbedingungen zu schaffen und den Produktentwicklungsprojekten die Mittel, die sie für die Zielerreichung benötigen, bedarfs- und zeitgerecht zur Verfügung zu stellen. Die Zielgrößen des Produktentwicklungsbereichs sind Entwicklungszeit, Entwicklungsleistung und Entwicklungskosten. Produktentwicklungsprojekte setzen Produkthanforderungen effizient in technisch realisierbare und wirtschaftlich verwertbare Lösungen um. Sie werden nach den technisch-wirtschaftlichen Zielgrößen Projektqualität, Projektkosten und Projekttermine gesteuert. Die Projektziele leiten sich aus den Zielen der Unternehmung und den Produktzielen ab. Die Zielabhängigkeit zwischen Produktentwicklungsprojekt-, Unternehmungs- und Produktzielen spielt neben der Zieldefinition auch im Fall von Zielabweichungen bei Produktentwicklungsprojekten eine wichtige Rolle.³⁶³ Produktneu- und Produktweiterentwicklungen können demnach nicht isoliert geplant, gesteuert und überwacht werden, sondern sind über die projekt-, produkt-, entwicklungsbereichs- und gesamtunternehmensbezogenen Zielobjekte und -größen in eine Ziel-Mittel-Beziehung eingebunden.

Die bei Schmelzer genannten Zielobjekte und -größen gelten grundsätzlich auch für die Standardsoftwareentwicklung.³⁶⁴ Dabei ist für das Zielsystem einer Standardsoftwareunternehmung die langfristige strategische Entwicklungsplanung kennzeichnend. In diesem Punkt liegt ein wesentlicher Unterschied zur Individualsoftwareentwicklung. Dieser Unterschied zwischen einem singulären Entwicklungsergebnis bei der Individualsoftwareentwicklung und einem standardisierten und marktfähigen Kaufobjekt als Ergebnis der Standardsoftwareentwicklung wird in der ersten Phase begründet, die Schmelzer wiederum in Produktdefinition und Anforderungsermittlung unterteilt.³⁶⁵

³⁶² Vgl. Sauerwald (2007), S. 57f.; vgl. Bea (2004), Sp. 1675; vgl. Spengler (1999), S. 36ff.

³⁶³ Vgl. Schmelzer (1991), S. 86f.

³⁶⁴ Vgl. Herzwurm (2000), S. 61-70

³⁶⁵ Vgl. Herzwurm, Pietsch (2009), S. 32

Sowohl in der Betriebswirtschaftslehre als auch in den Ingenieurwissenschaften wird der Entwicklungsprozess logisch von der Entwicklungsplanung und der Produktdefinition getrennt.³⁶⁶ Der Übergang von der strategischen Entwicklungsplanung und der Produktdefinition zum Entwicklungsprozess bildet aus projektorientierter Sicht gleichzeitig den Übergang vom strategischen zum operativen Projektmanagement. Aus dieser Perspektive wird auch deutlich, wie sich die Ziel-Mittel-Beziehungen aus Sicht der vorliegenden Arbeit äußern.

Der Bezugspunkt des strategischen Projektmanagements ist die Auswahl der ‚richtigen‘ Projekte, d.h. der Vorhaben mit dem voraussichtlich größten Beitrag zur Erreichung der Organisationsziele. Hieran wird auch der Erfolg des strategischen Projektmanagements gemessen. Im zweiten Schritt muss dann die Umsetzung des gesamten Projektportefeuilles so erfolgen, dass eine koordinierte Überführung der Projekte von der Multiprojektebene auf die Ebene der Einzelprojektplanung möglich wird. Hier findet zugleich der Übergang vom strategischen in das operativ angelegte Management von Projekten statt, bei dem die effiziente Abwicklung jedes einzelnen Projektes im Vordergrund steht. Das projektbezogene Zielsystem (Termin, Kosten, Leistungsumfang und Qualität) ist durch geeignete Methoden und Techniken adäquat zu realisieren. Entsprechend wird hier der Erfolg an der Erreichung dieses vorgegebenen Zielsystems gemessen.³⁶⁷

Der entscheidungsorientierte Ansatz der Organisationsgestaltung unterscheidet hinsichtlich der verfolgten Perspektive zwischen dem statischen und dem dynamischen Gestaltungsziel.³⁶⁸ Aus dieser Sicht ist die Gestaltung des strategischen Projektmanagements dem dynamischen Gestaltungsziel zuzuordnen. Hier wird das Ziel verfolgt, „Organisationsstrukturen zu etablieren, die den Aufbau zukünftiger Handlungspotenziale fördern. Es soll durch organisatorische Regelungen sichergestellt werden, dass die Unternehmung langfristig auf den Märkten eine Position einnimmt, die ihren Bestand sichert und eine den Erfolgszielen entsprechende Realisierung von Absatzchancen erlaubt.“³⁶⁹ Damit einhergehend ist das operative Projektmanagement incl. der Multiprojektebene der statischen Organisationsgestaltung zuzuordnen. Diese ist „darauf ausgerichtet, die durch die gültige Strategie aufgebauten Potenziale zielkonform auszuschöpfen.“³⁷⁰

³⁶⁶ Vgl. Herzwurm, Pietsch (2009), S. 32

³⁶⁷ Vgl. Marr, Steiner (2004), Sp. 1198; vgl. Schelle u.a. (2005), S. 447f.; vgl. Bea u.a. (2008), S. 17f.; vgl. Project Management Institute (Hrsg., 2006), S. 8

³⁶⁸ Vgl. Frese (2005), S. 3

³⁶⁹ Frese (2005), S. 12

³⁷⁰ Frese (2005), S. 3

Die gängigen Prozessmodelle der Softwareentwicklung und auch der klassische Softwarelebenszyklus beginnen mit der Phase ‚Requirements Analysis und Definition‘ und damit mit der statischen Organisationsgestaltung. Sie beinhalten keine allgemeine Produktdefinition, deren Aufbau- und Ablauforganisation der dynamischen Organisationsgestaltung zuzuordnen wäre.³⁷¹ Erst ein zunehmend aufkommendes, umfassendes Verständnis des Requirements Engineering als ein phasenübergreifender Prozess mit engen Interdependenzen zum strategischen und operativen Projektmanagement stellt die allgemeine Produktdefinition in den Vordergrund und integriert sie explizit in den Softwareentwicklungsprozess.³⁷²

In diesem Zusammenhang sind aus entscheidungsorientierter Sicht v.a. die konzeptionellen Arbeiten von Aurum u.a. erwähnenswert: Um eine Abstimmung mit den Zielen der Unternehmung zu erreichen schlagen sie vor, bei der Betrachtung von Entscheidungsprozessen beim Requirements Engineering grundsätzlich zwischen der

- produkt-, projekt- und geschäftsfeldbezogenen Perspektive, sowie zwischen der
- strategischen, taktischen und operativen Ebene

zu differenzieren. Unter dieser Prämisse analysieren Aurum u.a. die komplexen Entscheidungsprozesse beim Requirements Engineering und entwickeln ein konzeptionelles Beschreibungsmodell, das als Grundlage für weitere Arbeiten dienen soll. In dem Modell ergeben sich Entscheidungssituationen und Ziel-Mittel-Beziehungen zwischen diesen Entscheidungssituationen auf drei Abstraktionsebenen:³⁷³

- Geschäftsfeld-, produk- und projektbezogene Entscheidungssituationen;
- Strategisch, taktisch und operativ ausgerichtete Entscheidungssituationen;
- Entscheidungssituationen in der Vor-Projekt- und der Projekt-Phase.

Unter Rückgriff auf diese drei Abstraktionsebenen leiten Aurum u.a. des Weiteren eine Entscheidungstaxonomie des Requirements Engineering ab (vgl. Abb. 31). Neben der Betrachtung von Entscheidungsprozessen auf verschiedenen Abstraktionsebenen und aus verschiedenen Perspektiven an sich sehen die Autoren das Alleinstellungsmerkmal ihrer Entscheidungstaxonomie v.a. in der expliziten Unterscheidung zwischen der Vor-Projekt- und der Projekt-Phase in Zusammenhang mit dem Requirements Engineering. Gerade diese Unterscheidung ist für die vorliegende Arbeit von Bedeutung.

³⁷¹ In Anlehnung an Herzwurm, Pietsch (2009), S. 32

³⁷² Bei Pohl nicht nur als phasenübergreifendes, sondern gar als projektübergreifendes Requirements Engineering verstanden. Vgl. Pohl (2007), S. S. 35

³⁷³ Vgl. Aurum u.a. (2006), S. 800f.

RE decisions in three levels		Strategic Decision	Tactical Decision	Operational Decision
Business decisions	<i>Pre-project RE decisions</i>	<ul style="list-style-type: none"> • business strategy • competitiveness • technology • marketing • economic value of the product 	<ul style="list-style-type: none"> • pricing • planned benefits of the product 	<ul style="list-style-type: none"> • tradeoff between technology push and market-pull
Product level decisions		<ul style="list-style-type: none"> • software product road mapping • packaging requirements for a specific release • software product architectures 	<ul style="list-style-type: none"> • engineering decisions • personnel management 	<ul style="list-style-type: none"> • chance management • requirements volatility, e. g. whether a particular requirement is subject to a syntactic or semantic change
Project level decisions	<i>In-project RE decisions</i>	<ul style="list-style-type: none"> • project planning • feasibility study • recruiting people 	<ul style="list-style-type: none"> • project management • quality control 	<ul style="list-style-type: none"> • validation in terms of which requirements will go to the next release

Abb. 31: Entscheidungstaxonomie des Requirements Engineering³⁷⁴

Abb. 32 stellt die Zusammenhänge aus allen angesprochenen Perspektiven zusammenfassend dar. Die Darstellung ist fokussiert auf einen entscheidenden Punkt aus Sicht des situativen Koordinationsansatzes, zu dem die vorstehenden Ausführungen hinführen: Die Ergebnisse strategischer Entscheidungen bzw. die Zielformulierung auf den Ebenen des strategischen Projektmanagements und der Projektprogrammplanung stellen aus Sicht (der Gestaltung) von einzelnen Entwicklungsprojekten innerhalb eines Projektprogramms Gestaltungsbedingungen dar. Es sind übergeordnete, den Entwicklungsprojekten vorgegebene Ziele, zu welchen die einzelnen Entwicklungsprojekte in einer Ziel-Mittel-Beziehung stehen.³⁷⁵ Damit stellen insbesondere auch die an den vier Ecken des Teufelsquadrates angetragenen Projektziele aus Sicht (der Gestaltung) von einzelnen Entwicklungsprojekten Gestaltungsbedingungen dar. Sie sind als Ausgangssituation bzw. als eine Problemdefinition auf hohem Abstraktionsniveau zu begreifen und werden im Verlauf des Entwicklungsprojektes weiter konkretisiert und in Problemdimensionen auf niedrigeren Abstraktionsebenen sowie in Lösungsdimensionen überführt (vgl. Abb. 32).³⁷⁶

³⁷⁴ Aurum u.a. (2006), S. 801

³⁷⁵ In Anlehnung an Herzwurm (2000), S. 53, 56

³⁷⁶ Vgl. Kap. 2.3.2.1 zur Softwareentwicklung als ein kontinuierlicher und konfliktärer Zielbildungsprozess

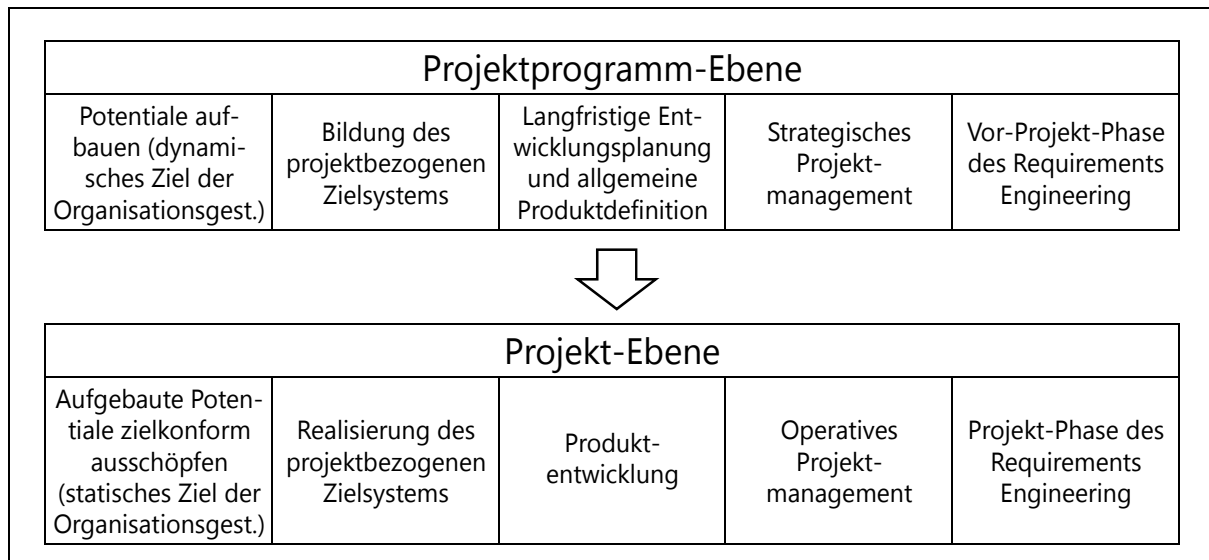


Abb. 32: Zielsystem einer Standardsoftwareunternehmung

2.3.3.2 Übergeordnete Projektzielpräferenzen als Gestaltungsbedingung für die Koordinationseffektivität

Im voranstehenden Kap. 2.3.3.1 ist begründet worden, dass Projektziele aufgrund der beschriebenen Ziel-Mittel-Relationen im Zielsystem einer Standardsoftwareunternehmung aus Projektsicht Gestaltungsbedingungen darstellen. Im Folgenden ist noch zu klären, warum

- bei der Gestaltung von Koordinationsprozessen der Standardsoftwareentwicklung
- mit dem Gestaltungsziel der Koordinationseffektivität
- Projektziele i.e.S. nur den Ausgangspunkt für Gestaltungsbedingungen

darstellen.

Wie bereits erläutert stellen die zu Projektbeginn definierten Ziele eine erste Problemdefinition für die nachfolgenden Phasen der Softwareentwicklung dar. Sie werden in einem kontinuierlichen und konfliktären Zielbildungsprozess in Lösungsbeschreibungen überführt, welche zugleich Problemdefinitionen auf niedrigeren Abstraktionsniveaus darstellen. Bildlich gesprochen stellen somit die an den vier Ecken des Teufelsquadrates angetragenen Projektziele das zu Projektbeginn gewünschte Projektergebnis dar. Dieses Ergebnis kann nur bestmöglich erreicht werden,³⁷⁷ wenn Entscheidungen in den Koordinationsprozessen im Projektverlauf im Sinn der dem Projekt übergeordneten Produkt- bzw. Projektprogrammziele getroffen werden. Die Koordinationsmechanismen der Standardsoftwareentwicklung müssen die Abstimmung der widersprüchlichen Ziele und Interessenslagen der Projektbeteiligten, also die ‚Navigation im

³⁷⁷ Aufgrund der Unsicherheit bei der Softwareentwicklung und der damit zusammenhängenden, tendenziell zu optimistischen Planung zu Projektbeginn, kann das Ziel der Softwareentwicklung nur im bestmöglichen Erreichen der Projektziele bestehen.

Teufelsquadrat‘, auf die übergeordneten Produkt- und Projektprogrammziele ausrichten. Die Ausgestaltung der Koordinationsprozesse muss dazu führen, dass die einzugehenden Kompromisse immer auf diesen Projektvorgaben basieren.

In Zusammenhang mit derartigen einzugehenden Kompromissen im Projektverlauf („Trade-Offs“) definiert Kerzner drei typische Kategorien von Projektvorgaben. Er geht dabei von den drei Projektzieldimensionen Kosten, Zeit und Leistung³⁷⁸ aus und definiert die drei Kategorien von Projektvorgaben wie folgt (vgl. auch Abb. 33):³⁷⁹

- Ein Projekt hat jeweils ein fixes Zielelement und zwei variable Zielelemente als Ansatzpunkte für einzugehende Kompromisse;
- Ein Projekt hat jeweils zwei fixe Zielelemente und ein variables Zielelement als Ansatzpunkt für einzugehende Kompromisse;
- Ein Projekt hat drei fixe oder drei variable Zielelemente und damit keine bzw. unbeschränkte Ansatzpunkte für einzugehende Kompromisse.

	Zeit	Kosten	Leistung
A. Jeweils ein fixes Element			
A-1	Fix	Variabel	Variabel
A-2	Variabel	Fix	Variabel
A-3	Variabel	Variabel	Fix
B. Jeweils zwei fixe Elemente			
B-1	Fix	Fix	Variabel
B-2	Fix	Variabel	Fix
B-3	Variabel	Fix	Fix
C. Drei Elemente fix oder variabel			
C-1	Fix	Fix	Fix
C-2	Variabel	Variabel	Variabel

Abb. 33: Zielkonfliktbasierte Kategorien von Projektvorgaben³⁸⁰

Die Situationen A und B stellen für Kerzner typische Kompromisse für das Projektmanagement dar, wobei v.a. die Situation A-3 auf die meisten Forschungs- und Entwicklungsprojekte zutrefte. Analog ordnet Kerzner u.a. auch typische Vertragstypen wie bspw. ‚Festpreisvertrag‘ oder ‚Festpreis mit Leistungsanreiz‘ als zwei von insg. 7 Vertragstypen den Kategorien von Projektvorgaben zu.³⁸¹ Auch im Umfeld der Softwareentwicklung ist von solchen typischen Kategorien von Projektvorgaben auszugehen, wie sie Kerzner annimmt.

³⁷⁸ I.w.S., also incl. Qualität.

³⁷⁹ Vgl. Kerzner (2003), S. 560f., 566f.

³⁸⁰ Kerzner (2003), S. 560f.

³⁸¹ Vgl. Kerzner (2003), S. 578

Wie bereits in Kap. 2.1.3.2 angesprochen fordert Boehm in Zusammenhang mit den sog. ‚SCQAIV‘³⁸²-Prozessmodellen, dass situativ entweder der Fertigstellungstermin, die Kosten oder die Qualität als fix oder als variabel und damit als ‚Stellschrauben‘ im Verlauf eines Softwareentwicklungsprojektes betrachtet werden.³⁸³ Eine solche Forderung wird auch durch die Ergebnisse einer empirischen Untersuchung zum Risikomanagement bei Softwareentwicklungsprojekten unterstützt. Wallace und Keil verdeutlichen hier, dass bspw. bei Projekten, bei welchen Kosten- und Terminziele hoch priorisiert sind bzw. die fixen Zielelemente darstellen, andere Risikofaktoren minimiert werden müssen, als das bspw. bei Projekten mit fixen Vorgaben bei den Zieldimensionen Qualität und Leistungsumfang der Fall ist.³⁸⁴ Abb. 34 stellt eine Auswahl der identifizierten erfolgskritischen Risiken bei den beiden genannten Projektvorgaben dar.

Hoch priorisierte Projektziel-dimension	Zu minimierende Risikofaktoren (u.a.)
Kosten und Zeit	Konfliktäre Anforderungen; häufige Anforderungsänderungen; häufige Änderungen des Projektumfanges; unklare oder unkorrekte Anforderungen
Qualität und Leistungsumfang	Qualifikation der Projektbeteiligten nicht auf Projektziele ausgerichtet; hohe technologische Unsicherheiten; starke Interdependenzen zu anderen Projekten

Abb. 34: Zielkonfliktbasierte Behandlung von Risikofaktoren bei Softwareentwicklungsprojekten³⁸⁵

Auch Yourdon fordert ausgehend von der Qualitätskritikalität schon länger ein Umdenken hin zu einer situativen Priorisierung von Projektzielen. Zwar werde bspw. bei Flugzeugen die Qualität und hier insb. die Zuverlässigkeit immer die kritische Projektzieldimension bilden. Doch sei es in vielen anderen Domänen der Softwareentwicklung schlussendlich von den Kundenanforderungen³⁸⁶ im Einzelfall abhängig, ob der Leistungsumfang kritischer eingestuft wird als ein schneller Projektabschluss und/oder die Qualität.³⁸⁷

McConnell argumentiert analog ausgehend von der Zeitkritikalität. Verschiedene Projektsituationen erfordern seines Erachtens mehr oder weniger Bekenntnis zu einem schnellstmöglichen Projektabschluss als übergeordnetes Projektziel. Daher sei es in bestimmten Situationen zielführend, die Entwicklungszeit um jeden Preis zu verkürzen, in anderen Situationen sollte dies dagegen nur erfolgen, falls dadurch keine über-

³⁸² SCQAIV: schedule/cost/quality as independent variable

³⁸³ Vgl. Boehm u.a. (2002), S. 1f.; vgl. Boehm u.a. (2000)

³⁸⁴ Vgl. Wallace, Keil (2004), S. 73

³⁸⁵ Vgl. Wallace, Keil (2004), S. 72f.

³⁸⁶ Interne oder externe Kunden

³⁸⁷ Vgl. Yourdon (1995), S. 79f.

mäßigen Kosten und/oder Verminderungen beim Leistungsumfang entstünden.³⁸⁸ Somit gelte für Projekte grundsätzlich: „efficient development represents a sensible optimization of cost, schedule, and product characteristics.“³⁸⁹

Die Kategorien von Projektvorgaben bei Kerzner sowie die kurz angesprochenen softwarespezifischen Ansätze stellen positive Ausnahmen dar. Ansonsten ist ein wesentliches Defizit bei der Formulierung von Projektzielen im gesamten Umfeld der Softwareentwicklung zu beobachten: Die einzelnen Projektziele innerhalb der vier Zieldimensionen bilden in der Summe das Projektzielsystem. Ein Zielsystem muss jedoch nach dem Postulat der Zielklarheit nicht nur die Ziele an sich beinhalten, sondern auch eine Präferenzordnung zwischen den Zielen, sofern mehr als ein Ziel angestrebt wird.³⁹⁰ Solche Präferenzordnungen³⁹¹ stellen Vorgaben dar, ob und in welchem Umfang ein Ziel der Erreichung eines anderen Zieles vorgezogen oder nachgeordnet werden soll. Da alle Projektziele in den seltensten Fällen mit den beschränkten Ressourcen gleichzeitig in dem Ausmaß verfolgt werden können, wie das wünschenswert wäre, ist eine Präferenzordnung nahezu immer notwendig.³⁹² In diesem Sinn stellt auch Schmelzer in Zusammenhang mit seiner Beschreibung der Zielobjekte und Zielgrößen der Produktentwicklung fest: „Konkurrierende Ziele erfordern eine Konfliktordnung, die über die Festlegung von Zielgewichtungen (Prioritäten) oder Nebenbedingungen erreicht werden kann.“³⁹³

Die Handhabung von Zielkonflikten in einem Zielsystem ist besonders dann kompliziert, wenn dieses, wie es in der Standardsoftwareentwicklung der Fall ist, auf unterschiedlichen Zielvorstellungen verschiedener Entscheidungsträger beruht. Das ist aus entscheidungsorientierter Sicht der zentrale Punkt beim Verständnis des Standardsoftwareentwicklungsprozesses als ein kontinuierlicher und hoch konfliktärer Zielbildungsprozess. In diesem Verständnis bildet

- die durch die übergeordneten Produkt- und Projektprogrammziele vorgegebene Präferenzordnung für die konkurrierenden Projektziele, im Folgenden als ‚Projektzielpräferenzen‘ bezeichnet, die zentrale Gestaltungsbedingung

³⁸⁸ Vgl. McConnell (1996), S. 18f.

³⁸⁹ McConnell (1996), S. 19

³⁹⁰ Vgl. Hauschildt (1981), S. 308-310; vgl. Hauschildt (1977); Der Ausdruck ‚die Ziele an sich‘ ist vereinfacht: Konkret handelt es sich dabei um ein abgegrenztes Zielobjekt, bestimmte Zielvariablen, eindeutige Zielmaßstäbe, explizite und präzise Angaben zum angestrebten Ausmaß sowie eine Angabe zum zeitlichen Bezug des Ziels. Hinzu kommt wie erläutert eine Präferenzordnung. Die Aufzählung zeigt, wie hoch die Anforderungen an ein vollständig formuliertes und operationales Entscheidungsziel sind.

³⁹¹ Auch: Präferenzrelationen

³⁹² Vgl. Bea u.a. (2008), S. 118

³⁹³ Schmelzer (1991), S. 85; vgl. auch Hauschildt (1980), Sp. 2421f.

- für die Ausgestaltung der Koordinationsprozesse als Gestaltungsmaßnahmen
- mit dem Ziel, Koordinationseffektivität zu erreichen.

Aufgrund der erläuterten Ziel-Mittel-Beziehungen im Zielsystem einer Standardsoftwareunternehmung muss bei der Bildung der Präferenzordnung auf Projektebene auf das Konzept der Opportunitätskosten zurückgegriffen werden. Mit Opportunitätskosten sind dem Wortsinn nach Kosten im Sinne ‚entgangener Gelegenheiten‘ gemeint. Opportunitätskosten entstehen immer dadurch, dass Ressourcen nur einmal verwendet und nicht gleichzeitig anderen nutzenstiftenden Zwecken zugeführt werden können. Deren Zielbeitrag geht daher verloren, sodass Opportunitätskosten alternative Kosten sind, die einem entgangenen Nutzen gleichkommen. Nutzenentgang setzt dabei voraus, dass mehrere Verwendungsmöglichkeiten miteinander konkurrieren. Fällt die Entscheidung auf eine Alternative, so kann der bei einer anderen Alternative entstandene Nutzenentgang als Opportunitätskosten verstanden werden. Nach dieser Charakterisierung können Opportunitätskosten für beliebige Zielfunktionen definiert werden.³⁹⁴

Mit ansteigenden potentiellen Opportunitätskosten im Fall einer Zielverfehlung in einer der vier Projektzieldimensionen steigt deren Kritikalität. Anders ausgedrückt müssen die Projektzieldimensionen mit den potentiell höchsten Opportunitätskosten bei der Zielverfehlung auch die höchste Stellung in der Präferenz- bzw. Konfliktordnung erfahren. An dieser Stelle ist es notwendig, die vier Projektzieldimensionen näher zu betrachten.

Aus der Ingenieurssicht ist das Ziel eines Softwareentwicklungsprojektes die Bereitstellung von (Teilbestandteilen einer) Software. Dieses Ziel wird durch die Umsetzung der Summe der sog. funktionalen Anforderungen sowie der sog. Qualitätsanforderungen festgelegt, die von der zu entwickelnden Software erfüllt werden müssen.³⁹⁵

„Eine funktionale Anforderung definiert eine vom System bzw. von einer Systemkomponente bereitzustellende Funktion oder einen bereitzustellenden Service.“³⁹⁶ Sie kann als Kundenanforderung noch sehr allgemein beschrieben sein; als Bestandteil einer Spezifikation ist eine funktionale Anforderung dagegen sehr detailliert beschrieben. Zielverfehlungen bei der Projektzieldimension Leistungsumfang resultieren aus der nicht erfolgten Umsetzung von funktionalen Anforderungen. Qualitätsanforderun-

³⁹⁴ Vgl. Ewert, Wagenhofer (2008), S. 114; Die obige Charakterisierung stellt verdrängte Alternativen als wesentliches Merkmal heraus. Tatsächlich wird der Opportunitätskostenbegriff jedoch vielschichtiger verwendet. Weitere Differenzierungen stellen beim outputbezogenen Opportunitätskostenbegriff neben Alternativkosten die sog. Optimalkosten in den Vordergrund. Darüber hinaus existieren neben dem outputbezogenen Opportunitätskostenbegriff auch die inputbezogenen Opportunitätskosten. Vgl. dazu Ewert, Wagenhofer (2008), S. 114f.

³⁹⁵ Vgl. Mellis (2004), S. 47

³⁹⁶ Pohl (2007), S. 15

gen definieren die gewünschten Qualitätsmerkmale der zu entwickelnden Software wie bspw. die Performanz des Softwaresystems, seine Zuverlässigkeit oder die gewünschte Ausfallsicherheit. Die Qualitätsmerkmale können sowohl für das gesamte Softwaresystem, als auch für eine bestimmte Funktion oder eine Komponente definiert werden.³⁹⁷ Analog zum Leistungsumfang resultieren Zielverfehlungen bei der Projektzieldimension Qualität aus der nicht erfolgten Umsetzung von Qualitätsanforderungen.

Die Qualität und der funktionale Leistungsumfang des Entwicklungsgegenstandes bzw. die Umsetzung der Anforderungen in der Summe werden auch als die Produktziele oder Sachziele des Entwicklungsprojektes bezeichnet. Hinzu kommen Formalziele der beiden Zieldimensionen Kosten als geplanter und bewerteter Ressourceneinsatz in Form von Aufwand in Mann-Monaten, sowie Zeit als geplanter Fertigstellungstermin.³⁹⁸ Entsprechend stellen Überschreitungen des geplanten Projektaufwandes und des geplanten Fertigstellungstermins Zielverfehlungen bei den Projektzieldimensionen Kosten und Zeit dar. Empirische Erkenntnisse zur Bedeutung der Kostenkritikalität bei der Standardsoftwareentwicklung machen allerdings deutlich, dass hier Kostenzielverfehlungen eine relativ untergeordnete Rolle im Verhältnis zu Zielverfehlungen bei den anderen Projektzieldimensionen spielen.

Wird eine Standardsoftware entwickelt, die an eine große Zahl von Kunden verkauft werden soll, so wird die Einhaltung der geplanten Entwicklungskosten beim Abwägen gegenüber der Einhaltung von Entwicklungszeit, Qualität und Leistungsumfang am niedrigsten präferiert. Wäre dagegen das Ziel der Softwareentwicklung bspw. eine Individuallösung, die nur an einen Kunden und zudem zu einem vorher festgelegten Preis verkauft wird (sog. Festpreisprojekt), wäre die Einhaltung der geplanten Entwicklungskosten für die Erlössituation des Auftragnehmers zweifellos von entscheidender Bedeutung und somit hoch präferiert.³⁹⁹ Diese grundsätzlich relativ niedrige Kostenkritikalität bei der Standardsoftwareentwicklung legt es daher nahe, diese bei der Bildung der Präferenz- bzw. Konfliktordnung des Projektzielsystems im Einzelfall nicht direkt, sondern mittelbar aus den anderen Zieldimensionen abzuleiten: Ist ein Projekt hinsichtlich der anderen Zieldimensionen bereits hoch kritisch, so kann es aus der konzeptionellen Perspektive nicht gleichzeitig kostenkritisch sein.

Abb. 35 stellt die übergeordneten Projektzielpräferenzen als Gestaltungsbedingung in Zusammenhang mit den Gestaltungsmaßnahmen der Koordination dar.

³⁹⁷ Vgl. Pohl (2007), S. 15f.

³⁹⁸ In Anlehnung an Schmelzer (1991), S. 83

³⁹⁹ Vgl. Carmel (1995), S. 119; vgl. Gerhardt (1992), S. 115

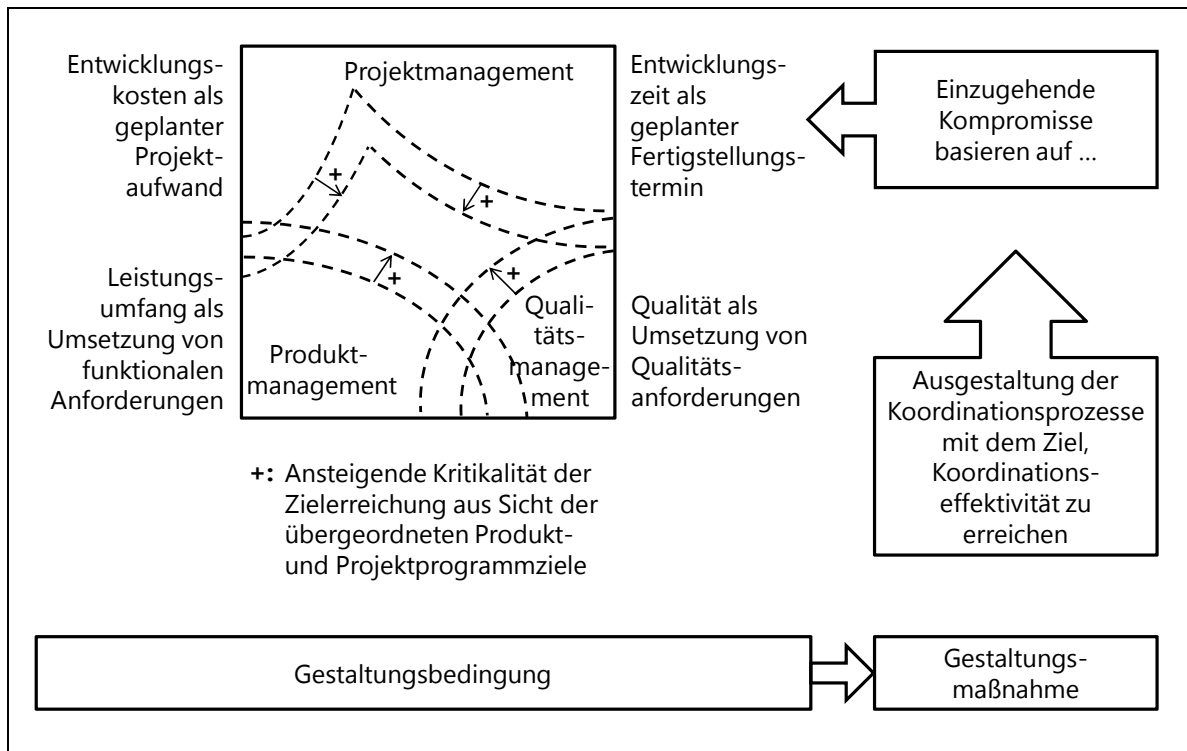


Abb. 35: Übergeordnete Projektzielpräferenzen als Gestaltungsbedingung für Gestaltungsmaßnahmen der Koordination

2.3.4 Organisatorische Umsetzung des situativen Koordinationsansatzes als Koordination durch Programme

Der Situative Ansatz fordert die Stimmigkeit zwischen der Situation (Gestaltungsbedingungen) und der Struktur (Gestaltungsmaßnahmen). In den Kap. 2.3.2 und 2.3.3 wurden die Gestaltungsmaßnahmen und -bedingungen des situativen Koordinationsansatzes entwickelt. Es wurde soweit nicht darauf eingegangen, wie die erforderliche Stimmigkeit hergestellt werden kann. Diese Fragestellung wird durch die Art und Weise der organisatorischen Umsetzung des situativen Koordinationsansatzes beantwortet.

Bei der organisatorischen Umsetzung des Situativen Ansatzes nimmt die Koordination durch Programme eine Vorrangstellung ein. Von Koordination durch ein Programm wird gesprochen, wenn „die Ausführung bestimmter Aktivitäten in einer bestimmten Art und Weise, nach einem bestimmten Verfahren oder Handlungsschema erfolgen soll.“⁴⁰⁰ Die Steuerungsfunktion von Programmen kann sich sowohl auf die Aufgabenerfüllung von einer einzelnen Stelle als auch von mehreren organisatorischen Einheiten beziehen. Koordinierende Wirkung besitzen Programme jedoch erst, wenn sie sich nicht isoliert auf einzelne Aktivitäten einer Stelle beziehen, sondern eine Abstimmung der Aktivitäten von verschiedenen organisatorischen Einheiten verfolgen. Da die

⁴⁰⁰ Vgl. Mellis (2004), S. 147

Abstimmung dabei vorausschauend erfolgt, sind Programme immer der Vorauskoordination zuzurechnen.⁴⁰¹

Ein Programm besteht genau genommen aus zwei Komponenten. Es beinhaltet zum einen ein Klassifikationsschema, das allgemein ausgedrückt die möglicherweise auftretenden Probleme definiert und zu einzelnen Problemklassen zusammenfasst. Zum anderen beinhaltet ein Programm, wiederum allgemein, die entsprechenden Verfahren zur Lösung der einzelnen Problemklassen. Durch ein Klassifikationsschema wird die Realität strukturiert und deren Komplexität reduziert. Programme legen hierüber fest, wie differenziert die Umwelt wahrgenommen wird. Je mehr Klassen ein Klassifikationsschema aufweist, desto mehr Unterschiede zwischen den realen Fällen können identifiziert werden. Ob zwei Objekte als gleich oder ungleich erkannt werden, hängt nämlich davon ab, wie fein das Raster zur Erfassung der Realität ist. Ein anschauliches Beispiel soll dies verdeutlichen: Wenn die Größe von Menschen in Zentimetern erfasst wird, so erweisen sich relativ viele Menschen als gleich groß. Wenn die Größe jedoch in Millimetern gemessen wird, verringert sich die Anzahl gleich großer Menschen erheblich.⁴⁰²

Mit der Entscheidung zugunsten der Koordination durch Programme muss zunächst geklärt werden, welchen Differenzierungsgrad des Klassifikationsschemas die organisatorische Umsetzung des situativen Koordinationsansatzes aufweisen soll.⁴⁰³

Jedes Konzept der organisatorischen Gestaltung beruht in Anbetracht der Problemkomplexität auf der Anwendung von heuristischen Vereinfachungsprinzipien. Anders können die methodischen Anforderungen von Gestaltungsentscheidungen und die Mehrdeutigkeit angesichts der begrenzten empirischen Absicherung von Wirkungszusammenhängen zwischen organisatorischen Gestaltungsvariablen nicht bewältigt werden.⁴⁰⁴ Die Entscheidung darüber, welchen Differenzierungsgrad das Klassifikationsschema aufweisen soll, ist eine Entscheidung über den Komplexitätsgrad und damit über das Ausmaß der Anwendung von heuristischen Vereinfachungsprinzipien.

Laux bezeichnet die Entscheidung über die Anwendung von heuristischen Vereinfachungsprinzipien als ein Meta-Entscheidungsproblem.⁴⁰⁵ Kern dieses Meta-Entscheidungsproblems ist der Konflikt zwischen den beiden Zielen Entscheidungsqualität und Modellierungskosten der Entscheidung. Denn durch die Modellierung zusätzlicher Details steigt die Realitätstreue eines Modells und damit seine Prognose-

⁴⁰¹ Vgl. Kieser, Walgenbach (2003), S. 116

⁴⁰² Vgl. Kieser, Walgenbach (2003), S. 76, 117

⁴⁰³ Vgl. Kap. 3.4.1

⁴⁰⁴ Vgl. Frese (2005), S. 5f.

⁴⁰⁵ Vgl. Laux, Liermann (2003), S. 36

und Gestaltungsqualität. Eine zunehmende Realitätstreue führt aber auch zu steigenden Kosten der Modellierung und damit der Entscheidungsfindung.⁴⁰⁶ Auf der anderen Seite hat jede Reduzierung eines Entscheidungsproblems zur Folge, „dass die Realisierung der für das ursprüngliche Entscheidungsproblem existierenden optimalen Lösung nicht garantiert werden kann.“⁴⁰⁷

Hinsichtlich der Frage, in welchem Umfang heuristische Vereinfachungen bei einem Entscheidungsproblem zweckmäßig sind, gibt es kein systematisches Verfahren, das es erlaubt, den optimalen Komplexitätsgrad auch nur annähernd exakt zu ermitteln.⁴⁰⁸ Ebenso ist diese Frage in der Literatur ohne eine konkrete und über die Angabe einiger allgemeiner Gesichtspunkte hinausgehende Antwort geblieben.⁴⁰⁹ In diesem Zusammenhang wird auch vom sog. Dilemma der Organisationstheorie gesprochen. Es besagt, dass die Konsequenzen von Vereinfachungen bei komplexen Entscheidungsproblemen nicht im Detail analysiert werden können.⁴¹⁰

In der Literatur finden sich zahlreiche heuristische Prinzipien, die insbesondere für den Umgang mit komplexen Entscheidungsproblemen als geeignet angesehen werden. Ihre Anwendung führt aus entscheidungstheoretischer Sicht zu der Lösung komplexer organisatorischer Entscheidungsprobleme mithilfe einschneidender Vereinfachungen bei allen (Entscheidungs-)Modellbestandteilen. Denn erst durch das Weglassen von Modellbestandteilen erhält ein Entscheidungsmodell eine handhabbare Komplexität, was die Voraussetzung für dessen Lösung ist.⁴¹¹ Im konkreten Fall des Entscheidungsproblems bei der Koordination durch Programme führt das Weglassen von Modellbestandteilen bzw. die Anwendung von heuristischen Vereinfachungsprinzipien zu einem geringeren Differenzierungsgrad des Klassifikationsschemas. Mögliche heuristische Vereinfachungsprinzipien sind in diesem Zusammenhang

- die Nichtberücksichtigung oder Abstraktion von Elementen der Situationsdefinition,
- die verdichtete Aggregation von Variablen sowie
- die horizontale oder vertikale Dekomposition eines Problems.⁴¹²

⁴⁰⁶ Vgl. Lang (2004), S. 137f.

⁴⁰⁷ Frese (2005), S. 125

⁴⁰⁸ Vgl. Bitz (1977), S. 429f.

⁴⁰⁹ Vgl. Bitz (1977), S. 402

⁴¹⁰ Vgl. Laux, Liermann (2003), S. 27f.

⁴¹¹ Vgl. Laux, Liermann (2003), S. 137

⁴¹² Vgl. zu den allgemeinen Vereinfachungsprinzipien im Folgenden Frese (2005), S. 125ff.; Lang (2004), S. 138ff., mit zahlreichen weiteren Verweisen.

Bezogen auf die Koordination durch Programme ist ersteres als die Nichtberücksichtigung oder Abstraktion von Gestaltungsbedingungen oder -maßnahmen zu interpretieren. Bei Gestaltungsmaßnahmen wirkt sich dieses Reduzierungsprinzip durch eine Einschränkung der betrachteten Handlungsmenge aus. Im weiteren Sinn gehört hierzu auch die unvollständige Konkretisierung von Handlungsalternativen, wenn also auf die Spezifikation von grundsätzlich relevanten Handlungsalternativen verzichtet wird.⁴¹³

Eine verdichtete Aggregation von Variablen kommt durch die repräsentative Auswahl eines oder mehrerer Stellvertreter bzw. durch die Konstruktion eines oder mehrerer (fiktiver) Stellvertreter zustande. Hierdurch erfolgt schlussendlich eine Zusammenfassung einzelner Variablen zu einem Variablenkomplex, der als eine Einheit betrachtet wird. Bei der Koordination durch Programme kann dies sowohl bei den Gestaltungsbedingungen als auch bei den Gestaltungsmaßnahmen erfolgen. D.h. als Variablen können Gestaltungsbedingungen oder auch Gestaltungsmaßnahmen interpretiert werden.

Bei der Koordination durch Programme spielt weniger die horizontale, sondern vielmehr die vertikale Dekomposition eines Problems in mehrere untergeordnete Teilprobleme eine Rolle. Dabei werden aus einem vage wahrgenommenen Problem präziser bestimmte Teilprobleme abgeleitet, die in einer vertikalen Beziehung zueinander stehen. D.h. die Ergebnisse des übergeordneten Teilproblems stellen Restriktionen für das untergeordnete Teilproblem dar. Bei diesem sind somit nur solche Lösungen relevant, die effizient hinsichtlich des übergeordneten Teilproblems sind. Insbesondere an diesem Prinzip wird eine der Hauptfunktionen von Heuristiken deutlich, nämlich die Einschränkung des Lösungsraums.

Die zweite zu klärende Fragestellung bei der Koordination durch Programme bezieht sich auf den Detaillierungsgrad der Programmvorgaben.⁴¹⁴ Dieser muss zwar ein bestimmtes Mindestmaß aufweisen – nämlich jenes, das es noch erlaubt, den Differenzierungsgrad des Klassifikationsschemas abzubilden. Die Problemlösungsvorgaben der einzelnen Problemklassen müssen sich also schlichtweg voneinander unterscheiden. Wie detailliert die Programmvorgaben darüber hinaus sein sollen, wird aber mit dem Detaillierungsgrad des Klassifikationsschemas nicht festgelegt. In dieser Hinsicht kann zwischen konkreten und abstrakten Programmen unterschieden werden.⁴¹⁵ Konkrete Programme legen ein Vorgehen detailliert fest, während abstrakte Programme nur grobe Anleitungen darstellen, die für ihre Umsetzung erst interpretiert und konkre-

⁴¹³ Die unvollständige Konkretisierung von Handlungsalternativen spielt auch bei der Frage zum Detaillierungsgrad der Projektvorgaben eine wesentliche Rolle; vgl. hierzu im Folgenden.

⁴¹⁴ Vgl. Kap. 3.3.3

⁴¹⁵ Vgl. Mellis (2004), S. 148

tisiert werden müssen. Analog zum Differenzierungsgrad gilt auch hier, dass je starrer und konkreter ein Programm ist, desto einheitlicher erfolgt die Ausführung der Aktivitäten durch die verschiedenen zu koordinierenden organisatorischen Einheiten.

Der Zusammenhang zwischen dem Differenzierungsgrad des Klassifikationsschemas und dem Detaillierungsgrad der Vorgaben lässt sich sehr anschaulich anhand des Tailoring⁴¹⁶ der Softwareentwicklung darstellen. Hinsichtlich der Gestaltungsmöglichkeiten des Tailoring muss allen voran der formale Rahmen abgeklärt werden, in welchem das Tailoring stattfinden soll. Wie in Kap. 2.1.3.1 bereits ausgeführt, kann hier grob zwischen einem informalen, d.h. pragmatischen und einfachen Tailoring-Prozess, und einem mehr oder weniger formalen Tailoring unterschieden werden. Zu den formalen Ansätzen zählt dabei das systematische Projekt-Tailoring über den Zwischenschritt der Definition von Projekttypen, welches in Analogie zu Programmen aus den beiden Elementen Klassifikationsschema und Vorgaben für den Projektablauf besteht. Abb. 36 stellt die möglichen Ansätze zur projektspezifischen Adaption eines Softwareentwicklungsprozesses dar.

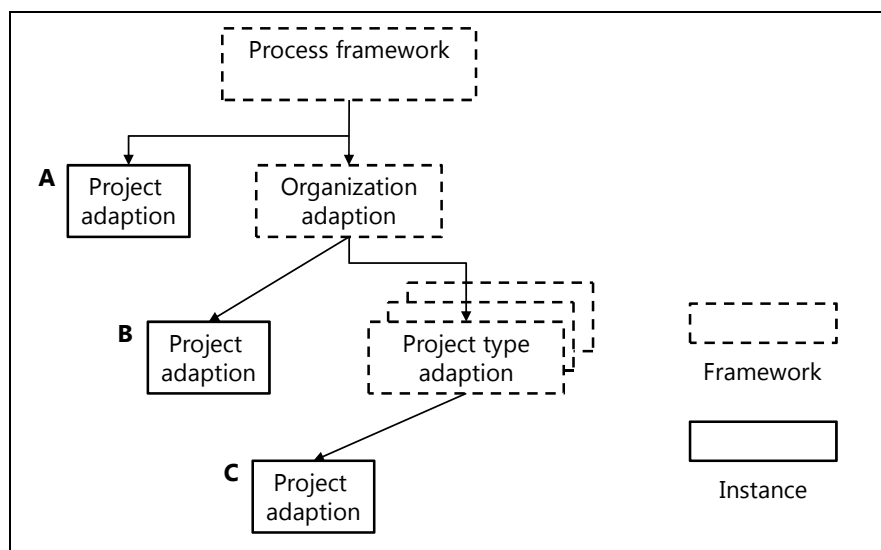


Abb. 36: Ansätze zur projektspezifischen Adaption eines Softwareentwicklungsprozesses⁴¹⁷

Die Adaption eines Softwareentwicklungsprozesses auf ein bestimmtes Projekt kann direkt oder über einen oder mehrere Zwischenschritte erfolgen. In Abb. 36 werden ausgehend von einem generellen, nicht unternehmensspezifischen Softwareentwicklungsprozess drei mögliche Adaptionswege aufgezeigt: Im Fall A geschieht das Tailoring einzeln für jedes Projekt in der Softwareunternehmung. In der Praxis hat das einen großen Aufwand und bei sich wiederholbaren Projekten nicht vertretbare Doppelarbeiten zur Folge. Diese werden umgangen, indem der Softwareentwicklungsprozess wie im Fall B zunächst unternehmensspezifisch angepasst wird. Ausgehend davon kann

⁴¹⁶ Vgl. Kap. 2.1.3.1

⁴¹⁷ Hanssen u.a. (2005), S. 316

dann das Projekt-Tailoring mit erheblich weniger Anpassungsbedarf und damit auch weniger Aufwand durchgeführt werden. Im Fall C wird mit der Bildung von Projekttypen nochmals ein Zwischenschritt eingefügt. Das Projekt-Tailoring erfolgt hier ausgehend von definierten Projekttypen.⁴¹⁸

Der Differenzierungsgrad des Klassifikationsschemas ist umso größer, je mehr Projekttypen definiert werden (Anzahl der gestrichelten Kästen ‚Project type adaption‘ in Abb. 36). Der Detaillierungsgrad der Projektvorgaben ist wiederum umso größer, je weniger Anpassungsbedarf bzw. Spielraum zwischen den Projekttypvorgaben und der projektspezifischen Adaption verbleibt (in der Abbildung als Gesamtlänge der Pfeile zwischen ausgehendem Framework und der Instanz⁴¹⁹). Damit wird auch ersichtlich, dass der Detaillierungsgrad der Projektvorgaben stark vom Ausmaß der vertikalen Dekomposition des Entscheidungsproblems abhängt. Denn die projektspezifische Adaption und die Projekttyp- bzw. unternehmensweite Adaption stehen in einer vertikalen Beziehung zueinander: Letzteres stellt Restriktionen bzw. Einschränkungen des Lösungsraumes für das Entscheidungsproblem auf projektspezifischer Ebene dar.

Je höher der Detaillierungsgrad der Vorgaben für die definierten Projekttypen ist, desto weitreichender sind diese Restriktionen und desto mehr werden die nachfolgenden, untergeordneten Teilentscheidungen vorstrukturiert. Dagegen wird bei einem geringen Detaillierungsgrad der Projektvorgaben auf der Projekttyp-Ebene noch keine realisationsreife Lösung des gegebenen Entscheidungsproblems formuliert. Frese spricht in diesem Zusammenhang von der Begrenzung bzw. Strukturierung des Entscheidungsproblems.⁴²⁰ Das Entscheidungsergebnis ist dabei noch nicht bis in alle Details bestimmt, sondern es sind noch gewisse Spielräume vorhanden. Das so strukturierte Problem wird an eine weitere Entscheidungseinheit, in diesem Fall auf der Projekt-Ebene, zur weiteren Strukturierung übertragen.

Entscheidungen äußern sich nach dem gerade beschriebenen Strukturierungsmodell demnach in ihrem „Beitrag ... zur stärkeren Strukturierung der Komponenten eines gegebenen Entscheidungsproblems. Jede auf diese Weise auf einer Stufe des Entscheidungsprozesses vorgenommene Strukturierung der Entscheidungskomponenten verringert den Lösungsraum und schränkt den Spielraum für die Entwicklung von Alternativen auf den nachfolgenden Stufen ein. Das Entscheidungsproblem ist gelöst, wenn eine realisationsreife Handlung formuliert ist“⁴²¹ (hier Projektadaption).

⁴¹⁸ Vgl. Hanssen u.a. (2005), S. 316f.

⁴¹⁹ In der Originalabbildung hat die Pfeillänge keine Bedeutung.

⁴²⁰ Vgl. Frese (2005), S. 130f.

⁴²¹ Frese (2005), S. 131

2.4 Konzeption des situativen Koordinationsansatzes im Überblick

Abb. 37 stellt die Konzeption des situativen Koordinationsansatzes im Überblick dar. Den Rahmen für die Konzeption bildet das Grundmodell der pragmatischen Variante des Situativen Ansatzes der Organisationsgestaltung.⁴²²

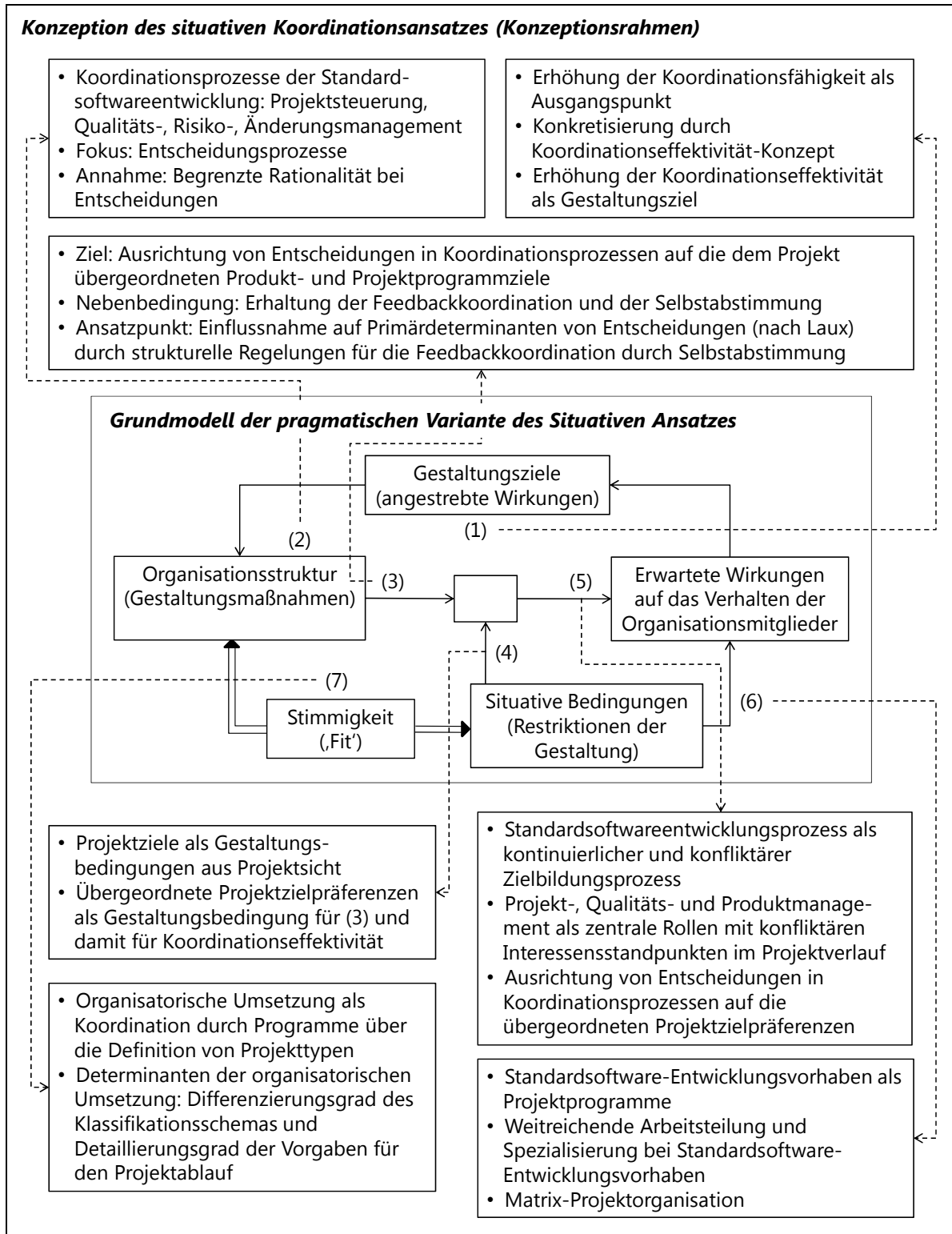


Abb. 37: Konzeption des situativen Koordinationsansatzes im Überblick

⁴²² Vgl. Kap. 2.1.1

- (1) Der situative Koordinationsansatz geht von dem allgemeinen Ziel ‚Erhöhung der Koordinationsfähigkeit‘ aus. Die mangelnde Koordinationsfähigkeit ist ein Hauptgrund für die vielfach zitierte Softwarekrise. Dieses allgemeine Ziel wird mit der Entwicklung des Koordinationseffektivität-Konzeptes weiter konkretisiert. Bei diesem stehen die grundsätzliche Wirksamkeit (Effektivität) der Koordination und das Entscheidungssystem der Unternehmung im Vordergrund. Koordinationseffektivität wird dabei als ein Stimmigkeitskonstrukt definiert. Stimmigkeit eignet sich als Konstrukt zur Erklärung von organisatorischer Effektivität insbesondere wenn eine Arbeit den Situativen Ansatz der Organisationsgestaltung verfolgt. Koordinationseffektivität zielt auf die Stimmigkeit zwischen den einzelnen Gestaltungsmaßnahmen bei den Koordinationsprozessen untereinander (2) und zu den übergeordneten Projektzielpräferenzen (4) ab. Entsprechend bewirkt hier eine Erhöhung der Stimmigkeit die Erhöhung der Koordinationseffektivität und damit eine bessere Ausrichtung von Entscheidungen im Projektverlauf auf die dem Projekt übergeordneten Produkt- und Projektprogrammziele.⁴²³
- (2) Die Koordinationseffektivität als Gestaltungsziel muss durch eine geeignete Organisationsstruktur herbeigeführt werden. Die Überlegungen hierzu gehen davon aus, dass es nicht den einen bestimmten Koordinationsprozess in der Standardsoftwareentwicklung gibt. Zum (abstrakten) Koordinationsprozess müssen vielmehr alle Teilprozesse gezählt werden, die eng mit der Abstimmung interdependenter, abweichungsinduzierter Entscheidungen auf ein übergeordnetes Ziel hin verknüpft sind. Bei der Standardsoftwareentwicklung gehören hierzu die unterstützenden phasenübergreifenden Teilprozesse Projektsteuerung sowie Risiko-, Änderungs- und Qualitätsmanagement. Dabei werden in dieser Arbeit ganz im Sinne des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung die Entscheidungsprozesse in den Mittelpunkt der Betrachtung gestellt und es wird angenommen, dass diese unter der begrenzten Rationalität der Entscheider stattfinden.⁴²⁴
- (3) Die Organisationsstruktur beinhaltet Vorgaben für die Organisationsmitglieder und schreibt bestimmte Verhaltensweisen vor. Beim situativen Koordinationsansatz stellt sich hier die Frage, wie sichergestellt werden kann, dass die im Projektverlauf stattfindenden Entscheidungen in den Koordinationsprozessen Projektsteuerung sowie Risiko-, Änderungs- und Qualitätsmanagement im Sinne der dem Projekt übergeordneten Produkt- und Projektprogrammziele getroffen werden. Durch die Vorgaben dürfen jedoch die Entscheidungen im Projektverlauf nicht vorwegge-

⁴²³ Vgl. Kap. 2.3.1

⁴²⁴ Vgl. Kap. 2.2.1 und Kap. 2.3.2.2

nommen werden, d.h. aus der Feedbackkoordination im Sinne der Reaktion auf Störungen darf nicht eine ausschließliche Vorauskoordination vor dem Beginn der Aufgabenerfüllung entstehen. Desweiteren darf die Selbstabstimmung in den Koordinationsprozessen nicht grundsätzlich durch eine hierarchische Entscheidungsfindung bzw. persönliche Weisung durch hierarchisch übergeordnete Einheiten ersetzt werden. Die Festlegung von bestimmten strukturellen Regelungen für die Feedbackkoordination durch Selbstabstimmung ist ein Ansatzpunkt, der die beiden voranstehenden Bedingungen erfüllt. Als konzeptionelle Grundlage für die Festlegung der konkreten Regelungen dient dabei das Strukturierungskonzept der entscheidungsorientierten Organisationslehre nach Laux. Danach können Entscheidungen anderer Organisationsmitglieder dadurch beeinflusst und gesteuert werden, indem auf mindestens eine der Primärdeterminanten von Entscheidungen Einfluss genommen wird. D.h. die Beeinflussung des Typs des Entscheidungsmodells, der Menge der erwogenen Handlungsalternativen, der Informationsstruktur, der Prognosefunktion, der Zielfunktion sowie der Ergebnisse der Alternativen sind als Zielkomponenten der Steuerung von Entscheidungen zu sehen.⁴²⁵

- (4) Die Ergebnisse strategischer Entscheidungen bzw. die Zielformulierung auf den Ebenen des strategischen Projektmanagements und der Projektprogrammplanung stellen aus Sicht der Entwicklungsprojekte in einem Projektprogramm Gestaltungsbedingungen dar. Es sind übergeordnete Ziele, zu welchen die einzelnen Entwicklungsprojekte in einer Ziel-Mittel-Beziehung stehen. Damit stellen insbesondere auch die Projektziele aus Sicht der einzelnen Entwicklungsprojekte Gestaltungsbedingungen dar.⁴²⁶

Die Ausgestaltung der Koordinationsprozesse (3) muss dazu führen, dass die Entscheidungen im Projektverlauf trotz der widersprüchlichen Ziele und Interessenslagen der zentralen Projektrollen (5) auf den Projektzielen und somit auch auf den übergeordneten Produkt- und Projektprogrammzielen basieren. Nach diesem Verständnis bilden

- die übergeordneten Projektzielpräferenzen die zentrale Gestaltungsbedingung
- für die Ausgestaltung der Koordinationsprozesse als Gestaltungsmaßnahmen
- mit dem Ziel, Koordinationseffektivität zu erreichen.

⁴²⁵ Vgl. Kap. 2.3.2.3

⁴²⁶ Vgl. Kap. 2.3.3.1

Die übergeordneten Projektzielpräferenzen stellen Vorgaben dafür dar, welches der Projektziele im Zweifelsfall der Erreichung eines anderen Projektziels vorgezogen oder nachgeordnet werden soll. Steigen die potentiellen Opportunitätskosten aus Produkt- und Projektprogramm-Sicht im Falle der Zielverfehlung in einer der vier Projektzieldimensionen an, so steigt damit auch die Kritikalität der Zielerreichung in ebendieser Zieldimension an.⁴²⁷

(5) Die erwarteten Wirkungen von organisatorischen Regelungen auf das Verhalten der Organisationsmitglieder werden als Kombination von (3) und (4) begriffen. Entscheidend für das Verständnis dieser Wirkungen ist das Verständnis des Standardsoftwareentwicklungsprozesses als ein kontinuierlicher und konfliktärer Zielbildungsprozess. In einem solchen Umfeld ist die Handhabung von Zielkonflikten besonders kompliziert. Denn hier basiert das Zielsystem auf unterschiedlichen Zielvorstellungen verschiedener Entscheidungsträger. Dabei lassen sich mit dem Projekt-, dem Qualitäts-, und dem Produktmanagement drei am Projekt beteiligte zentrale Rollen mit konfliktären Interessenstandpunkten festmachen. Projektleiter neigen dazu, Kosten- und Terminziele am stärksten zu priorisieren, v.a. stärker als Qualitätsziele. Das Qualitätsmanagement bzw. ein Qualitätsverantwortlicher im Projekt kann als der ‚Gegenspieler‘ des Projektleiters bezeichnet werden. Dieser ist für die Erfüllung der Qualitätsziele verantwortlich. Dem Produktmanager wird die interne Auftraggeber-Rolle für die Entwicklung der Standardsoftware übertragen. Er vertritt in einem Entwicklungsprojekt ‚sein Produkt‘. Nichts desto trotz müssen die Koordinationsmechanismen der Standardsoftwareentwicklung die Abstimmung dieser widersprüchlichen Interessenslagen der Projektbeteiligten auf die übergeordneten Projektzielpräferenzen und damit die Produkt- und Projektprogrammziele ausrichten.⁴²⁸

(6) Über die in (4) aufgeführten Gestaltungsbedingungen hinaus haben auch direkte, d.h. von der Organisationsstruktur unabhängige Gestaltungsbedingungen Einfluss auf die erwarteten Wirkungen der Gestaltungsmaßnahmen. Hierunter sind die grundsätzlichen Gestaltungsbedingungen zu verstehen, die bei der Entwicklung von Standardsoftware vorherrschen. Aus Sicht des situativen Koordinationsansatzes ist dabei zentral, dass die zumeist umfangreichen Entwicklungsvorhaben die Bildung von mehreren Projekten innerhalb eines Projektprogramms erfordern. Somit sind Arbeitsteilung und Spezialisierung bei Entwicklungsvorhaben im Bereich Standardsoftware sehr weitgehend. Die Entwicklungsvorhaben werden dabei in-

⁴²⁷ Vgl. Kap. 2.3.3.2

⁴²⁸ Vgl. Kap. 2.3.2.1

nerhalb von oder im Zusammenspiel von Produktbereichen durchgeführt. Das führt zu der weiten Verbreitung von mehrdimensionalen Organisationsformen bei Standardsoftwareunternehmen und insbesondere zur Matrix-Projektorganisation mit ihrem inhärenten Potential für Kompetenzkonflikte.⁴²⁹

- (7) Es wurde soweit nicht darauf eingegangen, wie die notwendige Stimmigkeit zwischen der Situation (Gestaltungsbedingungen) und der Struktur (Gestaltungsmaßnahmen) hergestellt werden kann. Hierzu wird die organisatorische Umsetzung des situativen Koordinationsansatzes als Koordination durch Programme vorgeschlagen. Ein Programm besteht aus zwei Komponenten. Es beinhaltet zum einen ein Klassifikationsschema, das allgemein ausgedrückt die möglicherweise auftretenden Probleme definiert und zu einzelnen Problemklassen zusammenfasst. Zum anderen beinhaltet ein Programm, wiederum allgemein, die entsprechenden Verfahren zur Lösung der einzelnen Problemklassen. Mit der Entscheidung zugunsten der Koordination durch Programme muss zunächst geklärt werden, welchen Differenzierungsgrad des Klassifikationsschemas die organisatorische Umsetzung des situativen Koordinationsansatzes aufweisen soll. Die Entscheidung darüber ist eine Entscheidung über den Komplexitätsgrad und damit über das Ausmaß der Anwendung von heuristischen Vereinfachungsprinzipien. Die zweite zu klärende Fragestellung bezieht sich auf den Detaillierungsgrad der Programmvorgaben. Konkrete Programme legen ein Vorgehen detailliert fest, während abstrakte Programme nur grobe Anleitungen darstellen, die für ihre Umsetzung erst interpretiert und konkretisiert werden müssen. Bei der Standardsoftwareentwicklung kann die Koordination durch Programme über die Definition von Projekttypen stattfinden. Das sog. Projekt-Tailoring über Projekttypen besteht in Analogie zu Programmen aus den beiden Elementen Klassifikationsschema und Vorgaben für den Projektablauf. Der Differenzierungsgrad des Klassifikationsschemas ist dabei umso größer, je mehr Projekttypen definiert werden. Der Detaillierungsgrad der Vorgaben für den Projektablauf ist wiederum umso größer, je weniger Anpassungsbedarf bzw. Spielraum zwischen den Projekttypvorgaben und der projektspezifischen Adaption verbleibt.⁴³⁰

⁴²⁹ Vgl. Kap. 2.2.2

⁴³⁰ Vgl. Kap. 2.3.4

3 Umsetzung des situativen Koordinationsansatzes

Im vorliegenden Kap. 3 der Arbeit wird der entwickelte Konzeptionsrahmen zu einem sog. Entscheidungsrahmen weiterentwickelt. Im Entscheidungsrahmen werden die theoretischen Aussagen aus Kap. 2 auf praktische Problemstellungen hin ausgerichtet, d.h. zu solchen Aussagen weiterentwickelt, die sich zur Unterstützung realer organisatorischer Problemlösungen heranziehen lassen.

Während das Gestaltungsziel des situativen Koordinationsansatzes (Koordinationseffektivität)⁴³¹ hinreichend geklärt ist, müssen die anderen Komponenten noch konkretisiert werden. Denn die in Kap. 2 erarbeiteten konzeptionellen Antworten zu den Forschungsfragen stellen noch keine Entscheidungshilfen für die Praxis dar – sie müssen vielmehr zu solchen weiterentwickelt werden. Dazu muss angegeben werden, welche konkreten Gestaltungsmaßnahmen (Kap. 3.2) unter den jeweils herrschenden Gestaltungsbedingungen (Kap. 3.3) bei Berücksichtigung der Wirkungen (Organisatorische Umsetzung; Kap. 3.4) ergriffen werden können, um das Gestaltungsziel des situativen Koordinationsansatzes (Koordinationseffektivität) zu erreichen. Kap. 3.5 stellt den Konzeptionsrahmen im Überblick dar.

Mit dem Übergang vom Konzeptions- zum Entscheidungsrahmen erfolgt auch der Übergang von der sachlich-analytischen zur empirischen Forschungsstrategie, auf die im nachfolgenden Kap. 3.1 eingegangen wird.

3.1 Fallstudiendesign

Die Explorationsstrategie der vorliegenden Arbeit erfordert beim empirischen Teil Rückgriff auf Methoden der qualitativen empirischen Forschung. Konkret kommt bei der Entwicklung des Entscheidungsrahmens die Einzelfallstudie zum Einsatz. Wie in Kap. 1.3 dargelegt gibt die Einzelfallstudie an sich noch keine Auskunft darüber, auf welche Forschungsmethoden im Einzelnen zurückgegriffen wird. In der vorliegenden Arbeit besteht die Einzelfallstudie aus einer umfassenden Dokumentenanalyse und anschließenden Experteninterviews. Beide Forschungsmethoden dienen im Zusammenspiel zur Beantwortung der Forschungsfragen, die an den empirischen Teil der Arbeit gerichtet sind. Dabei soll bereits eine Absicherung der Erkenntnisse erfolgen. Die Vorgehensweise hierzu lehnt sich an die sog. analytische Induktion an.

Die analytische Induktion setzt nach der Entwicklung eines vorläufigen Ergebnisses an. Ein vorläufiges Ergebnis kann im Allgemeinen ein vorläufiges Theoriegerüst, Modell, Ansatz etc. sein. In dieser Arbeit sind es konkret die Ergebnisse der Dokumenten-

⁴³¹ Vgl. Kap. 2.3.1

analyse. Die analytische Induktion ist an der Absicherung der gewonnenen Erkenntnisse durch die Analyse und die anschließende Integration abweichender Fälle, die die vorläufigen Ergebnisse in Frage stellen, orientiert.⁴³² Durch die Analyse dieser Fälle sollen gerade solche Aussagen erarbeitet werden, die nicht durch Sammeln großer Datenmengen hervorkommen wären. Diese Ausnahmefälle sollen zum umfassenderen Verständnis, zur besseren Theorie, und schließlich zum wissenschaftlichen Fortschritt führen.⁴³³ In der vorliegenden Arbeit wird die analytische Induktion im Rahmen der Experteninterviews umgesetzt.

Bei der Fallstudienunternehmung handelt es sich um einen international führenden Hersteller von betrieblicher Standardsoftware. Da das Kerngeschäft der Fallstudienunternehmung in der Entwicklung von Software liegt, die als ein eigenständiges Produkt vermarktet wird, gehört sie der sog. Software-Primärbranche an. Gerade die Entwicklung von betrieblicher Standardsoftware bildet den Schwerpunkt der Software-Primärbranche: Ca. 5000 bzw. 55% der Softwareunternehmungen erstellen Software, die diesem Spektrum zuzuordnen ist.⁴³⁴ Hinsichtlich der Zusammensetzung des Softwaremarktes aus Anbietern und Nachfragern agiert die Fallstudienunternehmung auf dem B2B-Softwaremarkt (Business-To-Business).⁴³⁵

Dem Softwareentwicklungsbereich der Fallstudienunternehmung gehören ca. 15.000 Mitarbeiter an mehreren weltweit verteilten Entwicklungszentren an,⁴³⁶ wobei sich das größte hiervon am Hauptsitz der Unternehmung in Deutschland befindet. Die für die vorliegende Arbeit relevanten organisatorischen Aspekte des Softwareentwicklungsbereiches entsprechen den Ausführungen in Kap. 2.2.2.1 und 2.2.2.2. Das Produktportfolio der Fallstudienunternehmung umfasst v.a. die folgenden betrieblichen Standardsoftwarelösungen, die über verschiedene Betriebs- und Nutzungsmodelle weltweit vermarktet und vertrieben werden:⁴³⁷

⁴³² Vgl. hierzu auch Kap. 1.4 zur hermeneutischen Spirale.

⁴³³ Vgl. Bühler-Niederberger (1991), S. 447; vgl. Flick (2006), S. 430f.; vgl. Steinke (2004), S. 329; vgl. Lamnek (2005), S. 161

⁴³⁴ Vgl. GfK Marktforschung u.a. (Hrsg., 2000), S. 67; Die weiteren Bereiche der Software-Primärbranche sind: Multimedia- und Internet-Software (ca. 4200 Softwareunternehmungen bzw. 46%), technische Software (ca. 3000 Softwareunternehmungen bzw. 33%), Systemsoftware (ca. 1500 Softwareunternehmungen bzw. 17%), sonstige kommerzielle Software (ca. 3300 Softwareunternehmungen bzw. 36%, hierunter v.a. Anbieter von Dokumentenverwaltungs- und Wissensmanagementsoftware).

⁴³⁵ Vgl. Kremer (2010)

⁴³⁶ Die Gesamtanzahl der Mitarbeiter beträgt ca. 48.000.

⁴³⁷ Die Ausführungen zur Fallstudienunternehmung basieren auf der offiziellen Unternehmensdarstellung.

- Eine integrierte Standardsoftwarelösung mit Anwendungen zur Unterstützung von nahezu allen administrativen Bereichen und betrieblichen Kernprozessen einer großen Unternehmung;
- Mehr als 25 branchenspezifische Varianten der erstgenannten Standardsoftwarelösung;
- Mehrere integrierte Standardsoftwarelösungen für kleine und mittelgroße Unternehmungen;
- Eine Standardsoftwarelösung für die Anforderungen von Entscheidungsträgern und Prozessverantwortlichen in Unternehmungen aller Größenordnungen (sog. ‚Business Intelligence‘);
- Eine Plattformlösung, die Informationen, Anwendungen und Geschäftsprozesse über verschiedene Technologien und Organisationsformen hinweg integriert.

Abb. 38 stellt das Fallstudiendesign beim empirischen Teil der Arbeit schematisch dar.

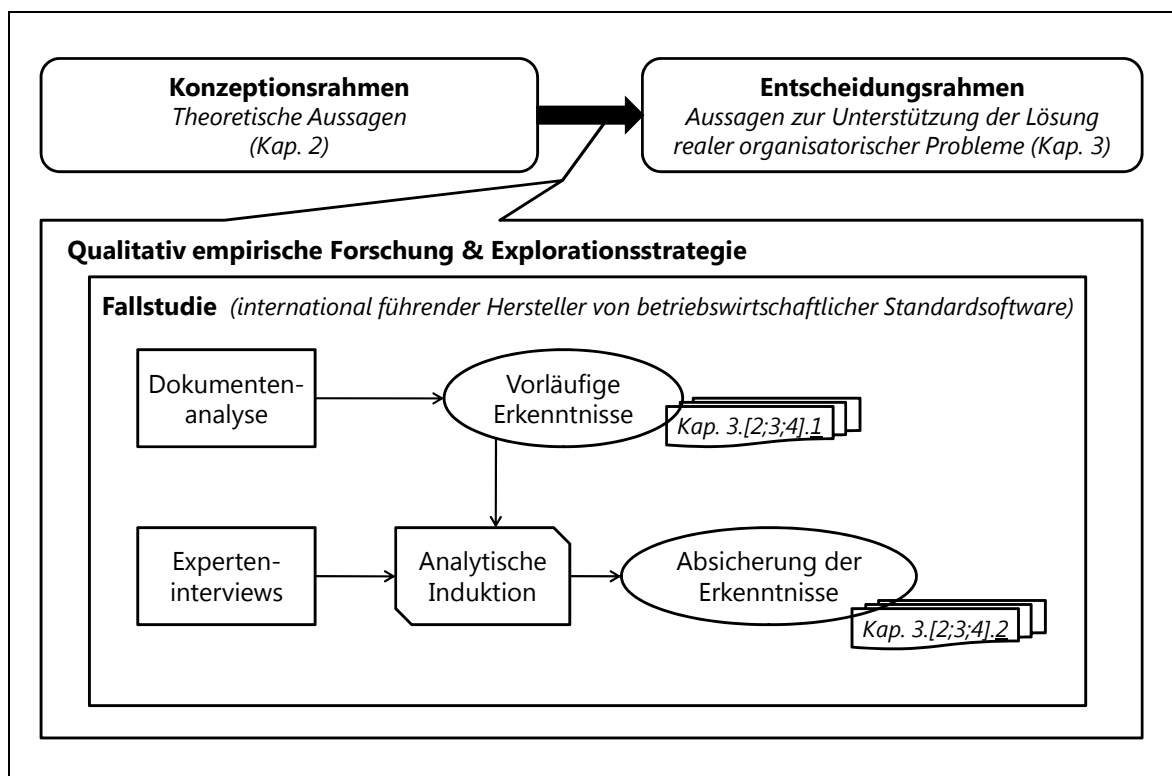


Abb. 38: Fallstudiendesign beim empirischen Teil der Arbeit

3.1.1 Vorgehensweise bei der Dokumentenanalyse

Die Vorgehensweise bei der Dokumentenanalyse orientiert sich jeweils an der zu beantwortenden Forschungsfrage. Zur Ableitung der Gestaltungsmaßnahmen für den situativen Koordinationsansatz (Forschungsfrage 2) wurden alle Prozess- und Rollenbeschreibungen in der Fallstudienunternehmung analysiert, die im Zusammenhang mit den hier relevanten Koordinationsprozessen Projektsteuerung, Risiko-, Änderungs-

und Qualitätsmanagement stehen. Diese 40 elektronischen Dokumente sowie auf ‚ARIS‘-Prozessmodellen basierenden Prozessbeschreibungen⁴³⁸ wurden in die ausführliche Dokumentenanalyse einbezogen. Diese fokussierte auf die Identifizierung von Ansatzpunkten für die situative Ausrichtung der Koordinationsprozesse in Abhängigkeit von den übergeordneten Projektzielpräferenzen. Solche Steuerungsmaßnahmen müssen die folgenden Voraussetzungen erfüllen:

- Die Steuerungsmaßnahme beeinflusst mindestens eine der Primärdeterminanten von Entscheidungen nachgeordneter Entscheidungsträger⁴³⁹ und weist dabei einen begründeten Zusammenhang zu einer der Projektzieldimensionen auf;
- Die Steuerungsmaßnahme findet im Rahmen der Projektsteuerung oder des Risiko-, Änderungs- oder Qualitätsmanagements statt;
- Die Steuerungsmaßnahme findet im Projektverlauf zeitlich zwischen Projektbeginn und der Phase Implementierung statt;
- Die Steuerungsmaßnahme ist situativ einsetzbar.

Alle Steuerungsmaßnahmen, welche diese vier Voraussetzungen erfüllen, wurden zunächst anhand eines festen Erfassungsschemas dokumentiert (vgl. Abb. 39).

Erfassungsschema	Erläuterung
Steuerungsmaßnahme	Kurzbeschreibung der Steuerungsmaßnahme bzw. der identifizierten Ansatzpunkte für eine Steuerungsmaßnahme.
Zuordnung zum Koordinationsprozess	Die Steuerungsmaßnahme muss im Rahmen der Projektsteuerung oder des Risiko-, Änderungs- oder Qualitätsmanagements stattfinden.
Tailoring der Steuerungsmaßnahme	Die Steuerungsmaßnahme an sich findet zwar im Projektverlauf statt, doch das Tailoring der Steuerungsmaßnahme muss bereits zu Projektbeginn möglich sein. Daher wird hier auf die situative Umsetzung bzw. das Tailoring eingegangen (evtl. Klassifikationsschema, Prozessvarianten, etc.).
Bezug zur Projektzieldimension	Die Steuerungsmaßnahme muss in einem unmittelbaren Zusammenhang zu einer der vier Projektzieldimensionen Zeit, Kosten, Leistungsumfang und Qualität stehen.

Abb. 39: Erfassungsschema für Gestaltungsmaßnahmen bei der Dokumentenanalyse

⁴³⁸ ARIS („Architektur integrierter Informationssysteme“) ist ein umfassender Ansatz zur Modellierung und zum Management von Geschäftsprozessen; Zugang zu den Dokumenten erfolgte über das unternehmensinterne Prozessportal im Intranet.

⁴³⁹ Vgl. Kap. 2.3.2.3

In einem zweiten Schritt wurden alle auf Basis der Dokumentenanalyse abgeleiteten Steuerungsmaßnahmen zu sog. Strukturdimensionen der Organisationsgestaltung zusammengefasst. Die Ableitung dieser Strukturdimensionen als übergeordneter Rahmen, in welchen sich die möglichen Steuerungsmaßnahmen mit Einfluss auf Entscheidungen in den Koordinationsprozessen einordnen lassen, erfolgt in der vorliegenden Arbeit somit induktiv⁴⁴⁰. Die der Dokumentenanalyse zeitlich nachgelagerten Experteninterviews beziehen sich auf diese Strukturdimensionen und nicht auf die Steuerungsmaßnahmen im Einzelnen.

Zweck der Dokumentenanalyse hinsichtlich der Gestaltungsbedingungen beim situativen Koordinationsansatz war es, die Kriterien zu identifizieren, mit welchen die Kritikalität der jeweiligen Projektzieldimension bestimmt werden kann. Auf dieser Basis können die übergeordneten Projektzielpräferenzen bestimmt werden. Die Ausgangssituation in der Fallstudienunternehmung erlaubte es, mithilfe von Meta-Dokumenten zur Verwaltung und Beschreibung von Mustervorlagen für die zu erbringenden (Zwischen-)Ergebnisse im Softwareentwicklungsprozess, alle relevanten Dokumente für die vorliegende Fragestellung zu analysieren. Die Untersuchungsbasis bilden 145 Mustervorlagen in Form von elektronischen Dokumenten⁴⁴¹ sowie 9 datenbankgestützte Mustervorlagen als Bestandteile von CASE-Tools. Aufgrund dieser großen Untersuchungsbasis hat die Dokumentenanalyse als ein zweistufiger Prozess stattgefunden. Im ersten Schritt wurde die Zahl der relevanten Quellen auf 46 elektronische Dokumente und 5 datenbankgestützte Mustervorlagen eingeschränkt, welche die folgenden Voraussetzungen zur weiteren Analyse erfüllten:

- Die Mustervorlage muss einen begründeten Zusammenhang zu mindestens einer der Projektzieldimensionen Zeit, Leistungsumfang oder Qualität aufweisen, sodass die darin enthaltenen Informationen potentiell für die Bestimmung der Kritikalität der jeweiligen Projektziele herangezogen werden können. Bspw. erfüllen Mustervorlagen zum Wissens- oder zum Patentmanagement diese Voraussetzung nicht;
- Die Mustervorlage bzw. vielmehr die Informationen, die diese im konkreten Fall enthält, müssen zu Projektbeginn, d.h. spätestens zu Beginn der Phase ‚Analyse‘⁴⁴², verfügbar sein. Bspw. erfüllen Mustervorlagen, die in den Phasen ‚Grob- und Feinentwurf‘ verwendet werden, diese Anforderung nicht.

⁴⁴⁰ Induktive Schlüsse sind Schlüsse vom Besonderen auf das Allgemeine.

⁴⁴¹ Textverarbeitungs-, Tabellenkalkulations- und Präsentationsdokumente.

⁴⁴² Vgl. hierzu sowie zu den Phasen Grob- und Feinentwurf Kap. 2.1.2.1

Die verbliebenen 51 Mustervorlagen wurden in die ausführliche Dokumentenanalyse einbezogen. Dabei sind für die vorliegende Fragestellung nicht die Inhalte auf der Abstraktionsebene der gesamten Mustervorlage von Interesse (bspw. Mustervorlage für den Projektplan), sondern die Inhalte der einzelnen Informationsfelder in einer Mustervorlage (bspw. ein Informationsfeld zu zeitlichen Restriktionen als Kriterium für die Terminzielkritikalität im Projektplan). Die identifizierten Kriterien für die Bestimmung der Projektzielkritikalitäten wurden anhand eines festen Erfassungsschemas dokumentiert (vgl. Abb. 40).

Erfassungsschema	Erläuterung
Mustervorlage	Elektronisches Dokument oder datenbankgestützte Mustervorlage als Bestandteil eines CASE-Tools.
Informationsfeld	Eine Mustervorlage kann mehrere relevante Informationsfelder beinhalten, die alle separat erfasst werden.
Wertebereich der Information	Der Wertebereich definiert die Menge der möglichen Werte, welche das Informationsfeld aufweisen kann. Dabei kann es sich um verschiedene Skalierungen handeln, bspw. ordinal (hoch/mittel/niedrig o.ä.) oder nominal (natürlich-sprachliche Beschreibung eines Sachverhaltes o.ä.).
Bezug zur Projektzieldimension	Die Information lässt sich als Kriterium zur Bestimmung der Qualitäts-, Leistungsumfangs- oder Zeitzielpräferenzen heranziehen.
Normierung auf Projektebene	Alle Informationen, die nicht projektweit gültig sind, müssen valide und nachvollziehbar auf Entwicklungsprojekte bezogen werden können. Diese Normierung ist erforderlich, da schlussendlich Projektzielpräferenzen bestimmt werden sollen.

Abb. 40: Erfassungsschema für Gestaltungsbedingungen bei der Dokumentenanalyse

Um Anhaltspunkte für die mögliche organisatorische Umsetzung des Gesamtansatzes in einer Standardsoftwareunternehmung zu erhalten (Forschungsfrage 3), wurden im Rahmen der Dokumentenanalyse die in der Fallstudienunternehmung bereits vorhandenen Tailoringansätze identifiziert und hinsichtlich ihrer organisatorischen Einbindung analysiert. Grundlage bildeten dabei auch hier Prozess- und Rollenbeschreibungen, wobei im Unterschied zur Ableitung der einzelnen Steuerungsmaßnahmen an dieser Stelle umfassende Tailoringansätze auf der Projekt- und Projektprogrammebene im Fokus standen. In der Fallstudienunternehmung wurden drei relevante Ansätze vorgefunden und analysiert. Abb. 41 stellt das dabei zugrundegelegte Erfassungsschema dar.

Erfassungsschema	Erläuterung
Gestaltungsziel des Tailoringansatzes	Darstellung des Gestaltungsziels, das bei dem Tailoringansatz als ein situativer Ansatz verfolgt wird.
Gestaltungsbedingungen des Tailoringansatzes	Darstellung des Klassifikationsschemas, das die Gestaltungsbedingungen des Tailoringansatzes abbildet.
Gestaltungsmaßnahmen des Tailoringansatzes	Darstellung der vorgesehenen Verfahren zur Anwendung bei den einzelnen Problemklassen.
Vorgehensweise beim Tailoring	Wie in Kap. 2.1.3.1 erläutert kann Tailoring grundsätzlich <ul style="list-style-type: none"> • durch Weglassen, Hinzufügen, Ändern oder Konkretisieren von Prozesselementen, • auf verschiedenen Ebenen (bspw. unternehmensweit oder projektbezogen), • als ein mehr oder weniger formaler Prozess oder gänzlich informal und ad hoc erfolgen. Hierauf wird an dieser Stelle eingegangen.

Abb. 41: Erfassungsschema für Anhaltspunkte zur organisatorischen Umsetzung des Gesamtansatzes bei der Dokumentenanalyse

3.1.2 Vorgehensweise bei den Experteninterviews

Experteninterviews können in das explorative, systematisierende und theoriegenerierende Experteninterview eingeteilt werden. In der vorliegenden Arbeit wurden bei den Experteninterviews als Teil der Fallstudie die Ergebnisse der Dokumentenanalyse, sowie implizit auch das theoretische Vorwissen aus der Entwicklung des Konzeptionsrahmens, offen gelegt und in den Fragekomplexen des Interviews zum Ausdruck gebracht. Die mithilfe der Experteninterviews gewonnenen Erkenntnisse wurden in den Forschungsprozess wieder eingebracht. Damit handelt es sich hier um theoriegenerierende Experteninterviews. In diesem Fall dient der Experte nicht nur zur Gewinnung von sachdienlichen Informationen, sondern es wird ausgehend von der Vergleichbarkeit der Expertenäußerungen, die methodisch durch den Interviewleitfaden und empirisch durch die gemeinsame organisatorisch-institutionelle Anbindung der Experten gesichert ist, eine theoretisch gehaltvolle Konzeptualisierung von Wissensbeständen angestrebt.⁴⁴³

Im Rahmen der Fallstudie fanden 17 Experteninterviews in einem Umfang von jeweils 60-75 min. statt. Die qualitative empirische Forschung kann keine Zufallsstichproben praktizieren, weil diese Verfahren eine große Zahl von Untersuchungsobjekten voraussetzen. Dies wäre bei Experteninterviews auch nicht realisierbar.⁴⁴⁴ Daher erfolgte ein sog. theoretical sampling: Die Experten wurden vom Verfasser der Arbeit mit der Bitte

⁴⁴³ Vgl. Bogner, Menz (2002), S. 37-39

⁴⁴⁴ Vgl. Lamnek (2005), S. 265

um Teilnahme kontaktiert.⁴⁴⁵ Die Auswahl der Teilnehmer orientierte sich dabei ausschließlich an inhaltlichen Relevanzkriterien.⁴⁴⁶ Als Kriterium diente hier die Rolle des Experten – es wurden nur Mitarbeiter in den Rollen ‚Projektprogrammleiter‘ oder ‚Prozessverantwortlicher‘ als Experten herangezogen (vgl. Abb. 42).

Inter-view	Rolle des Experten	Organisationseinheit des Experten⁴⁴⁷
Experte ‚A.D.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Enterprise Resource Planning‘
Experte ‚B.E.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Supply Chain Management‘
Experte ‚C.S.‘	Prozessverantwortlicher	Stabsbereich ‚Process Governance‘ für den Entwicklungsbereich ‚Branchenspezifische Varianten der integrierten Standardsoftwarelösung‘
Experte ‚F.E.‘	Prozessverantwortlicher	Stabsbereich ‚Process Governance‘ für den Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen‘
Experte ‚J.C.‘	Projektprogrammleiter	Entwicklungsbereich ‚Branchenspezifische Varianten der integrierten Standardsoftwarelösung‘
Experte ‚J.K.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Supply Chain Management‘
Experte ‚J.W.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Supplier Relationship Management‘
Experte ‚K.H.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Customer Relationship Management‘
Experte ‚L.G.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Supplier Relationship Management‘
Experte ‚M.G.‘	Prozessverantwortlicher	Zentraler Stabsbereich ‚Process Governance‘
Experte ‚O.S.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Enterprise Resource Planning‘
Experte ‚R.L.‘	Prozessverantwortlicher	Stabsbereich ‚Process Governance‘ für den Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen – Anwendung: Enterprise Resource Planning‘
Experte ‚T.D.‘	Prozessverantwortlicher	Stabsbereich ‚Process Governance‘ für den Entwicklungsbereich ‚Plattformlösung‘
Experte ‚T.M.‘	Projektprogrammleiter	Entwicklungsbereich ‚Branchenspezifische Varianten der integrierten Standardsoftwarelösung‘
Experte ‚T.S.‘	Projektprogrammleiter	Entwicklungsbereich ‚Plattformlösung‘
Experte ‚T.Si.‘	Projektprogrammleiter	Entwicklungsbereich ‚Integrierte Standardsoftwarelösung für große Unternehmen - Anwendung: Supply Chain Management‘
Experte ‚T.W.‘	Prozessverantwortlicher	Zentraler Stabsbereich ‚Process Governance‘

Abb. 42: Übersicht der durchgeführten Experteninterviews

⁴⁴⁵ Kontaktaufnahme über E-Mail; Der Verfasser der Arbeit war zum Forschungszeitraum Mitarbeiter (Doktorand) bei der Fallstudienunternehmung im zentralen Stabsbereich ‚Process Governance‘.

⁴⁴⁶ Vgl. hierzu Froschauer, Lueger (2003), S. 54f.

⁴⁴⁷ Vgl. Kap. 3.1 zum Produktportfolio der Fallstudienunternehmung

Da nach der Durchführung der 17 Experteninterviews die sog. theoretische Sättigung eintrat, konnte das Sampling beendet werden. D.h., dass die bei den letzten Experteninterviews neu gewonnenen empirischen Befunde nicht mehr zu einer Veränderung oder Weiterentwicklung der Forschungsergebnisse beitragen konnten. Vielmehr stabilisierten sich die Interpretationen und ließen sich unter die bereits entwickelten Forschungsergebnisse fassen. Die theoretische Sättigung ist ein valides forschungspragmatisches Abbruchkriterium. Ab dem Zeitpunkt der theoretischen Sättigung würde der Aufwand für weitere Analysen den daraus zu ziehenden Erkenntnisgewinn übersteigen.⁴⁴⁸

Obwohl das Experteninterview im Grunde lediglich durch Verweis auf die Qualität des Interviewpartners als Experte spezifiziert ist, ist darunter üblicherweise, und so auch in der vorliegenden Arbeit, ein Leitfadeninterview zu verstehen.⁴⁴⁹ Damit sind Experteninterviews insgesamt in dreifacher Hinsicht gekennzeichnet.⁴⁵⁰

- Zielgruppe: Die Befragten sind Experten für einen bestimmten Gegenstandsreich. Dabei interessiert weniger die Person des Befragten, sondern ausschließlich sein Erfahrungswissen über das untersuchte Thema sowie sein Wissen über die Strukturzusammenhänge, organisationalen Entscheidungsstrukturen und Problemlösungen in der Fallstudienunternehmung;
- Relativ starke Fokussierung der Inhalte des Interviews: Experteninterviews haben eine klar definierte inhaltliche Ausrichtung, die durch einen Leitfaden vorgegeben ist. Der hier verwendete Leitfaden orientiert sich an den zu beantwortenden Forschungsfragen und damit an Themenblöcken, die analog zum Aufbau des Entscheidungsrahmens gestaltet sind (vgl. Abb. 43);
- Hohes Maß an Pragmatik der Interviewführung: So kann bspw. bei Bedarf durch eine entsprechend offene Handhabung des Leitfadens auch eine selbstläufige Schilderung der Experten gewährleistet werden. Dieser Aspekt ist bei theoriegenerierenden Experteninterviews besonders relevant.

⁴⁴⁸ Vgl. Lamnek (2005), S. 189

⁴⁴⁹ Vgl. Bogner, Menz (2002), S. 19f.; vgl. Liebold, Trinczek (2002), S. 33f.

⁴⁵⁰ Vgl. Flick (2006), S. 218f.; vgl. Liebold, Trinczek (2002), S. 39, 41

Themenblock	Frage hinsichtlich ..
Gestaltungsbedingungen des situativen Koordinationsansatzes	A1: .. der Projektziele als Gestaltungsbedingung A2: .. übergeordneter Projektzielpräferenzen als Gestaltungsbedingung A3: .. der Präferenzbestimmung der Projektzieldimension Zeit A4: .. der Präferenzbestimmung der Projektzieldimension Leistungsumfang A5: .. der Präferenzbestimmung der Projektzieldimension Qualität
Gestaltungsmaßnahmen des situativen Koordinationsansatzes	B1: .. der Strukturdimension Artefaktspezifikation B2: .. der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘ B3: .. der Strukturdimension Kompetenzverteilung
Organisatorische Umsetzung des situativen Koordinationsansatzes	C1: .. der organisatorischen Umsetzung durch systematisches Tailoring über Projekttypen
Abschließende Bewertung des situativen Koordinationsansatzes	D1: .. der Umsetzbarkeit und Vorteilhaftigkeit des Gesamtansatzes

Abb. 43: Interviewleitfaden für die Experteninterviews

Die konkrete Ausgestaltung und Transkription der Interviews hängen vom angestrebten Analyseverfahren und Auswertungszweck ab. Die qualitative empirische Forschung verfügt über ein vielfältiges Repertoire an Analyseverfahren, die sich nicht alle für jede Gesprächsführung und für jedes Erkenntnisinteresse eignen.⁴⁵¹ Bei den theoriegenerierenden Experteninterviews in der vorliegenden Arbeit liegt das Erkenntnisinteresse beim manifesten Gehalt der Aussagen. Als Auswertungsverfahren dienen hier themenanalytische Verfahren. Sie sind aus methodischer Sicht am wenigsten anspruchsvoll, denn hier sind lediglich die zusammenfassende Aufbereitung von Inhalten zu den verschiedenen Themen angezeigt. Fragen nach der Rechtfertigung von Aussagen der befragten Personen, Bedeutungen von Wortbetonungen, Bedeutungen von bestimmten sprachlichen Besonderheiten⁴⁵² oder auch von gewählten grammatikalischen Konstruktionen⁴⁵³ sind hier nicht von Interesse. Entsprechend bedarf es auch keiner Transkription von Informationen über die Interviewaussagen hinaus für die eben genannten Fragestellungen.⁴⁵⁴ Ebenso orientiert sich auch die konkrete Ausgestaltung der Interviewführung und des Interviewleitfadens am Auswertungszweck:

⁴⁵¹ Vgl. Froschauer, Lueger (2003), S. 89f.

⁴⁵² Wortverwendungen, Wiederholungen, Satzabbrüche, Versprecher, Sprachpausen etc.

⁴⁵³ Aktiv, Passiv, Konditionalform etc.

⁴⁵⁴ Vgl. Froschauer, Lueger (2003), S. 90f., 114, 117, 158f.

- Die Interviews wurden als neutrale Interviews geführt. Bei diesem Modell des gesamten Interviewverhaltens werden durch eine sachliche und unpersönliche Beziehung uniforme Interviewsituationen angestrebt. Damit soll die Vergleichbarkeit der Informationen erhöht werden;⁴⁵⁵
- Es wurden nicht-standardisierte Interviews mit offenen Fragen geführt: Die Unterscheidung zwischen einem standardisierten und einem nicht-standardisierten Interview bezieht sich auf die Verwendungsweise von Antwortkategorien. Als standardisiert werden Fragen bezeichnet, deren Antworten in Kategorien zusammengefasst werden. Bei nicht-standardisierten Fragen wird entweder auf eine Kategorisierung der Antworten verzichtet oder sie wird, wie zum Teil in der vorliegenden Arbeit, später vollzogen. Offene Fragen enthalten grundsätzlich keine festen Antwortkategorien und die befragte Person formuliert ihre Antwort selbständig. Der Interviewer hat die Aufgabe, die Antworten zunächst zu notieren. Sie werden ggf. erst im Nachhinein bei der Auswertung bestimmten Antwortkategorien zugeordnet.⁴⁵⁶

Bei der Durchführung der Experteninterviews stand einer Tonbandaufzeichnung das explizite Interesse der Interviewpartner und der Fallstudienunternehmung entgegen. Die Interviewaussagen wurden daher unter pragmatischen Gesichtspunkten und ausgerichtet auf den Auswertungszweck dokumentiert. So wurden nur die für die Fragestellung relevanten Teile der Experteninterviews selektiv transkribiert und teilweise auch direkt paraphrasiert. Dabei wurde ein wenig aufwendiges Notationssystem verwendet, das auf parasprachliche Elemente etc. nicht eingeht. Denn mit dem manifesten Gehalt der Aussagen sind, wie bereits erläutert, alle Informationen enthalten, die bei der Interpretation genutzt werden.⁴⁵⁷

Die Datenaufbereitung erfolgte in Form einer dreistufigen Inhaltsanalyse:⁴⁵⁸

- Im ersten Schritt wurde der Ausgangstext durch die sog. zusammenfassende Inhaltsanalyse auf einen Umfang reduziert, der nur noch die relevanten Inhalte umfasst. Zu den Arbeitsgängen der zusammenfassenden Inhaltsanalyse gehören Paraphrasierung⁴⁵⁹, Generalisierung⁴⁶⁰ und Bündelung⁴⁶¹;

⁴⁵⁵ Vgl. Atteslander (2003), S. 149-152

⁴⁵⁶ Vgl. Atteslander (2003), S. 160ff.

⁴⁵⁷ Vgl. Liebold, Trinczek (2002), S. 47-49; vgl. Flick (2006), S. 220

⁴⁵⁸ Vgl. Mayring (2002), S. 94ff.; vgl. Bortz, Döring (2006), S. 332

⁴⁵⁹ Wegstreichen ausschmückender Redewendungen, Transformation auf grammatikalische Kurzformen.

⁴⁶⁰ Konkrete Beispiele werden verallgemeinert.

⁴⁶¹ Ähnliche Paraphrasen werden zusammengefasst.

- Einhergehend mit der zusammenfassenden Inhaltsanalyse erfolgte die sog. explizierende Inhaltsanalyse. Hier wurden bspw. unklare Begriffe und Sätze verständlich gemacht, indem zusätzliche Materialien wie andere Interviewpassagen oder weitere Informationen im Nachhinein auf Nachfrage von den Befragten herangezogen wurden;
- Die sog. strukturierende Inhaltsanalyse ist als fließender Übergang zwischen der Datenaufbereitung und -analyse zu sehen. Hier wurde die zusammenfassende und explizierte Kurzversion der Interviewdokumentation wie folgt geordnet und ausgewertet.

Wie bereits erwähnt können Experteninterviews im Prozess der Dateninterpretation je nach Erkenntnisabsicht unterschiedlich verwendet werden. Die erste Grundsatzfrage stellt sich bez. der Analyseebene: Mitunter interessiert der einzelne Experte als besonderer Fall, oder, wie in der vorliegenden Arbeit, stellt vielmehr die Organisation die Analyseebene dar und die durchgeführten Experteninterviews werden zu einer Organisationsfallstudie verdichtet. Diese Verdichtung erfolgte im Rahmen einer sog. querdimensionalen Analyse.⁴⁶²

Bei der querdimensionalen Analyse liegt am Ende der Datenaufbereitung das gesamte Interviewmaterial inhaltlich nach den jeweiligen Themenblöcken des Leitfadens geordnet vor. „Dies bedeutet nicht nur, dass die Sequenzialität des einzelnen Interviews aufgehoben ist, sondern auch, dass das Einzelinterview zwar immer noch als Fallgestalt präsent ist, gleichzeitig aber auch textlich aufgelöst wird und im weiteren Fortgang der Arbeit teilweise nicht mehr als weitere Arbeitsgrundlage dient. Gearbeitet wird nun wesentlich mit dem aus allen Interviews zu jeweils einem spezifischen Teil-Thema extrahierten Material, wobei .. immer wieder zur weiteren Kontextualisierung der Einzeltexte auf die Fallgestalt zurückgegriffen wird.“⁴⁶³

Die Teil-Themen werden als einzelne inhaltliche Dimensionen des Gesamtmaterials bezeichnet. Bei der querdimensionalen Analyse geht es entsprechend um den systematischen Vergleich der inhaltlichen Passagen und um die Rekonstruktion von ihren typischen Ausprägungen. Bei der Interpretation des Materials wurden in der vorliegenden Fallstudie in einem ersten Schritt für jedes Teil-Thema die entsprechenden Passagen aus den verschiedenen Interviews systematisch inhaltlich miteinander in Bezug gesetzt, um jeweils Gemeinsamkeiten und Unterschiede herauszuarbeiten und diese für jedes Teil-Thema typologisierend zu verdichten.⁴⁶⁴ Bei einer ausgesprochen qualitati-

⁴⁶² Vgl. im Folgenden Liebold, Trinczek (2002), S. 50-55

⁴⁶³ Liebold, Trinczek (2002), S. 52f.

⁴⁶⁴ In Anlehnung an Liebold, Trinczek (2002)

ven Analyse wie der querdimensionalen Analyse interessiert es weniger, „wie ein Problem statistisch verteilt ist, sondern welche Probleme es tatsächlich gibt und wie sie beschaffen sind. Während im Rahmen einer quantitativ angelegten Studie wenige abweichende Fälle gleichsam untergehen, weil sie quantitativ unbedeutend erscheinen, können gerade sie innerhalb einer qualitativen Studie die Theorieentwicklung weiter vorantreiben.“⁴⁶⁵ In diesem Zusammenhang orientiert sich die vorliegende Fallstudie bei der querdimensionalen Analyse am Verfahren der analytischen Induktion.⁴⁶⁶

3.2 Gestaltungsmaßnahmen der Koordination

Das im Konzeptionsrahmen zugrunde gelegte Strukturierungskonzept von Laux verdeutlicht zwar grundsätzlich, wie über die verschiedenen Entscheidungsdeterminanten Einfluss auf nachgeordnete Entscheidungen in Koordinationsprozessen genommen werden kann, ohne diese aber vorwegzunehmen. Jede Steuerungsmaßnahme setzt sich nämlich aus bestimmten Ausprägungen dieser ‚Basisvariablen organisatorischer Gestaltung‘ zusammen und lässt sich so in die Vielfalt der potentiellen Steuerungsmöglichkeiten einordnen. Es bleibt jedoch weitgehend offen, welche konkreten Gestaltungsmaßnahmen dafür in Frage kommen. Es können Anweisungen und Richtlinien jeder erdenkbaren Art sein, mit denen der Entscheidungsprozess eines nachgeordneten Entscheidungsträgers beeinflusst werden kann.⁴⁶⁷

Im vorliegenden Kap. 3.2 werden aufbauend auf dem Konzeptionsrahmen und damit auf dem Strukturierungskonzept von Laux die konkreten Gestaltungsmaßnahmen für die situative Ausrichtung der Koordinationsprozesse entwickelt. Ausgehend von den identifizierten (potentiellen) Steuerungsmaßnahmen in der Fallstudienunternehmung werden dabei drei sog. Strukturdimensionen gebildet, auf die in den folgenden Kapiteln 3.3.1 bis 3.3.3 eingegangen wird. Diese Strukturdimensionen sind „als unterschiedliche Arten von organisatorischen Regelungen zu begreifen, deren jeweilige Kombination die Organisationsstruktur ausmacht.“⁴⁶⁸ Durch die Bildung von Strukturdimensionen wird ein Möglichkeits- bzw. Merkmalsraum geschaffen, in dem die entsprechenden Ausprägungen realer Organisationsstrukturen abgebildet werden können. Dabei wird eine Komplexitätsreduktion der betrachteten Phänomene angestrebt und nicht die Wiedergabe der Realität in ihrer gesamten Vielfalt und Komplexität.⁴⁶⁹

⁴⁶⁵ Lamnek (2005), S. 188

⁴⁶⁶ Vgl. Kap. 3.1.1

⁴⁶⁷ Vgl. Laux, Liermann (2003), S. 151f., 154

⁴⁶⁸ Kieser, Walgenbach (2003), S. 71

⁴⁶⁹ Vgl. Kieser, Walgenbach (2003), S. 71f., 74; Eine Zusammenstellung von Ansätzen zur Systematisierung von Koordinationsinstrumenten i.S.v. Strukturdimensionen findet sich bei Lühring (2006), S. 85f., mit zahlreichen weiteren Verweisen.

Streng genommen können Strukturdimensionen der Organisationsgestaltung weder begründet, noch abgeleitet werden. Ihre Rechtfertigung erhalten sie aus der Fruchtbarkeit ihrer Anwendung in der Analyse, Erklärung und Präskription der Organisationsgestaltung. Dazu müssen alle Kombinationen von Gestaltungsmaßnahmen innerhalb der Strukturdimensionen sinnvoll, wenn auch nicht immer vorteilhaft sein. Voraussetzung hierzu ist, dass die Strukturdimensionen möglichst unabhängig voneinander sind, sowie so voneinander abgegrenzt werden können, dass substantielle und überprüfbare Hypothesen über die Wirkung der verschiedenen Ausprägungskombinationen formuliert werden können.⁴⁷⁰

Beim situativen Koordinationsansatz werden die drei Strukturdimensionen Artefakt-spezifikation, Kompetenzverteilung, sowie ‚Personelle und organisatorische Unabhängigkeit‘ gebildet. Abb. 44 greift den nachfolgenden Ausführungen vor, indem sie die drei Strukturdimensionen definiert und zeigt, über welche der Entscheidungs-determinanten nach Laux in den jeweiligen Strukturdimensionen Einfluss auf die nachgeordneten Entscheidungen in Koordinationsprozessen genommen wird.

Struktur-dimension	Definition	Einflussnahme primär über Entscheidungs-determinante⁴⁷¹
Artefakt-spezifikation	Organisatorische Regelungen zur Informationsstruktur von Arbeitsergebnissen (Artefakten) der Koordinationsprozesse	Informationsstruktur, Menge der erwogenen Alternativen
Kompetenz-verteilung	Organisatorische Regelungen zur Verteilung von Entscheidungskompetenzen zwischen dem Projekt-, Qualitäts- und Produktmanagement in einem Entwicklungsvorhaben	Typ des Entscheidungsmodells
‚Personelle und organisatorische Unabhängigkeit‘	Organisatorische Regelungen zur personellen und organisatorischen Unabhängigkeit des Projekt-, Qualitäts- und Produktmanagements in einem Entwicklungsvorhaben	Zielfunktion

Abb. 44: Strukturdimensionen der Organisationsgestaltung beim situativen Koordinationsansatz

3.2.1 Strukturdimension Artefakt-spezifikation

3.2.1.1 Steuerungsmaßnahmen in der Strukturdimension Artefakt-spezifikation

Spezifikation von Arbeitsergebnissen bezeichnet bei Mellis ganz allgemein die Forderung, dass alle Arbeitsergebnisse in Abhängigkeit von den zu sichernden Erfordernissen durch die Koordination bestimmte Merkmale besitzen und in einer bestimmten

⁴⁷⁰ Vgl. Mellis (2004), S. 62f.

⁴⁷¹ Vgl. Kap. 2.3.2.3

Anzahl bereitgestellt werden müssen.⁴⁷² In der Fallstudienunternehmung wurden drei Steuerungsmaßnahmen identifiziert, die in diesem Sinn die Bildung der Strukturdimension Artefaktspezifikation begründen. Beim situativen Koordinationsansatz wird Artefaktspezifikation allerdings konkreter als bei Mellis gefasst: Die Informationsstruktur der Arbeitsergebnisse (Artefakte) der Koordinationsprozesse muss in Abhängigkeit von den übergeordneten Projektzielpräferenzen gestaltet werden. Artefaktspezifikation bündelt damit alle konkreten Gestaltungsmaßnahmen, die nach dem Strukturierungskonzept von Laux über die Beeinflussung der Informationsstruktur und der Menge der erwogenen Alternativen Koordinationseffektivität sicherstellen sollen (vgl. Abb. 45). Im Folgenden erfolgt die Auseinandersetzung mit der Strukturdimension Artefaktspezifikation aus der Sicht von jedem der vier Koordinationsprozesse der Standardsoftwareentwicklung.

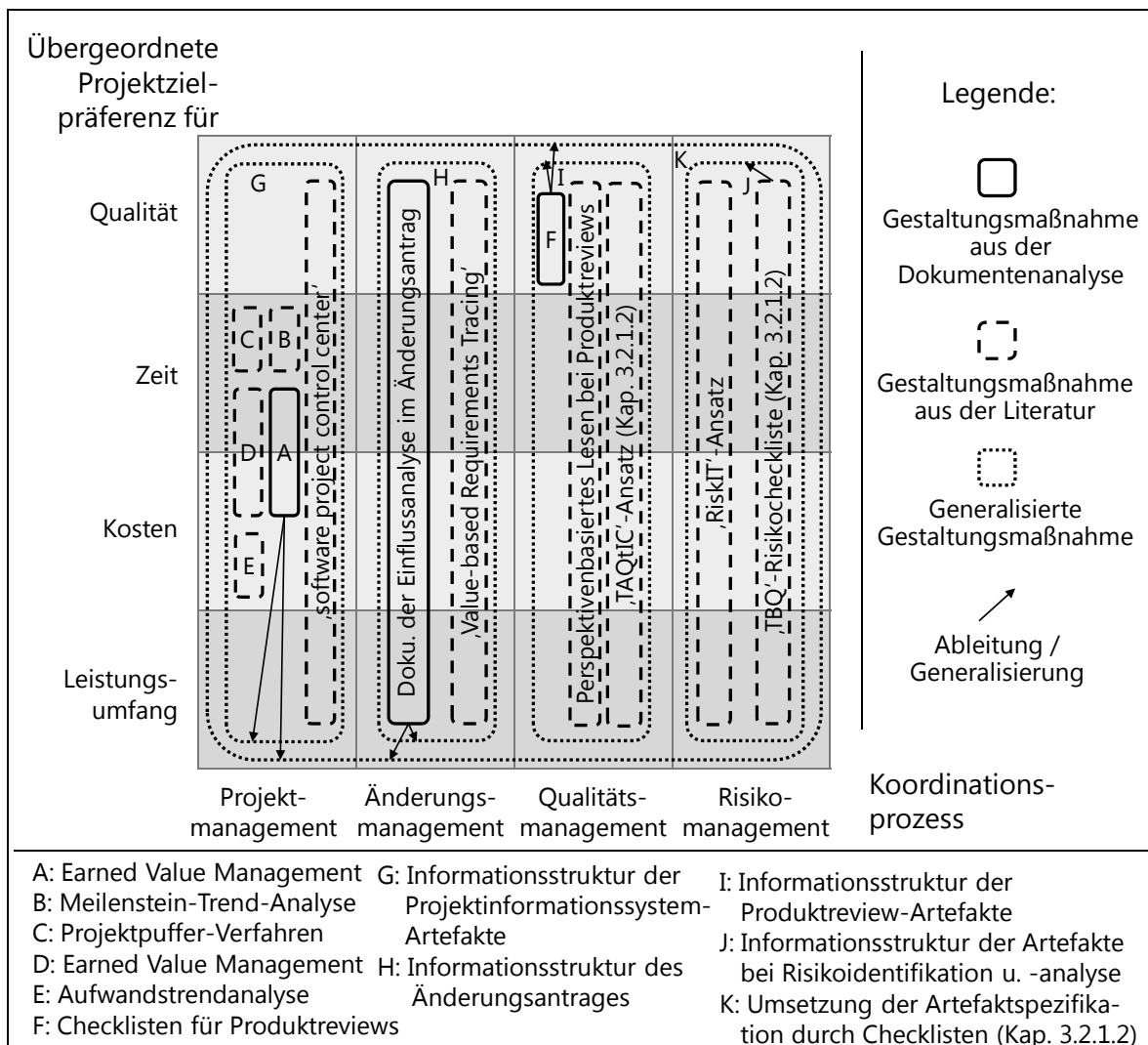


Abb. 45: Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation

⁴⁷² Vgl. Mellis (2004), S. 135

Grundlage der Projektsteuerung sind Pläne der laufenden Projekte, auf deren Basis die Ermittlung der Ist-Daten und ihre Gegenüberstellung mit den Plandaten, die Untersuchung der aufgetretenen Abweichungen, sowie die Planung und Einleitung von Gegenmaßnahmen erfolgen. Dabei müssen die vier Projektzieldimensionen bei Abweichungsanalysen zwar grundsätzlich gemeinsam betrachtet werden,⁴⁷³ doch im Rahmen des Projektinformationssystems⁴⁷⁴ ergeben sich Ansatzpunkte zur situativen Spezifikation der Artefakte und damit implizit auch zur situativen Anwendung der entsprechenden Methoden.

So ist bei der Fallstudienunternehmung die Überwachung des Projektfortschrittes mit der Methode Earned Value Management (EVM) situativ bei zeit- und kostenkritischen Entwicklungsprojekten vorgesehen. Als solche werden Projekte mit über 200 geplanten Manntagen Aufwand für Entwurf und Implementierung der Standardsoftware definiert (vgl. Abb. 46). EVM bedient sich der Kennzahlen ‚Ist-Aufwand‘, ‚Planaufwand‘ (‚Earned Value‘) und ‚Soll-Aufwand‘ und ermittelt damit durch festgelegte Bewertungsmethoden die Kosten- und Zeitabweichung eines Projektes. Mit den beiden errechneten Kennzahlen ‚Cost Variance‘ und ‚Schedule Variance‘ lassen sich verlässliche Voraussagen über den voraussichtlichen Fertigstellungstermin und Gesamtaufwand des Projektes machen.⁴⁷⁵

Steuerungsmaßnahme	Situative Überwachung und visuelle Darstellung des Projektfortschrittes mittels EVM bei zeit- und kostenkritischen Entwicklungsprojekten
Zuordnung zum Koordinationsprozess	Projektmanagement
Tailoring der Steuerungsmaßnahme	Verpflichtend für Projekte mit mehr als 200 geplanten Manntagen Aufwand für Entwurf und Implementierung
Bezug zur Projektzieldimension	Zeit; Kosten

Abb. 46: EVM als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation

Ähnlich wie bei der Fallstudienunternehmung ist EVM bei Softwareentwicklungsvorhaben des US-Amerikanischen Verteidigungsministeriums für Entwicklungsprojekte ab einem bestimmten Projektumfang verpflichtend. Je nach Ausmaß der Zeit- und Kostenkritikalität kommt hier aber nochmals eine situative Anpassung von EVM in-

⁴⁷³ Vgl. Fiedler (2010), S. 176f.

⁴⁷⁴ Auch: Projektberichtswesen

⁴⁷⁵ Vgl. Drews, Hillebrand (2007), S. 232; Fiedler schlägt die Anwendung der EVA für kostenkritische Projekte grundsätzlich vor, d.h. abgesehen von der Terminkritikalität. Vgl. Fiedler (2010), S. 176; zur Darstellung der Methode vgl. bspw. Drews, Hillebrand (2007), S. 231ff.

nerhalb des verpflichtenden Projektumfangs hinzu. EVM muss dann ggf. zahlreichen vorgegebenen Anforderungen wie bspw. einer kontinuierlichen Kontrolle genügen. In definierten Fällen muss gar eine formale Validierung der Anwendung von EVM erfolgen.⁴⁷⁶

Neben EVM wird in der Projektmanagement-Literatur die Anwendung einiger weiterer Methoden in Abhängigkeit von den übergeordneten Projektzielpräferenzen vorgeschlagen:

- Meilenstein-Trend-Analyse bei zeitkritischen Projekten,⁴⁷⁷ da hier eine einfache und übersichtlich visualisierte Darstellung des Trends von Meilensteinterminen und dessen Exploration in die nähere Zukunft erfolgt;⁴⁷⁸
- Aufwandstrendanalyse bei kostenkritischen Projekten, da hier eine kontinuierliche Bewertung der erbrachten Leistung im Verhältnis zum Aufwand erfolgt;⁴⁷⁹
- Projektpuffer-Verfahren bei zeitkritischen Projekten, da diese Methode ein differenziertes Kontrollinstrument hinsichtlich der Termine bietet.⁴⁸⁰

Deutlich weniger Raum in der Literatur nehmen dagegen umfassende situative Projektinformationssystem-Ansätze ein. Hierzu liefert die Arbeit von Münch und Heidrich erste Ansatzpunkte. Die Autoren stellen einen Referenzrahmen für im situativen Sinn verstandene Projektinformationssysteme vor, dort ‚software project control center‘ genannt. Der Referenzrahmen ist als eine zweidimensionale Matrix aufgebaut, bei der die beiden Dimensionen ‚Art der Datenverarbeitung‘ und ‚Art der Datenvisualisierung‘ als Dimensionen der Situativität von Projektinformationssystemen zu verstehen sind. Sie können jeweils drei Ausprägungen einnehmen, die sich in ihrem Grad der Situativität unterscheiden (vgl. Abb. 47).⁴⁸¹ Münch und Heidrich ordnen in ihrer Arbeit existierende Projektinformationssystem-Ansätze in den Referenzrahmen ein und legen so Forschungsbedarf ganz im Sinn der Strukturdimension Artefaktspezifikation offen: „Purpose- and role-oriented usages based on a flexible set of techniques and methods are not comprehensively supported.“⁴⁸²

⁴⁷⁶ Vgl. Patton (2007), S. 11

⁴⁷⁷ Vgl. Fiedler (2010), S. 176

⁴⁷⁸ Vgl. Drews, Hillebrand (2007), S. 247ff.; vgl. Fiedler (2010), S. 191f.

⁴⁷⁹ Vgl. Drews, Hillebrand (2007), S. 219ff.

⁴⁸⁰ Vgl. Drews, Hillebrand (2007), S. 260ff.

⁴⁸¹ Vgl. Münch, Heidrich (2004), S. 10

⁴⁸² Vgl. Münch, Heidrich (2004), S. 17

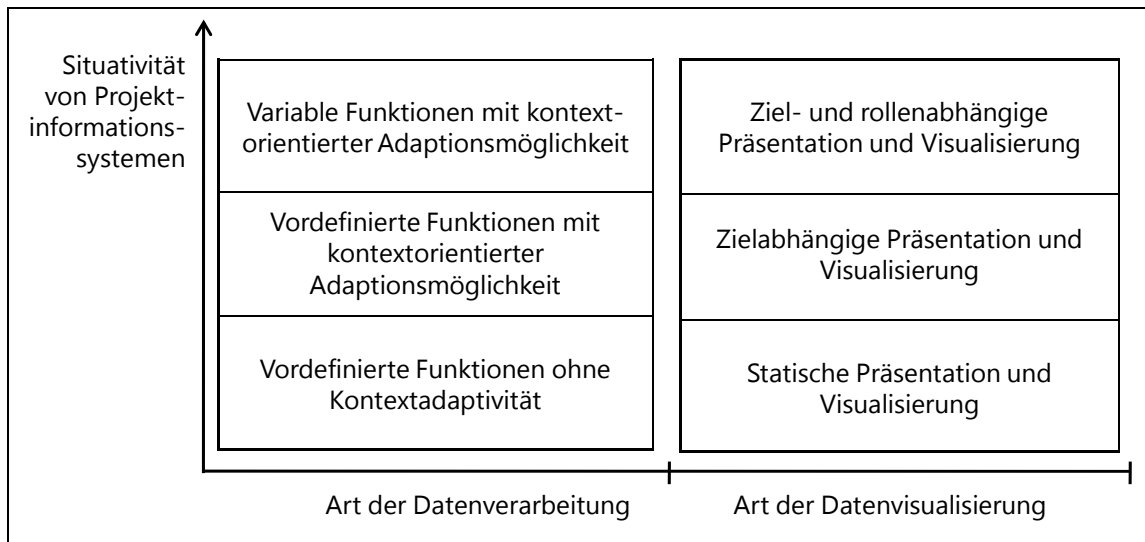


Abb. 47: Referenzrahmen für situative Projektinformationssystem-Ansätze⁴⁸³

Im Rahmen eines systematischen Änderungsverfahrens müssen alle notwendigen Informationen zur Anforderungsänderung erfasst werden. Das wird üblicherweise über die Erstellung eines Änderungsantrags sichergestellt.⁴⁸⁴ Insbesondere müssen im Änderungsantrag die Auswirkungen der Änderung auf andere zu erstellende Leistungen sowie auf die Kosten- und Terminsituation im Projekt analysiert werden. Dies ist der Gegenstand der Einflussanalyse einer Anforderungsänderung.⁴⁸⁵ Bei der Fallstudienunternehmung muss die Dokumentation der Einflussanalyse im Änderungsantrag einer vorgegebenen Struktur folgen, welche sich auch auf diverse kritikalitätsbezogene Informationen zur Anforderungsänderung bezieht (vgl. Abb. 48). Durch eine derartige situative Informationsstruktur des Änderungsantrages in Abhängigkeit von den übergeordneten Projektzielpräferenzen ergibt sich hier der Ansatzpunkt zur Umsetzung der Strukturdimension Artefaktspezifikation beim Änderungsmanagement.

Mit der situativen Ausgestaltung der Informationsstruktur von Änderungsanträgen muss notwendigerweise auch eine Situativität der Anforderungsnachvollziehbarkeit einhergehen, welche v.a. die Einflussanalyse erst ermöglicht. Dies ist als Teil der festzulegenden Nutzungsstrategie für Nachvollziehbarkeitsinformationen zu sehen. Die Nutzungsstrategie legt fest, in welcher Situation welche Nachvollziehbarkeitsinformationen durch welche Rolle zur Durchführung welcher Aktivität benötigt werden. Eine situative Nutzungsstrategie erfordert eine ebenso situative Aufzeichnungsstrategie. Diese legt wiederum fest, in welcher Situation welche Nachvollziehbarkeitsinformationen durch welche Rolle in welcher Form aufgezeichnet werden müssen.⁴⁸⁶

⁴⁸³ Vgl. Münch, Heidrich (2004), S. 10

⁴⁸⁴ Auch: Change Request; vgl. Bea u.a. (2008), S. 266; vgl. Schelle u.a. (2005), S. 239, 241

⁴⁸⁵ Vgl. Jönsson, Lindvall (2005), S. 117f.

⁴⁸⁶ Vgl. Pohl (2007), S. 505f., 516-521

Steuerungsmaßnahme	Situative Informationsstruktur des Änderungsantrages und darin insb. der Einflussanalyse in Abhängigkeit von den übergeordneten Projektzielpräferenzen. Folgende, auf die Projektzielpräferenzen bezogene Informationen zur Anforderungsänderung werden erfasst: <ul style="list-style-type: none"> • Auswirkungen der Anforderungsänderung auf interne und externe Terminzusagen, Meilensteintermine etc. (Zeitkritikalität); • Auswirkungen der Anforderungsänderung auf Kosten (Kostenkritikalität); • Auswirkungen der Anforderungsänderung auf Konformität zum unternehmensweiten Qualitätsmodell und auf die Qualitätsziele, sowohl bei Annahme, als auch bei Ablehnung der Anforderungsänderung (Qualitätskritikalität); • Auswirkungen auf den Leistungsumfang bei Ablehnung der Anforderungsänderung (Kritikalität des Leistungsumfanges).
Zuordnung zum Koordinationsprozess	Änderungsmanagement
Tailoring der Steuerungsmaßnahme	Bei der Fallstudienunternehmung ad hoc; systematisches Tailoring der Steuerungsmaßnahme zu Projektbeginn jedoch grundsätzlich möglich
Bezug zur Projektziel-dimension	Zeit, Kosten, Qualität, Leistungsumfang

Abb. 48: Informationsstruktur des Änderungsantrages als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation

Eine Vorgehensweise zur situativen Festlegung von Aufzeichnungsstrategien, bei der auch die Projektzielpräferenzen eine Gestaltungsbedingung darstellen, wird von Heindl und Biffel vorgestellt. Ihr Ansatz ‚Value-based Requirements Tracing‘ ist dem Forschungsprogramm VBSE zuzuordnen. „The goal of the value-based requirements tracing .. is to identify traces based on prioritized requirements and thus to identify which traces are more important and valuable than others.“⁴⁸⁷ Hier wird also die Wichtigkeit einer Nachvollziehbarkeitsinformation v.a. aus der Kritikalität der Anforderung heraus bestimmt, deren Lebenszyklus nachverfolgt werden soll. Hiervon abhängig sind dann als Gestaltungsmaßnahmen der Aufzeichnungsstrategie die Anzahl und der Detaillierungsgrad der Nachvollziehbarkeitsinformationen.⁴⁸⁸

Produktreviews stellen die zentralen Koordinationsphasen innerhalb der Qualitätslenkung dar. Bei der Fallstudienunternehmung werden Produktreviews durch Checklisten unterstützt, die in einer generischen Form für jedes der unternehmensweit definierten Qualitätsteilmerkmale vorliegen. Sie werden im Rahmen der Produktreview-Planung situativ in Abhängigkeit von der zu Projektbeginn durchgeführten Qualitätszielbestimmung zusammengestellt.⁴⁸⁹ So soll erreicht werden, dass Produktreviews v.a. die hinsichtlich der kritischen Qualitätsteilmerkmale relevanten potentiellen Fehler adressieren (vgl. Abb. 49).

⁴⁸⁷ Heindl, Biffel (2005), S. 62

⁴⁸⁸ Vgl. Heindl, Biffel (2005), S. 62ff.

⁴⁸⁹ Zum Qualitätsmodell der Fallstudienunternehmung u. den ‚focus product standards‘ vgl. Kap. 3.3.2

Steuerungsmaßnahme	Unterstützung von Produktreviews durch Checklisten, die in einer generischen Form für jedes der unternehmensweit definierten Qualitätsteilmerkmale vorliegen
Zuordnung zum Koordinationsprozess	Qualitätsmanagement
Tailoring der Steuerungsmaßnahme	Situative Zusammenstellung im Rahmen der Produktreview-Planung und in Abhängigkeit von der zu Projektbeginn durchgeführten Qualitätszielbestimmung
Bezug zur Projektziel-dimension	Qualität

Abb. 49: Produktreview-Checklisten als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation

In der Praxis haben sich zahlreiche Varianten von Produktreviews, teilw. auch als Inspektionen bezeichnet, herausgebildet.⁴⁹⁰ Damit ist gleichzeitig auch wesentlicher Forschungsbedarf entstanden. So bemängeln Denger und Olsson das Fehlen von Ansätzen zur Entscheidungsunterstützung für die folgenden zentralen Fragestellungen und Entscheidungssituationen in Zusammenhang mit dem Produktreview-Prozess:⁴⁹¹

- Welche Gestaltungsbedingungen beeinflussen den Produktreview-Prozess?
- Welche der am Entwicklungsvorhaben beteiligten Rollen sind in welchen Fällen in den Produktreview-Prozess mit einzubeziehen?
- Wie soll das Tailoring des Produktreview-Prozesses stattfinden, um diesen auf die kritischen Qualitätsmerkmale des Prüflings auszurichten?
- Welche konkrete Produktreview- bzw. Inspektionstechnik⁴⁹² soll in welchen Fällen verwendet werden und wie können die verschiedenen Techniken ggf. kombiniert werden?

Die vorliegende Arbeit kann diese softwaretechnischen Fragestellungen nicht beantworten. Im Gegenteil, sie reiht sich mit der Forderung der Strukturdimension Artefaktspezifikation, die Informationsstruktur der bei Produktreviews zugrundegelegten Artefakte in Abhängigkeit von den übergeordneten Projektzielpräferenzen zu gestalten, in den Forschungsbedarf ein.

⁴⁹⁰ Vgl. allgemein zu Produktreviews bspw. Ludewig, Lichter (2006), S. 258ff.; Im Detail bestehen zwischen Reviews, Inspektionen und weiteren Varianten wie bspw. Walkthroughs Unterschiede, auf die hier nicht weiter eingegangen werden soll.

⁴⁹¹ Vgl. Denger, Olsson (2005), S. 181; Der im nachfolgenden Kap. 3.2.1.2 kurz vorgestellte Ansatz ‚Taqtic‘ adressiert die genannten Fragestellungen teilweise.

⁴⁹² Bspw. checklist-based reading, scenario-based reading, usage-based reading, perspective-based reading, defect-based reading, traceability-based reading etc.; vgl. hierzu auch Thelin u.a. (2003), S. 688, mit zahlreichen weiteren Verweisen zu den einzelnen Inspektionstechniken.

Die Produktreview-Checklisten bei der Fallstudienunternehmung fokussieren auf die Kritikalität der Projektzieldimension Qualität. Produktreviews dürfen jedoch nicht hierauf reduziert werden, sondern können die Untersuchung des Prüflings hinsichtlich aller vier Projektzieldimensionen umfassen. Einen Ansatz hierzu bietet das sog. Perspektivenbasierte Lesen, bei welchem ein Artefakt aus unterschiedlichen Perspektiven überprüft wird. Jedem Prüfer wird dabei eine Perspektive mit detaillierten spezifischen Prüfanweisungen und Fragestellungen für die Durchführung des Reviews vorgegeben. Das Perspektivenbasierte Lesen beruht auf der Annahme, dass die Prüfer aufgrund der Fokussierung auf einzelne Perspektiven unterschiedliche Fehler finden und in der Summe über alle Prüfer hinweg mehr Fehler aufgedeckt werden, als das bei der Fehlersuche ohne differenzierte Perspektiven der Fall wäre.⁴⁹³ Die typischen Perspektiven können wie in Abb. 50 dargestellt charakterisiert und für die Artefaktspezifikation einbezogen werden (vgl. Abb. 50).

Perspektive	Beschreibung ⁴⁹⁴	Bezug zur Projektziel-dimension
Kunde	Aus der Kundenperspektive ist zu prüfen, ob Anforderungsartefakte die gewünschte Funktionalität und Qualität beschreiben.	Leistungs- umfang
Software-architekt	Aus der Perspektive eines Softwarearchitekten ist zu prüfen, ob Anforderungsartefakte die Anforderungen an den Architekturentwurf beschreiben (architekturbezogene Qualitätsanforderungen wie bspw. Zeit- und Verbrauchsverhalten oder Übertragbarkeit).	Qualität
Tester	Aus der Perspektive eines Testers ist zu prüfen, ob Anforderungsartefakte als Grundlage für die Erstellung von Testfällen für die Phase ‚Softwaretest‘ nutzbar sind.	Leistungs- umfang und/oder Qualität
Entwicklungsprozessaspekte	Weitere sinnvolle Perspektiven für das Perspektivenbasierte Lesen lassen sich u.a. aus den sog. Kontextfacetten eines Softwaresystems ableiten. Eine mögliche Kontextfacette stellen dabei Entwicklungsaspekte dar. Die Entwicklungsfacette umfasst alle Kontextaspekte, welche die Entwicklung der geplanten Software betreffen. Hierzu gehören Aspekte des Entwicklungsprozesses, Entwicklungswerkzeuge, Reifegradmodelle, Qualitätssertifizierungen etc. ⁴⁹⁵	Zeit- und/oder Kostenziel

Abb. 50: Perspektivenbasiertes Lesen als Steuerungsmaßnahme der Strukturdimension Artefaktspezifikation

⁴⁹³ Vgl. Pohl (2007), S. 454

⁴⁹⁴ Zur Charakterisierung der Perspektiven vgl. Shull u.a. (2000)

⁴⁹⁵ Zur Charakterisierung der Entwicklungsfacette vgl. Pohl (2007), S. 39f.

Ansatzpunkte zur Artefaktspezifikation beim Risikomanagement lassen sich in den Phasen Risikoidentifikation und -analyse implementieren. Auch hier sollte nämlich der Prämisse gefolgt werden, dass die Informationsstrukturen der zugrundegelegten Artefakte in Abhängigkeit von den übergeordneten Projektzielpräferenzen gestaltet werden müssen. Mit ‚RiskIT‘⁴⁹⁶ kann auf einen situativen und zielorientierten Risikomanagement-Ansatz zurückgegriffen werden, der entsprechende Ansatzpunkte zur Umsetzung der Artefaktspezifikation bietet. ‚RiskIT‘ stellt dem (wiederkehrenden) Risikomanagement-Prozess eine Initialisierungsphase voran. Diese legt in den Prozessschritten ‚Risk management mandate definition‘, ‚Goal review‘ und ‚Risk identification‘ die Grundlagen für die Ausführung der wiederkehrenden Prozessschritte Risikoanalyse, -bewertung, -steuerung und -überwachung fest. Das ‚Risk management mandate‘ definiert Ziele, Umfang und Ausgestaltung des Risikomanagement-Prozesses. Im zweiten Prozessschritt ‚Goal review‘ wird analysiert, für welche Schlüsselrolle im Projekt welche Projektziele wie bedeutend sind⁴⁹⁷ – ganz im Sinne der vorliegenden Arbeit.

Das Ergebnis der Risikoidentifikation ist bei ‚RiskIT‘ eine Liste mit potentiellen Risiken für das Softwareentwicklungsprojekt. Grundlage dafür bilden die festgelegten Rahmenbedingungen im ‚Risk management mandate‘ sowie Risikochecklisten, die hier ganz in Analogie zum auf Checklisten basierten Produktreview angewendet werden.⁴⁹⁸ Projektziele und Schlüsselrollen spielen auch im Prozessschritt ‚Risk analysis‘ eine entscheidende Rolle. Hier wird für jedes identifizierte und in die Risikoanalyse aufgenommene Risiko analysiert,⁴⁹⁹ welche Auswirkungen es auf die Projektziele hat. Die möglichen Risikoereignisse werden in Form von sog. Risikoszenarien dokumentiert. Da in der Initialisierungsphase festgelegt wurde, welches Projektziel für welche Schlüsselrolle wichtig ist, lässt sich abschätzen, welche Schlüsselrolle den größten Schaden beim Eintreten eines Risikoereignisses hat.⁵⁰⁰

3.2.1.2 Umsetzung der Steuerungsmaßnahmen durch Checklisten

Checklisten als Mittel zur Umsetzung der Strukturdimension Artefaktspezifikation sind bisher, ohne auf dieses Instrument näher einzugehen, im Rahmen des Qualitäts- und des Risikomanagements angesprochen worden. Checklisten lassen sich jedoch bei allen vier Koordinationsprozessen zur Umsetzung der Artefaktspezifikation heranziehen. Das liegt an der grundsätzlichen Wirkungsweise ihrer Anwendung. Keil u.a. un-

⁴⁹⁶ Vgl. Kontio (2001)

⁴⁹⁷ Vgl. Kontio (2001), S. 55-62

⁴⁹⁸ Vgl. Kontio (2001), S. 56, 65

⁴⁹⁹ In vielen Fällen kann nur ein Bruchteil der identifizierten Risiken in die Risikoanalyse und die weiteren Phasen des Risikomanagement-Prozesses aufgenommen werden. Vgl. Freimut u.a. (2001), S. 284

⁵⁰⁰ Vgl. Kontio (2001), S. 49, 56, 62, 66f., 70f.

tersuchten diesen Aspekt anhand der Anwendung von Risikochecklisten bei der Softwareentwicklung.

Als Untersuchungsmethode dient Keil u.a. ein Laborexperiment, bei welchem 128 Teilnehmer mit verschiedenen Projektszenarios konfrontiert und gebeten wurden, Risiken zu bewerten sowie Folgemaßnahmen zu treffen. Das Experiment soll u.a. konkret Aufschluss über den Einfluss von Risikochecklisten auf die Wahrnehmung von Risiken und auf die Entscheidungsfindung beim Risikomanagement in der Softwareentwicklung geben.⁵⁰¹ Mit dieser Untersuchungszielsetzung sind fünf zu prüfende Hypothesen verbunden, wovon zwei relevante Erkenntnisse für die Umsetzung von Steuerungsmaßnahmen in der Strukturdimension Artefaktspezifikation liefern.⁵⁰²

- a) Teilnehmer, die eine Risikocheckliste verwenden, identifizieren mehr Risiken als Teilnehmer, die keine Risikocheckliste verwenden;
- b) Risikochecklisten formen die Aufmerksamkeit so, dass Teilnehmer, die eine Risikocheckliste verwenden, Risiken anders wahrnehmen als Teilnehmer, die keine Risikocheckliste verwenden.

In der Untersuchung konnte Hypothese a) bestätigt werden: Teilnehmer, die eine Risikocheckliste verwendeten, identifizierten tatsächlich wesentlich mehr Risiken als Teilnehmer, die keine Risikocheckliste erhalten haben. Diese Erkenntnis ist insofern relevant, als dass ungeachtet des Einflusses von Checklisten auf die Effektivität (Hypothese b), diese in jedem Fall die Effizienz erhöhen.

Tatsächlich aber erhöhen Checklisten auch die Effektivität: Um die Hypothese b) zu prüfen, basieren die in der Untersuchung verwendeten Risikochecklisten auf 14 Risikokategorien. Das Projektszenario der entsprechenden Untersuchungsgruppen ist mit 7 konkreten Risiken aus ebendiesen 14 Risikokategorien behaftet. Die Untersuchungsergebnisse zu Hypothese b) zeigen dabei, dass Teilnehmer, welche die Risikocheckliste verwendeten, sowohl mehr eingestreuete als auch mehr sonstige Risiken identifiziert haben als Teilnehmer, die keine Risikocheckliste verwendeten.⁵⁰³

Die Umsetzung der Artefaktspezifikation über Checklisten, welche in Abhängigkeit von den übergeordneten Projektzielpräferenzen gestaltet sind, d.h. welche sich bei den vorgegebenen Prüfpunkten bzw. Fragen schwerpunktmäßig auf die kritische(n) Projektzieldimension(en) ausrichten, wirkt sich analog zur Hypothese b) aus. Dies bestätigen auch die Erkenntnisse aus den Experteninterviews:

⁵⁰¹ Vgl. Keil u.a. (2006), S. 2

⁵⁰² Vgl. Keil u.a. (2006), S. 4

⁵⁰³ Vgl. Keil u.a. (2006), S. 5, 7

- Die Zweckmäßigkeit von Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation wird grundsätzlich nicht angezweifelt, sondern als ein notwendiger Schritt gesehen, eine bei den Koordinationsprozessen bereits vorhandene ‚intuitive‘ bzw. ‚nicht bewusst gelebte‘ Ausrichtung auf die übergeordneten Projektzielpräferenzen zu institutionalisieren.⁵⁰⁴
- Checklisten werden explizit als die bevorzugte Form der Unterstützung bzw. Institutionalisierung der o.g. ‚Intuition‘ und damit als die bevorzugte Form der Umsetzung von Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation gesehen.⁵⁰⁵

Um Checklisten projektspezifisch für die einzelnen Koordinationsprozesse in Abhängigkeit von den übergeordneten Projektzielpräferenzen zusammenzustellen, bieten sich jeweils zahlreiche Ansatzpunkte an. Eine Zusammenstellung von einigen Checklisten für das Risikomanagement in der Softwareentwicklung findet sich bspw. bei Keil u.a.⁵⁰⁶ McConnell befasst sich ausführlich mit der Steuerung von zeitzielbezogenen Risiken der Softwareentwicklung und stellt dazu eine umfassende Liste von potentiellen Risiken zusammen, die unmittelbar als Grundlage für eine entsprechende Checkliste dienen kann.⁵⁰⁷

Carr u.a. präsentieren mit dem ‚Taxonomy-Based Questionnaire (TBQ)‘ eine sehr umfassende Risikocheckliste mit insgesamt 194 Fragen. Diese Checkliste ist allerdings für die praktische Anwendung zu umfangreich und muss daher projektspezifisch auf die im Einzelfall relevanten Fragestellungen reduziert werden.⁵⁰⁸ So verwenden Freimut u.a. einen Ausschnitt aus dem ‚TBQ‘ als Checkliste für die Risikoidentifizierung bei einer pilothaften Anwendung von ‚RiskIT‘.⁵⁰⁹ Abb. 51 zeigt am Beispiel des ‚TBQ‘, wie eine solche umfassende Checkliste auf die übergeordneten Projektzielpräferenzen ausgerichtet werden kann.

⁵⁰⁴ Experte ‚B.E.‘ zur Frage B1, Experte ‚K.H.‘ zur Frage B1, Experte ‚T.Si.‘ zur Frage B1, Experte ‚O.S.‘ zur Frage B1

⁵⁰⁵ Experte ‚C.S.‘ zur Frage B1, Experte ‚J.W.‘ zur Frage B1, Experte ‚J.C.‘ zur Frage B1, Experte ‚T.S.‘ zur Frage B1, Experte ‚T.D.‘ zur Frage B1, Experte ‚T.W.‘ zur Frage B1, Experte ‚L.G.‘ zur Frage B1

⁵⁰⁶ Vgl. Keil u.a. (2006), S. 3, mit weiteren Verweisen

⁵⁰⁷ Vgl. McConnell (1996), S. 86-91, 98f.

⁵⁰⁸ Vgl. Carr u.a. (1993), S. 11

⁵⁰⁹ Vgl. Freimut u.a. (2001), S. 281

Projektzielpräferenz	Fragen aus dem ‚Taxonomy-Based Questionnaire‘ ⁵¹⁰
Kosten	C: PROGRAM CONSTRAINTS [154] Is the budget stable? [157] Is there anything for which adequate budget was not allocated? [158] Do budget changes accompany requirement changes? (Yes) (158.a) Is this standard part of the change control process?
Zeit	C: PROGRAM CONSTRAINTS [143] Has the schedule been stable? [144] Is the schedule realistic? (Yes) (144.a) Is the estimation method based on historical data? (Yes) (144.b) Has the method worked well in the past? [145] Is there anything for which adequate schedule was not planned?
Leistungsumfang; Qualität	A: PRODUCT ENGINEERING [3] Are requirements missing or incompletely specified? [37] Are the design specifications in sufficient detail to write the code? [38] Is the design changing while coding is being done?

Abb. 51: Ausrichtung einer Risikocheckliste auf die übergeordneten Projektzielpräferenzen

Die in der Praxis vorherrschende Produktreview-Variante ist das auf Checklisten basierte Produktreview.⁵¹¹ Brykczynski kommt bei seiner Untersuchung zur Anwendung von Checklisten beim Produktreview auf eine Untersuchungsbasis von 117 Checklisten.⁵¹² Eine Checkliste kann dabei auf zwei verschiedene Arten verwendet werden. Zum einen kann die Checkliste als ein Hilfsmittel dienen, das durch den Prüfer nach eigenem Ermessen verwendet wird. Zum anderen kann eine Dokumentationspflicht bestehen – in diesem Fall muss der Prüfer bei jedem Element des untersuchten Anforderungsartefaktes auf alle in der Checkliste definierten Fragestellungen eingehen und dies auch im Detail dokumentieren.⁵¹³

Ein erwähnenswerter situativer Produktreview-Ansatz wird von Denger und Shull vorgestellt. Zu den Kernkonzepten ihres Ansatzes ‚TAQtIC‘ (‚Tailoring Approach for Quality- Driven Inspections‘) gehört das Tailoring des Review-Prozesses auf diverse spezifische Bedingungen im Umfeld des Produktreviews sowie auf die kritischen Qualitätsteilmerkmale des Prüflings. Dabei wird das Perspektivenbasierte Lesen mit der Anwendung von Checklisten kombiniert. Die verschiedenen Überprüfungsperspektiven, die jedem Prüfer vorgegeben werden und die detaillierte spezifische Prüfanwei-

⁵¹⁰ Vgl. hierzu Carr u.a. (1993), S. D1ff.

⁵¹¹ Vgl. Thelin u.a. (2003), S. 688

⁵¹² Vgl. Brykczynski (1999)

⁵¹³ Vgl. Pohl (2007), S. 464

sungen und Fragestellungen für die Durchführung des Produktreviews beinhalten, sind hier in Form von Checklisten umgesetzt.⁵¹⁴

Checklisten für das Änderungsmanagement und hier insbesondere für die Einflussanalyse nehmen in der Literatur deutlich weniger Raum ein als Checklisten für das Risikomanagement oder für Produktreviews. Trotzdem ist hier die Anwendung von Checklisten, wie auch die Vorgehensweise in der Fallstudienunternehmung zeigt, möglich und vorteilhaft. Eine entsprechende Checkliste mit Fragen hinsichtlich der möglichen Auswirkungen eines Änderungsantrages findet sich bei Wiegers. Abb. 52 stellt am Beispiel von einigen Fragen aus dieser Checkliste dar, wie die Einflussanalyse durch die Verwendung von Checklisten auf die übergeordneten Projektzielpräferenzen ausgerichtet werden kann (vgl. Abb. 52).

Projektzielpräferenz	Checklistenfragen zu möglichen Auswirkungen eines Änderungsantrages⁵¹⁵
Kosten	<ul style="list-style-type: none"> • Welche negativen Nebenwirkungen oder sonstigen Risiken bestehen [hinsichtlich Kosten], wenn die Änderung durchgeführt wird? • Welche Auswirkungen hat die Änderung auf Reihenfolge, Abhängigkeiten, Aufwand oder Dauer der einzelnen Aufgaben, die momentan im Projektplan verzeichnet sind? • Wie viel Aufwand, der bereits in das Projekt investiert wurde, geht verloren, wenn die Änderung genehmigt wird?
Zeit	<ul style="list-style-type: none"> • Welche negativen Nebenwirkungen oder sonstigen Risiken bestehen [hinsichtlich Zeit], wenn die Änderung durchgeführt wird? • Welche Auswirkungen hat die Änderung auf Reihenfolge, Abhängigkeiten, Aufwand oder Dauer der einzelnen Aufgaben, die momentan im Projektplan verzeichnet sind?
Leistungsumfang	<ul style="list-style-type: none"> • Bestehen irgendwelche Konflikte zwischen bestehenden [funktionalen] Anforderungen in der Baseline und der vorgeschlagenen Änderung? • Welche negativen Nebenwirkungen oder sonstigen Risiken bestehen [hinsichtlich Leistungsumfang], wenn die Änderung durchgeführt wird?
Qualität	<ul style="list-style-type: none"> • Bestehen irgendwelche Konflikte zwischen bestehenden [Qualitäts-] Anforderungen in der Baseline und der vorgeschlagenen Änderung? • Welche [...] technischen Folgen hat es, wenn die Änderung nicht vorgenommen wird? • Wirkt sich die Änderung nachteilig auf die Leistungsanforderungen oder andere Qualitätsattribute aus? • Welche negativen Nebenwirkungen oder sonstigen Risiken bestehen [hinsichtlich Qualitätsanforderungen], wenn die Änderung durchgeführt wird?

Abb. 52: Auf Projektzielpräferenzen bezogene Checklistenfragen für die Einflussanalyse beim Änderungsmanagement

⁵¹⁴ Vgl. Denger, Shull (2007), S. 79ff.; Spezifische Bedingungen im Umfeld des Produktreviews sind bspw. Fehlerhistorie des Prüflings oder projektspezifische Rahmenbedingungen. Eine weitere Umsetzung des Checklisten-basierten Produktreviews findet sich bspw. bei Lee, Boehm (2005).

⁵¹⁵ Vgl. hierzu Wiegers (2005), S. 322f.

3.2.2 Strukturdimension Kompetenzverteilung

3.2.2.1 Steuerungsmaßnahmen in der Strukturdimension Kompetenzverteilung

Damit eine organisatorische Einheit ihre Aufgaben erfüllen kann, muss sie das Recht haben, jene Maßnahmen zu ergreifen, die zur ordnungsgemäßen Aufgabenerfüllung notwendig sind. Diese Handlungsrechte werden als Kompetenzen bezeichnet. Sie stellen die entscheidende Grundlage positionsspezifischer Autorität dar und sind damit die wesentliche Basis für die Einflussnahme einer organisatorischen Einheit auf die Gesamtaufgabe.⁵¹⁶

Die Gestaltungsmaßnahmen der Strukturdimension Kompetenzverteilung zielen auf die Verteilung dieser endlichen Handlungsrechte auf die drei bei der Standardsoftwareentwicklung zentralen Rollen mit konfliktären Interessenstandpunkten zueinander.⁵¹⁷ Projekt-, Qualitäts- und Produktmanagement müssen in Abhängigkeit von den übergeordneten Projektzielpräferenzen mit entsprechenden Kompetenzen bei den Koordinationsprozessen ausgestattet werden.

Kompetenzverteilung fasst damit alle konkreten Gestaltungsmaßnahmen zusammen, welche nach dem Strukturierungskonzept von Laux über die Beeinflussung des Entscheidungsmodells Koordinationseffektivität sicherstellen sollen. Entsprechende Ansatzpunkte konnten in der Fallstudienunternehmung beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen sowie beim Änderungsmanagement identifiziert werden. Sie bilden den Ursprung der Strukturdimension Kompetenzverteilung (vgl. Abb. 53).

⁵¹⁶ Vgl. Hill u.a. (1994), S. 124

⁵¹⁷ Vgl. hierzu Kap. 2.3.2.1

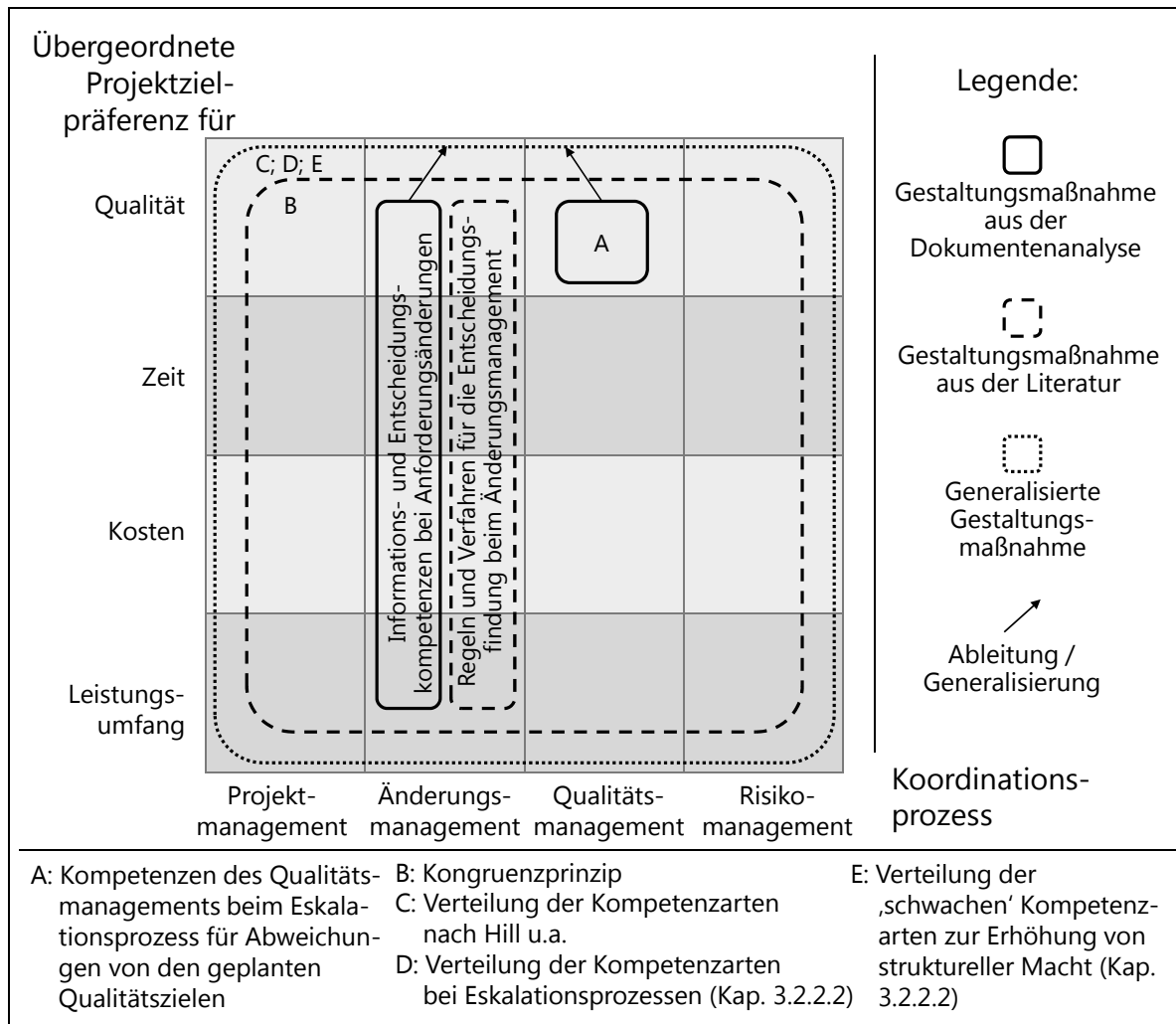


Abb. 53: Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung

Bei der Fallstudienunternehmung verfügt das Qualitätsmanagement im Rahmen des Eskalationsprozesses bei Abweichungen von den geplanten Qualitätszielen bei besonders kritischen Abweichungen - und auch nur in diesem Fall - über die Kompetenz, einen alternativen Gegenmaßnahmenplan zu dem des Projektprogrammleiters zur weiteren Entscheidung vorzulegen. Hier wird also die sog. Entscheidungskompetenz für ein und dasselbe Sachproblem gleich in zweierlei Hinsicht umfangmäßig abgestuft: Mit der Festlegung eines Eskalationsprozesses an sich werden Entscheidungskompetenzen für den Normalfall, d.h. für den Projektablauf ohne wesentliche Abweichungen von den geplanten Qualitätszielen, sowie für den Ausnahmefall, d.h. für einen Projektablauf mit Qualitätszielabweichungen und damit einhergehender Eskalation, verteilt. Durch den Übergang vom Normal- in den Ausnahmefall kann sich der Kompetenzumfang der beteiligten Rollen in einem wesentlichen Ausmaß ändern. Zum zweiten wird nochmals im Fall der Ausnahme abgestuft, nämlich nach ihrer Tragweite (vgl. Abb. 54).

Steuerungsmaßnahme	Bei geringfügigen Abweichungen von den geplanten Qualitätszielen legen der Projektprogrammleiter und das Qualitätsmanagement einen gemeinsamen Gegenmaßnahmenplan zur weiteren Entscheidung vor. Bei schwerwiegenden Abweichungen hat das Qualitätsmanagement jedoch die Kompetenz, einen eigenen Gegenmaßnahmenplan vorzulegen, v.a. wenn keine Übereinstimmung mit dem Projektprogrammleiter erzielt werden konnte.
Zuordnung zum Koordinationsprozess	Qualitätsmanagement
Tailoring der Steuerungsmaßnahme	Anwendung der Steuerungsmaßnahme in Abhängigkeit von der Kritikalität der Qualitätszielabweichung: geringfügige Abweichungen vs. schwerwiegende Abweichungen; Evaluierung mithilfe einer entsprechenden Mustervorlage.
Bezug zur Projektziel-dimension	Qualität

Abb. 54: Kompetenzverteilung beim Eskalationsprozess für Qualitätszielabweichungen als Steuerungsmaßnahme der Strukturdimension Kompetenzverteilung

Beim Änderungsmanagement in der Fallstudienunternehmung wird die Kompetenzverteilung zwischen dem Projekt-, Qualitäts- und Produktmanagement situativ über die Mustervorlage für den Änderungsantrag gehandhabt. Diese listet zahlreiche Rollen auf, die von einer Anforderungsänderung grundsätzlich betroffen sein können. Aus diesen Rollen müssen im Zuge des Änderungsmanagements diejenigen bestimmt werden, welche in dem bestimmten Fall tatsächlich betroffen sind. Anschließend findet eine weitere Auswahlentscheidung statt, die dem Vorschlag von Wiegers entspricht.⁵¹⁸ Alle Rollen, deren Arbeit von der Entscheidung über den Änderungsantrag betroffen ist, müssen zwar über den Entscheidungsverlauf zu der Anforderungsänderung informiert werden, doch sie müssen nicht alle an den Entscheidungen teilhaben, d.h. hier die entsprechenden (Mit-)Entscheidungskompetenzen besitzen (vgl. Abb. 55).

Steuerungsmaßnahme	Kompetenzverteilung beim Änderungsmanagement: Festlegung von Rollen, welche bei einer konkreten Anforderungsänderung zu informieren sind, welche Einwände vorbringen können und welche die Anforderungsänderung genehmigen müssen.
Zuordnung zum Koordinationsprozess	Änderungsmanagement
Tailoring der Steuerungsmaßnahme	Bei der Fallstudienunternehmung ad hoc; systematisches Tailoring zu Projektbeginn jedoch grundsätzlich möglich.
Bezug zur Projektziel-dimension	Zeit, Kosten, Qualität, Leistungsumfang

Abb. 55: Kompetenzverteilung beim Änderungsmanagement als Steuerungsmaßnahme der Strukturdimension Kompetenzverteilung

⁵¹⁸ Vgl. Wiegers (2005), S. 316

Neben der Kompetenzverteilung hinsichtlich (Mit-)Entscheidungskompetenz oder lediglich der Kompetenz, informiert zu werden, sind bei einem systematischen und institutionalisierten Änderungsmanagement weitere Kompetenzen zu verteilen – angefangen mit der generellen Kompetenz zur Einbringung einer Anforderungsänderung. Sie verkörpern in der Summe die Regeln und Verfahren für die Entscheidungsfindung beim Änderungsmanagement:⁵¹⁹

- Anzahl der anwesenden Mitglieder des Änderungsgremiums, damit eine verbindliche Entscheidung getroffen werden kann;
- Anwesenheit bestimmter Rollen, damit eine verbindliche Entscheidung getroffen werden kann;
- Art des Entscheidungsverfahrens, das bei der Entscheidungsfindung zum Einsatz kommt (Abstimmung, Einstimmigkeit, Beratung, etc.);
- Eventuelle Veto- oder Überstimmungskompetenz des Vorsitzenden des Änderungsgremiums oder einer anderen (übergeordneten) Instanz.

Den o.g. Regeln und Verfahren für die Entscheidungsfindung beim Änderungsmanagement liegen grundsätzliche Kompetenzarten zugrunde, die bei allen vier Koordinationsprozessen die Basis für die Ableitung der Gestaltungsempfehlungen in der Strukturdimension Kompetenzverteilung bilden. Die folgende Aufzählung ordnet die für diese Arbeit relevanten Kompetenzarten nach ihrer ‚Stärke‘, d.h. dem Ausmaß der Partizipation an der Entscheidungsfindung, an:⁵²⁰

- Antragskompetenz beinhaltet ein Initiativrecht zur Auslösung von Entscheidungsprozessen über einen bestimmten Sachverhalt an einer anderen, hierzu befugten Stelle. Je nach Ausgestaltung kann die Antragskompetenz zumindest indirekt auch die Kompetenz beinhalten, festzulegen, in welche Richtung entschieden werden soll – wenn nämlich auch die Entscheidungsvorbereitung mehr oder weniger vollständig an die beantragende Stelle abgegeben wird.
- Verfügungskompetenz beinhaltet das Zugriffs- und Verfügungsrecht über bestimmte Arbeitsobjekte, Sachmittel etc. Die Verfügungskompetenz kann sich auch auf Informationen beziehen, was die in Zusammenhang mit der Strukturdimension Kompetenzverteilung relevante Ausprägung der Verfügungskompetenz darstellt. In diesem Fall beinhaltet die Verfügungskompetenz das Recht, Informationen von ei-

⁵¹⁹ Vgl. Wiegers (2005), S. 317

⁵²⁰ Vgl. Schulte-Zurhausen (2002), S. 142f.; vgl. Hill u.a. (1994), S. 125-129; vgl. Picot u.a. (2005), S. 230f.; vgl. Jost (2009), S. 310f. Daneben werden in der Literatur auch Ausführungs-, Kontroll- und Anordnungskompetenz (letzteres auch: Weisungskompetenz) aufgeführt (vgl. Hill u.a. (1994), S. 125, 128f.). Sie werden hier allerdings für die Strukturdimension Kompetenzverteilung nicht instrumentalisiert.

ner andern Stelle zu verlangen – wie es bspw. beim Änderungsmanagement in der Fallstudienunternehmung umgesetzt ist.

- Mitberatungskompetenz ist die schwächste Form der Mitsprachekompetenz, d.h. der generellen Kompetenz zur Mitwirkung bei Entscheidungen. Von einer Entscheidungskompetenz i.e.S. kann hier noch nicht gesprochen werden, denn die Mitberatungskompetenz besagt lediglich, dass die Rolle mit Mitberatungskompetenz vor der Entscheidungsfindung zu konsultieren ist.
- Die Mitentscheidungskompetenz einer oder mehrerer Rollen führt zu einem Kollegialentscheid, der anstelle einer Entscheidung durch eine Person eintritt. Die Stärke der Mitentscheidungskompetenz hängt von den Entscheidungsmodalitäten ab. Mögliche Ausgestaltungen sind hier bspw. Einstimmigkeit, d.h. es wird beraten, bis ein Konsens unter allen Beteiligten erzielt ist, oder der Entscheid mit einfacher oder qualifizierter Mehrheit. Eine besonders stark ausgeprägte Mitentscheidungskompetenz liegt vor, wenn eine Rolle über ein Vetorecht verfügt und somit keine Entscheidung gegen die Präferenzen dieser Rolle gefällt werden kann (Vetokompetenz).
- Fachentscheidungskompetenz ist die Kompetenz, einen bestimmten Teilaspekt einer Entscheidung abschließend festlegen zu können. Bei komplexen Entscheidungen kann für jeden Teilaspekt einer Entscheidung jeweils eine andere Rolle zuständig sein und damit jeweils die Fachentscheidungskompetenz bei einer und derselben Entscheidung besitzen. Im Gegensatz zur Mitentscheidungskompetenz kann hier ein Entscheid erst zustande kommen, wenn jede Rolle einzeln positiv zu ihrem Teilaspekt entschieden hat - d.h. keine Rolle mit Fachentscheidungskompetenz kann überstimmt werden. In der Realität ist der Übergang zwischen der Mitentscheidungs- und Fachentscheidungskompetenz fließend, v.a. im Fall der Mitentscheidungskompetenz als Vetokompetenz.
- Richtlinienkompetenz beinhaltet das Recht, Richtlinien oder grundsätzliche Regelungen vorzugeben, innerhalb derer die betroffenen Rollen Entscheidungen treffen können. Die Richtlinienkompetenz ist damit eine Entscheidungskompetenz höherer Ordnung.

Mit der Zuweisung von Aufgaben und Kompetenzen werden die organisatorischen Einheiten, bzw. im Fall der Projektorganisation die Rollen,⁵²¹ zugleich verpflichtet, diese Aufgaben und Kompetenzen ‚richtig‘ zu erfüllen. Diese Verpflichtung ist zugleich ihre Verantwortung. Hier setzt eines der traditionsreichsten und für die Praxis

⁵²¹ Da sich die Ausführungen im Folgenden auf die Projektorganisation beziehen, wird nur noch auf Rollen Bezug genommen, auch wenn die zitierte Literatur allgemeiner gefasst ist und sich auf organisatorische Einheiten bezieht.

bedeutsamsten Prinzipien der betriebswirtschaftlichen Organisations- und Führungslehre ein – das Kongruenzprinzip: Es besagt, dass bei der Kompetenzverteilung grundsätzlich eine Übereinstimmung (Kongruenz) von Aufgabe, Kompetenz und Verantwortung anzustreben ist. Nur wenn eine Rolle die für ihre Aufgabenerfüllung erforderlichen Kompetenzen besitzt, kann sie auch für die Ergebnisse ihrer Tätigkeit zur Verantwortung gezogen werden.⁵²²

Illustrative und zahlreiche Fälle in der Praxis, bei welchen vom Kongruenzprinzip abgewichen wird, sind bspw. der sog. ‚Frühstücksdirektor‘ als eine Rolle mit Aufgaben ohne Kompetenz und Verantwortung, die ‚Amtsanmaßung‘ als Kompetenzausübung außerhalb des eigenen Aufgabengebietes oder auch der ‚Sündenbock‘ als eine Rolle mit Verantwortung ohne Aufgaben und Kompetenzen.⁵²³ Nichts desto trotz ist die Forderung des Kongruenzprinzips unumstritten. „Allerdings stellt sich die Frage, ob die Kongruenz von Aufgabe, Person und Verantwortung immer eindeutig hergestellt werden kann.“⁵²⁴

Jede umfassende Gesamtaufgabe, so auch die Standardsoftwareentwicklung, erfordert grundsätzlich Arbeitsteilung und Spezialisierung. Ohne die Differenzierung hochkomplexer Vorgänge in eine Vielzahl von Teilaufgaben, welche insbesondere wegen der damit verbundenen Spezialisierung nur auf Subsystemebene erbracht werden können, ist die Gesamtaufgabe nicht zu lösen. Hierdurch ergeben sich aus Sicht des Kongruenzprinzips vielfältige Probleme.⁵²⁵ Sie treffen auf die Standardsoftwareentwicklung mit ihrer weitreichenden Arbeitsteilung und Spezialisierung, der Matrix-Projektorganisation und ihrem Charakter als ein kontinuierlicher und konfliktärer Zielbildungsprozess besonders zu:

- Die Teilaufgaben auf der Subsystemebene können kaum nur den Kompetenzen der jeweils zuständigen organisatorischen Einheit unterworfen werden. Sie müssen meistens durch eine Reihe bewusst geschaffener Kompetenzen anderer organisatorischer Einheiten wie bspw. Informations-, Beratungs- und Mitentscheidungsrechte abgeschwächt werden, um der Gefahr einer suboptimalen Aufgabenerfüllung entgegenzuwirken. Damit wird jedoch die Autonomie des Aufgabenvollzuges eingeschränkt und das Kongruenzprinzip verletzt;

⁵²² Vgl. Bleicher (1980), Sp. 1057f., mit weiteren Verweisen; vgl. Schulte-Zurhausen (2002), S. 143f.; vgl. Wicher (1993), S. 579; vgl. Hill u.a. (1994), S. 124

⁵²³ Vgl. Krüger (1992), S. 47f.

⁵²⁴ Bleicher (1980), Sp. 1058

⁵²⁵ Vgl. hierzu auch im Folgenden Bleicher (1980), Sp. 1058

- Mit der zunehmenden Komplexität und Varianz der Teilaufgaben sind die Möglichkeiten hinsichtlich einer überschneidungsfreien Kompetenzabgrenzung der organisatorischen Einheiten auf Subsystemebene nur sehr begrenzt und laufen im Wesentlichen auf die Begründung überlappender Zuständigkeiten hinaus, wie es bspw. bei der Matrix-Projektorganisation der Fall ist.

Die Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung können allesamt auch als Ansatzpunkte interpretiert werden, um bei der Standardsoftwareentwicklung dem Kongruenzprinzip entsprechen zu können. Hierzu lassen sich mit dem zuvor beschriebenen Kanon von Kompetenzarten je nach den übergeordneten Projektzielpräferenzen die Kompetenzen

- des Qualitätsmanagements bei qualitätskritischen Projekten,
- des Projektmanagements bei zeit- und kostenkritischen Projekten, sowie
- des Produktmanagements bei leistungsumfangskritischen Projekten

bei den Koordinationsprozessen entsprechend dem Kongruenzprinzip jeweils in Relation zu den anderen Rollen mit entgegengesetzten Interessensstandpunkten deutlich ausgeprägter festlegen.

3.2.2.2 Entscheidungs- und Mitwirkungskompetenzen bei Eskalationsprozessen

Bei der Standardsoftwareentwicklung als ein kontinuierlicher und konfliktärer Zielbildungsprozess ist eine Selbstabstimmung der Projektbeteiligten unentbehrlich und darf nicht durch eine hierarchische Entscheidungsfindung ersetzt werden.⁵²⁶ Strukturelle Regelungen, die einen am Selbstabstimmungsprozess partizipierenden Projektbeteiligten mit weitreichenden Kompetenzen ausstatten, müssen kritisch daraufhin betrachtet werden, ob eine solche Entscheidungsfindung den Charakter der Selbstabstimmung behält und nicht stattdessen die Rolle mit weitreichenden Kompetenzen de facto hierarchisch eine Entscheidung herbeiführen kann. Die Ergebnisse der Experteninterviews zielen hinsichtlich der Strukturdimension Kompetenzverteilung v.a. auf diese Problematik ab. Sie führen dabei zu einer differenzierten Sichtweise, bei der das Konzept ‚Management by Exception‘ sowie in diesem Zusammenhang die Definition von Eskalationsprozessen im Vordergrund stehen.

Beim ‚Management by Exception‘ werden Projektbeteiligten fest umrissene Aufgaben- und Kompetenzbereiche übertragen, in die im Normalfall hierarchisch nicht eingegriffen werden darf. Daher gewinnt hier zwangsläufig die Unterscheidung zwischen Normal- und Ausnahmefällen an Bedeutung. Bei Letzteren bleibt ein Eingriff in die

⁵²⁶ Vgl. Kap. 2.3.2.1

Entscheidungsfindung vorbehalten. Differenzierungskriterien für einen Normal- und Ausnahmefall können ganz allgemein das Mengen- oder Wertvolumen (bspw. Projektgröße oder Auftragswert), der Grad der Wiederholbarkeit, Ansprüche an die Vertraulichkeit, oder auch das Eintreten von potentiellen Störungen des Projektverlaufes sein.⁵²⁷

Das ‚Management by Exception‘-Konzept wird beim Projektmanagement-Standard ‚PRINCE2‘ sehr konsequent umgesetzt. Danach muss das (Projektprogramm-) Management stets über den Projektstatus informiert werden, doch nur bei wichtigen Entscheidungen oder großen Planabweichungen wird von außen auf die Entscheidungsfindung der Projektbeteiligten eingegriffen. Bewegt sich die Planabweichung dagegen innerhalb der Toleranzgrenzen, bleibt die Entscheidungsfindung weiterhin in Form von Selbstabstimmung bei den Projektbeteiligten. Das Differenzierungskriterium ist hier also genau das o.g. Eintreten von potentiellen Störungen des Projektverlaufes.⁵²⁸ Sobald die vorgegebenen Toleranzgrenzen überschritten werden könnten, ist die Erstellung von einem Ausnahmeplan notwendig. D.h. ein Projekt bzw. eine Projektphase befindet sich bereits in einer Ausnahmesituation, wenn die gesetzten Phasentoleranzgrenzen voraussichtlich überschritten werden. In dieser Situation muss der Lenkungsausschuss von der voraussichtlichen oder bestehenden Verletzung der Toleranzgrenzen durch einen Ausnahmebericht verständigt werden sowie einen Ausnahmeplan vorgelegt bekommen, welcher den gegenwärtigen Phasenplan ersetzen soll. Innerhalb des Ausnahmeplans ist also ein neuer Projektplan enthalten, der u.a. auch neue Regeln zur Entscheidungsfindung und Kompetenzverteilung enthalten kann.⁵²⁹ Abb. 56 stellt den ‚PRINCE2‘-Subprozess ‚Erstellen eines Ausnahmeplans‘ schematisch dar.

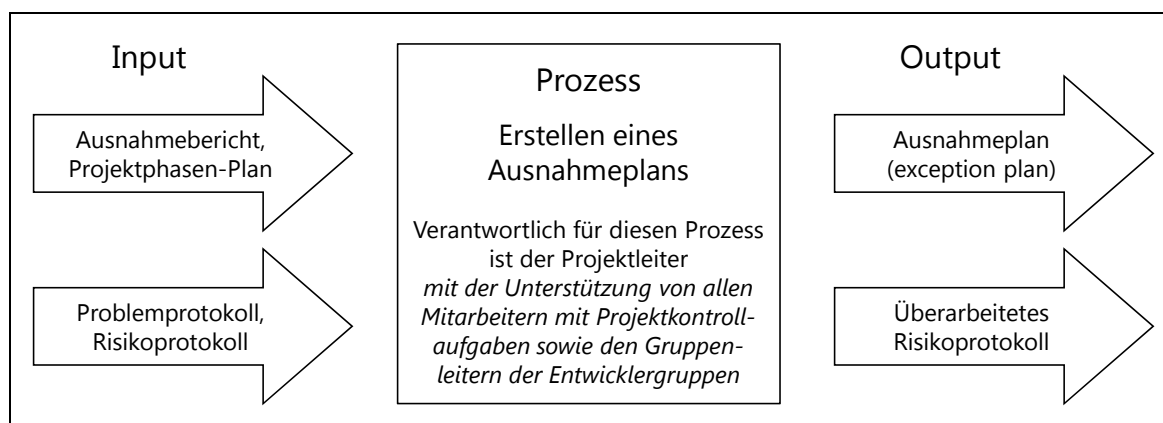


Abb. 56: ‚PRINCE2‘-Subprozess ‚Erstellen eines Ausnahmeplans‘⁵³⁰

⁵²⁷ Vgl. Schanz (1994), S. 217

⁵²⁸ Vgl. Köhler (2006), S. 115

⁵²⁹ Ausnahmeplan ggf. mit Gültigkeit für nur eine bzw. die nächste Projektphase; vgl. Köhler (2006), S. 135, 209f.

⁵³⁰ Vgl. Köhler (2006), S. 210

In den Experteninterviews wird vorgeschlagen, die Regelungen zur Kompetenzverteilung auf den Ausnahmefall i.S.d. ‚Management by Exception‘ zu beschränken. Mithilfe dieser Regelungen sollen komplexere, den Entscheidungsproblemen angemessene Eskalationsprozesse definiert werden können.⁵³¹ Als Differenzierungskriterium für den Normal- und Ausnahmefall kann hier, wie beim ‚PRINCE2‘-Standard, das Eintreten von potentiellen Störungen des Projektverlaufes gelten. Zum Zeitpunkt der Untersuchung ist bei der Fallstudienunternehmung mit dem Eskalationsprozess bei Abweichungen von den geplanten Qualitätszielen nur ein Eskalationsprozess definiert, der als dem Entscheidungsproblem angemessen bezeichnet werden kann. Eskalationsprozesse für Zielabweichungen bei den anderen Projektzieldimensionen sind nicht explizit definiert. Nichts desto trotz sieht die Mustervorlage für den Projektplan einen projektspezifisch festgelegten Eskalationsplan vor.

Auf der anderen Seite weisen die Experteninterviews auch auf eine Problematik bei der Umsetzung der Strukturdimension Kompetenzverteilung über Eskalationsprozesse hin: Eskaliert in der Praxis ein Projekt, so ist bereits eine gewisse Zeit verstrichen, ohne dass andere, mehr proaktive Maßnahmen ergriffen wurden.⁵³² Zudem verlaufen eskalierende Projekte in der Regel ineffizienter als Projekte im Normalfall.⁵³³ Ein anderer zielführender Ansatzpunkt, gewissermaßen ein Mittelweg zur Umsetzung der Strukturdimension Kompetenzverteilung ist es daher, die Regelungen zur Kompetenzverteilung zwar (auch) auf den Normalfall zu beziehen, sich dabei aber nur auf die ‚schwachen‘ Kompetenzarten⁵³⁴ und/oder in einer Projektzieldimension besonders kritische Projekte zu beschränken.⁵³⁵ Dieser Ansatzpunkt kann mithilfe des Macht-Konzeptes begründet werden.

Ausgangspunkt der Überlegungen soll die klassische Definition von Macht nach Weber bilden: „Macht bedeutet jede Chance, innerhalb einer sozialen Beziehung den eigenen Willen auch gegen Widerstreben / Widerstand durchzusetzen, gleichviel worauf diese Chance beruht.“⁵³⁶ Eine genaue Analyse dieser Definition erbringt die folgenden Kennzeichen von Macht und verdeutlicht dabei, dass das Macht-Konzept Parallelen zur Intention der Strukturdimension Kompetenzverteilung aufweist.⁵³⁷

⁵³¹ Experte ‚T.Si.‘ zur Frage B3, Experte ‚T.Si.‘ zur Frage D1, Experte ‚T.M.‘ zur Frage B3, Experte ‚B.E.‘ zur Frage B3, Experte ‚T.W.‘ zur Frage B3

⁵³² Experte ‚C.S.‘ zur Frage D1

⁵³³ Experte ‚J.K.‘ zur Frage B3

⁵³⁴ Experte ‚F.E.‘ zur Frage B3

⁵³⁵ Experte ‚T.M.‘ zur Frage B3

⁵³⁶ Weber (1972), S. 28, zitiert nach Neubauer, Rosemann (2006), S. 42 und Miebach (2007), S. 74

⁵³⁷ Vgl. zu den Kennzeichen von Macht Neubauer, Rosemann (2006), S. 42

- Zentrales Merkmal von Macht ist die Möglichkeit der eigenen Willensdurchsetzung in einer sozialen Beziehung. Da in einer sozialen Beziehung mindestens zwei beteiligte Parteien (Einzelpersonen, organisatorische Einheiten, Organisationen etc.) zueinander stehen, ist Macht ein relationales Konzept – ebenso wie die Strukturdimension Kompetenzverteilung;
- Macht ist als ein dynamisches System aufzufassen, bei dem die Parteien mit mehr Macht, d.h. die Machtbesitzer, vergleichsweise mehr Einfluss besitzen. Dabei können temporäre und situative Bedingungen eine Rolle spielen – bei der Strukturdimension Kompetenzverteilung handelt es sich dabei, wie bei allen Gestaltungsmaßnahmen des vorliegenden Ansatzes, um die übergeordneten Projektzielpräferenzen als situative Bedingung;
- Die Möglichkeit, den eigenen Willen auch gegen das Widerstreben anderer Parteien durchzusetzen, kann sehr unterschiedlich fundiert sein. Grundlage ist dabei immer die Verfügbarkeit und Steuerbarkeit von Qualität und/oder Quantität wie auch immer gearteter Ressourcen – als solche sind auch die verschiedenen Kompetenzarten zu verstehen.

Je nach Art der Ressourcen, die eine Machtbeziehung begründen, können diverse Varianten von Macht unterscheiden werden. Die an dieser Stelle relevante Unterscheidung ist die nach personaler und struktureller Macht: Die personale Macht äußert sich grundsätzlich in einer direkten Machtausübung, d.h. hier setzt eine Partei als Machtbesitzer ihren Einfluss unvermittelt auf die anderen Parteien aus. Daneben kann Macht durch bestimmte strukturelle Festlegungen in einer weniger offenkundigen Form ausgeübt werden (strukturelle Macht).⁵³⁸ Durch die Nichtberücksichtigung der ‚starken‘ Kompetenzarten wie bspw. der Vetokompetenz, die massiv in die Entscheidungsfindung eingreifen, kann der nunmehr institutionalisierte kontinuierliche Zielbildungsprozess immer noch in Form von Selbstabstimmung stattfinden und wird auch so von den Projektbeteiligten wahrgenommen. Der Grund hierfür ist, dass es sich bei den ‚schwachen‘ Kompetenzarten wie Antrags-, Informations- oder Beratungskompetenz weniger um die erstgenannte offenkundige und personalisierte Form der Machtausübung handelt, sondern mehr um die strukturelle Macht.

Nichts desto trotz ist die Wirkung der Machtausübung durch die ‚schwachen‘ Kompetenzarten nicht zu vernachlässigen. Ist mit der strukturellen Macht, bspw. in Form der Beratungskompetenz, die Vorbereitung von Verhandlungen, die Erstellung eines Kriterienkatalogs, die Formulierung der Tagesordnung o.ä. verbunden, so kann diese Form der Machtausübung durch die entsprechende Einflussnahme sogar zu einer Vor-

⁵³⁸ Vgl. Neubauer, Rosemann (2006), S. 43

wegnahme von Entscheidungen führen.⁵³⁹ Dies gilt analog für die Antragskompetenz, sofern auch hier die Kompetenz zur Entscheidungsvorbereitung inbegriffen ist.

Ein weiteres Beispiel für die zielführende Wirkung einer vermeintlich ‚schwachen‘ Kompetenzart stellt die Antragskompetenz hinsichtlich sog. Sofortmaßnahmen dar. Mit diesen erhält ein Projekt vorübergehend höchste Priorität, um bspw. ein Leistungsdefizit, welches sich prinzipiell auf jede der vier Projektzieldimensionen beziehen kann, zeitnah zu beheben. Damit Sofortmaßnahmen ihre gewünschte Wirkung entfalten können, sind bestimmte Regeln strikt einzuhalten. U.a. müssen sich Anordnungen im Rahmen einer Sofortmaßnahme deutlich von allen anderen Vorgängen abheben. Zu jedem Zeitpunkt darf dabei höchstens eine Sofortmaßnahme in Kraft sein, wobei bei Bedarf alle anderen Arbeiten unterbrochen werden müssen, um die Sofortmaßnahme ausführen.⁵⁴⁰

Die Ausübung der Informationskompetenz ist aus machttheoretischer Sicht ebenfalls nicht zu unterschätzen. Sie wird bei der Klassifikation von Machtquellen nach Crozier und Friedberg explizit als eine Machtquelle aufgeführt. In ihrem (spieltheoretischen) Machtmodell ist Macht das kontingente Ergebnis der Mobilisierung der von den Akteuren in einer gegebenen Situation kontrollierten Ungewissheitszonen für ihre Beziehungen und Verhandlungen mit den anderen Beteiligten. Hierbei verleiht die Kontrolle von Informationen und Kommunikationskanälen Macht und wird für die anderen Akteure zu einer Ungewissheitsquelle.⁵⁴¹

3.2.3 Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘

3.2.3.1 Steuerungsmaßnahmen in der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘

Die anfallenden Aufgaben und Verantwortlichkeiten im Rahmen eines Projektes werden personenunabhängig durch das Rollen-Konzept beschrieben. Rollenbeschreibungen besitzen generellen Charakter, d.h. sie bilden als Bestandteil eines Softwareentwicklungsprozesses den Ausgangspunkt für das Projekt-Tailoring. Beim V-Modell XT ist das Konzept der Rolle neben konkreten Vorgehensweisen und den zugehörigen (Zwischen-)Ergebnissen eine von drei Abstraktionsebenen für die Vorgaben.⁵⁴²

Erst im Rahmen des Projekt-Tailoring erfolgt die Zuordnung von Personen zu den Rollen, wobei in einem konkreten Projekt nicht alle unternehmensweit definierten Rollen besetzt werden müssen. Die projektspezifisch adaptierten Rollen müssen dabei

⁵³⁹ Vgl. Neubauer, Rosemann (2006), S. 43f.

⁵⁴⁰ Vgl. Fiedler (2010), S. 187

⁵⁴¹ Vgl. Crozier, Friedberg (1993), S. 17, 50, zitiert nach Miebach (2007), S. 77f.

⁵⁴² Vgl. Bund (Hrsg., 2008), Teil 8, S. 36

auch nicht zwingend jeweils durch einen Projektbeteiligten übernommen werden. Eine Rolle kann ebenso durch mehrere Personen besetzt werden, wie auch umgekehrt ein Projektbeteiligter mehrere Rollen ausführen kann. Die Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘ befasst sich mit diesem Schritt der projektspezifischen Übernahme und Zuordnung von zentralen Projektrollen im Umfeld des Projekt-, Qualitäts- und Produktmanagements. Die Relevanz der personellen und organisatorischen Unabhängigkeit ergibt sich hier aufgrund der konfliktären Interessenstandpunkte der Rollen beim kontinuierlichen Zielbildungsprozess in der Standardsoftwareentwicklung. Damit fasst diese Strukturdimension alle konkreten Gestaltungsmaßnahmen zusammen, die nach dem Strukturierungskonzept von Laux über die Beeinflussung der Zielfunktion Koordinationseffektivität sicherstellen sollen (vgl. Abb. 57).

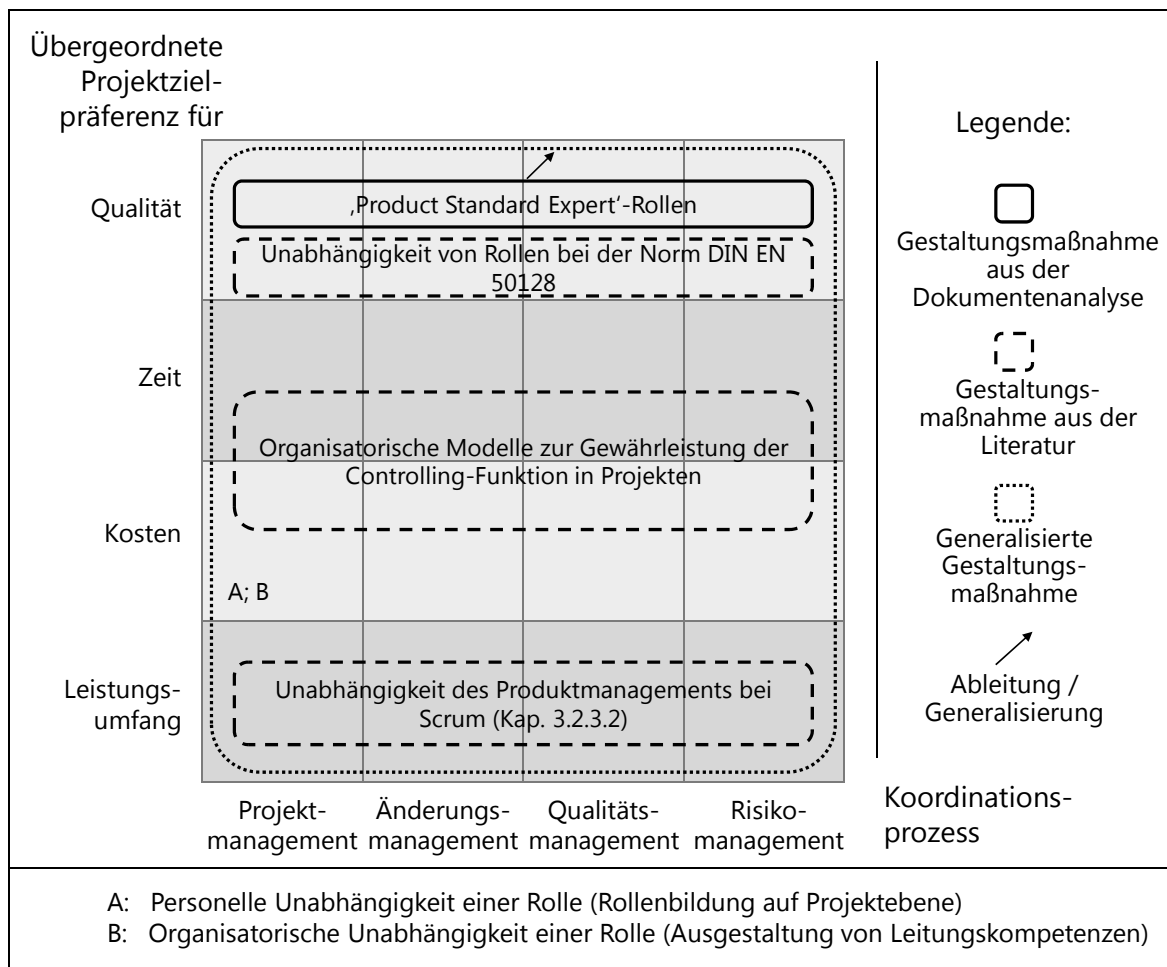


Abb. 57: Gestaltungsmaßnahmen in der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘

Für die personelle Unabhängigkeit einer Rolle im Projektverlauf müssen zwei Voraussetzungen erfüllt sein. Zum ersten muss die unternehmensweit definierte Rolle für ein konkretes Projekt überhaupt übernommen worden sein. Zum zweiten darf die Rolle nicht in Personalunion durch einen Projektbeteiligten übernommen werden, der zudem im selben Projekt eine andere, mit entgegengesetzten Interessen behaftete Rolle aus-

führt.⁵⁴³ In der Fallstudienunternehmung setzt die projektspezifische Übernahme der sog. ‚Product Standard Expert‘-Rollen an beiden Voraussetzungen an. Der unternehmensweite Softwareentwicklungsprozess der Fallstudienunternehmung definiert 16 dieser Rollen. Damit existiert für jedes Qualitätsteilmerkmal des unternehmenseigenen Qualitätsmodells eine Rolle auf Projektebene, deren Aufgaben und Verantwortlichkeiten im Rahmen des Projektes auf einen Zweck fokussiert sind: Erreichung des jeweiligen Qualitätsteilmerkmals in der definierten Qualitätsstufe. Ob eine der ‚Product Standard Expert‘-Rollen für ein konkretes Projekt übernommen wird oder nicht, hängt von der Kritikalität der Zielerreichung hinsichtlich des entsprechenden Qualitätsteilmerkmals ab (vgl. Abb. 58).⁵⁴⁴

Steuerungsmaßnahme	Projektspezifische Übernahme von ‚Product Standard Expert‘-Rollen als Verantwortliche für die Umsetzung von Qualitätsteilmerkmalen
Zuordnung zum Koordinationsprozess	Qualitäts-, Projekt-, Risiko-, Änderungsmanagement
Tailoring der Steuerungsmaßnahme	Entscheidung zu Projektbeginn in Abhängigkeit von der Kritikalität der Zielerreichung hinsichtlich des entsprechenden Qualitätsteilmerkmals
Bezug zur Projektziel-dimension	Qualität

Abb. 58: Projektspezifische Übernahme von ‚Product Standard Expert‘-Rollen als Gestaltungsmaßnahme der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘

Eine entsprechende Umsetzung der personellen Unabhängigkeit lässt sich hinsichtlich aller vier Koordinationsprozesse und Projektzieldimensionen verwirklichen. In Analogie zur Kompetenzverteilung zwischen dem Projekt-, Qualitäts- und Produktmanagement muss sich auch deren personelle Unabhängigkeit an den übergeordneten Projektzielpräferenzen orientieren, sodass bei den situativ kritischen Projektzieldimensionen intrapersonale Interessenskonflikte ausgeschlossen werden können. In diesem Fall sollen Negativbeispiele zur Verdeutlichung dienen. So darf

- bei qualitätskritischen Projekten die Qualitätsmanagement-Rolle weder unbesetzt, noch vom Projekt- oder Produktmanagement wahrgenommen werden, wobei für besonders kritische Qualitätsaspekte eigenständige, ebenso personell unabhängig zu besetzende Rollen definiert werden sollten (bspw. ‚Product Standard Expert‘-Rollen);

⁵⁴³ Bspw. Projektleiter ist gleichzeitig Qualitätsmanager.

⁵⁴⁴ auch Experte ‚R.L.‘ zur Frage B3

- bei leistungsumfangskritischen Projekten die Produktmanagement-Rolle weder unbesetzt noch vom Qualitäts- oder Projektmanagement wahrgenommen werden, wobei für besonders kritische Aspekte hinsichtlich Leistungsumfang eigenständige, ebenso personell unabhängig zu besetzende Rollen definiert werden sollten (bspw. Verantwortlicher für ein bestimmtes Feature oder Anwendungsszenario);
- bei zeit- und kostenkritischen Projekten die Projektmanagement-Rolle weder unbesetzt noch vom Qualitäts- oder Produktmanagement wahrgenommen werden, wobei für besonders kritische Termin- oder Kostenaspekte eigenständige, ebenso personell unabhängig zu besetzende Rollen definiert werden sollten (bspw. Kosten-Controller⁵⁴⁵).

Zwischen der Unabhängigkeit einer Projektrolle bei der Entscheidungsfindung und der Verteilung von Kompetenzen besteht über die sog. Leitungskompetenzen ein Zusammenhang. Als Leitungskompetenzen werden Kompetenzarten bezeichnet, über welche das Verhalten anderer Rollen bestimmt werden kann. Daher begründen Leitungskompetenzen Über- und Unterordnungsverhältnisse.⁵⁴⁶ Die in dieser Arbeit definierte Strukturdimension Kompetenzverteilung beinhaltet jedoch keine der Leitungskompetenzen. Stattdessen beruht die organisatorische Unabhängigkeit einer Rolle bei der Entscheidungsfindung auf einer zielführenden Ausgestaltung hinsichtlich der Leitungskompetenzen Fremdentscheidungs-, Weisungs- und Kontrollkompetenz.⁵⁴⁷

- Die Fremdentscheidungskompetenz umfasst das Recht, für andere Rollen verbindliche Entscheidungen zu fällen;
- Weisungskompetenzen beinhalten das Recht, andere Rollen in bestimmten Fragen zu einem Tun oder Unterlassen anzuweisen. Weisungskompetenzen leiten sich unmittelbar aus der Fremdentscheidungskompetenz ab. Sie sind „nötig, um nach der Willensbildung (Entscheidungen) die Willensdurchsetzung der getroffenen Entscheidungen zu sichern;“⁵⁴⁸
- Mit Weisungskompetenzen sind gleichzeitig auch Kontrollkompetenzen verbunden. Letztere beinhalten das Recht, die richtige Ausführung der Anweisungen zu kontrollieren. Dies kann sich als Ergebniskontrolle auf die Arbeitsergebnisse be-

⁵⁴⁵ Vgl. Patzak, Rattay (2004), S. 350; Dort werden die folgenden, mehr oder weniger personell und organisatorisch unabhängigen, Modelle zur Gewährleistung der Controlling-Funktion in Projekten vorgeschlagen: Der Projektcontroller ist vollständig einem Projekt zugeordnet, d.h. Projektcontrolling wird durch eine eigene Person im Projektteam übernommen; Projektcontrolling wird vom Projektleiter übernommen; Projektcontrolling wird vom Projektauftraggeber übernommen; Der Projektcontroller steht gleichzeitig mehreren Projekten zur Verfügung.

⁵⁴⁶ Vgl. Schulte-Zurhausen (2002), S. 143

⁵⁴⁷ Vgl. Schulte-Zurhausen (2002), S. 143; vgl. Hill u.a. (1994), S. 128f.

⁵⁴⁸ Hill u.a. (1994), S. 128

schränken, aber auch als Verfahrenskontrolle den Entwicklungsprozess zum Gegenstand haben.

Aufgrund der Über- und Unterordnungsverhältnisse, welche durch die drei Leitungskompetenzarten begründet werden, ist durch die personelle Unabhängigkeit einer Rolle insgesamt noch keine Unabhängigkeit bei der Entscheidungsfindung garantiert. Diese kommt erst zustande, wenn auch die organisatorische Unabhängigkeit gegeben ist, d.h. wenn keine andere Rolle mit entgegengesetzten Interessensstandpunkten etwaige Fremdentscheidungs-, Weisungs- oder Kontrollkompetenzen gegenüber dem Entscheidungsträger ausüben kann. Beispielhaft (Negativbeispiel) sei hier die Konstellation genannt, in welcher bei einem qualitätskritischen Projekt das Projektmanagement dem Qualitätsmanagement gegenüber eine oder mehrere der Leitungskompetenzarten ausüben kann. Die gesamte Handlungsautonomie einer Rolle lässt sich also zweidimensional darstellen (vgl. Abb. 59).⁵⁴⁹

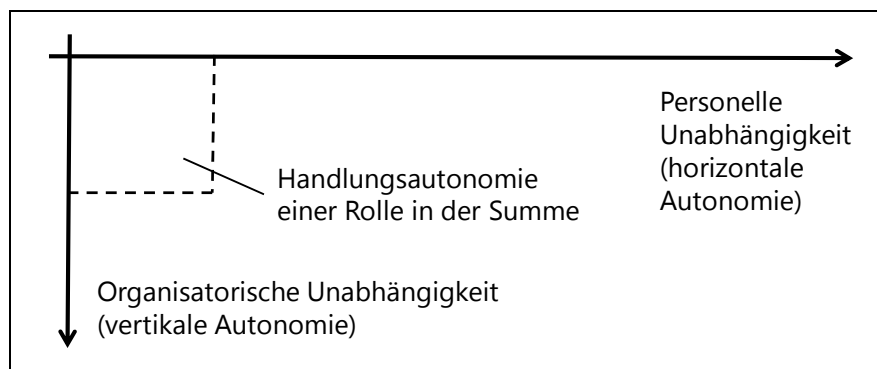


Abb. 59: Handlungsautonomie einer Rolle⁵⁵⁰

Eine musterhafte und integrierte Umsetzung der personellen und organisatorischen Unabhängigkeit von Rollen im Projektverlauf findet sich bei der Entwicklung von sicherheitskritischen eingebetteten Softwaresystemen. Im Gegensatz zu fachlich-technischen Aspekten wie bspw. der Verwendung von formalen Methoden zur Spezifikation und Verifikation der sicherheitskritischen Softwaresysteme sind dortige organisatorische Ansätze auf die Entwicklung von Standardsoftware ohne Weiteres übertragbar.

Einige Normen für sicherheitskritische eingebettete Software sehen abgestufte Regelungen in Abhängigkeit von der Sicherheitskritikalität einer Softwarekomponente vor. Diese Regelungen beziehen sich sowohl auf fachlich-technische, als auch auf organi-

⁵⁴⁹ In Anlehnung an Hill u.a. (1994), S. 224 ff. Die dortigen Ausführungen beziehen sich allgemein, d.h. nicht projektspezifisch, auf die Handlungsautonomie eines (Sub-)Systems. Die horizontale Autonomie bezieht sich dabei auf die Dezentralisation von Entscheidungen; die vertikale Autonomie auf die Delegation von Entscheidungen.

⁵⁵⁰ In Anlehnung an Hill u.a. (1994), S. 225

satorische Aspekte. So werden bei der Norm IEC 61508⁵⁵¹, einer umfassenden Norm zum Thema Sicherheit elektrisch bzw. elektronisch programmierbarer, sicherheitskritischer Systeme, u.a. zahlreiche Regelungen zur organisatorischen Unabhängigkeit der Qualitätssicherung definiert. Diesen Regeln liegt das folgende Prinzip zugrunde: Mit ansteigender Sicherheitskritikalität einer Softwarekomponente, die in Form von definierten Kritikalitätsstufen⁵⁵² bestimmt wird, muss die personelle und organisatorische Unabhängigkeit der Qualitätssicherung zunehmen.⁵⁵³

Die Norm DIN EN 50128, die sich auf eingebettete Software für Eisenbahnsteuerungs- und Überwachungssysteme bezieht, ist weitgehend von der Norm IEC 61508 abgeleitet.⁵⁵⁴ Die DIN EN 50128 unterscheidet im Umfeld der Qualitätssicherung die Rollen des Verifizierers und des Validierers. Der Verifizierer überprüft die fehlerfreie Umsetzung der Anforderungen einer Phase in das Ergebnis der Phase. Der Validierer stellt sicher, dass das Produkt seine Anforderungen erfüllt.⁵⁵⁵ Beide Rollen müssen im gesamten Projektverlauf in dem durch die Software-Sicherheitsanforderungsstufe geforderten Umfang personell und organisatorisch unabhängig sein (vgl. Abb. 60).⁵⁵⁶

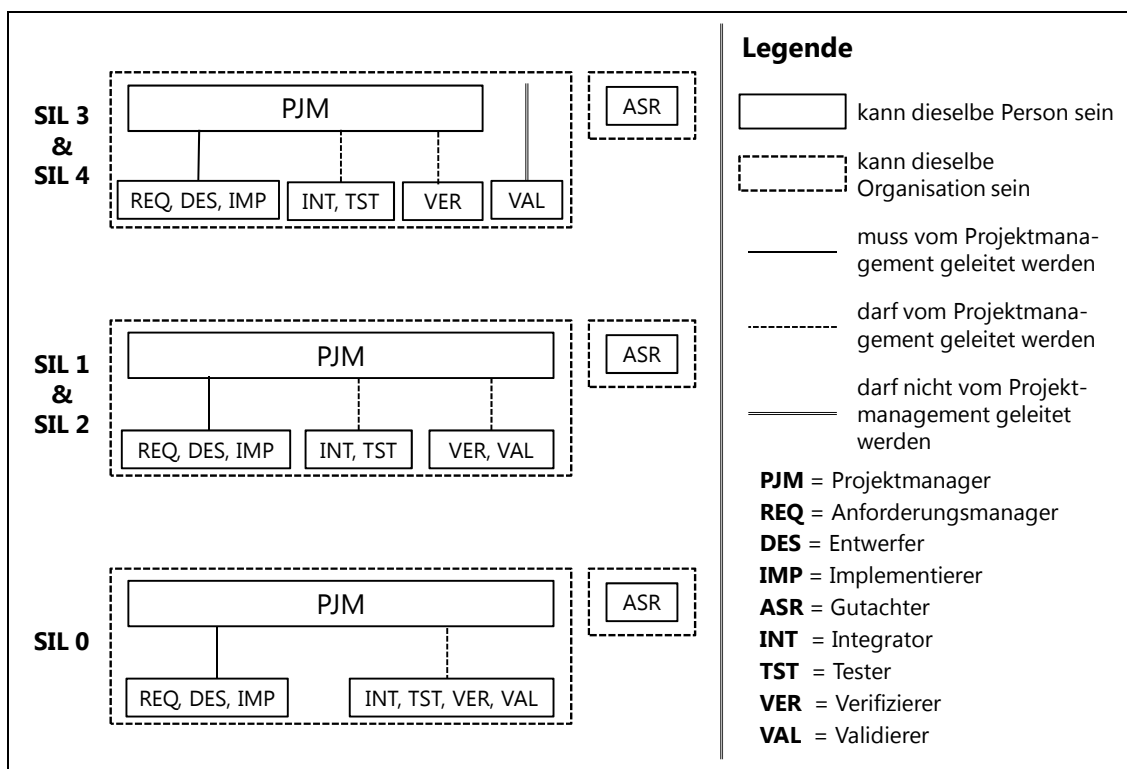


Abb. 60: Personelle und organisatorische Unabhängigkeit bei der Norm DIN EN 50128⁵⁵⁷

⁵⁵¹ Vgl. DIN (Hrsg., 2001); Software wird insbesondere in der IEC 61508-3 behandelt.

⁵⁵² Sog. SIL: Safety Integrity Level

⁵⁵³ Vgl. Liggesmeyer, Rombach (2005), S. 8f.; vgl. Liggesmeyer (1992), S. 361f., 369

⁵⁵⁴ Vgl. Liggesmeyer, Rombach (2005), S. 2

⁵⁵⁵ Vgl. Liggesmeyer, Rombach (2005), S. 8; vgl. Liggesmeyer (1992), S. 361

⁵⁵⁶ Vgl. DIN (Hrsg., 2009), S. 18

⁵⁵⁷ Vgl. DIN (Hrsg., 2009), S. 19

Die Gestaltungsoption ‚kann dieselbe Person sein‘ bezieht sich auf die personelle Unabhängigkeit. Die Gestaltungsoptionen ‚kann dieselbe Organisation sein‘ sowie ‚[muss; darf; darf nicht] vom Projektmanager geleitet werden‘ beziehen sich auf die organisatorische Unabhängigkeit der Rollen. Abb. 60 stellt die bevorzugten Organisationsstrukturen für die verschiedenen Software-Sicherheitsanforderungsstufen dar. Es dürfen jedoch auch weitere, in der Norm definierte Alternativen angewendet werden.⁵⁵⁸ Diese verschiedenen Ausgestaltungen verdeutlichen in der Summe die zahlreichen Möglichkeiten und Wirkungsweisen der Umsetzung von Regelungen zur personellen und organisatorischen Unabhängigkeit.

So dürfen bei den Sicherheitsanforderungsstufen 3 und 4 Validierer und Verifizierer alternativ auch dieselbe Person sein, sofern dabei die organisatorische Unabhängigkeit vom Projektmanager gewahrt bleibt.⁵⁵⁹ Desweiteren muss in diesem Fall die Verifikation von einer anderen fachkundigen Person mit demselben Unabhängigkeitsniveau wie dem des Validierers überprüft werden. Eine zweite, alternative Ausgestaltung für die Sicherheitsanforderungsstufen 3 und 4 ermöglicht entgegen der bevorzugten Organisationsstruktur die personelle Zusammenführung der Verifizierer- mit der Integrator- und/oder der Tester-Rolle. Der Validierer bleibt hierbei aber weiterhin personell und organisatorisch unabhängig. In der Praxis hat sich für Software der Sicherheitsanforderungsstufe 4 bei größeren Projekten zudem die konsequente Trennung der Projektmanager- von der sog. ‚Safety Manager‘-Rolle bewährt. Die dadurch erlangte Unabhängigkeit des ‚Safety Managers‘ ermöglicht diesem die Verweigerung der Freigabe von Phasenübergängen aufgrund von Sicherheitsbedenken, bis aus seiner Sicht alle Anforderungen an die Sicherheit erfüllt sind.⁵⁶⁰

Bei den Sicherheitsanforderungsstufen 1 und 2 darf der Verifizierer alternativ auch dieselbe Person wie der Integrator und/oder Tester sein.⁵⁶¹ Da in diesem Fall die Rolle des Validierers weiterhin die Überprüfung der Leistungen einschließt, bleiben zwei Ebenen der Kontrolle innerhalb der gesamten Projektorganisation erhalten. Eine zweite alternative Ausgestaltung ermöglicht gar die personelle Zusammenführung der Validierer- mit der Verifizierer-, Integrator- und Tester-Rolle. Diese weitgehende Aufgabe der personellen Unabhängigkeit des Validierers muss jedoch über eine Erweiterung

⁵⁵⁸ Vgl. auch im Folgenden DIN (Hrsg., 2009), S. 20

⁵⁵⁹ Bei der bevorzugten Organisationsstruktur der Sicherheitsanforderungsstufen 3 und 4 muss der Validierer personell vollkommen unabhängig sein; vgl. Abb. 60

⁵⁶⁰ Vgl. Liggesmeyer, Rombach (2005), S. 52

⁵⁶¹ Bei der bevorzugten Organisationsstruktur der Sicherheitsanforderungsstufen 1 und 2 darf die Verifizierer- lediglich mit der Validierer-Rolle personell zusammengeführt werden. Im Vergleich zu den Sicherheitsanforderungsstufen 3 und 4 wird hier also die vollkommene personelle Unabhängigkeit der Validierer-Rolle aufgegeben.

seiner organisatorischen Unabhängigkeit gewissermaßen kompensiert werden: Die personell zusammengeführte Validierer-Rolle muss hier organisatorisch unabhängig vom Projektmanager sein. Desweiteren, ebenfalls im Sinne einer ‚Wiederherstellung‘ der insgesamt Unabhängigkeit, müssen hier die Verifizierungsleistungen von einer anderen fachkundigen Person mit demselben Unabhängigkeitsniveau wie dem des Validierers überprüft werden.

Auch bei Software der Sicherheitsanforderungsstufe 0, deren Einfluss auf die Sicherheit des Gesamtsystems nur gering ist, sind alternativ zu der bevorzugten Organisationsstruktur weitere Ausgestaltungen möglich. Bspw. dürfen hier Verifizierer, Integrator und Tester auch an den Validierer berichten.

3.2.3.2 Personelle und organisatorische Unabhängigkeit des Produktmanagements

Das Produktmanagement hat in der Praxis trotz vieler Verantwortlichkeiten und Aufgaben oftmals nicht die notwendige Autorität, um seinen Verpflichtungen nachkommen zu können.⁵⁶² Diese Autorität begründet sich in der entsprechenden formalen und hierarchischen Einordnung der Rolle. Kann sich der Produktmanager in einem Projekt nicht oder nur begrenzt auf seine organisatorisch geregelte Autorität berufen, da ihm bspw. ein funktionelles Weisungsrecht fehlt, hängt der Erfolg seiner Arbeit nur von persönlichen Fähigkeiten wie bspw. Ausstrahlung oder Überzeugungskraft ab.⁵⁶³

Die geschilderte Situation des Produktmanagements in der Praxis trifft auch auf die Fallstudienunternehmung zu, denn bei den Experteninterviews wird bei der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘ v.a. auf die Produktmanagement-Rolle Bezug genommen. Gerade diese Rolle müsse durch die personelle- und organisatorische Unabhängigkeit situativ gestärkt werden⁵⁶⁴ und so die Partizipation des Produktmanagements an der Entscheidungsfindung im Projektverlauf de facto erst ermöglicht werden. Letzteres ist nach der Wahrnehmung mehrerer Interviewpartner bei der Fallstudienunternehmung nicht gegeben.⁵⁶⁵ In diesem Zusammenhang wird mit der Rolle des ‚Product Owners‘ argumentiert,⁵⁶⁶ einem zentralen Bestandteil der agilen Methode ‚Scrum‘.⁵⁶⁷ Durch die ‚Product Owner‘-Rolle erfährt das Produktma-

⁵⁶² Vgl. Van de Weerd u.a. (2006), S. 7

⁵⁶³ Vgl. Herzwurm, Pietsch (2009), S. 57

⁵⁶⁴ Experte ‚J.W.‘ zur Frage D1, Experte ‚C.S.‘ zur Frage D1, Experte ‚T.W.‘ zur Frage B3, Experte ‚A.D.‘ zur Frage A2, Experte ‚T.Si.‘ zur Frage B3

⁵⁶⁵ Experte ‚C.S.‘ zur Frage C1, Experte ‚K.H.‘ zur Frage C1, Experte ‚J.W.‘ zur Frage C1

⁵⁶⁶ Experte ‚T.W.‘ zur Frage B3, Experte ‚A.D.‘ zur Frage A2

⁵⁶⁷ Scrum wird in der Fallstudienunternehmung situativ angewendet; zur Agilen Softwareentwicklung vgl. Kap. 2.1.3.2

nagement bei ‚Scrum‘ eine sowohl personell als auch organisatorisch äußerst unabhängige Stellung im gesamten Projektverlauf.

‚Scrum‘ ist ein „agiles Managementframework zur Entwicklung von Software, das aus wenigen klaren Regeln besteht.“⁵⁶⁸ Ein ‚Scrum‘-Projekt verläuft in einer Reihe kurzer Arbeitszyklen, den ‚Sprints‘. Jeder ‚Sprint‘ wandelt Anforderungen, die in einem ‚Product Backlog‘ stets aktuell und priorisiert festgehalten sind, in ein auslieferbares Produktinkrement um. Dazu findet zu Beginn jedes ‚Sprints‘ eine Planungssitzung statt, in der auf Grundlage des ‚Product Backlogs‘ das sog. ‚Sprint Backlog‘ erstellt wird. Dieses beschreibt alle notwendigen Aktivitäten zur Umsetzung der Anforderungen in ein Produktinkrement im aktuellen Sprint. Auf dieser Basis startet das Projektteam mit der Umsetzung des ‚Sprint Backlogs‘. Die weitere Koordination beschränkt sich ab hier auf eine kurze tägliche Besprechung, die ‚Daily Scrum‘. Die Dauer eines ‚Sprints‘ soll nicht länger als 30 Tage betragen, sodass bei ‚Scrum‘ in relativ kurzen Abständen immer wieder lauffähige, getestete und dokumentierte Produktinkremente fertiggestellt werden (vgl. Abb. 61).⁵⁶⁹

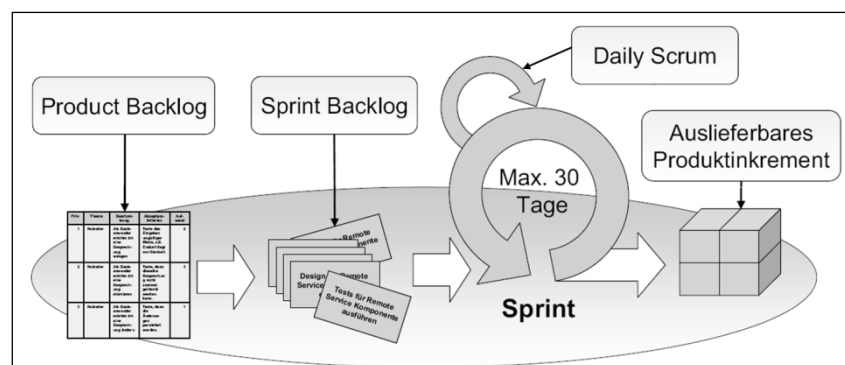


Abb. 61: ‚Scrum‘ im Überblick⁵⁷⁰

Zu den wenigen Regeln in ‚Scrum‘ zählt auch die konsequente Anwendung der drei definierten Rollen Projektteam, ‚Scrum-Master‘ und ‚Product Owner‘. Sie weisen im Wesentlichen die folgenden Verantwortlichkeiten auf: Das Projektteam entwickelt die auslieferbaren Produktinkremente und organisiert sich dabei weitgehend selbstständig. Der ‚Scrum-Master‘ hilft allen Projektbeteiligten, ‚Scrum‘ richtig anzuwenden und unterstützt das Projektteam dabei, seine Leistungsfähigkeit kontinuierlich zu verbessern. Der ‚Product Owner‘ nimmt in Scrum eine zentrale Stellung ein - er steuert als Repräsentant der Endkundenbedürfnisse über verschiedene Ansatzpunkte das gesamte ‚Scrum‘-Projekt.⁵⁷¹

⁵⁶⁸ Pichler (2008a), S. 1; Als ein agiler Ansatz verkörpert Scrum die Werte des Agilen Manifestes, vgl. hierzu Kap. 2.1.3.2

⁵⁶⁹ Vgl. Pichler (2008a), S. 7f.

⁵⁷⁰ Pichler (2008a), S. 7

⁵⁷¹ Vgl. Pichler (2008a), S. 9

„Scrum“ adressiert primär kleinere Softwareentwicklungsvorhaben, die kein Projektprogrammmanagement benötigen. Bereits ab einer Projekt(programm)größe von zwei Projektteams müssen zusätzliche Praktiken eingesetzt werden, um „Scrum“ erfolgreich umzusetzen. Hierzu gehören u.a. die Bildung von Feature- und Komponententeams⁵⁷² sowie der Einsatz projektübergreifend verbindlicher Normen und Standards. Letztere betreffen sowohl Entwicklungspraktiken wie bspw. Programmierrichtlinien und Testtools als auch Managementpraktiken wie bspw. sog. „Scrum of Scrums“-Sitzungen, welche der Koordination von mehreren Projektteams dienen.⁵⁷³ Weitere zusätzliche „Scrum“-Praktiken bei umfangreichen Entwicklungsvorhaben tangieren die Rolle des „Product Owners“. Da je Projektteam ein „Product Owner“ vorgesehen ist, führen umfangreiche Entwicklungsvorhaben zwangsläufig zu der Bildung von einem „Product Owner Team“, wobei einer der „Product Owner“ für das Gesamtprodukt verantwortlich gemacht wird und damit die Rolle des „Chief Product Owners“ einnimmt. Die Skalierung von „Scrum“ verläuft auf diese Weise analog bis hin zur Bildung von einem „Program Owner Team“, das aus den „Chief Product Owners“ der einzelnen Teilprodukte besteht.⁵⁷⁴

Die personelle Besetzung der „Product Owner“-Rolle bzw. der eben genannten Rollen bei einem skalierten „Scrum“ erfolgt typischerweise durch das Produktmanagement.⁵⁷⁵ Entsprechend decken sich die „Product Owner“- und die klassische Produktmanagement-Rolle im Entwicklungsprojekt zum großen Teil: Wie auch der klassische Produktmanager repräsentiert der „Product Owner“ die Endkundenbedürfnisse. Er arbeitet dabei mit dem Projektteam über den gesamten Projektverlauf eng zusammen und schlägt somit die Brücke zwischen den Endkunden und der Softwareentwicklung. Zu seinen Aufgaben und Verantwortlichkeiten zählen u.a. das Erfassen der Kundenbedürfnisse und die Detaillierung und Verfeinerung der im jeweils nächsten Sprint umzusetzenden Anforderungen.⁵⁷⁶

Der „Product Owner“ nimmt jedoch bei Weitem nicht nur Verantwortlichkeiten und Aufgaben des Produktmanagements wahr. "The product owner is a new, multifaceted role that unites the authority and responsibility traditionally scattered across separate roles, including the customer or sponsor, the product manager, and the project manager."⁵⁷⁷ Der „Product Owner“ entscheidet nicht nur über die produktmanagementrelevanten Fragestellungen und Zielsetzungen. Er steuert über die Formulierung von An-

⁵⁷² Vgl. hierzu auch Kap. 2.2.2.2

⁵⁷³ Vgl. Pichler (2008a), S. 125, 136ff., 152, 170

⁵⁷⁴ Vgl. Pichler (2008a), S. 134f.

⁵⁷⁵ Vgl. Pichler (2010), S. 2f.

⁵⁷⁶ Vgl. Pichler (2008a), S. 9ff.; vgl. Pichler (2010), S. 5f.

⁵⁷⁷ Pichler (2010), S. 2f.

forderungen und deren Priorisierung im ‚Product Backlog‘ den gesamten Softwareentwicklungsprozess.⁵⁷⁸ „Anders als traditionelle Produktmanager delegiert er die Leitung des Entwicklungsprojekts also nicht an einen Projektmanager ... Im Gegenteil: Der Product Owner führt wichtige Projektmanagementaufgaben selbst aus.“⁵⁷⁹ Hierzu weist ‚Scrum‘ vehement auf die Notwendigkeit hin, den ‚Product Owner‘ adäquat zu bevollmächtigen.⁵⁸⁰ „Er allein entscheidet über Auslieferungszeitpunkt, Funktionalität und Kosten.“⁵⁸¹

Insgesamt führt die ‚Product Owner‘-Rolle zu einer grundlegenden Veränderung des Zusammenspiels zwischen dem Produkt- und dem Projektmanagement.⁵⁸² Diese mündet in der eingangs erwähnten personell und organisatorisch sehr unabhängigen Position des Produktmanagements im gesamten Projektverlauf bei einer gleichzeitigen Aufgabe der personellen Unabhängigkeit des Projektmanagements. Zwar sind alle traditionellen Projektmanagementaufgaben auch in ‚Scrum‘-Projekten notwendig, doch „ist Scrum in einem Unternehmen vollständig etabliert, wird die Rolle des Projektleiters überflüssig.“⁵⁸³

Den Vertretern von ‚Scrum‘ ist dabei sehr wohl bewusst, dass das Zusammenlegen üblicherweise auf mehrere Personen verteilter Rollen eine Herausforderung für die Praxis darstellt.⁵⁸⁴ Nicht thematisiert wird jedoch, dass diese Aufgabe der personellen Unabhängigkeit nicht in jedem Fall zielführend ist.⁵⁸⁵ Da die ‚Product Owner‘-Rolle einen festen und zentralen Bestandteil von ‚Scrum‘ darstellt, ist damit auch der Einsatz von ‚Scrum‘ nicht generell zielführend, selbst wenn die Rahmenbedingungen grundsätzlich für eine agile Methode sprechen.⁵⁸⁶ Diese Ansicht wird auch in den Experteninterviews vertreten.

Nach Expertenmeinung ist ‚Scrum‘ gerade aufgrund der ‚Product Owner‘-Rolle nur bei leistungsumfangskritischen Projekten einsetzbar.⁵⁸⁷ Qualität-, zeit- oder kostenkritische Projekte erfordern dagegen eine andere Ausgestaltung der personellen und organisatorischen Unabhängigkeit der zentralen Rollen. Für qualitätskritische Projekte ist bspw. die Konstellation denkbar, bei welcher das Qualitätsmanagement dem Pro-

⁵⁷⁸ Der ‚Product Owner‘ ist die einzige Person, die für das Management des ‚Product Backlog‘ verantwortlich ist; vgl. Schwaber, Sutherland (2010), S. 7

⁵⁷⁹ Pichler (2008a), S. 10

⁵⁸⁰ Vgl. Pichler (2008b), S. 15; vgl. Pichler (2010), S. 6

⁵⁸¹ Pichler (2008a), S. 10

⁵⁸² Vgl. Pichler (2008b), S. 15

⁵⁸³ Pichler (2008a), S. 24

⁵⁸⁴ Vgl. Pichler (2008b), S. 14

⁵⁸⁵ Vgl. vorstehendes Kap. 3.2.3.1

⁵⁸⁶ Vgl. Kap. 2.1.3.2

⁵⁸⁷ Experte ‚T.W.‘ zur Frage B3, Experte ‚A.D.‘ zur Frage A2

jektmanagement gegenüber weisungsbefugt ist und somit das Projekt de facto leitet. Der Projektmanager ist hier ‚die rechte Hand‘ des Qualitätsmanagers und nimmt vielmehr eine Kosten- und Zeit-Controller-Rolle wahr.⁵⁸⁸ Als für ‚Scrum‘ ebenso wenig geeignet werden zeit- und kostenkritische Projekte angesehen, was offensichtlich an der Stellung des Projektmanagements in ‚Scrum‘ liegt. Als weiterer Grund wird hier aber auch die fehlende Transparenz und Vorhersehbarkeit über den Projektfortschritt genannt.⁵⁸⁹

Die Expertenaussagen werfen zwei weitere diskussionsbedürftige Aspekte auf. Zum einen ist es die Rolle des Qualitätsmanagements, auf die bei ‚Scrum‘ kaum eingegangen wird. Es wird zwar offensiv kommuniziert, dass die ‚klassische‘ Projektmanagement-Rolle bei ‚Scrum‘ nicht mehr vorhanden ist. Weniger im Mittelpunkt steht aber die Tatsache, dass es ebenso wenig ein personell wie organisatorisch unabhängiges Qualitätsmanagement gibt. Stattdessen ist es auch hier der ‚Product Owner‘ und damit schlussendlich das Produktmanagement, welches das Qualitätsmanagement verantwortet und weswegen bei ‚Scrum‘ die Tendenz zur Vernachlässigung von Qualitätsaspekten besteht.⁵⁹⁰

Der ‚Product Owner‘ überprüft am Ende von jedem ‚Sprint‘ die entstandenen Arbeitsergebnisse im ‚Sprint‘-Review, beurteilt ihre Angemessenheit und entscheidet über ihre Annahme. Des Weiteren unterliegen weitere wesentliche Qualitätsmanagement-Aufgaben implizit über das ‚Product Backlog‘ dem Kompetenzbereich des ‚Product Owners‘. Denn das ‚Product Backlog‘ enthält neben den funktionalen Anforderungen auch die Qualitätsanforderungen an die zu entwickelnde Software. Außerdem kann es Arbeitsergebnisse wie das Aufsetzen der Testumgebung und das Beseitigen von Defekten enthalten. Ob sonstige qualitätssichernde Maßnahmen angewendet werden, bleibt bei ‚Scrum‘ dem Projektteam im Rahmen seiner Selbstorganisation überlassen.⁵⁹¹

Der zweite diskussionsbedürftige Aspekt ist die Transparenz. Nach den Verfechtern von ‚Scrum‘ gilt als einer von drei Leitsätzen: „Die erste Säule ist Transparenz.“⁵⁹² Tatsächlich aber wird die hierzu im Widerspruch stehende o.g. Expertenmeinung auch durch die Studie von Pikkarainen u.a. gestützt. Demnach haben die agilen Praktiken bei Scrum wie ‚Sprint Planning‘ und ‚Daily Scrum‘ zwar positive Effekte auf die interne Kommunikation und Informationstransparenz innerhalb des Kernteams der Ent-

⁵⁸⁸ Experte ‚T.W.‘ zur Frage B3

⁵⁸⁹ Experte ‚A.D.‘ zur Frage A2

⁵⁹⁰ Vgl. Pichler (2010), S. 35

⁵⁹¹ Vgl. Pichler (2008a), S. 2, 8, 24, 27

⁵⁹² Vgl. Schwaber, Sutherland (2010), S. 3

wicklung (Projektteam, ‚Product Owner‘ und ‚Scrum Master‘). Doch im Hinblick auf die Kommunikation mit Interessensvertretern außerhalb des Kernteams stellen dieselben Praktiken große Hürden dar. ‚Scrum‘ bietet hierzu auch keine anderen Mechanismen, sodass der gesamte Projektverlauf allen außerhalb des Kernteams völlig intransparent erscheint.⁵⁹³

Die mangelnde Transparenz nach außen ist auf das Agile Manifest zurückzuführen, das ‚Scrum‘ als ein agiler Ansatz umsetzt. Dieses fordert nämlich, dass die informale persönliche Kommunikation im Team ausgeweitet wird und die formalisierte Dokumentation von (Zwischen-)Ergebnissen ersetzen soll. Vor allem die Umsetzung dieser Forderung, aber auch der weiteren Punkte des Agilen Manifesto und der dahinterstehenden Prinzipien, führen zu der mangelnden Transparenz von ‚Scrum‘-Projekten aus Sicht von außerhalb des Kernteams. Ein weiterer Grund ist dagegen weniger offensichtlich: So sind die vier Koordinationsprozesse in ‚Scrum‘ de facto entweder nicht institutionalisiert oder unter der alleinigen Verantwortung des ‚Product Owners‘. Neben den bereits angesprochenen Aufgaben und Verantwortlichkeiten im Umfeld des Qualitäts- und Projektmanagements ist dieser auch für das Reporting und das Risikomanagement verantwortlich. Da sich das ‚Product Backlog‘ über die gesamte Projektlaufzeit verändert und Anforderungsänderungen somit den Normalfall darstellen, kennt ‚Scrum‘ kein Änderungsverfahren.⁵⁹⁴

3.3 Bestimmung der übergeordneten Projektzielpräferenzen als Gestaltungsbedingung

Die Frage, an welchen Gestaltungsbedingungen die Koordination der Standardsoftwareentwicklung ausgerichtet werden muss (Forschungsfrage 3), wurde auf konzeptioneller Ebene in Kap. 2.3.3 beantwortet. Demnach bilden die

- übergeordneten Projektzielpräferenzen für die Projektziele die zentrale Gestaltungsbedingung
- für die Ausgestaltung der Koordinationsprozesse als Gestaltungsmaßnahmen
- mit dem Ziel, Koordinationseffektivität zu erreichen.

Die übergeordneten Projektzielpräferenzen stellen Vorgaben dar, welches der Projektziele der Erreichung eines anderen Projektzieles vorgezogen oder nachgeordnet werden soll. Solche Vorgaben können über die Festlegung von Zielgewichtungen bzw. Prioritäten erreicht werden.⁵⁹⁵ An dieser Stelle wurde mit dem Opportunitätskosten-

⁵⁹³ Vgl. Pikkarainen u.a. (2008), S. 304, 328f., 332

⁵⁹⁴ Vgl. Pichler (2008a), S. 24, 28

⁵⁹⁵ Vgl. Schmelzer (1991), S. 85; vgl. Hauschildt (1980), Sp. 2421f.

Konzept argumentiert: Steigen die potentiellen Opportunitätskosten im Falle der Zielverfehlung in einer der vier Projektzieldimensionen an, so steigt damit auch die Kritikalität der Zielerreichung in ebendieser Zieldimension an. Daher müssen die Projektzieldimensionen mit den potentiell höchsten Opportunitätskosten bei der Zielverfehlung auch die höchste Gewichtung bei der Bildung der Präferenzordnung für die Projektziele erfahren.

Im vorliegenden Kapitel wird dargestellt, anhand welcher konkreten Kriterien die übergeordneten Projektzielpräferenzen im Sinne des Opportunitätskosten-Konzeptes bestimmt werden können. Dabei wird, wie bereits in Kap. 2.3.3 erläutert, davon ausgegangen, dass bei der Standardsoftwareentwicklung Kostenzielverfehlungen eine relativ untergeordnete Rolle im Verhältnis zu Zielverfehlungen bei den anderen Projektzieldimensionen spielen.⁵⁹⁶ Diese grundsätzlich relativ niedrige Kostenkritikalität bei der Standardsoftwareentwicklung legt es nahe, die Kostenzielpräferenzen im Einzelfall nicht direkt, sondern mittelbar aus den anderen Zieldimensionen abzuleiten.⁵⁹⁷ Insofern sind für die Bestimmung der Kostenzielpräferenz keine expliziten Kriterien notwendig.

3.3.1 Projektzieldimension Leistungsumfang

3.3.1.1 Ableitung der Präferenz aus dem Priorisierungskriterium ‚Schaden‘

Für die Präferenzbestimmung der Projektzieldimension Leistungsumfang spielt v.a. die Anforderungsanalyse, eine Kernaktivität des primären Teilprozesses ‚Requirements Analysis and Definition‘, eine zentrale Rolle. Bei der Anforderungsanalyse findet aufbauend auf der Ermittlung von potenziellen Anforderungen⁵⁹⁸ an die zu erstellende Software deren Analyse statt. Hier sollen die ermittelten und bisher unstrukturierten Anforderungen klassifiziert und anschließend priorisiert werden. Durch die Klassifikation entstehen Gruppen von eng zusammenhängenden Anforderungen, wobei redundante, sich überlappende und konfliktäre Anforderungen dabei deutlich werden. Die Anforderungen können nun priorisiert werden, um festzulegen, welche in ein bestimmtes Inkrement⁵⁹⁹ eingehen sollen. Das Ergebnis der Anforderungsanalyse ist als ein optimales Set von Anforderungen für die sukzessive Implementierung innerhalb von Inkrementen.⁶⁰⁰

⁵⁹⁶ Diese Annahme wird sowohl durch andere (vgl. Carmel (1995), S. 119; vgl. Gerhardt (1992), S. 115), als auch durch die eigenen empirischen Ergebnisse (Experte ‚B.E.‘ zur Frage A1, Experte ‚T.M.‘ zur Frage A1, Experte ‚K.H.‘ zur Frage A1, Experte ‚O.S.‘ zur Frage A1) gestützt.

⁵⁹⁷ Vgl. Kap. 3.4.1.2

⁵⁹⁸ Zu Anforderungen vgl. Kap. 2.3.3.2

⁵⁹⁹ Ausbaustufe einer Software; vgl. hierzu im Detail Kap. 2.2.2.2

⁶⁰⁰ Vgl. Pohl (2007), S. 311ff., 526

Abb. 62 enthält Priorisierungsinformationen aus der Anforderungsanalyse, die bei der Fallstudienunternehmung zu Projektbeginn (weiterhin) verfügbar sind. Es handelt sich dabei um die Priorisierung von Anforderungen nach ihrer Wichtigkeit. Aus diesen Informationen kann auf die Präferenz für die Projektzieldimension Leistungsumfang geschlossen werden. Die grundsätzliche Annahme ist dabei, dass je höher der Anteil von bei der Anforderungsanalyse hoch priorisierten funktionalen Anforderungen am gesamten Projektumfang ist, desto kritischer ist in diesem Projekt die Zielerreichung in der Projektzieldimension Leistungsumfang. Durch diese Annahme findet hier eine zusammenfassende Betrachtung von vielen Anforderungen als eine ‚funktionale Projektanforderung‘ und somit schlussendlich eine Anforderungspriorisierung auf Projektebene statt. Ein solches Verfahren bringt zwar Informationsverlust und einen gewissen Grad an Heuristik mit sich, es ist jedoch insbesondere bei sehr vielen einzelnen Anforderungen üblich. Die aggregierte Betrachtung reduziert nicht nur die Anzahl der zu priorisierenden Anforderungen, sondern implizit auch die Anzahl der Abhängigkeiten zwischen ihnen.⁶⁰¹

Mustervorlage	Datenbankgestützte Mustervorlage, die als Basis für die Produktdefinition, Produktplanung sowie das ‚commitment management‘ hinsichtlich Ressourcen und Fragen der Softwarearchitektur dient. Die Informationen werden zum Teil für die Erstellung und ständige Aktualisierung anderer Dokumente übernommen, so bspw. für das Lastenheft.
Informationsfeld	Priorisierte funktionale Anforderungen an die zu erstellende Standardsoftware nach ihrer Wichtigkeit aus Marktsicht
Wertebereich der Information	Ordinal: sehr hoch/hoch/mittel
Bezug zur Projektzieldimension	Leistungsumfang
Normierung auf Projektebene	Je höher der Anteil von bei der Anforderungsanalyse hoch priorisierten funktionalen Anforderungen am gesamten Projektumfang ist, desto kritischer ist in diesem Projekt die Zielerreichung in der Projektzieldimension Leistungsumfang.

Abb. 62: Priorisierungskriterium ‚Wichtigkeit‘ zur Präferenzbestimmung der Projektzieldimension Leistungsumfang

Das Priorisierungskriterium ‚Wichtigkeit‘ ist als Grundlage für die Präferenzbestimmung der Projektzieldimension Leistungsumfang aus einem wesentlichen Grund diskussionsbedürftig: Die Wichtigkeit einer Anforderung kann viele verschiedene Facetten adressieren, so bspw.

⁶⁰¹ Vgl. Berander, Andrews (2005), S. 82f.

- die Wichtigkeit im Hinblick auf eine frühzeitige Implementierung (d.h. Dringlichkeit),
- die Wichtigkeit der Anforderung für die Akzeptanz der Software aus Kundensicht,
- die Wichtigkeit der Anforderung in Bezug auf die Gestalt der Systemarchitektur, oder auch
- die strategische Wichtigkeit der Anforderung im Hinblick auf die Marktposition des Unternehmens.⁶⁰²

Insbesondere ist die Kritikalität der Erfüllung einer funktionalen Anforderung nicht zwangsläufig mit ihrer ‚Wichtigkeit‘ aus der Anforderungsanalyse gleichzusetzen. Hat bspw. eine funktionale Anforderung ihren Ursprung in der Sicherstellung der Konformität mit einem Industriestandard oder mit gesetzlichen Rahmenbedingungen, ist die Kritikalität der Erfüllung dieser Anforderung sehr hoch, bis hin zum verweherten Markteintritt für die Softwarelösung. Auf der anderen Seite muss diese funktionale Anforderung aus Sicht der verschiedenen o.g. Facetten des Priorisierungskriteriums ‚Wichtigkeit‘ nicht zwingend hoch priorisiert sein.⁶⁰³

Zur Priorisierung von Anforderungen auf Grundlage eines oder mehrerer Kriterien kommen neben der Wichtigkeit auch zahlreiche weitere Kriterien wie bspw. Kosten, Zeitdauer, Risiko oder Volatilität⁶⁰⁴ einer Anforderung in Frage.⁶⁰⁵ Alternativ können diese indirekt, d.h. als Facetten des Priorisierungskriteriums Wichtigkeit, in die Priorisierung eingehen. Für die Präferenzbestimmung der Projektzieldimension Leistungsumfang ist jedoch ein weiteres Priorisierungskriterium von zentraler Bedeutung: Es ist nämlich möglich, den potentiell eintretenden Schaden direkt zu evaluieren, wenn eine Anforderung nicht erfüllt werden kann. Das Priorisierungskriterium ‚Schaden‘ betrachtet in diesem Sinn direkt „den Umfang des entstehenden Schadens oder Nachteils, sollte die Anforderung nicht berücksichtigt bzw. implementiert werden. Der eintretende Schaden kann sich z.B. auf eventuelle Vertragsstrafen, verringerte Absatzmöglichkeiten des Systems am Markt oder Prestigeverlust beziehen.“⁶⁰⁶ Es kann daher unmittelbar als Kriterium für die Präferenzbestimmung der Projektzieldimension Leistungsumfang verwendet werden.

⁶⁰² Vgl. Pohl (2007), S. 528; vgl. Lehtola, Kauppinen (2006), S. 8; vgl. Berander, Andrews (2005), S. 72

⁶⁰³ Vgl. Berander, Andrews (2005), S. 73

⁶⁰⁴ Wahrscheinlichkeit dafür, dass sich eine Anforderung im Verlauf des Entwicklungsprozesses verändert.

⁶⁰⁵ Vgl. Pohl (2007), S. 528; vgl. Berander, Andrews (2005), S. 72-74

⁶⁰⁶ Pohl (2007), S. 528

Das Priorisierungskriterium ‚Schaden‘ ist für sich genommen für die Zwecke der Phase ‚Anforderungsanalyse‘ nicht geeignet und kommt daher üblicherweise nur als Bestandteil von multikriteriellen Priorisierungsverfahren vor. Ein solches sehr bekanntes Verfahren ist das Priorisierungsverfahren nach Wiegers. Die Anforderungspriorisierung wird hier mit vier Kriterien vorgenommen, u.a. ‚Wichtigkeit‘ und ‚Schaden‘, was nochmals an dieser Stelle verdeutlicht, dass beide Priorisierungskriterien nicht das Selbe sind. Die Basisannahme von Wiegers ist, dass sich die Priorität einer Anforderung proportional zu deren Nutzen und umgekehrt proportional zu den Kriterien ‚Kosten‘, ‚Schaden‘ und ‚Risiko‘ verhält. Im Mittelpunkt des Verfahrens steht eine Priorisierungsmatrix, in der die Anforderungsprioritäten auf Grundlage der vier genannten Kriterien und der Basisannahme berechnet werden.⁶⁰⁷

Ein weiteres Priorisierungsverfahren, in welches das Priorisierungskriterium ‚Schaden‘ eingeht, ist die Anforderungspriorisierung auf Grundlage des sog. Kano-Modells. Das Kano-Modell⁶⁰⁸ beschreibt den Zusammenhang zwischen verschiedenen Leistungskomponenten eines Produktes und der wahrgenommenen Produktqualität und Zufriedenheit des Kunden. Die im Kano-Modell definierte Bewertungs- und Klassifikationstechnik kann auch sehr effektiv zur Priorisierung von Anforderungen verwendet werden, v.a. in Zusammenhang mit der Klassifizierung von Produktmerkmalen im Hinblick auf deren Marktwirkung. Ausgangspunkt der Klassifizierung ist die Einordnung von Produktmerkmalen in die drei folgenden Merkmalsklassen:⁶⁰⁹

- Ein Basismerkmal liegt vor, wenn es von den Kunden als selbstverständlich erachtet wird. Daher muss die Softwarelösung dieses Merkmal aufweisen, wobei es nicht zwingend von Kundenseite artikuliert wird;
- Ein Zusatzmerkmal liegt vor, wenn es von den Kunden bewusst gefordert wird. Zusatzmerkmale der Software bestimmen den Zufriedenheitsgrad der Kunden. Zunehmendes Ausmaß an Zusatzmerkmalen führt zu einer Steigerung der Kundenzufriedenheit;
- Ein Begeisterungsmerkmal liegt vor, wenn es den Kunden nicht bewusst war bzw. wenn sie die Umsetzung dieses Merkmals im Softwaresystem nicht erwartet haben. Die Kundenzufriedenheit wächst überproportional, falls die Software ein solches Merkmal umsetzt.

⁶⁰⁷ Vgl. Wiegers (2005), S. 236-239; vgl. Pohl (2007), S. 536-538; vgl. Lehtola, Kauppinen (2006), S. 9f.

⁶⁰⁸ Vgl. Kano u.a. (1984)

⁶⁰⁹ Vgl. Pohl (2007), S. 531f.

Abb. 63 zeigt den von Kano empirisch ermittelten Einfluss des Erfüllungsgrades von Basis-, Zusatz- und Begeisterungsmerkmalen auf die Kundenzufriedenheit. Wie bereits kurz erwähnt, werden die Merkmale teilweise von den Kunden nicht artikuliert und nur unbewusst vorausgesetzt. Daher wird im Kano-Modell ein systematisches, vierstufiges Vorgehen zur Klassifizierung von Produktmerkmalen vorgeschlagen:⁶¹⁰

- 1) Identifikation von Produktmerkmalen, die klassifiziert werden sollen;
- 2) Entwurf eines Fragebogens, in dem für jedes Produktmerkmal erfragt wird, wie ein potenzieller Kunde empfindet, wenn (1) das Merkmal im Softwaresystem enthalten ist (sog. funktionale Frage) und (2) das Merkmal nicht im Softwaresystem enthalten ist (sog. dysfunktionale Frage);
- 3) Auswertung der Befragungsergebnisse und Ermittlung der Durchschnittswerte;
- 4) Zuordnung des Produktmerkmals zu der zugehörigen Merkmalsklasse.

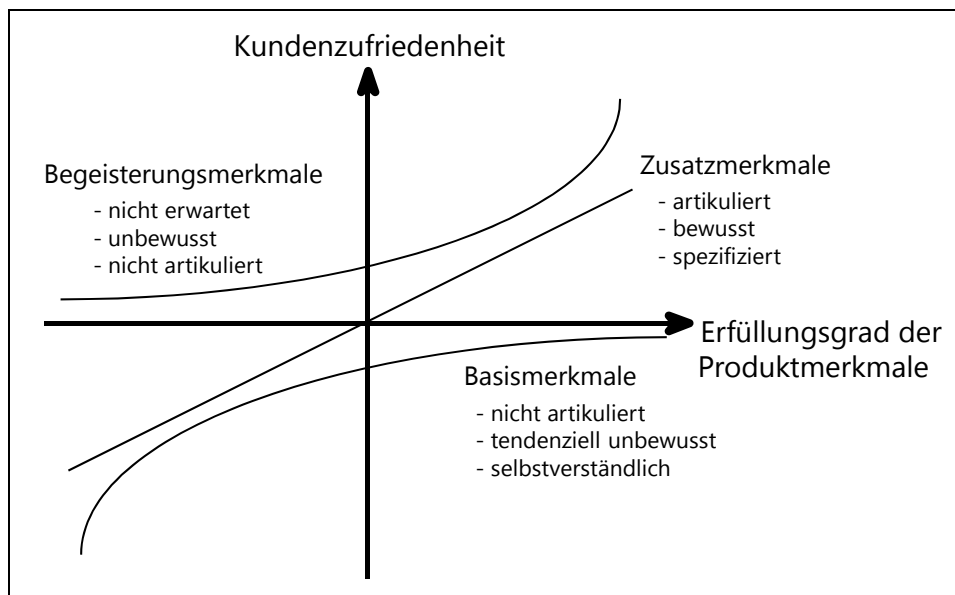


Abb. 63: Einfluss des Erfüllungsgrades von Basis-, Zusatz- und Begeisterungsmerkmalen auf die Kundenzufriedenheit im Kano-Modell⁶¹¹

Durch die Klassifikation von Produktmerkmalen werden implizit auch die Anforderungen klassifiziert, deren Implementierung zur Realisierung des jeweils betrachteten Merkmals dient. Die daraus resultierende Klassifikation von Anforderungen nach solchen, die zur Realisierung von Basis-, Zusatz- oder Begeisterungsmerkmalen der Software dienen, kann bspw. zur Festlegung eingesetzt werden, welche Basis-, Zusatz- oder Begeisterungsmerkmale in den einzelnen Inkrementen zu realisieren sind. Dabei ist zu beachten, dass zur Realisierung des entsprechenden Produktmerkmals alle aus diesem Merkmal abgeleiteten Anforderungen realisiert werden müssen.⁶¹²

⁶¹⁰ Vgl. Pohl (2007), S. 533

⁶¹¹ Pohl (2007), S. 532; zur Abb. im Original vgl. Kano u.a. (1984)

⁶¹² Vgl. Pohl (2007), S. 534

In der Fallstudienunternehmung wird das Kano-Modell in einem ähnlichen Zusammenhang direkt im Anschluss an die Anforderungsanalyse eingesetzt (vgl. Abb. 64). In einer sog. Concept Selection Scorecard (CSSC) werden zusammenhängende funktionale Anforderungen als sog. Anforderungsgruppen⁶¹³ nach Kano klassifiziert sowie mit einer Priorität versehen. Die Priorität leitet sich dabei aus der Priorisierung in der Anforderungsanalyse ab. Darüber hinaus geht eine Aufwandsschätzung in drei Ausprägungen je Anforderungsgruppe in die CSSC ein:

- Aufwand für die Anforderungsgruppe bei einer ‚Minimallösung‘;
- Aufwand für die Anforderungsgruppe bei einer Lösung mit ‚normalem‘ Aufwand;
- Aufwand für die Anforderungsgruppe bei einer ‚Optimallösung‘.

Mithilfe der CSSC und der Anforderungsklassifizierung nach dem Kano-Modell soll festgelegt werden, welche der drei möglichen Umsetzungsalternativen gewählt werden soll. Da das Priorisierungskriterium ‚Schaden‘ in das Kano-Modell nur indirekt über die dysfunktionalen Fragen im zweiten Klassifizierungsschritt eingeht, kann daraus auch nicht unmittelbar auf die Kritikalität der Projektzieldimension Leistungsumfang geschlossen werden. Allerdings lassen sich sowohl aus der Klassifizierung nach dem Kano-Modell, als auch aus ihrer Kombination mit der Priorisierung und der gewählten Aufwandsintensität, sinnvolle Konventionen ableiten, um auf die Kritikalität der Leistungszieldimension schließen zu können (vgl. Abb. 64).

Mustervorlage	Concept Selection Scorecard: Mustervorlage zur Evaluation möglicher Umsetzungsalternativen für zusammenhängende funktionale Anforderungen
Informationsfeld	Klassifizierung der Anforderungsgruppe nach Kano
Wertebereich der Information	Drei Klassen: Basis-, Zusatz-, Begeisterungsmerkmal
Bezug zur Projektzieldimension	Leistungsumfang
Normierung auf Projektebene	Je höher der Anteil von hoch priorisierten funktionalen Anforderungen hinsichtlich des Priorisierungskriteriums ‚Schaden‘ am gesamten Projektumfang, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Leistungsumfang.

Abb. 64: Kano-Modell und Priorisierungskriterium ‚Schaden‘ zur Präferenzbestimmung der Projektzieldimension Leistungsumfang

⁶¹³ Die Abstraktionsebene einer Anforderungsgruppe liegt zwischen einem Feature oder Use Case und einzelnen Kundenanforderungen.

3.3.1.2 Wiederverwendung von Priorisierungsinformationen

Die Priorisierungsinformationen aus der Anforderungsanalyse sind in einem anderen Kontext entstanden, und originär auch in einem anderen Kontext verwendet worden als es beim situativen Koordinationsansatz der Fall sein soll. Unter welchen Umständen sie tatsächlich für die Ableitung der Kritikalität der Projektzieldimension Leistungsumfang herangezogen werden können, muss kritisch hinterfragt werden. In diesem Zusammenhang hat sich bei den Experteninterviews sehr eindeutig eine generelle und kritische Diskussion zur Wiederverwendung von Priorisierungsinformationen zu verschiedenen Zwecken hervorgerufen.

Die Anforderungspriorisierung unterstützt nicht ausschließlich Aktivitäten im Rahmen der Anforderungsanalyse. Die Priorität einer Anforderung dokumentiert ganz allgemein die Bedeutung der Anforderung bezüglich eines oder mehrerer ausgewählter Priorisierungskriterien. Die Bestimmung der Priorität kann dabei isoliert für jede Anforderung oder durch paarweise Vergleiche erfolgen. Das Ergebnis der Anforderungspriorisierung ist „eine (partielle) Ordnung der betrachteten Anforderungen in Bezug auf die für die Priorisierung ausgewählten Kriterien.“⁶¹⁴ Eine solche Information kann grundsätzlich als Input für eine Reihe von Aktivitäten der Softwareentwicklung dienen, wengleich die Planung von Inkrementen den zentralen Zweck der Anforderungspriorisierung darstellt.⁶¹⁵

Da sich je nach unterstützter Aktivität das Ziel der Priorisierung unterscheidet, sind zwangsläufig auch jedes Mal andere Priorisierungskriterien und andere Abstraktionsebenen der Priorisierung notwendig. Unter dieser Prämisse ist die Anforderungspriorisierung auch in Zusammenhang mit den folgenden Aktivitäten erforderlich:⁶¹⁶

- Im Rahmen der Gewinnung von Anforderungen und damit noch vor der Anforderungsanalyse wird mittels Priorisierung festgelegt, welche Anforderungen als nächstes detailliert und welche Stakeholder besonders intensiv in die Anforderungsgewinnung einbezogen werden sollen;
- Konfliktäre Anforderungen und v.a. die größtenteils konfliktären nicht-funktionalen Anforderungen können explizit gegenübergestellt und in diesem Rahmen priorisiert werden, um so zunächst Übereinstimmung bei den für kritisch erachteten Anforderungen herbeizuführen;

⁶¹⁴ Pohl (2007), S. 525

⁶¹⁵ Vgl. Pohl (2007), S. 525f.

⁶¹⁶ Vgl. Berander, Andrews (2005), S. 70f.; vgl. Pohl (2007), S. 526; jeweils mit weiteren Verweisen

- Im Rahmen der Validierung von Anforderungen kann durch eine Priorisierung festgelegt werden, in welcher Reihenfolge und mit welcher Intensität die Anforderungen validiert werden sollen;
- Beim Änderungsmanagement wird eine Priorisierung von Änderungsanträgen durchgeführt, um über wichtige Änderungen möglichst zeitnah entschieden und diese umsetzen zu können.

Der Priorisierungszweck ist nicht nur die entscheidende Determinante bei der Wahl der Priorisierungskriterien, sondern auch bei der Bestimmung der richtigen Abstraktionsebene von Priorisierungsinformationen. Die o.g. Informationen werden für Zwecke der Anforderungsanalyse verwendet. Zu diesem Zeitpunkt können die Anforderungen noch sehr grob formuliert sein.⁶¹⁷ Für den originären Zweck der Priorisierung stellt dies zwar kein Problem dar. Auch wird es in der Praxis explizit begrüßt, dass auf Priorisierungsinformationen aus der Anforderungsanalyse über diese Projektphase hinaus weiter zurückgegriffen wird.⁶¹⁸ Doch an dieser Stelle müssen zwingend die Qualität und der Detaillierungsgrad von Anforderungen, welche priorisiert worden sind, kritisch überprüft werden.⁶¹⁹

Bei den durchgeführten Interviews hat sich keine herrschende Expertenmeinung zur richtigen Abstraktionsebene für den vorliegenden Priorisierungszweck herausgebildet. Einige Vorschläge zielen auf die Präferenzbestimmung der Projektzieldimension Leistungsumfang auf der Ebene von sog. Anwendungsfällen ab.⁶²⁰ Anwendungsfälle beschreiben zielorientiert und aus der Außensicht, d.h. aus Sicht der beteiligten Akteure an der Interaktion mit dem Softwaresystem, was dieses leisten soll und welche Interaktionen hierzu notwendig sind. Durch einen Anwendungsfall wird ein vollständiger und sinnvoller Ablauf von Ereignissen dargestellt. Anwendungsfälle werden natürlichsprachlich in einer strukturierten Form beschrieben.⁶²¹

Als weitere mögliche Abstraktionsebenen für die Präferenzbestimmung werden bei den Experteninterviews Entwicklungsobjekte sowie die einzelnen konkreten Systemanforderungen genannt.⁶²² Alle bisher angesprochenen Abstraktionsebenen für die Priorisierung von Anforderungen werden auch in der Literatur vorgeschlagen. Je nach Projektgröße gelten als angemessene Abstraktionsebene für die Priorisierung sowohl Funktionen bzw. Features, als auch Anwendungsfälle und funktionale Anforderungen.

⁶¹⁷ Experte ‚J.W.‘ zur Frage A3

⁶¹⁸ Experte ‚T.M.‘ zur Frage A4, Experte ‚L.G.‘ zur Frage A4, Experte ‚J.K.‘ zur Frage A4, Experte ‚B.E.‘ zur Frage A4

⁶¹⁹ Experte ‚J.W.‘ zur Frage A4

⁶²⁰ Experte ‚T.W.‘ zur Frage A4, Experte ‚F.E.‘ zur Frage A4

⁶²¹ Vgl. Ludewig, Lichter (2006), S. 357f.

⁶²² Experte ‚B.E.‘ zur Frage A1, Experte ‚B.E.‘ zur Frage A4

Möglich ist auch eine kombinierte Vorgehensweise, bei der erst eine erste Priorisierung der Funktionen bzw. Features durchgeführt wird, um anschließend die funktionalen Anforderungen innerhalb dieser zu priorisieren. In diesem Fall, sowie grundsätzlich dann, wenn auf mehreren Abstraktionsebenen priorisiert wird, müssen bestimmte Konvention definiert werden. Diese können bspw. festlegen, ob eine hohe Anforderungspriorität automatisch an die untergeordneten Anforderungen weitergegeben wird, oder ob jede einzelne funktionale Anforderung eine eigene Priorität besitzen soll.⁶²³

Wie auch für Anwendungsfälle gilt für Entwicklungsobjekte und Systemanforderungen allerdings, dass die notwendigen Informationen nicht unmittelbar zu Projektbeginn vorliegen, sondern erst in der Phase Grobanalyse erarbeitet werden. D.h., dass die Präferenzbestimmung der Projektzieldimension Leistungsumfang in diesen Fällen nicht unmittelbar zu Projektbeginn stattfinden könnte. Informationen, die zu diesem Zeitpunkt vorliegen, finden sich üblicherweise in der Projektvereinbarung (auch Projektauftrag oder Project Charter), im Lastenheft und im Projektplan. Desweiteren kommen zahlreiche weitere Dokumente in Betracht, auf welche aus dem Lastenheft, dem Projektplan und der Projektvereinbarung heraus referenziert wird.⁶²⁴

Die Begriffe ‚Projektvereinbarung‘ sowie ‚Projektauftrag‘ sind nicht genormt und werden dementsprechend unterschiedlich verwendet. Da eine Vereinbarung nicht den verbindlichen Charakter eines Auftrages hat, wird in der Regel bei internen Projekten, welche bei der Betrachtung der Standardsoftwareentwicklung grundsätzlich vorliegen, von einer Projektvereinbarung gesprochen. In der Projektvereinbarung werden die Sach- und Formalziele des Projekts, die Rollen im Projekt, sowie die sonstigen wesentlichen Rahmenbedingungen des Projekts festgelegt.⁶²⁵

Das Lastenheft beschreibt die "Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines (Projekt-)Auftrags."⁶²⁶ Analog ist in der Softwareentwicklung das Lastenheft „ein Dokument, in dem die grundlegenden Anforderungen an ein zu erstellendes oder zu beschaffendes Software-System niedergelegt werden.“⁶²⁷ Ein Lastenheft sollte grundsätzlich ergebnisorientiert und vom Auftraggeber formuliert werden. Da dies bereits in der Planungsphase erfolgt, sind Lastenhefte weniger detailliert als sog. Pflichtenhefte. Es soll lediglich eine Konzentration auf fundamentale Eigenschaften des Produktes auf einem genügend hohen Abstrakti-

⁶²³ Vgl. Wiegers (2005), S. 299

⁶²⁴ Die genannten Dokumente können in mehr oder weniger umfassender und/oder (un)verbindlicher Form, sowie in unterschiedlicher Reihenfolge und im unterschiedlichen Zusammenhang erstellt werden. Auf diese Aspekte im Detail braucht hier nicht näher eingegangen zu werden.

⁶²⁵ Vgl. Angermeier (2005), S. 368f.

⁶²⁶ Vgl. DIN (Hrsg., 1997), zitiert nach Angermeier (2005), S. 213

⁶²⁷ Myrach (2010)

onsgrad erfolgen, sodass die Anforderungen präzise beschrieben, aber nicht bis ins Detail spezifiziert sind. Lastenhefte können in dieser Hinsicht als grobe Pflichtenhefte aufgefasst werden, die sich weiterentwickeln und detaillieren lassen, um so zu einem tatsächlichen Pflichtenheft zu gelangen.⁶²⁸

Nach DIN 69905 ist der Projektplan die "Gesamtheit aller im Projekt vorhandenen Pläne".⁶²⁹ Dabei wird der Begriff ‚Plan‘ nicht weiter spezifiziert. Zum Mindestumfang eines Projektplans gehören der Projektstrukturplan, der Terminplan und der Kostenplan. Dabei definiert der Projektstrukturplan den Projektgegenstand und das Ergebnis, der Terminplan legt die zeitlichen Rahmenbedingungen fest und der Kostenplan den Aufwand.⁶³⁰ Einen detaillierteren Einblick in die Inhalte eines Projektplans ermöglicht auch die Norm IEEE Std. 1058.1 mit ihrer Mustervorgabe für die Inhaltsstruktur eines Projektplans. Zum Umfang gehören hier u.a. Ausführungen zu den Arbeitspaketen, Projektergebnissen und Ressourcenanforderungen sowie zum Zeitplan und Budget.⁶³¹

3.3.2 Projektzieldimension Qualität

3.3.2.1 Ableitung der Präferenz aus Qualitätsmodellen

Der Begriff der Softwarequalität ist ein mehrdimensionales Konstrukt. So definiert der bekannte Softwarequalitätstandard ISO/IEC 9126 Softwarequalität als die Gesamtheit von Merkmalen und Merkmalswerten eines Softwareprodukts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.⁶³² Eine solche allgemeine Definition ist jedoch für die praktische Anwendung nicht ausreichend. Hierfür muss Softwarequalität erst mithilfe eines Qualitätsmodells durch Ableiten von Unterbegriffen operationalisiert werden. D.h. es muss eine Verfeinerung der Merkmale auf der obersten Abstraktionsebene durch Teilmerkmale erfolgen, welche ihrerseits durch Qualitätsindikatoren mess- und bewertbar gemacht werden. Ein so aufgebautes Modell wird als ein factor-criteria-metrics-model (FCM-Modell) bezeichnet. Da mehrere Qualitätsmerkmale gemeinsame Teilmerkmale besitzen können, sind sowohl baum- als auch netzartige FCM-Modelle möglich. Desweiteren können Teilmerkmale selbst eine Hierarchie bilden, sodass ein ‚FCM‘-Modell mehr als drei Ebenen enthalten kann.⁶³³

⁶²⁸ Vgl. Myrach (2010); Balzert (2001), S. 62 ff.

⁶²⁹ Vgl. DIN (Hrsg., 1997)

⁶³⁰ Vgl. Angermeier (2005), S. 347f.

⁶³¹ Vgl. IEEE (Hrsg., 1988)

⁶³² Vgl. ISO/IEC (Hrsg., 2001)

⁶³³ Vgl. Balzert (2008), S. 461f.; vgl. Herzwurm, Mikusz (2010a)

In der Praxis und Forschung existiert eine Vielzahl von Qualitätsmodellen.⁶³⁴ Zahlreiche Softwareunternehmen verfügen über ihre eigenen, unternehmens- und domänen-spezifischen Qualitätsmodelle. Diese lehnen sich oftmals an den o.g. Softwarequalitätsstandard ISO/IEC 9126 an. In diesem kommen zwei FCM-Modelle zur Anwendung – eines zur Definition der sog. Nutzungsqualität sowie ein weiteres FCM-Modell zur Definition der sog. externen und internen Qualität des Softwareproduktes. Letzteres besteht aus 6 Merkmalen, die weiter in eine Vielzahl von Teilmerkmalen aufgespalten werden (vgl. Abb. 65).⁶³⁵

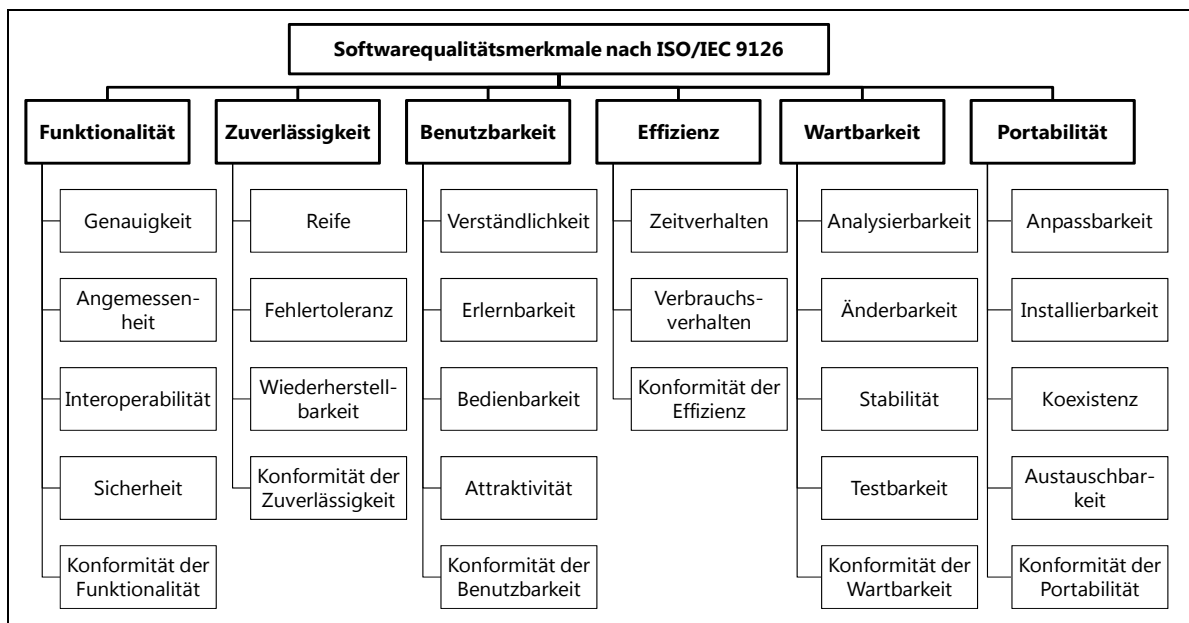


Abb. 65: Merkmale zur Definition der externen und internen Qualität des Softwareproduktes nach ISO/IEC 9126⁶³⁶

Nicht an jedes Softwareprodukt und damit zwangsläufig auch nicht an jeden Softwareentwicklungsprozess und jedes Softwareentwicklungsprojekt werden dieselben Qualitätsanforderungen gestellt. In der Regel ist eine Qualitätszielbestimmung auf der Softwareprodukt-Ebene erforderlich. In bestimmten Fällen kann es jedoch auch nötig sein, je Teilbestandteil des Softwareproduktes eine Qualitätszielbestimmung vorzunehmen. Hier führt die Qualitätszielbestimmung zu Qualitätszielen, aus welchen sich weiter Qualitätsanforderungen ableiten lassen. Qualitätsanforderungen legen fest, welche Qualitätsmerkmale im konkreten Fall als relevant erachtet werden und in welcher Qualitätsstufe sie erreicht werden müssen. Die vorgegebenen Qualitätsanforderungen beeinflussen wesentlich die Gestaltung des Softwareentwicklungsprozesses.⁶³⁷

⁶³⁴ Eine Zusammenstellung von atomaren Merkmalen der Produktqualität (normenunabhängig) findet sich in Ludewig, Lichter (2006), S. 66ff.

⁶³⁵ Vgl. Balzert (2008), S. 462ff.; vgl. Herzwurm, Mikusz (2010a)

⁶³⁶ Vgl. ISO/IEC (Hrsg., 2001)

⁶³⁷ Vgl. Balzert (2008), S. 471-473

Bei der Standardsoftwareentwicklung macht der gewöhnlich große Umfang der Entwicklungsvorhaben eine Qualitätszielbestimmung je Teilprodukt und damit je Entwicklungsprojekt erforderlich. Denn die verschiedenen Teilbestandteile einer Standardsoftware erfordern hier verschiedene Kombinationen von Qualitätsmerkmalen. Bspw. kann die Änderbarkeit für bestimmte Komponenten wichtig sein, während für andere Komponenten die Bedienbarkeit entscheidend ist.⁶³⁸ Wird dabei ein Qualitätsmodell benutzt, das sich auf Kennzahlen abstützt, dann sind Qualitätsstufen zu definieren und es ist im Einzelfall festzulegen, welche Qualitätsstufen erreicht werden müssen. Konkrete Qualitätsstufen können bspw. durch die Vorgabe von Relativmaßen definiert werden. Bezogen auf die Einhaltung von Standards könnte die Qualitätsstufe bei normaler Produktqualität zwischen einschließlich 0 und einschließlich 0,1 definiert werden. D.h. von zehn betrachteten Elementen darf bei dieser Qualitätsstufe höchstens ein Element den Standard nicht erfüllen.⁶³⁹

Bei der Fallstudienunternehmung werden die Qualitätsziele ‚top down‘ ausgehend von der Softwareprodukt-Ebene, bzw. aus der projektbezogenen Sicht ausgehend von der Projektprogramm-Ebene, bestimmt. Auf dieser Ebene werden zunächst relativ pauschal Qualitätsstufen definiert. Ausgangspunkt ist dabei das unternehmenseigene Qualitätsmodell, das sehr stark an das Modell der ISO/IEC 9126 zur Definition der externen und internen Qualität des Softwareproduktes angelehnt ist. Aus der Vielzahl von Teilmerkmalen der Qualität werden dabei für jedes Entwicklungsvorhaben diejenigen Teilmerkmale bestimmt, die situationsspezifisch als besonders relevant erachtet werden und damit in einer absolut höchsten Qualitätsstufe erreicht werden müssen. Sie werden als ‚focus product standards‘ bezeichnet und im Qualitätsplan des Projektprogramms dokumentiert.

Die Qualitätszielbestimmung auf der Projektprogramm-Ebene dient als Anhaltspunkt für die Qualitätskritikalität von Projekten. Über die Verknüpfung der ‚focus product standards‘ mit den einzelnen Features, Komponenten und funktionalen Anforderungen, und damit implizit mit den Projekten in einem Projektprogramm, lassen sich nun diejenigen Projekte bestimmen, die in einem unmittelbaren Zusammenhang zu dem fokussierten Qualitätsteilziel stehen. Dieser Punkt wird in der Fallstudienunternehmung bei der weiteren Detaillierung der Qualitätsziele auf der Projektebene aufgegriffen. Direkt zu Projektbeginn ist für jedes Entwicklungsprojekt ein sog. ‚product standard compliance statement‘ obligatorisch.

⁶³⁸ Vgl. Wiegers (2005), S. 203

⁶³⁹ Vgl. Balzert (2008), S. 471-473

Das ‚product standard compliance statement‘ umfasst

- die für das jeweilige Projekt relevanten ‚focus product standards‘,
- den Nachweis über ihre Berücksichtigung bei der initialen Aufwandsschätzung und Projektplanung, sowie insb. auch
- Informationen über die Folgen einer nicht erreichten Umsetzung dieser ‚focus product standards‘ (vgl. Abb. 66).

Mustervorlage	‘Product standard compliance statement‘
Informationsfeld	Auflistung der für das jeweilige Projekt relevanten ‚focus product standards‘ incl. von Informationen über die Folgen einer nicht erreichten Umsetzung dieser ‚focus product standards‘
Wertebereich der Information	Nominal; natürlich-sprachliche Beschreibung
Bezug zur Projektzieldimension	Qualität
Normierung auf Projektebene	Die Informationen beziehen sich bereits auf die Projekt-Ebene; Je mehr Qualitätsanforderungen in einem Projekt umzusetzen sind und je höher dabei die zu erreichenden Qualitätsstufen sind, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Qualität.

Abb. 66: Umzusetzende Qualitätsanforderungen als Kriterium für die Präferenzbestimmung der Projektzieldimension Qualität

Eine weitere Möglichkeit der Dokumentation von Detailinformationen zu den Qualitätsteilzielen ist neben der Formulierung einer Konformitätserklärung auf Projektebene zum unternehmenseigenen Qualitätsmodell, entsprechende Informationen direkt in die Anforderungsdokumente mit aufzunehmen. Eine weit verbreitete Strukturierung von Anforderungsdokumenten basiert auf zwei Standards der Organisation IEEE.⁶⁴⁰ Beide definieren Referenzstrukturen für Anforderungsdokumente, die allgemeingültig für alle Abstraktionsebenen sind und sich somit sowohl auf Markt- bzw. Kundenanforderungen (Features), als auch auf bereits sehr genau spezifizierte Anforderungen anwenden lassen.⁶⁴¹ Abb. 67 stellt den für die Detaildokumentation von Qualitätsteilzielen relevanten Ausschnitt aus dem Aufbau von Anforderungen nach dem IEEE-Standard 830 dar (vgl. Abb. 67).

⁶⁴⁰ Es handelt sich hierbei um die beiden Standards ‚IEEE Std. 1233-1998 Systems Requirements Specification‘ sowie ‚IEEE Std. 830-1998 Software Requirements Specification‘; vgl. IEEE (Hrsg., 1998a) und vgl. IEEE (Hrsg., 1998b)

⁶⁴¹ Vgl. Pohl (2007), S. 252f.

Rubrik	Beschreibung
Einschränkungen	Bezug zu Qualitätsanforderungen oder sonstigen Einschränkungen aus Normen, Gesetzen, Rahmenverträgen etc. Für häufige Einschränkungen sollten separate Felder benutzt werden. Einschränkungen können sich ebenso aus der Qualitätszielbestimmung ergeben bzw. sich auf diese beziehen.
Erläuterung	Präzise, verständliche und mit Projektbezug gehaltene Erläuterungen. Die Erläuterungen können auch auf weitere externe Dokumente referenzieren – wie bspw. auf weitere Detailinformationen hinsichtlich der Qualitätsteilziele.
...	...

Abb. 67: Ausschnitt aus dem Aufbau von Anforderungen nach dem IEEE-Standard 830⁶⁴²

3.3.2.2 Abgrenzung der Projektzieldimensionen Qualität und Leistungsumfang

In der Literatur zur Softwareentwicklung wird teilweise die Ansicht vertreten, dass bei der Betrachtung von Zielkonflikten eine explizite Qualitätszieldimension keine Berechtigung hat. Der Grund für diese Ansicht liegt in der nicht eindeutigen Abgrenzbarkeit und Relation der Qualitätszieldimension zu den anderen Projektzieldimensionen. So könnten Qualitätsaspekte sowohl positive als auch negative Auswirkungen auf die Erreichung von Termin- und Kostenzielen haben. Die Qualitätszieldimension solle daher laut McConnell zum Leistungsumfang gezählt werden und als eine solche integrierte Produktzieldimension in die Betrachtung von Zielkonflikten mit den Termin- und Kostenzielen eingehen.⁶⁴³

Eine solche Meinung spiegeln die durchgeführten Experteninterviews zwar nicht gänzlich wieder, doch das grundsätzliche Problem der Abgrenzung von Qualitätsanforderungen und funktionalen Anforderungen, bzw. allgemeiner, das Problem der Abgrenzung der Projektzieldimensionen Leistungsumfang und Qualität voneinander, wird auch hier gesehen.⁶⁴⁴ Denn insbesondere in den frühen Projektphasen existieren oftmals nur wenig detailliert beschriebene Marktanforderungen, die erst im weiteren Projektverlauf in konkretere funktionale Anforderungen und Qualitätsanforderungen überführt werden.⁶⁴⁵ Dabei können zu den Marktanforderungen auch Anforderungen gehören, die nicht vom Produktmanagement herangetragen wurden, sondern aus dem Umfeld des Qualitätsmanagements stammen. Anforderungen, deren Umsetzung zur Steigerung der Wartbarkeit oder zu architektonischen Verbesserungen führen soll, sind

⁶⁴² Vgl. IEEE (Hrsg., 1998a); vgl. Ebert (2008), S. 153

⁶⁴³ Vgl. McConnell (1996), S. 126

⁶⁴⁴ Experte ‚M.G.‘ zur Frage A1, Experte ‚B.E.‘ zur Frage A1

⁶⁴⁵ Experte ‚M.G.‘ zur Frage A4

Beispiele hierfür.⁶⁴⁶ Für eine Abgrenzung der Projektzieldimensionen Qualität und Leistungsumfang sind daher so früh wie möglich genau beschriebene Qualitätsanforderungen eine wesentliche Voraussetzung.⁶⁴⁷

Einer im Projektverlauf relativ frühen Beschreibung von Qualitätsanforderungen steht in der Praxis wiederum ein weiteres Problem entgegen, nämlich die Unterspezifizierung von funktionalen Anforderungen. Pohl argumentiert in diesem Zusammenhang neben funktionalen Anforderungen und Qualitätsanforderungen mit einer dritten Klasse von Anforderungen, nämlich den nicht-funktionalen Anforderungen. Die Klasse der nicht-funktionalen Anforderungen gliedert sich dabei in zwei Unterklassen auf:⁶⁴⁸

- Unterspezifizierte funktionale Anforderungen, die bei einer entsprechenden Detaillierung in funktionale Anforderungen sowie ggf. in Qualitätsanforderungen überführt werden können;
- Qualitätsanforderungen, die sich, wie bereits beschrieben, auf qualitative Eigenschaften des Gesamtsystems, einzelner Funktionen oder Funktionsgruppen beziehen können.

Bei zahlreichen in der Literatur beschriebenen und in realen Spezifikationen enthaltenen nicht-funktionalen Anforderungen handelt es um unterspezifizierte funktionale Anforderungen. Sie lassen in Bezug auf die Eigenschaften des zukünftigen Software-systems ein weites Spektrum von Interpretationen zu. Das Problem der Unterspezifizierung beschränkt sich in der Praxis aber nicht nur auf die Beschreibung von funktionalen Anforderungen an sich, sondern besteht auch bei sonstigen und mit allen Anforderungsklassen zusammenhängenden Informationen wie bspw. Priorisierungsinformationen. Daher lässt sich auch bei der Identifikation von Kriterien für die Präferenzbestimmung hinsichtlich der beiden Projektzieldimensionen Leistungsumfang und Qualität der Bezug von einigen Kriterien zu einer der beiden Dimensionen wegen Unterspezifizierung nicht einwandfrei festlegen. Ein Beispiel dazu stellt in der Fallstudienunternehmung die Bestimmung und Dokumentation des Erfüllungsrisikos im Projekt-Qualitätsplan (vgl. Abb. 68).

⁶⁴⁶ Experte ‚A.D.‘ zur Frage A4, Experte ‚J.K.‘ zur Frage A4

⁶⁴⁷ Experte ‚F.E.‘ zur Frage A5

⁶⁴⁸ Vgl. auch im Folgenden Pohl (2007), S. 16

Mustervorlage	Projekt-Qualitätsplan
Informationsfeld	Erfüllungsrisiko im Sinne des entstandenen Schadens bei einer nicht in der geplanten Qualität erfolgten Umsetzung eines Entwicklungsgegenstandes
Wertebereich der Information	Ordinal: hoch (bei zentralen Entwicklungsgegenständen aus Marktsicht) / mittel / niedrig (bei Zusatzfunktionalität mit Relevanz für eine kleine Kundengruppe)
Bezug zur Projektzieldimension	Leistungsumfang oder Qualität
Normierung auf Projektebene	Je höher der Anteil von hoch priorisierten Entwicklungsgegenständen hinsichtlich des Erfüllungsrisikos am gesamten Projektumfang, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Leistungsumfang / Qualität.

Abb. 68: Erfüllungsrisiko als Kriterium für die Präferenzbestimmung der Projektzieldimensionen Qualität und Leistungsumfang

Besonders problematisch, nicht nur hinsichtlich Unterspezifizierung, sondern allgemein hinsichtlich der Abgrenzbarkeit der beiden Projektzieldimensionen Qualität und Leistungsumfang stellt sich das Qualitätsmerkmal Funktionalität dar. Es ist eines von sechs Softwarequalitätsmerkmalen auf der obersten Ebene des Qualitätsmodells der ISO/IEC 9126 und einer der Hauptgründe für die Kritik an diesem Standard als ein unklares und doppeldeutiges Konstrukt.⁶⁴⁹ Die Funktionalität ist bei der ISO/IEC 9126 aus 5 Teilmerkmalen zusammengesetzt, die sich bei näherem Hinsehen sowohl dem Leistungsumfang, als auch der Qualitätszieldimension zuordnen lassen (vgl. Abb. 69).

Teilmerkmal der Funktionalität nach ISO/IEC 9126 ⁶⁵⁰	Projektzieldimension
Angemessenheit: Eignung von Funktionen für spezifizierte Aufgaben, z.B. aufgabenorientierte Zusammensetzung von Funktionen aus Teilfunktionen.	Leistungsumfang
Richtigkeit: Liefern der richtigen oder vereinbarten Ergebnisse oder Wirkungen, z.B. die benötigte Genauigkeit von berechneten Werten.	Qualität
Interoperabilität: Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken.	Qualität
Sicherheit: Fähigkeit, unberechtigten Zugriff, sowohl versehentlich als auch vorsätzlich, auf Programme und Daten zu verhindern.	Qualität
Ordnungsmäßigkeit: Merkmale von Software die bewirken, dass anwendungsspezifische Normen, gesetzliche Bestimmungen und ähnliche Vorschriften erfüllt werden.	Leistungsumfang / Qualität

Abb. 69: Zuordnung von Teilmerkmalen der Funktionalität nach ISO/IEC 9126 zu den Projektzieldimensionen Qualität und Leistungsumfang

⁶⁴⁹ Vgl. Al-Kilidar u.a. (2005), S. 131

⁶⁵⁰ Vgl. ISO/IEC (Hrsg., 2001)

Die Abgrenzungsproblematik bei den Projektzieldimensionen Qualität und Leistungsumfang ist bisher mit dem Fokus auf die Unterspezifizierung von funktionalen Anforderungen sowie auf das unklare und doppeldeutige Qualitätsteilmerkmal Funktionalität vorrangig aus der Perspektive der Praxis betrachtet worden. Daher soll der Betrachtungsfokus im Folgenden auf die konzeptionelle Ebene verlagert werden. Allerdings handelt es sich hierbei um ein softwaretechnisches Problem, sodass es bei der Ausrichtung der vorliegenden Arbeit, also der Betrachtung der Standardsoftwareentwicklung aus der organisatorischen Perspektive heraus, nicht abschließend geklärt werden kann. Zumindest aus der organisatorischen Sicht scheint es aber, ausgehend von den folgenden Unterschieden zwischen funktionalen Anforderungen und Qualitätsanforderungen, eine klare konzeptionelle Lösung zu geben.

- Funktionale Anforderungen beziehen sich auf eine jeweils spezifische Funktionalität, während Qualitätsanforderungen üblicherweise mehrere Funktionen oder Features, bis hin zum gesamten Softwaresystem, adressieren;
- Die Umsetzung von Qualitätsanforderungen führt zu bestimmten Eigenschaften des Softwaresystems, das jedoch ohne das Vorhandensein von umgesetzten funktionalen Anforderungen zwecklos ist;
- Funktionale Anforderungen werden durch das Softwaresystem entweder erfüllt oder nicht erfüllt, während Qualitätsanforderungen mehr oder weniger erfüllt werden können, v.a. wenn Qualitätsstufen durch die Vorgabe von Relativmaßen definiert werden.⁶⁵¹

Weiterhin können zwei Unterklassen von Qualitätsanforderungen unterschieden werden. Qualitätsanforderungen können sich nämlich auf Qualitätseigenschaften beziehen, welche mittels Ausführung der Software beobachtbar sind, oder welche architekturbezogen und daher nicht mittels Ausführung beobachtbar sind.⁶⁵²

- Mittels Ausführung der Software beobachtbare Qualitätseigenschaften sind u.a. Zuverlässigkeit, Zeit- und Verbrauchsverhalten, Sicherheit und Benutzbarkeit;
- Zu den nicht mittels Ausführung der Software beobachtbaren Qualitätseigenschaften gehören u.a. Änderbarkeit, Portabilität, Skalierbarkeit und Testbarkeit. Diese Qualitätseigenschaften stellen die primären Treiber für den Softwarearchitekturentwurf dar, da sie die Struktur einer Softwarearchitektur stark beeinflussen.

Benutzbarkeit und Änderbarkeit sollen die Einordnung in die beiden Unterklassen von Qualitätsanforderungen beispielhaft verdeutlichen. So hat die Gestaltung einer graphi-

⁶⁵¹ Vgl. Berander, Andrews (2005), S. 84

⁶⁵² Vgl. Bass u.a. (1998), S. 76ff.

schen Benutzeroberfläche als ein wesentlicher Aspekt der Benutzbarkeit primär mit der richtigen Auswahl und Anordnung der Steuerelemente (Dialogboxen, Auswahllisten etc.) zu tun. Diese Aspekte sind unabhängig von der Softwarearchitektur und gleichzeitig unmittelbar mittels Ausführung beobachtbar. Änderbarkeit ist dagegen ein primärer Treiber für den Softwarearchitekturentwurf. Denn damit die Modifikation von Funktionalität ohne Gefährdung des Gesamtsystems möglich ist, muss die Softwarearchitektur modular konzipiert sein und das Geheimnisprinzip umsetzen.⁶⁵³

Wie bereits im Zusammenhang mit der ‚Navigation im Teufelsquadrat‘ erläutert, werden die Projektbeteiligten immer eine ihren Interessenstandpunkten entsprechende Entscheidung anstreben. Daher muss bei der Abgrenzung zwischen Qualitätsanforderungen und funktionalen Anforderungen auch mitberücksichtigt werden, ob die jeweilige Anforderung aus dem Umfeld des Produkt- oder des Qualitätsmanagements stammt bzw. deren Interessenstandpunkte tangiert.⁶⁵⁴ Unter dieser Prämisse und mit Rückgriff auf die Unterschiede zwischen funktionalen Anforderungen und Qualitätsanforderungen sowie zwischen den Qualitätsanforderungen untereinander lassen sich aus der organisatorischen Perspektive auf die Standardsoftwareentwicklung die Projektzieldimensionen Qualität und Leistungsumfang konzeptionell klar trennen:

- Funktionale Anforderungen sowie Qualitätsanforderungen, welche sich nur lokal auf bestimmte Features oder Komponenten beziehen und mittels Ausführung beobachtbar und damit nicht architekturbezogen sind, werden der Projektzieldimension Leistungsumfang zugerechnet;
- Qualitätsanforderungen, welche sich nicht auf bestimmte Features oder Komponenten beziehen lassen⁶⁵⁵ oder mittels Ausführung nicht beobachtbar, sondern architekturbezogen sind, werden der Projektzieldimension Qualität zugerechnet.

3.3.3 Projektzieldimension Zeit

3.3.3.1 Ableitung der Präferenz aus Projektabhängigkeiten

Für die Terminplanung von Projekten und Projektprogrammen sind die logischen Abhängigkeiten zwischen den Arbeitspaketen in einer Vorgangsliste festzulegen. Daraus ergibt sich die Bearbeitungsreihenfolge, die in einem Netzplandiagramm dargestellt werden kann. Es kann die technischen, wirtschaftlichen und logischen Abhängigkeiten zwischen Arbeitspaketen verdeutlichen. Vor allem kann das Netzplandiagramm den

⁶⁵³ Vgl. Bass u.a. (1998), S. 78

⁶⁵⁴ Experte ‚C.S.‘ zur Frage A4, Experte ‚T.S.‘ zur Frage A4, Experte ‚T.D.‘ zur Frage A5, Experte ‚J.W.‘ zur Frage A4

⁶⁵⁵ Dieser Punkt ist in Zusammenhang mit der Ausrichtung des Qualitätsmanagements als Querschnittsfunktion zu sehen.

sog. kritischen Pfad in Abgrenzung zu Vorgängen mit Pufferzeiten anzeigen. Auf dem kritischen Pfad liegen Vorgänge, die keinen Puffer haben und sich damit nicht verzögern dürfen, da sich sonst das geplante Projektende nicht einhalten ließe. Ein wesentliches Ziel der Netzplantechnik ist es, solche zeitkritischen Vorgänge zu identifizieren.⁶⁵⁶ Um den kritischen Pfad zu bestimmen, müssen folgende Termine bekannt sein:

- 1) Frühester Termin, an dem ein Vorgang beginnen kann;
- 2) Frühester Termin, an dem ein Vorgang enden kann;
- 3) Spätester Termin, an dem ein Vorgang beginnen kann, ohne das Projektende zu verzögern;
- 4) Spätester Termin, an dem ein Vorgang enden kann, ohne das Projektende zu verzögern.

Der Puffer als die Zeitspanne, um die sich ein Vorgang gegenüber seinem frühesten Termin verspäten darf, ohne das geplante Projektende zu beeinflussen, kann entweder als Differenz zwischen 1) und 3) oder zwischen 2) und 4) berechnet werden. Bei Vorgängen, die auf dem kritischen Pfad liegen, sind die früheste und die späteste Lage identisch.⁶⁵⁷

Auch wenn die Netzplantechnik eine mathematische Optimierung zulässt, ist auf Grund der Unsicherheiten in jedem Projekt ein solcher Zeitplan nie eine exakte Vorausschau, sondern ein Hilfsmittel, um kritische Entwicklungen zu identifizieren.⁶⁵⁸ In diesem Sinn ist die Netzplantechnik keineswegs auf die Ebene von wohl definierten Arbeitspaketen beschränkt, sondern ebenso für ganze Projekte als Vorgänge in einem Projektprogramm einsetzbar. Auf dieser Ebene bestehen zahlreiche logische Abhängigkeiten (Interdependenzen), die für die Terminplanung und für die Kritikalität des Projektfertigstellungstermins entscheidend sind. Nach ihrer Intensität lassen sich dabei drei Arten von Interdependenzen unterscheiden (vgl. auch anschließend in Abb. 70):⁶⁵⁹

- Die gemeinsame Interdependenz⁶⁶⁰ liegt vor, wenn organisatorische Teilbereiche zwar unabhängig voneinander arbeiten, jedoch zu ein und demselben Unternehmen gehören und daher auf eine begrenzte Menge von gemeinsamen Ressourcen (bspw. spezialisierte Arbeitskräfte) zugreifen. Sie befinden sich in einem sog. Restriktionsverbund und müssen sich daher abstimmen, welcher Teilbereich wann und in welchem Umfang eine bestimmte Ressource in Anspruch nehmen kann;

⁶⁵⁶ Vgl. Fiedler (2010), S. 125f.

⁶⁵⁷ Vgl. Fiedler (2010), S. 128

⁶⁵⁸ Vgl. Fiedler (2010), S. 132

⁶⁵⁹ Vgl. Thompson (1967), S. 54f., zitiert nach Beck (1996), S. 207f.; vgl. Schulte-Zurhausen (2002), S. 203f.

⁶⁶⁰ Auch: Ressourceninterdependenz, Restriktionsverbund, pooled interdependence

- Bei der einseitigen Interdependenz⁶⁶¹ bilden die Leistungen eines vorgelagerten organisatorischen Teilbereiches die Voraussetzung für die Leistungen der nachgelagerten Organisationseinheit. Die Organisationseinheiten sind also im Prozess der betrieblichen Leistungserstellung hintereinander geschaltet, sodass die nachgelagerten Bereiche jeweils von der Leistungsqualität der vorgelagerten Arbeitsprozesse innerhalb des Geschäftsprozesses abhängig sind;
- Die intensivste Abhängigkeitsform sind wechselseitige Interdependenzen⁶⁶². Sie liegen vor, wenn sich Arbeitsprozesse gegenseitig bedingen, sodass zwischen zwei oder mehreren organisatorischen Einheiten ein gegenseitiger Austausch von Leistungen vorgenommen werden muss. D.h. jede Organisationseinheit arbeitet hier unmittelbar mit allen anderen zusammen an einer Aufgabe. Die Organisationseinheiten sind dabei nicht im eigentlichen Sinn vor- und nachgelagert, sondern bedingen sich gegenseitig: Die Leistungen einer Einheit sind Vorleistungen für die anderen, erfordern zugleich aber auch Leistungen der anderen Organisationseinheiten.

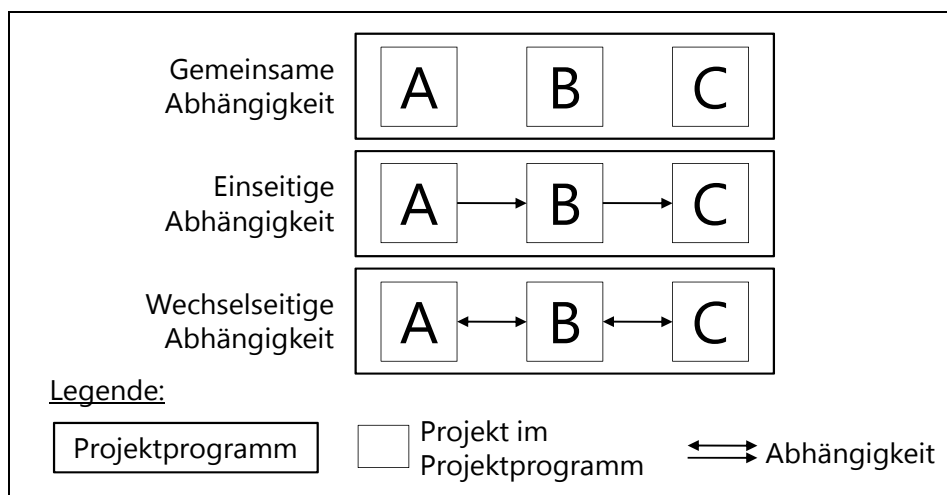


Abb. 70: Interdependenzenarten zwischen organisatorischen Einheiten⁶⁶³

Die Komplexität der Abhängigkeiten und der damit verbundenen Austauschbeziehungen steigt in der oben beschriebenen Reihenfolge von den gemeinsamen über die einseitigen bis hin zu den wechselseitigen Interdependenzen an. Damit einhergehend steigt auch der „Schwierigkeitsgrad, diese Abhängigkeitsbeziehungen zu bewältigen, da sie ein größeres Spektrum an Eventualitäten und Ausfallrisiken beinhalten.“⁶⁶⁴ Desweiteren führen sie dazu, dass die entsprechenden Vorgänge bzw. Projekte, sofern sie die Betrachtungsebene bilden, auf dem kritischen Pfad liegen und somit hinsichtlich der Zeitzieldimension hoch kritisch sind. Insofern hat auch die Interdependenzart

⁶⁶¹ Auch: sequentielle (Prozess-)Interdependenz

⁶⁶² Auch: beiderseitige oder reziproke (Prozess-)Interdependenz

⁶⁶³ In Anlehnung an Beck (1996), S. 208

⁶⁶⁴ Billing (2003), S. 49

einen direkten Einfluss auf die Lösung des Koordinationsproblems. Die Berücksichtigung von Abhängigkeitsbeziehungen ist insgesamt von großer Bedeutung bei der Fallstudienunternehmung. Daher erfolgt ihre Dokumentation bereits einleitend im Projektplan auf Projektebene (vgl. Abb. 71), sowie anschließend detaillierter im Projekt-Qualitätsplan auf Ebene der einzelnen Entwicklungsgegenstände (vgl. Abb. 72).

Mustervorlage	Projektplan
Informationsfeld	Beschreibung eventueller Abhängigkeitsbeziehungen zu anderen Projekten
Wertebereich der Information	Nominal; natürlich-sprachliche Beschreibung der Abhängigkeitsbeziehungen
Bezug zur Projektzieldimension	Zeit
Normierung auf Projektebene	Je intensiver die Interdependenzen zwischen dem betrachteten Projekt und den anderen Projekten im Projektprogramm, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Zeit.

Abb. 71: Abhängigkeitsbeziehungen auf Projektebene als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit

Mustervorlage	Projekt-Qualitätsplan
Informationsfeld	'Integrationsrisiko' - definiert als das Ausmaß an Abhängigkeiten und Austauschbeziehungen zwischen einem Entwicklungsgegenstand im betrachteten Projekt und anderen Projekten
Wertebereich der Information	Ordinal: ‚hoch‘ bei starken Interdependenzen zwischen dem Entwicklungsgegenstand und Entwicklungsgegenständen in anderen Projekten / ‚mittel‘ bei starken Interdependenzen, die sich jedoch auf Entwicklungsgegenstände im betrachteten Projekt beschränken / ‚niedrig‘ bei einem Entwicklungsgegenstand ohne nennenswerte Interdependenzen
Bezug zur Projektzieldimension	Zeit
Normierung auf Projektebene	Je höher der Anteil von hoch priorisierten Entwicklungsgegenständen hinsichtlich des Integrationsrisikos am gesamten Projektumfang, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Zeit.

Abb. 72: Abhängigkeitsbeziehungen auf Ebene der Entwicklungsgegenstände als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit

Die bisher diskutierten Abhängigkeitsbeziehungen können noch weiter heruntergebrochen werden – bis hin zu Abhängigkeiten zwischen den einzelnen umzusetzenden Anforderungen in den Entwicklungsprojekten innerhalb eines Projektprogramms. Nicht zuletzt aus diesem Grund herrscht zwischen dem Requirements Engineering und den Entwicklungsprojekten über den gesamten Projektlebenszyklus hinweg eine mehr oder weniger starke Wechselwirkung.⁶⁶⁵ Sehr deutlich zeigt sich dies bei der Anforderungsnachvollziehbarkeit. „Die Nachvollziehbarkeit einer Anforderung ist die Fähigkeit, den Lebenszyklus der Anforderung vom Ursprung der Anforderung über die verschiedenen Verfeinerungs- und Spezifikationsschritte bis hin zur Berücksichtigung der Anforderung in nachgelagerten Entwicklungsartefakten verfolgen zu können.“⁶⁶⁶ Anforderungsnachvollziehbarkeit befasst sich somit mit der Transformation und mit den Beziehungen zwischen verschiedenen Informationsebenen.⁶⁶⁷ Die Gründe für die Sicherstellung der Anforderungsnachvollziehbarkeit sind sehr vielfältig, liegen dabei aber zum großen Teil im Umfeld des Projektmanagements.⁶⁶⁸

- Einflussanalyse beim Änderungsmanagement: Anforderungsnachvollziehbarkeit ermöglicht es im Fall einer Anforderungsänderung festzustellen, wie sich diese Änderung auf welche anderen Artefakte auswirkt;
- Abdeckungsanalyse bei der Projektverfolgbarkeit: Nachvollziehbarkeitsinformationen ermöglichen die Verfolgung des Projektfortschritts und die Bestimmung des aktuellen Projektstatus. Bspw. muss hierzu durch eine Analyse der Nachvollziehbarkeitsinformationen ermittelt werden, wie viele und welche Anforderungen bereits implementiert oder getestet sind;
- Risikomanagement im Projekt: Anforderungsnachvollziehbarkeit unterstützt das Risikomanagement, indem die potenziell von einem Risiko betroffenen Anforderungen und damit zusammenhängenden Artefakte aufgrund der Nachvollziehbarkeit identifiziert werden können;
- Projektkontrolle und Zurechenbarkeit: Nachvollziehbarkeitsinformationen können dazu verwendet werden, um im Nachhinein einzelnen Anforderungen den Entwicklungsaufwand zuzuordnen und so eine Kosten-Nutzen Analyse auf der Ebene von Anforderungen zu ermöglichen.

⁶⁶⁵ Vgl. Pohl (2007), S. 34; s. dort auch zum phasenübergreifenden Requirements Engineering als eine Querschnittstätigkeit.

⁶⁶⁶ Pohl (2007), S. 503

⁶⁶⁷ Vgl. Hull u.a. (2005), S. 9

⁶⁶⁸ Vgl. Pohl (2007), S. 504f.; vgl. Ebert (2008), S. 266; vgl. Hull u.a. (2005), S. 11; weitere, aus Sicht des Projektmanagements weniger relevante Gründe sind bspw. die Nutzenanalyse zur Nachweisbarkeit und Akzeptanz der umgesetzten Anforderungen und die Vermeidung des sog. ‚gold plating‘.

Die Verwendungsmöglichkeiten von Informationen zur Anforderungsnachvollziehbarkeit im Projektverlauf verdeutlichen, dass alle Projektabhängigkeiten schlussendlich über Abhängigkeiten zwischen den einzelnen Anforderungen untereinander zustande kommen. Auf dieser Ebene kann die Anforderungsnachvollziehbarkeit sehr detailliert umgesetzt werden, mithin unter Berücksichtigung der verschiedenen Typen von Anforderungsbeziehungen. In Zusammenhang mit der Präferenzbestimmung der Projektzieldimension Zeit ist der Beziehungstyp ‚Vorbedingung/Benötigt‘ zentral.

Die Beziehung ‚Vorbedingung/Benötigt‘ von einer Anforderung als Ausgangsartefakt zu einer anderen Anforderung drückt aus, dass das Ausgangsartefakt eine Bedingung definiert, die erfüllt sein muss, bevor die andere Anforderung realisiert werden kann. Die Erfüllung einer Anforderung hängt also von der Erfüllung einer anderen Anforderung ab. Selbst an dieser Stelle gibt es noch mehrere Interpretationsmöglichkeiten, wobei eine davon die Auffassung der Vorbedingung als eine temporäre Beziehung ist. Dieser Beziehungstyp besagt, dass eine Anforderung zeitlich vor der Erstellung einer anderen Anforderung erbracht werden muss.⁶⁶⁹ In diesem Sinn ist auch die Dokumentation und Verwendung des Beziehungstyps ‚Vorbedingung/Benötigt‘ in der Fallstudienunternehmung zu verstehen (vgl. Abb. 73).

Mustervorlage	Datenbankgestützte Mustervorlage, die als Basis für die Produktdefinition, Produktplanung sowie das ‚commitment management‘ hinsichtlich Ressourcen und Fragen der Softwarearchitektur dient. Die Informationen werden zum Teil für die Erstellung und ständige Aktualisierung anderer Dokumente übernommen, so bspw. für das Lastenheft.
Informationsfeld	Informationen zur Anforderungsnachvollziehbarkeit
Wertebereich der Information	Nominal; mehrere definierte Beziehungstypen, u.a. ‚Vorbedingung/Benötigt‘
Bezug zur Projektzieldimension	Zeit
Normierung auf Projektebene	Je mehr Anforderungsabhängigkeiten vom Typ ‚Vorbedingung/Benötigt‘ als eine temporäre Beziehung auf das Projekt zurückgehen, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Zeit.

Abb. 73: Anforderungsabhängigkeiten vom Typ ‚Vorbedingung/Benötigt‘ als Kriterium für die Präferenzbestimmung der Projektzieldimension Zeit

⁶⁶⁹ Vgl. Pohl (2007), S. 509; vgl. Dahlstedt, Persson (2005), S. 103f.; vgl. Carlshamre u.a. (2001), S. 86

3.3.3.2 Anforderungsabhängigkeiten als Grundlage für die Bestimmung von Projektzielpräferenzen

Anforderungsabhängigkeiten sind nicht per se problematisch, sondern weil sie eine Reihe von Aktivitäten und Entscheidungen im Verlauf des Softwareentwicklungsprozesses beeinflussen – wie bereits die genannten Gründe für die Sicherstellung der Anforderungsnachvollziehbarkeit verdeutlicht haben. Im Umkehrschluss bedeutet das aber auch, dass zahlreiche Entscheidungen mittelbar über die Anforderungsabhängigkeiten verschiedene Anforderungen in einer nicht antizipierten oder gewünschten Weise betreffen können. Bspw. ist es bei der Planung der Inkremente entgegen der Zielsetzung nicht immer möglich, alle hoch priorisierten Anforderungen zur Umsetzung in das nächste Inkrement aufzunehmen. Der Grund hierfür sind u.a. Anforderungsabhängigkeiten: Die Implementierung einer wichtigen und hoch priorisierten Anforderung erfordert oftmals die vorherige Implementierung einer für sich genommen nicht hoch priorisierten Anforderung.⁶⁷⁰

Neben der Relevanz von Anforderungsabhängigkeiten bei wichtigen Entscheidungen im Verlauf der Softwareentwicklung an sich kommt erschwerend hinzu, dass sehr viele Abhängigkeitsbeziehungen zwischen den Anforderungen bestehen. Carlshamre u.a. kommen in ihrer empirischen Untersuchung zu dem Ergebnis, dass nur in etwa 20% aller Anforderungen in einem Softwareentwicklungsvorhaben singulär sind, d.h. nicht abhängig von anderen Anforderungen. Gleichzeitig stellen 20% aller Anforderungen den Grund für 75% aller Abhängigkeitsbeziehungen dar.⁶⁷¹ Der in Zusammenhang mit der Zeitzielpräferenz erläuterte Beziehungstyp ‚Vorbedingung/Benötigt‘ ist dabei nur ein möglicher Beziehungstyp. Es existieren weitere Beziehungstypen, auf die im Folgenden noch näher eingegangen wird. Notwendig macht es die bei den Experteninterviews gewonnene Erkenntnis, dass Anforderungsabhängigkeiten über die Projektzieldimension Zeit hinaus relevant sein können. So wird die Berücksichtigung von Anforderungsabhängigkeiten teilweise sowohl zur Bestimmung der Präferenz für die Projektzieldimension Leistungsumfang⁶⁷² als auch Qualität⁶⁷³ als notwendig erachtet.

Der Stand der Forschung zur Anforderungsnachvollziehbarkeit und insbesondere zu Anforderungsabhängigkeiten ist unbefriedigend und es existiert hier umfassender Forschungsbedarf.⁶⁷⁴ Auch das Modell zur Klassifizierung der fundamentalen Abhängigkeitstypen von Dahlstedt und Persson ist nicht als ein ausgereiftes Forschungsergebnis

⁶⁷⁰ Vgl. Dahlstedt, Persson (2005), S. 95f.

⁶⁷¹ Vgl. Carlshamre u.a. (2001)

⁶⁷² Experte ‚T.S.‘ zur Frage A4, Experte ‚J.C.‘ zur Frage A1

⁶⁷³ Experte ‚T.M.‘ zur Frage A5, Experte ‚A.D.‘ zur Frage A5

⁶⁷⁴ Vgl. Dahlstedt, Persson (2005), S. 96

zu verstehen. „The question marks in the model emphasize that the model needs further research, where more fundamental categories and interdependency types may be found“⁶⁷⁵ (vgl. Abb. 74).

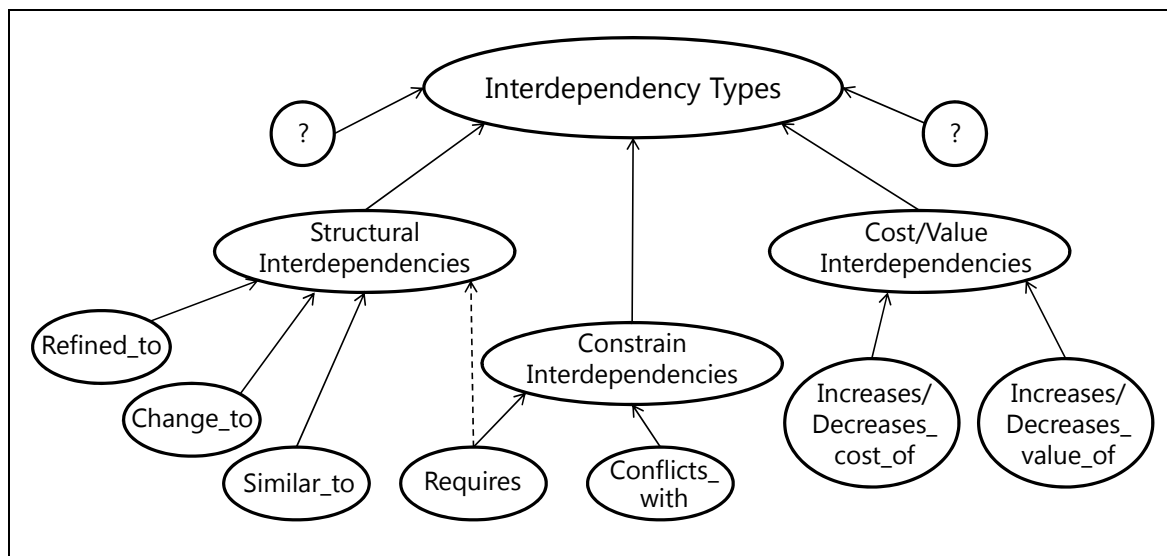


Abb. 74: Klassifizierung von fundamentalen Abhängigkeitstypen⁶⁷⁶

Für die Bestimmung der übergeordneten Projektzielpräferenzen haben sich bei den Experteninterviews die beiden strukturellen Abhängigkeitstypen (Structural Interdependencies) ‚Refined to‘ und ‚Requires‘, letzteres gleichzeitig auch ein Bedingungstyp (Constrain Interdependencies), als relevant herausgestellt. Die weiteren Abhängigkeitstypen sind dagegen für andere Probleme und Zwecke von Bedeutung und werden daher zuvor nur kurz umrissen:⁶⁷⁷

- ‚Changes to‘ beschreibt die Historie einer Anforderung und ermöglicht es, auf verschiedene Versionen einer Anforderung zu referenzieren;
- ‚Similar to‘ drückt aus, dass sich eine Anforderung mit einer oder mehreren anderen Anforderungen überlappt oder ihnen ähnelt;
- ‚Conflicts with‘ kann auf zwei unerwünschte Anforderungsabhängigkeiten hinweisen: Zum einen kann eine Anforderung im Konflikt zu einer anderen Anforderung stehen, sodass schlussendlich nur eine der beiden Anforderungen implementiert werden kann. Zum anderen kann die Implementierung einer Anforderung negativen Einfluss auf die Implementierung einer anderen Anforderung haben;
- ‚Increases / Decreases cost of‘ drückt aus, dass durch die Implementierung einer Anforderung die Implementierungskosten einer anderen Anforderung zu- bzw. abnehmen;

⁶⁷⁵ Dahlstedt, Persson (2005), S. 101

⁶⁷⁶ Dahlstedt, Persson (2005), S. 101

⁶⁷⁷ Vgl. Dahlstedt, Persson (2005), S. 102-105

- ‚Increases / Decreases value of‘ drückt aus, dass durch die Implementierung einer Anforderung der Kundennutzen, welcher seinerseits durch die Implementierung einer anderen Anforderung entsteht, zu- bzw. abnimmt.

Der Abhängigkeitstyp ‚Refined to‘ beschreibt mehrere Anforderungsbeziehungen, die alle eine Gemeinsamkeit aufweisen. Es handelt sich dabei immer um eine hierarchische Beziehung, bei welcher Anforderungen auf einer hohen Abstraktionsebene durch eine Weiterentwicklung bzw. Verfeinerung in detailliertere Anforderungen auf einer niedrigeren Abstraktionsebene überführt werden. ‚Refined to‘ kann in diesem Sinn bspw. ausdrücken, dass

- eine detaillierte Anforderung aus einer anderen abstrakten Anforderung abgeleitet ist,
- eine Anforderung die weitere Ausarbeitung einer anderen Anforderung darstellt,
- eine Anforderung ihren Ursprung in einer anderen Anforderung hat,
- eine Anforderung in mehrere Anforderungen aufgeteilt worden ist.⁶⁷⁸

Der Beziehungstyp ‚Requires‘ kann wie im voranstehenden Kapitel als eine temporäre Beziehung interpretiert werden, bei der eine Anforderung zeitlich vor einer anderen Anforderung implementiert werden muss und damit eine Determinante der Zeitzielkritikalität darstellen. Es kommen aber auch zwei weitere Definitionen in Frage, die weniger den strukturellen Charakter dieser Abhängigkeitsbeziehung, sondern vielmehr die Bedingtheit in den Vordergrund stellen:⁶⁷⁹

- ‚Requires‘ um abzubilden, dass die Implementierung einer Anforderung für die Implementierung einer anderen Anforderung notwendig ist, d.h. diese erst ermöglicht. Diese Beziehung kann ein- oder beidseitig bestehen;
- ‚Requires‘ um eine hierarchische Beziehung zwischen zwei Anforderungen zu beschreiben, die stärker ausgeprägt ist als die beschriebenen hierarchischen Beziehungen bei dem Abhängigkeitstyp ‚Refined to‘. ‚Requires‘ beschreibt in diesem Sinn, dass eine oder mehrere detaillierte Anforderungen gemeinsam implementiert werden müssen, um eine Anforderung auf höherer Abstraktionsebene umzusetzen.

Sollen Anforderungsabhängigkeiten im Rahmen der Anforderungsnachvollziehbarkeit für die Präferenzbestimmung aller vier Projektzieldimensionen herangezogen werden können, so sind alle Ausprägungen des Beziehungstyps ‚Requires‘, sowie der Beziehungstyp ‚Refined to‘ in einem Entwicklungsvorhaben nachzuverfolgen:

⁶⁷⁸ Vgl. Dahlstedt, Persson (2005), S. 102

⁶⁷⁹ Vgl. Dahlstedt, Persson (2005), S. 102f.

- ‚Requires‘ (strukturell) als temporäre Beziehung zur Bestimmung der Zeitzielpräferenz;⁶⁸⁰
- ‚Requires‘ als Bedingtheit zur Bestimmung der Präferenz für die Projektzieldimension Leistungsumfang;⁶⁸¹
- ‚Refined to‘ zur Weitergabe der Kritikalität einer Anforderung,⁶⁸² insbesondere zur Bestimmung der Qualitätspräferenz.⁶⁸³

Diese Forderung ist in der Praxis der Standardsoftwareentwicklung allerdings kaum zu erfüllen. Denn obwohl sich fundamentale Abhängigkeitstypen konzeptionell klassifizieren lassen, so sind sie in der Praxis voneinander nicht zu separieren.⁶⁸⁴ Zudem kann eine Standardsoftwareentwicklung mehrere zehntausend einzelne Anforderungen an die zu erstellende Software umfassen. Bei solchen Dimensionen können Nachvollziehbarkeits- und Abhängigkeitsinformationen nur in einem sehr begrenzten Umfang dokumentiert und gepflegt werden (vgl. Abb. 75).

Abbe- viation	Level	Order of magnitude	Interdependency management con- jectures with current requirements engineering technology
SSRE	Small-Scale Requirements Engineering	+10 require- ments	Managing a complete set of interde- pendencies requires small effort
MSRE	Medium-Scale Requirements Engineering	+100 require- ments	Managing a complete set of interde- pendencies is feasible, but requires large effort
LSRE	Large-Scale Requirements Engineering	+1000 requi- rements	Managing a complete set of interde- pendencies is practically unfeasible, but feasible among small bundles of re- quirements
VLSRE	Very Large-Scale Requirements Engineering	+10000 requi- rements	Managing a complete set of interde- pendencies among small bundles of re- quirements is unfeasible in practice

Abb. 75: Anforderungsnachvollziehbarkeit bei verschiedenen Dimensionen des Requirements Engineering⁶⁸⁵

⁶⁸⁰ Experte ‚T.S.‘ zur Frage A3, Experte ‚F.E.‘ zur Frage A3, Experte ‚T.Si.‘ zur Frage A3, Experte ‚T.M.‘ zur Frage A3, Experte ‚T.W.‘ zur Frage A3

⁶⁸¹ Experte ‚T.S.‘ zur Frage A4, Experte ‚J.C.‘ zur Frage A1

⁶⁸² Experte ‚T.M.‘ zur Frage A4, Experte ‚T.D.‘ zur Frage A4

⁶⁸³ Experte ‚T.M.‘ zur Frage A5, Experte ‚A.D.‘ zur Frage A5

⁶⁸⁴ Vgl. Dahlstedt, Persson (2005), S. 101

⁶⁸⁵ Regnell u.a. (2008), S. 124

Ähnlich wie bei der Abgrenzungsproblematik zwischen funktionalen Anforderungen und Qualitätsanforderungen in Kap. 3.3.1.2 handelt es sich bei der Anforderungsnachvollziehbarkeit um eine weitgehend softwaretechnische Problemstellung, auf die an dieser Stelle nicht abschließend eingegangen werden kann. Dabei ist insbesondere ab der Dimension ‚LSRE‘ wesentlicher Forschungsbedarf festzustellen, der das gesamte Aufgabenspektrum des Requirements Engineering betrifft. Es lassen sich nämlich nur wenige Forschungsarbeiten vorfinden, die sich mit dem ‚LSRE‘ befassen. Gleichzeitig ist aber gerade dies die Dimension, der sich die Praxis der Standardsoftwareentwicklung in vielen Fällen gegenüberstellen muss. Es ist sogar anzunehmen, dass wegen der ständig ansteigenden Komplexität und Größe der Entwicklungsvorhaben auch das Requirements Engineering komplexer und umfangreicher wird. Damit wird in der Praxis die Bedeutung und Verbreitung des ‚VLSRE‘ zunehmen.⁶⁸⁶

Nichts desto trotz soll im Folgenden ein möglicher Ansatzpunkt aufgezeigt werden, der auch bei umfangreichen Entwicklungsvorhaben die Einbeziehung von Anforderungsabhängigkeiten für die Präferenzbestimmung bei allen Projektzieldimensionen ermöglichen kann: Die direkte Attributierung von Kritikalitätsinformationen und ihre Nachverfolgung über einen eigenen Beziehungstyp auf der Abstraktionsebene von Features oder zusammenhängenden Anforderungsgruppen in einem Entwicklungsvorhaben.

Während der Softwareentwicklung fallen in Bezug auf eine Anforderung zahlreiche Informationen an, die strukturiert dokumentiert werden müssen. Hierzu hat sich die Attributierung von Anforderungen bewährt. Die Menge aller verwendbaren Attribute für eine bestimmte Klasse von Anforderungen (bspw. funktionale Anforderungen) wird dabei in einem Attributierungsschema definiert. Attributierungsschemata werden typischerweise projekt- oder unternehmensspezifisch festgelegt, wobei sich die Schemata für die verschiedenen Anforderungsklassen mehr oder weniger unterscheiden können. Einige Anforderungsattribute wie bspw. die Priorität einer Anforderung wurden bereits thematisiert. Insgesamt ist aber die potentielle Menge an Informationen, die attribuiert werden können, enorm, sodass Pohl die zahlreichen möglichen Anforderungsattribute in sieben Kategorien unterteilt:⁶⁸⁷

- Attribute zur Identifizierung einer Anforderung (bspw. Name und Bezeichner);
- Attribute für die Kontextbeziehungen einer Anforderung (bspw. Verantwortlicher für die Anforderung oder Stakeholder, die Nutzen aus der Anforderung ziehen);

⁶⁸⁶ Vgl. Regnell u.a. (2008), S. 124f.

⁶⁸⁷ Vgl. Pohl (2007), S. 257ff.; vgl. hierzu auch die Ausführungen zum IEEE-Standard 830 in Kap. 3.2.3.1

- Attribute für Dokumentationsaspekte (bspw. Dokumentations- oder Spezifizierungsregeln);
- Attribute für Inhaltsaspekte (bspw. Kurzbeschreibung der Anforderung oder Querbezüge);
- Attribute für Übereinstimmungsaspekte (bspw. Übereinstimmungsstatus oder erkannte Konflikte);
- Attribute für Validierungsaspekte (bspw. Status der Validierung);
- Attribute für Managementaspekte (bspw. Kritikalität oder Priorität).

Nach dieser Kategorisierung ist die Attributierung von Informationen zur Kritikalität den Managementaspekten zuzuordnen. Die Anforderungsattribute dieser Kategorie können aus Übersichtlichkeitsgründen weiter in zwei Unterkategorien aufgeteilt werden. Eine hiervon bilden die sog. Statusattribute wie ‚Risiko‘, ‚Kritikalität‘, ‚Priorität‘, etc.⁶⁸⁸ Sie bestimmen die Bedeutung einer Anforderung für das Projekt und können dabei in mehrfacher Hinsicht dokumentiert sein – so auch direkt hinsichtlich der übergeordneten Projektzielpräferenzen.

3.4 Organisatorische Umsetzung des situativen Koordinationsansatzes

In den Kap. 3.2 und 3.3 wurden die Ansatzpunkte für die situative Ausrichtung von Koordinationsprozessen (Gestaltungsmaßnahmen), sowie die konkreten Kriterien für die Bestimmung der übergeordneten Projektzielpräferenzen (Gestaltungsbedingungen) relativ unabhängig voneinander entwickelt. Es wurde soweit nicht darauf eingegangen, wie die notwendige Stimmigkeit zwischen der Situation (Gestaltungsbedingungen) und der Struktur (Gestaltungsmaßnahmen) hergestellt werden kann. Die erwarteten Wirkungen von organisatorischen Regelungen auf das Verhalten von Organisationsmitgliedern können jedoch nur als Kombination von Struktur- und Situationsausprägungen begriffen werden.⁶⁸⁹ Dabei ist die Gestaltungswirkung nicht nur von den Gestaltungsmaßnahmen und -bedingungen per se abgängig, sondern von der organisatorischen Umsetzung der Herbeiführung der Stimmigkeit zwischen Situation und Struktur.

In Kap. 2.4 wurde begründet, dass der hier entwickelte situative Koordinationsansatz in Form der Koordination durch Programme umgesetzt wird. Die beiden dort aufgeworfenen Fragestellungen zum Differenzierungsgrad des Klassifikationsschemas und zum Detaillierungsgrad der Programmvorgaben dienen im vorliegenden Kap. 3.4 zur Strukturierung der Ausführungen.

⁶⁸⁸ Vgl. Pohl (2007), S. 264f.

⁶⁸⁹ Vgl. Schulte-Zurhausen (2002), S. 27

3.4.1 Differenzierungsgrad des Klassifikationsschemas

3.4.1.1 Anwendung heuristischer Vereinfachungsprinzipien beim Tailoring der Softwareentwicklung

Der Differenzierungsgrad des Klassifikationsschemas bei der Koordination durch Programme wird maßgeblich durch das Ausmaß der Anwendung von heuristischen Vereinfachungsprinzipien bestimmt. Bei einer weitreichenden

- Nichtberücksichtigung bzw. Abstraktion von Gestaltungsbedingungen oder Gestaltungsmaßnahmen,
- Aggregation von Gestaltungsbedingungen oder -maßnahmen und
- vertikalen Dekomposition des Problems

muss von einem niedrigen Differenzierungsgrad gesprochen werden.⁶⁹⁰ Es gibt allerdings kein systematisches Verfahren zur Klärung der Frage, in welchem Umfang die Anwendung von heuristischen Vereinfachungsprinzipien bei einem konkreten Entscheidungsproblem zweckmäßig ist.⁶⁹¹ Vor diesem Hintergrund ist die hier gewählte Vorgehensweise zu sehen, den Komplexitätsgrad der bestehenden Tailoringansätze in der Fallstudienunternehmung als Anhaltspunkt für die Umsetzung des situativen Koordinationsansatzes heranzuziehen. Hierfür konnten drei relevante Tailoringansätze identifiziert werden. Es lässt sich vorab zusammenfassend festhalten, dass bei diesen Ansätzen die drei heuristischen Vereinfachungsprinzipien in einem enormen Ausmaß angewendet werden.

Die allgemeine Produktdefinition ist bei der Fallstudienunternehmung in den Softwareentwicklungsprozess und dabei insb. in die Phase ‚Requirements Analysis und Definition‘ integriert. Hierauf bezieht sich ein weitgehender Tailoringansatz auf der Projektprogramm-Ebene. Das Ziel ist es, die Produkt- und Anforderungsdefinition auf die Unsicherheit hinsichtlich der Marktanforderungen sowie auf die Anforderungskomplexität auszurichten. Dazu sind drei Vorhabensklassen definiert, die sich im Wesentlichen durch die zu erbringenden Artefakte unterscheiden (vgl. Abb. 76).

⁶⁹⁰ Vgl. Kap. 2.4

⁶⁹¹ Vgl. Bitz (1977), S. 429f.

Gestaltungsziel des Tailoringansatzes	Ausrichtung der Produkt- und Anforderungsdefinition auf Projektprogramm-Ebene auf die Unsicherheit hinsichtlich Marktanforderungen und auf die Anforderungskomplexität
Gestaltungsbedingungen des Tailoringansatzes	Klassifikationsschema: <ol style="list-style-type: none"> 1) Produkt- und Anforderungsdefinition für die Weiterentwicklung oder Verbesserung einer bestehenden Standardsoftware oder ihrer einzelnen Teilbestandteile; 2) Produkt- und Anforderungsdefinition für die Neuentwicklung einer Standardsoftware oder ihrer einzelnen Teilbestandteile; 3) Produkt- und Anforderungsdefinition für eine weitreichende Architekturänderung bei einer bestehenden Standardsoftware oder bei ihren einzelnen Teilbestandteilen.
Gestaltungsmaßnahmen des Tailoringansatzes	Vorgaben für die einzelnen Klassen: <ol style="list-style-type: none"> 1) ‚Standardprozess‘, der keine besonderen Vorgaben für die Produkt- und Anforderungsdefinition beinhaltet, da die Umsetzung der Anforderungen in einem bekannten Umfeld und ohne eine signifikante Änderung der bestehenden Standardsoftware und ihrer Softwarearchitektur vollzogen wird; 2) Produkt- und Anforderungsdefinition erfordern mehr Aufwand und mehr Dokumentation als beim ‚Standardprozess‘, da hier eine höhere Anforderungsunsicherheit und -komplexität gegeben ist. Es sind daher zahlreiche Hintergrundinformationen in zusätzlichen Artefakten zu dokumentieren; 3) Der Prozess der Produkt- und Anforderungsdefinition wird hier v.a. durch Fragen der Softwarearchitektur bestimmt, die auch in entsprechenden zusätzlichen Artefakten dokumentiert werden müssen.
Vorgehensweise beim Tailoring	Formaler Tailoringprozess, v.a. durch Hinzufügen, Weglassen oder Ändern der zu erbringenden Artefakte auf der Projektprogramm-Ebene.

Abb. 76: Tailoringansatz für die Produkt- und Anforderungsdefinition

Beginnend mit dem Abschluss der Produkt- und Anforderungsdefinition definiert der unternehmensweit gültige Softwareentwicklungsprozess der Fallstudienunternehmung auf der Projektprogramm-Ebene 11 unterschiedliche Klassen von Entwicklungsvorhaben. Es sind gewissenmaßen vordefinierte, projektprogrammspezifische ‚Pfade‘ durch den Softwareentwicklungsprozess. Das Ziel ist es, diesen ganz allgemein auf die übergeordneten Rahmenbedingungen und auf den Entwicklungsgegenstand auszurichten. Angesichts der Abstraktionsebene ist das Klassifikationsschema auch bei diesem Tailoringansatz sehr wenig differenziert (vgl. Abb. 77).

Gestaltungsziel des Tailoringansatzes	Ausrichtung des Softwareentwicklungsprozesses auf Projektprogramm-Ebene auf die übergeordneten Rahmenbedingungen und auf den Entwicklungsgegenstand
Gestaltungsbedingungen des Tailoringansatzes	11 Vorhabensklassen auf einer sehr hohen Abstraktionsebene, so u.a.: <ul style="list-style-type: none"> • ‚Standardsoftware-Neuentwicklung‘ • ‚Entwicklung von Zusatzfunktionalität‘ • ‚Pilotentwicklung‘ • ‚Branchenspezifische Variantenentwicklung‘ • ‚Partnerentwicklung‘ • ‚Entwicklung vorkonfigurierter Objekte für das Business Intelligence‘
Gestaltungsmaßnahmen des Tailoringansatzes	Jede Vorhabensklasse besteht aus <ul style="list-style-type: none"> • einer spezifischen Kombination von Prozessen (Prozessvarianten), • einem Satz von zu erbringenden Artefakten, sowie • einem Satz von vorgegebenen Meilensteinen. Gestaltungsmaßnahmen sind bspw.: <ul style="list-style-type: none"> • Keine vorangegangene Produktdefinition und kein Architektorentwurf bei ‚Pilotentwicklung‘ und bei ‚Entwicklung vorkonfigurierter Objekte für das Business Intelligence‘; • Kein sog. Integrationstest bei ‚Partnerentwicklung‘; • Vorhabensspezifische Prozessvariante für den Abnahmetest bei ‚Entwicklung von Zusatzfunktionalität‘; • Weglassen einiger für die ‚Standardsoftware-Neuentwicklung‘ vorgegebener Meilensteine bei den anderen Vorhabensklassen.
Vorgehensweise beim Tailoring	<ul style="list-style-type: none"> • Formaler Tailoringprozess durch Hinzufügen, Weglassen oder Ändern von Prozessen, Artefakten oder Meilensteinen auf der Projektprogramm-Ebene.

Abb. 77: Tailoringansatz für verschiedene Klassen von Entwicklungsvorhaben

Ein Szenario bildet einen betrieblichen Geschäftsprozess ab, den die zu erstellende Standardsoftware unterstützen soll. Vor allem in den Feature-Teilprojekten eines Projektprogramms müssen zahlreiche Szenarien implementiert werden. Wie dies im Einzelfall geschehen soll, wird in der Fallstudienunternehmung mithilfe eines Tailoringansatzes festgelegt. Die zu implementierenden Szenarien werden dabei nach ihrer Komplexität und Geschäftsprozesskritikalität klassifiziert (vgl. Abb. 78).

Die weitreichende Anwendung von heuristischen Vereinfachungsprinzipien und damit einhergehend ein niedriger Differenzierungsgrad des Klassifikationsschemas sind, wie die Experteninterviews verdeutlichen, auch bei der organisatorischen Umsetzung des situativen Koordinationsansatzes notwendig. Die Aussagen beziehen sich dabei v.a. auf die Aggregation von Gestaltungsmaßnahmen. So werden Bedenken hinsichtlich der entstehenden Komplexität durch die Prozessvariation geäußert, die jeder situative

Ansatz mit sich bringt. Sie werden teilweise auf den Gesamtansatz,⁶⁹² teilweise jedoch nur auf die einzelnen Strukturdimensionen Artefaktspezifikation,⁶⁹³ Kompetenzverteilung⁶⁹⁴ und ‚Personelle und organisatorische Unabhängigkeit‘⁶⁹⁵ bezogen.

Gestaltungsziel des Tailoringansatzes	Ausrichtung der Phasen ‚Implementierung‘ und ‚Softwaretest‘ auf die Komplexität und Geschäftsprozesskritikalität der zu implementierenden Szenarien bei Feature-Teilprojekten
Gestaltungsbedingungen des Tailoringansatzes	Klassifizierung der zu implementierenden Szenarien nach ‚Komplexität‘ (hoch/niedrig) und ‚Markteinfluss‘ (hoch/niedrig), wodurch sich insgesamt vier Vorhabensklassen ergeben
Gestaltungsmaßnahmen des Tailoringansatzes	Die Gestaltungsmaßnahmen unterscheiden sich v.a. darin, dass bei den Vorhabensklassen mit niedriger Komplexität der zu implementierenden Szenarien der sog. Integrationstest sowie bestimmte Validierungsaktivitäten nicht verbindlich vorgegeben sind
Vorgehensweise beim Tailoring	Formaler Tailoringprozess auf der Projekt-Ebene; Zur Bestimmung der Vorhabensklasse sind jeweils drei definierte Fragen mit vorgegebenen Antwortmöglichkeiten hinsichtlich der beiden Dimensionen ‚Komplexität‘ und ‚Markteinfluss‘ zu beantworten

Abb. 78: Tailoringansatz für die Implementierung von Szenarien

Gleichzeitig wird bei den Aussagen betont, dass die organisatorische Umsetzung des situativen Koordinationsansatzes ohne eine verdichtete Aggregation die (theoretische) Optimallösung darstellt. Der gegenwärtige Stand der Praxis und damit zusammenhängende Überlegungen zur Kosten-Nutzen-Relation sowie zur praktischen Umsetzbarkeit an sich machen jedoch die Abkehr von einem Optimierungsziel notwendig.⁶⁹⁶ Hinsichtlich der Frage, in wie weit die Aggregation der Gestaltungsmaßnahmen reichen soll, lassen sich weder eine herrschende Meinung, noch konkrete Lösungsvorschläge identifizieren. Es wird bei dieser Frage u.a. allgemein von einer notwendigen projektprogramm- oder entwicklungsbereichsweiten, gewissen ‚Homogenität der Projekte‘ gesprochen.⁶⁹⁷ Andere Äußerungen sind in dieser Frage differenziert und beziehen sich dabei auf die Größe des Projektprogramms als Rahmenbedingung.⁶⁹⁸

⁶⁹² Experte ‚M.G.‘ zur Frage A2, Experte ‚L.G.‘ zur Frage D1, Experte ‚J.C.‘ zur Frage D1, Experte ‚T.W.‘ zur Frage D1

⁶⁹³ Experte ‚C.S.‘ zur Frage B1

⁶⁹⁴ Experte ‚K.H.‘ zur Frage B3, Experte ‚T.Si.‘ zur Frage B3, Experte ‚F.E.‘ zur Frage B3, Experte ‚J.C.‘ zur Frage B3

⁶⁹⁵ Experte ‚M.G.‘ zur Frage B2, Experte ‚O.S.‘ zur Frage B2

⁶⁹⁶ Experte ‚M.G.‘ zur Frage B2, Experte ‚K.H.‘ zur Frage B2, Experte ‚T.Si.‘ zur Frage B2, Experte ‚F.E.‘ zur Frage B3, Experte ‚J.C.‘ zur Frage B3

⁶⁹⁷ Experte ‚M.G.‘ zur Frage A2, Experte ‚L.G.‘ zur Frage D1

⁶⁹⁸ Experte ‚T.S.‘ zur Frage C1, Experte ‚B.E.‘ zur Frage C1, Experte ‚J.K.‘ zur Frage C1, Experte ‚L.G.‘ zur Frage D1

3.4.1.2 Bildung des Klassifikationsschemas durch Definition von Projekttypen

Der niedrige Differenzierungsgrad des Klassifikationsschemas wird beim situativen Koordinationsansatz, wie im Folgenden noch näher zu erläutern, über 8 definierte Projekttypen umgesetzt. Einhergehend mit der Tatsache, dass die Koordination durch Programme ein Ansatz der Vorauskoordination ist, erfolgt das Tailoring der Projekte demnach nicht in jedem Einzelfall, sondern über den vorgelagerten Zwischenschritt der Definition von Projekttypen. Ein solches Tailoring über Projekttypen ist in der Praxis der Softwareentwicklung weit verbreitet: Unabhängig von der konkreten Umsetzung des Tailorings klassifizieren fünf von sechs Softwareunternehmen ihre Projekte zu den unterschiedlichsten Zwecken,⁶⁹⁹ welche Abb. 79 zu entnehmen sind.

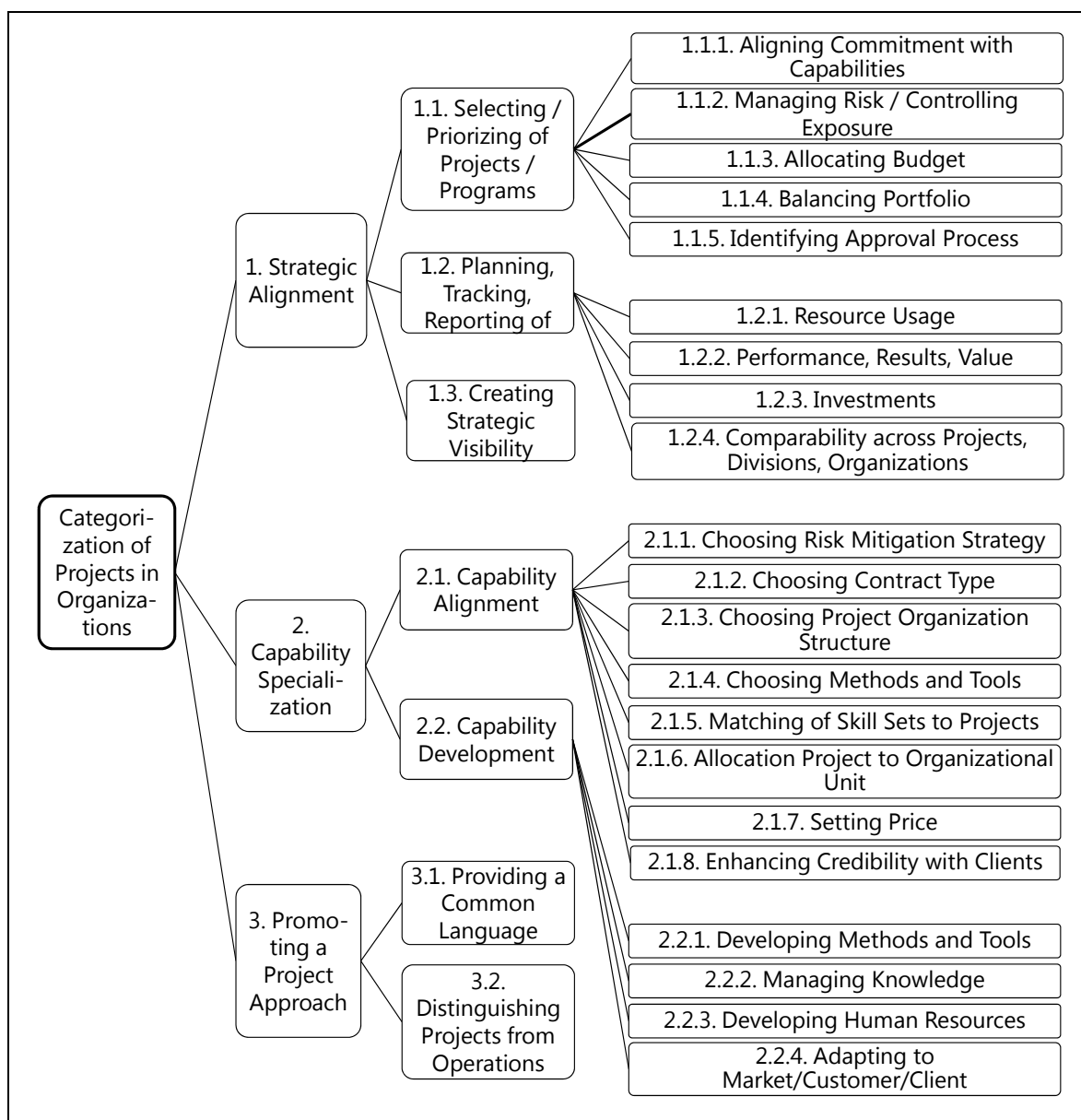


Abb. 79: Zwecke der Klassifikation von Projekten in der Softwareentwicklung⁷⁰⁰

⁶⁹⁹ Vgl. Crawford u.a. (2005), S. 23

⁷⁰⁰ Crawford u.a. (2005), S. 48

Bezogen auf diese Systematisierung kann der situative Koordinationsansatz in die Unterklasse ‚2.1. Capability alignment‘ eingeordnet werden. Zweck der Projektklassifizierung ist dort die Ausrichtung der gesamten Projektstruktur an den situativen Rahmenbedingungen der jeweiligen Projektklasse.⁷⁰¹ „A project that is assigned to a category will benefit from an interrelated set of methods and competencies.“⁷⁰²

Bei der Entscheidung über den Differenzierungsgrad des Klassifikationsschemas, d.h. hier konkret über die Anzahl der Projekttypen, muss bedacht werden, dass hierüber gleichzeitig auch die Gesamtanzahl und der Differenzierungsgrad der Vorgaben für den Projektablauf festgelegt werden. Denn je geringer die Anzahl der vorgegebenen Projekttypen ist, desto gleichförmiger erfolgt die Aufgabenerfüllung trotz einer Unterschiedlichkeit in der Realität. Auf der anderen Seite steigt mit der Anzahl der Projekttypen die Komplexität durch die Prozessvariation enorm.

Die bereits angesprochenen Bedenken der Praxis im Hinblick auf eine nicht mehr handhabbare Komplexität der Prozessvariation sind der Grund für eine wenig differenzierte Umsetzung des situativen Koordinationsansatzes anhand von acht definierten Projekttypen. Die Anzahl ergibt sich durch die Festlegung einer zweistufigen Gewichtungsskala für jede der vier Projektzieldimensionen. Ihre Kritikalität wird jeweils mit Hilfe der in Kap. 3.3 dargestellten Kriterien als ‚normal‘ oder ‚hoch‘ bewertet. Dabei ergeben sich rechnerisch 16 mögliche Präferenzordnungen im Projektzielsystem, wovon jedoch acht in Zusammenhang mit der impliziten Ableitung der in der Standardsoftwareentwicklung untergeordneten Kostenkritikalität aus den drei anderen Zieldimensionen ausgeschlossen werden: Ist ein Projekt in mindestens einer anderen Projektzieldimension hoch kritisch, so kann es sich dabei nicht gleichzeitig auch um ein hoch kostenkritisches Projekt handeln. Dagegen muss es sich um ein hoch kostenkritisches Projekt handeln, wenn keine der drei anderen Projektzieldimensionen als hoch kritisch eingestuft ist. Abb. 80 stellt die acht definierten Projekttypen dar.

⁷⁰¹ Vgl. hierzu die Unterkategorien von ‚2.1. Capability Alignment‘ in Abb. 79. Sie sind an dieser Stelle jedoch nicht überschneidungsfrei und nicht auf derselben Abstraktionsebene. Der Klassifizierungszweck ‚2.1.4. Choosing Methods and Tools‘ kann hier als übergeordnet angesehen werden und entspricht dem Klassifizierungszweck beim vorliegenden Ansatz: „Choosing risk mitigation strategy, contract type and organizational structure are all important examples of choosing methods and tools for managing projects differently. There are many other examples of tools and methods that are differentiated by project type including tools for .. planning, monitoring, controlling ... projects. Project management methodology and governance may also be adapted to suit project type.“ Crawford u.a. (2005), S. 143

⁷⁰² Crawford u.a. (2005), S. 142

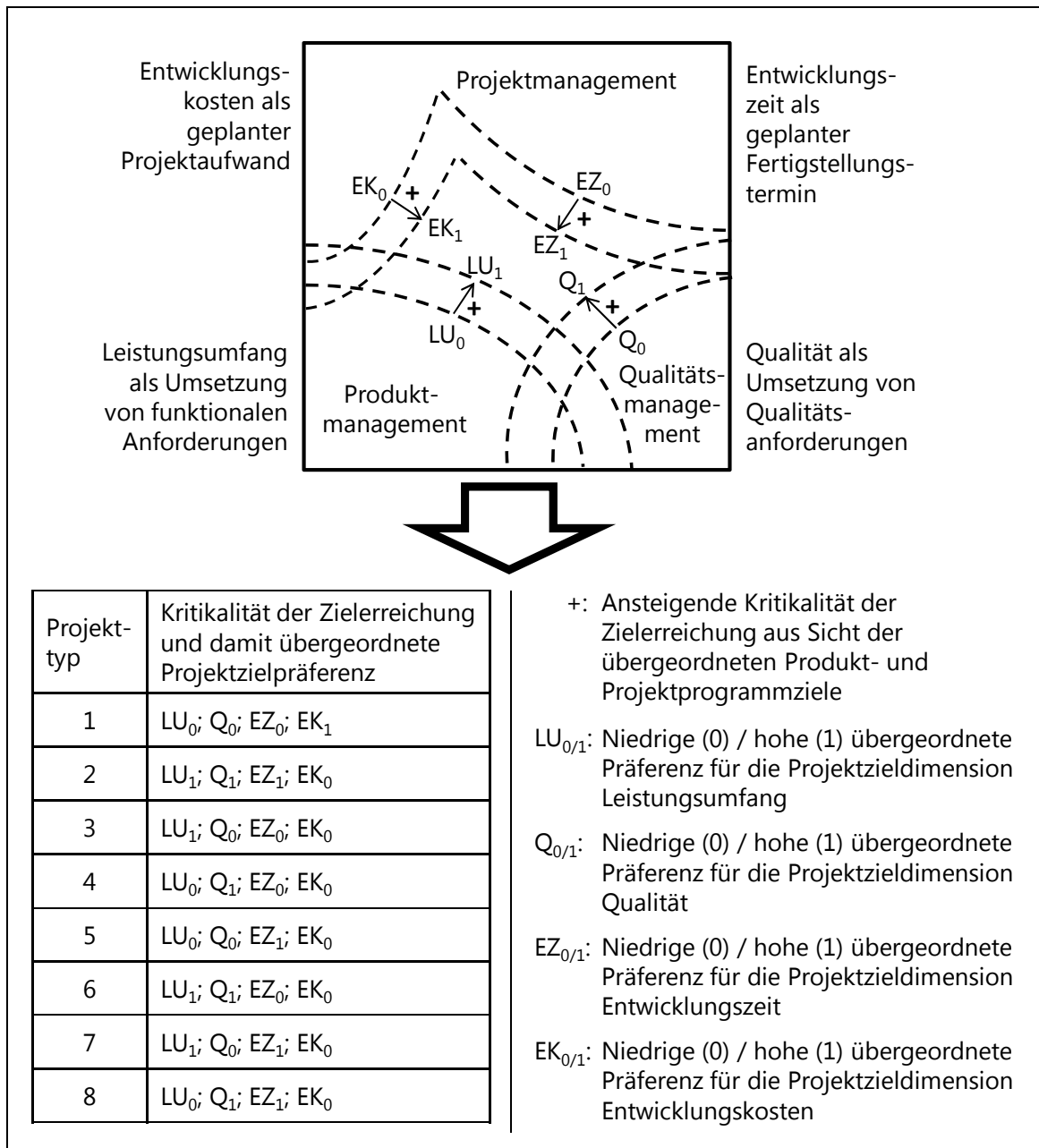


Abb. 80: Umsetzung des situativen Koordinationsansatzes durch definierte Projekttypen

Bei den Experteninterviews wurde an zahlreichen Stellen mithilfe von realen, d.h. in der Fallstudienunternehmung de facto vorhandenen, jedoch nicht zwingend formal definierten Projekttypen argumentiert. Die Auswertung der Interviewergebnisse hinsichtlich aller genannten Projekttypen bzw. Differenzierungsschemata spricht ebenfalls für wenige definierte Projekttypen beim situativen Koordinationsansatz. So sehen sich die befragten Experten in ihrem Softwareentwicklungsbereich jeweils mit nicht mehr als drei hinsichtlich der übergeordneten Projektzielpräferenzen unterschiedlichen Projekttypen konfrontiert. Als Differenzierungskriterium für die Projekttypen dient dabei überwiegend die Kritikalität von einer Projektziel-dimension. Über alle Differenzierungsschemata und Projektziel-dimensionen hinweg lassen sich 10 Projekttypen fest-machen (vgl. Abb. 81).

Differenzierungsschema und Projekttypen	Differenzierungsgrad und Differenzierungskriterium
‚Standard‘ (Projekt mit produkt- und prozessbezogenen Rahmenbedingungen, die üblicherweise bei Projekten im Entwicklungsbereich des Interviewpartners gelten; primär zeitkritisches Projekt); ‚Add On‘ (Projekt, das Zusatzfunktionalität entwickelt; weniger zeitkritisch als das ‚Standard‘-Projekt im Entwicklungsbereich); ‚Integrationsprojekt‘ (Projekt, bei welchem in enger Zusammenarbeit von mindestens zwei Entwicklungsbereichen sog. Integrationsszenarien entwickelt werden; hoch zeitkritisches Projekt, da es meistens auf dem kritischen Pfad eines Projektprogramms liegt). ⁷⁰³	Drei Projekttypen Primär nach Zeitzielkritikalität
‚Standard‘ (Projekt mit produkt- und prozessbezogenen Rahmenbedingungen, die üblicherweise bei Projekten im Entwicklungsbereich des Interviewpartners gelten; primär qualitäts- und zeitkritisches Projekt); ‚Pilot‘ (Projekt zur Entwicklung eines Prototypen oder eines Demonstrators der Standardsoftware zu verschiedenen Zwecken; primär zeitkritisch, weniger qualitätskritisch). ⁷⁰⁴	Zwei Projekttypen Primär nach Qualitäts- und Zeitzielkritikalität
‚Standard‘ (Projekt mit produkt- und prozessbezogenen Rahmenbedingungen, die üblicherweise bei Projekten im Entwicklungsbereich des Interviewpartners gelten; primär zeit- und qualitätskritisches Projekt); ‚Add On‘ (Projekt, das Zusatzfunktionalität entwickelt; weniger zeit- und qualitätskritisch als das ‚Standard‘-Projekt im Entwicklungsbereich). ⁷⁰⁵	Zwei Projekttypen Primär nach Zeit- und Qualitätszielkritikalität
‚Integrationsprojekt‘ (Projekt, bei welchem in enger Zusammenarbeit von mindestens zwei Entwicklungsbereichen sog. Integrationsszenarien entwickelt werden; primär zeit- und leistungsumfangskritisch). ⁷⁰⁶	Ein Projekttyp
‚Standard‘ (Projekt mit produkt- und prozessbezogenen Rahmenbedingungen, die üblicherweise bei Projekten im Entwicklungsbereich des Interviewpartners gelten; Kritikalität nicht näher bestimmt); ‚Ausnahme‘ (Projekt mit abweichenden Rahmenbedingungen von den eines ‚Standard‘-Projektes im Entwicklungsbereich; Kritikalität bei mindestens einer Projektzieldimension wesentlich anders als bei einem ‚Standard‘-Projekt im Entwicklungsbereich). ⁷⁰⁷	Zwei Projekttypen Unterscheidung nach Projektzielkritikalität (nicht näher bestimmt)

Abb. 81: Bei den Experteninterviews genannte Differenzierungsschemata und Projekttypen
 Bereits eine dreistufige Gewichtungsskala (Kritikalität jeweils niedrig/normal/hoch) würde die Zahl der notwendigen Projekttypen auf 27 ansteigen lassen. Zum einen ließen sich aber für ein derart differenziertes Klassifikationsschema kaum noch differenzierte Vorgaben für den Projektablauf definieren.⁷⁰⁸ Zum anderen würde dadurch eine

⁷⁰³ Experte ‚R.L.‘ zur Frage A1

⁷⁰⁴ Experte ‚T.S.‘ zur Frage A1, Experte ‚T.S.‘ zur Frage A5, Experte ‚A.D.‘ zur Frage A1

⁷⁰⁵ Experte ‚T.M.‘ zur Frage A1

⁷⁰⁶ Experte ‚J.C.‘ zur Frage A1

⁷⁰⁷ Experte ‚T.W.‘ zur Frage A1

⁷⁰⁸ Vgl. Pohl (2007), S. 536

in der Praxis nicht mehr handhabbare Komplexität der Prozessvariation entstehen. Des Weiteren muss bedacht werden, dass eine Organisation Kompetenz in der wiederholten Umsetzung der Projektvorgaben aufbauen muss, was bei 27 Projekttypen unrealistisch ist.⁷⁰⁹

Unabhängig davon, ob eine zwei- oder dreistufige Gewichtungsskala zugrundegelegt wird, sind derartige, einfache Bewertungsgrundlagen auch im Umfeld der Anforderungspriorisierung weit verbreitet, obwohl sie einige Fragen aufwerfen. So unterteilt das gebräuchlichste Priorisierungsverfahren in der Praxis Anforderungen in die drei Klassen ‚kritisch/standard/optional‘.⁷¹⁰ Die Kritik an diesem Verfahren richtet sich darauf, dass es zumeist ohne eine wesentliche Erläuterung der Klassen angewendet wird und somit sehr subjektiv ist. Die Beteiligten müssen sich darauf einigen, welche Bedeutung jede dieser Klassen hat. V.a. muss dabei definiert und transparent sein, welche Folgen sich aus der Zuordnung von Anforderungen zu einer der drei Klassen ergeben, d.h. worin der praktische Unterschied zwischen den Klassen besteht.⁷¹¹

Mit den jeweils zugrundegelegten Kriterien bei den vier Projektzieldimensionen und den klar zugeordneten Steuerungsmaßnahmen sind beim situativen Koordinationsansatz die beiden voranstehenden Kritikpunkte unzutreffend. Nichts desto trotz verbleiben offene Fragen aus der entscheidungstheoretischen Sicht. Denn ein solches pragmatisches Bewertungsverfahren erfüllt nicht die Gütekriterien der klassischen Messtheorie hinsichtlich Präzision, Operationalität und Relevanz. Hier muss jedoch entgegengehalten werden, dass diese strengen Anforderungen für die Bewertung betriebswirtschaftlich relevanter Sachverhalte nur begrenzt geeignet sind. Die Gründe hierfür sind vielfältig. Zentral ist dabei aber v.a., dass für betriebswirtschaftliche Beurteilungsprozesse die Wirtschaftlichkeit des Messvorgangs und die Orientierung am Verwendungszweck der Ergebnisse aus Sicht der Entscheidungsträger als entscheidend angesehen werden. Die technische Qualität des Messvorgangs und die Einhaltung der entsprechenden Anforderungen sind dagegen sekundär.⁷¹²

Diese Argumentation kann mit Rückgriff auf das ‚Konzept der problemgerechten Informationsstruktur‘ auch formaler dargestellt werden, wobei hier auf die Wiedergabe der Formeln verzichtet wird:⁷¹³ Ein Entscheidungsträger braucht eine auf sein Entscheidungsproblem zugeschnittene, eine problemgerechte Informationsstruktur, um es

⁷⁰⁹ Vgl. Fitzgerald u.a. (2003), S. 66

⁷¹⁰ O.ä. Bezeichnung; Es wird von den beiden sehr praxisrelevanten Standards ‚RFC 2119‘ und ‚IEEE Std. 830-1998‘ für die Anforderungspriorisierung vorgeschlagen.

⁷¹¹ Vgl. Wieggers (2005), S. 299; vgl. Lehtola, Kauppinen (2006), S. 9, 13f.; vgl. Berander, Andrews (2005), S. 76

⁷¹² Vgl. Omagbemi (1994), S. 153f.; vgl. Fischer (1989), S. 262f.

⁷¹³ Vgl. im Folgenden Marschak (1963), S. 722f., zitiert nach Frese (2005), S. 117-120

lösen zu können. Die problemgerechte Informationsstruktur stellt „ein gewisses Maß an Feinheit der Information über die Umwelt [dar], das dadurch gekennzeichnet ist, dass jede Handlung bei allen unterscheidbaren Umweltereignissen ... zu einem eindeutigen Ergebnis führen soll.“⁷¹⁴ Entsprechende Überlegungen gelten auch für die möglichen Handlungsalternativen. Auch diese „müssen so weit differenziert werden, dass jede unterscheidbare Handlung für alle Umweltzustände zu einem eindeutigen Ergebnis führt.“⁷¹⁵

Wie Marschak, zeigt existiert für jedes deterministische Entscheidungsproblem genau eine problemgerechte Informationsstruktur. Diese weist sowohl ein hinreichendes Maß an Feinheit auf, um eindeutige Ergebnisse zu garantieren, als auch ist sie hinreichend grob, sodass gleichzeitig keine überflüssigen Informationen geliefert werden. Die Überlegungen zur problemgerechten Informationsstruktur lassen sich sehr konkret auf den situativen Koordinationsansatz anwenden und werden von diesem auch berücksichtigt: Die Anzahl der Handlungsalternativen determiniert die Granularität der Klassifikation und umgekehrt, sodass nicht mehr Handlungsalternativen als klassifizierbar und beherrschbar, sowie gleichzeitig nicht weniger Handlungsalternativen (und Klassen) als es die Komplexität erfordert, vorhanden sein dürfen.

3.4.2 Detaillierungsgrad der Vorgaben für die definierten Projekttypen

3.4.2.1 Projekttypen als Konfigurationen

Die zweite zu klärende Kernfrage neben dem Differenzierungsgrad des Klassifikationsschemas bezieht sich auf den Detaillierungsgrad der Vorgaben für den Projektablauf. Hier ist durch den festgelegten Differenzierungsgrad lediglich ein notwendiges Mindestmaß gegeben, bei welchem sich die Vorgaben für die 8 Projekttypen unterscheiden müssen. Wie detailliert sie darüber hinaus sein sollen, wird mit dem Differenzierungsgrad des Klassifikationsschemas nicht festgelegt.⁷¹⁶

Die drei in der Fallstudienunternehmung analysierten Tailoringansätze zeichnen sich durch einen sehr geringen Detaillierungsgrad der Vorgaben aus (vgl. Abb. 82). Insbesondere werden nur die zu verwendeten Verfahren und/oder Mustervorlagen vorgeschrieben und es erfolgt kein Vorgriff auf die Projektplanung. Letzteres kann nur bei einem sehr hohen Detaillierungsgrad der Vorgaben für den Projektablauf der Fall sein, bspw. durch die Spezifizierung einer maximalen Zeitspanne zwischen zwei Meilensteinen. Stattdessen wird bei einem geringen Detaillierungsgrad der Planungsprozess

⁷¹⁴ Frese (2005), S. 119

⁷¹⁵ Frese (2005), S. 119

⁷¹⁶ Vgl. Kap. 2.3.4 zur Abgrenzung zwischen dem Differenzierungsgrad des Klassifikationsschemas und dem Detaillierungsgrad der Vorgaben.

nur indirekt über das Programm gesteuert. D.h. Pläne werden hier auf der Basis von Programmen erstellt, die festlegen, welche Informationen in welcher Art und Weise zu Plänen zu verarbeiten sind.⁷¹⁷

Tailoringansatz⁷¹⁸	Detaillierungsgrad der Vorgaben
Tailoringansatz für die Produkt- und Anforderungsdefinition	Niedrig: Die Vorgaben beziehen sich lediglich auf die zu erbringenden Artefakte und stellen im Wesentlichen bestimmte Anpassungen zum ‚Standardprozess‘ als Ausgangssituation dar. Es erfolgt kein Vorgriff auf die Projektplanung.
Tailoringansatz für verschiedene Klassen von Entwicklungsvorhaben	Niedrig: Die Vorgaben leiten sich aus dem grundsätzlichen Ziel ab, den Softwareentwicklungsprozess ganz allgemein auf die übergeordneten Rahmenbedingungen und auf den Entwicklungsgegenstand auszurichten. Es erfolgt eine indirekte Steuerung über die Verfahrensspezifikation und kein Vorgriff auf die Projektplanung, insbesondere auch nicht durch die Vorgabe der Meilensteine. Denn hier wird lediglich vorgegeben, welche Meilensteine grundsätzlich zu definieren sind und nicht, wann sie erreicht werden müssen.
Tailoringansatz für die Implementierung von Szenarien	Niedrig: Die Vorgaben unterscheiden sich v.a. hinsichtlich des Verbindlichkeitsgrades der Empfehlung zur Durchführung des sog. Integrationstests und bestimmter Validierungsaktivitäten. Es erfolgt kein Vorgriff auf die Projektplanung.

Abb. 82: Detaillierungsgrad der Vorgaben bei den Tailoringansätzen in der Fallstudienunternehmung

Im Vergleich zum Vorgriff auf die Projektplanung ist eine indirekte Steuerung über die Programmierung von Planungsprozessen wesentlich flexibler. Denn Programme legen den Ablauf von Aktivitäten auf Dauer fest. Wenn bestimmte Probleme auftauchen, so sind diese immer mit dem im Programm festgelegten Verfahren zu lösen. Programme spezifizieren also Verfahren und keine Ziele. Pläne enthalten dagegen immer Zielvorgaben für eine bestimmte Periode.⁷¹⁹ Aus Sicht des Strukturierungsmodells von Entscheidungsproblemen ist das Verhältnis zwischen Planung und (indirekter) Programmierung wie folgt: Die bei der (indirekten) Programmierung vorgenommene Strukturierung der Entscheidungskomponenten verringert den Lösungsraum bei der Planung. Damit verbleibt die Planung der letzte Schritt zur Erstellung einer realisationsreifen Lösung.

Der situative Koordinationsansatz orientiert sich auch beim Detaillierungsgrad der Vorgaben an den analysierten Tailoringansätzen in der Fallstudienunternehmung. Auch hier soll kein Vorgriff auf die Projektplanung erfolgen. Dies ist nicht notwendig,

⁷¹⁷ Vgl. Kieser, Walgenbach (2003), S. 119f.

⁷¹⁸ Vgl. zur Beschreibung der Ansätze Abb. 76-78 in Kap. 3.4.1.1

⁷¹⁹ Vgl. Kieser, Walgenbach (2003), S. 119f.

um Koordinationseffektivität zu erreichen – hierfür reicht bereits ein geringer Detaillierungsgrad der Vorgaben für den Projektablauf ab: Die Vorgaben müssen sicherstellen, dass die Stimmigkeit⁷²⁰ zwischen den einzelnen Gestaltungsmaßnahmen bei den Koordinationsprozessen⁷²¹ untereinander und zu den übergeordneten Projektzielpräferenzen⁷²² gegeben ist. Das ist bereits der Fall, wenn

- bei den vier Koordinationsprozessen Projektsteuerung sowie Qualitäts-, Risiko- und Änderungsmanagement
- alle Gestaltungsmaßnahmen in den drei Strukturdimensionen Artefaktspezifikation, Kompetenzverteilung sowie ‚Personelle und organisatorische Unabhängigkeit‘
- stets auf die Kritikalität der Projektziele und damit auf die Präferenzordnung im Projektzielsystem

ausgerichtet sind. Die acht definierten Projekttypen bilden somit Konfigurationen im Sinne des Konfigurationsansatzes der Organisationsgestaltung (vgl. Abb. 83).⁷²³ Im Anhang der Arbeit finden sich beispielhafte Konfigurationen zu den definierten Projekttypen.⁷²⁴

Je höher der Detaillierungsgrad der Vorgaben auf Projekttyp-Ebene ist, desto weniger Anpassungsbedarf verbleibt bei der anschließenden projektspezifischen Adaption.⁷²⁵ Damit ist es zwar umso wahrscheinlicher, dass auch die realisationsreifen Vorgaben auf Projektebene nach der Projektplanung stimmig sind. Auf der anderen Seite muss aber bedacht werden, dass weniger Anpassungsbedarf auf Projektebene gleichzeitig auch weniger Spielraum, bzw. konkreter, umso weitreichende Einschränkungen des Lösungsraumes für das Entscheidungsproblem auf projektspezifischer Ebene (Projektplanung) bedeutet. In diesem Spannungsverhältnis bewegt sich der ‚richtige‘ Detaillierungsgrad der Vorgaben auf Projekttyp-Ebene, um Koordinationseffektivität sicherzustellen. Die Annahme, dass es ein tendenziell niedriger Detaillierungsgrad ist, der keinen Vorgriff auf die Projektplanung darstellt, wird sowohl durch die Experteninterviews als auch durch die empirische Arbeit von Abdel-Hamid u.a. gestützt.

⁷²⁰ Vgl. Kap. 2.3.1

⁷²¹ Vgl. Kap. 3.2

⁷²² Vgl. Kap. 3.3

⁷²³ Konfigurationen sind im Allgemeinen repräsentative Muster bzw. Kombinationen von stimmigen Ausprägungen der zahlreichen interdependenten Merkmale der Situation und der Organisationsstruktur, wobei das insb. auch die interne Konsistenz zwischen den Gestaltungsparametern und der Organisationsstruktur mit einbezieht; vgl. hierzu Kap. 2.2.1

⁷²⁴ Vgl. Anhang: Beispielhafte Konfigurationen zu den definierten Projekttypen

⁷²⁵ Vgl. Kap. 2.3.4

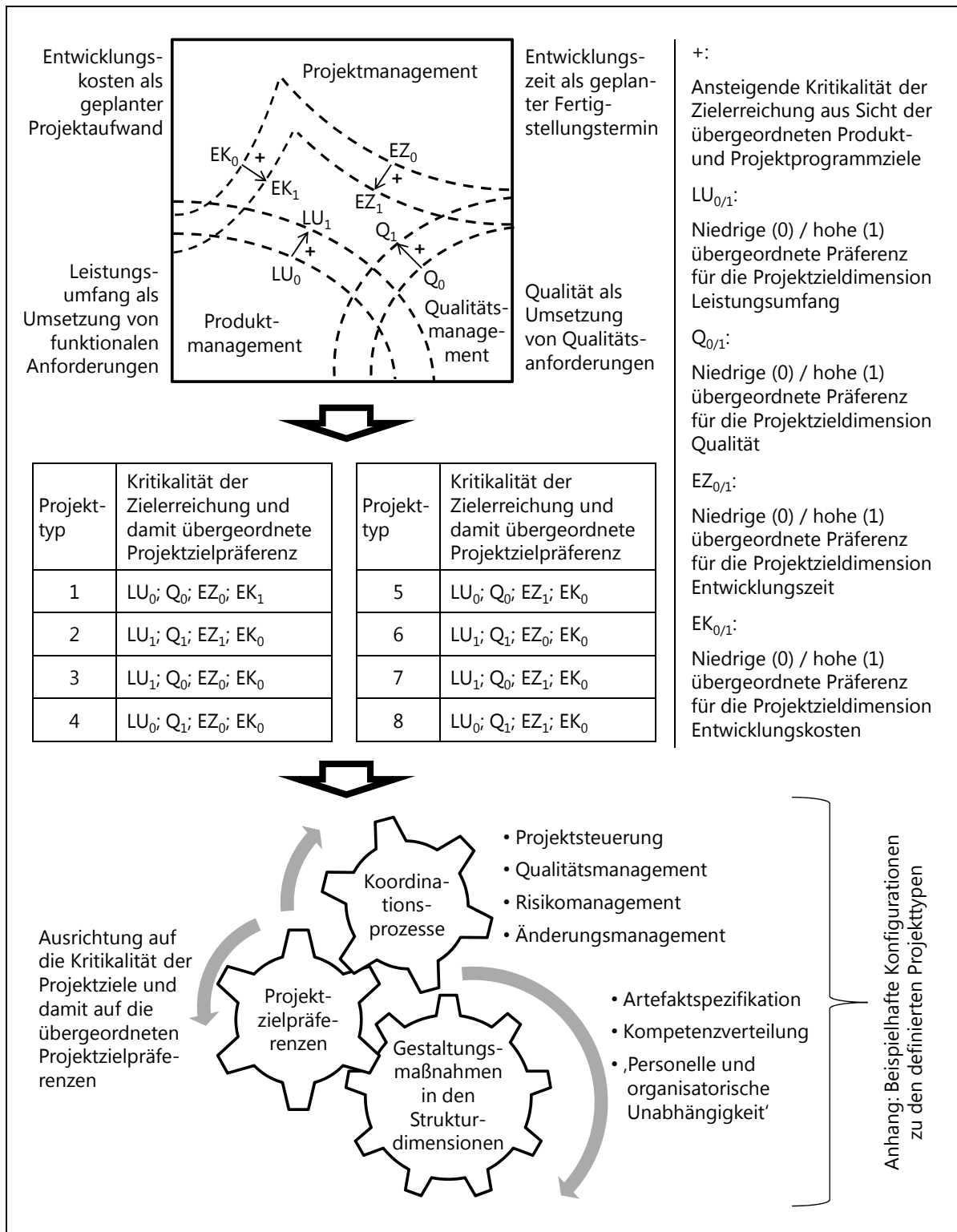


Abb. 83: Projekttypen des situativen Koordinationsansatzes als Konfigurationen im Sinne des Konfigurationsansatzes der Organisationsgestaltung

Abdel-Hamid u.a. zeigen, dass schon alleine die klare Vorgabe von übergeordneten Projektzielpräferenzen die Projektplanung und den weiteren Projekttablauf zielführend beeinflusst. Dazu wurden im Rahmen eines Laborexperimentes zwei alternative Präferenzordnungen definiert und als die unabhängige Variable ‚Zielsetzung‘ den beiden

Untersuchungsgruppen, jeweils Experimentteilnehmer in der Projektleiter-Rolle⁷²⁶ eines Softwareentwicklungsprojektes, vorgegeben.⁷²⁷

- Zielsetzung 1: Minimierung von Kosten- und Zeitzielabweichungen;
- Zielsetzung 2: Erreichung der Qualitätsziele und sofern möglich, Minimierung von Kosten- und Zeitzielabweichungen.

Die Ergebnisse des Experimentes zeigen, dass bei Kenntnis der Projektzielpräferenzen Entscheidungen bei der Projektplanung, sowie auch im weiteren Projektverlauf, so getroffen werden, dass das Softwareentwicklungsprojekt seine hoch präferierten Ziele erreicht. So wurden bei beiden Untersuchungsgruppen signifikante Unterschiede in der Entscheidungsfindung bei Problemstellungen, die direkt die Zielerreichung beeinflussen, festgestellt. Sie führten dazu, dass die Untersuchungsgruppe mit Zielsetzung 1 das Projekt ohne Kosten- und Zeitzielabweichungen abgeschlossen hat, während die Untersuchungsgruppe mit Zielsetzung 2 die Qualitätsziele erreicht hat, jedoch unter Inkaufnahme von Kosten- und Zeitzielabweichungen.⁷²⁸

Abdel-Hamid u.a. kommen bei ihrer Interpretation der Ergebnisse zum Schluss, dass Zielvorgaben, die im Projekt u.a. eine Steuerungsfunktion haben, von den oftmals unrealistischen anfänglichen Schätzwerten (v.a. aus der Aufwandsschätzung) abgekoppelt werden müssen. Stattdessen sollten die Zielsetzungen auf Größen basieren, welche die Steuerungsfunktion von Zielen besser erfüllen und somit die Entscheidungsfindung im Projektverlauf zielführend beeinflussen.⁷²⁹ Hierin sehen Abdel-Hamid u.a. auch weiteren Forschungsbedarf: „It would be a worthwhile topic for future research to identify goals that can help shape strategy and thus free managers from having to persevere with minimizing overruns from the initial cost estimate.“⁷³⁰

Die explizite Vorgabe der Präferenzordnung im (Projekt-)Zielsystem, wie auch von der normativen Entscheidungstheorie gefordert, beeinflusst die Entscheidungsfindung im Projektverlauf zielführend und ganz im Sinne der Forderung von Abdel-Hamid u.a. Dies stellen auch zahlreiche Aussagen bei den Experteninterviews heraus. Es wird in diesem Zusammenhang zumeist von einer hohen ‚Transparenz‘ gesprochen, die bei der Projektplanung und im weiteren Projektverlauf durch die konsequente Ausrichtung von Organisations- und Objektentscheidungen an den übergeordneten Projektzielprä-

⁷²⁶ Im Experiment mit der Gesamtverantwortung für die Planung und Zielerreichung hinsichtlich aller Projektzieldimensionen ausgestattet.

⁷²⁷ Vgl. Abdel-Hamid u.a. (1999), S. 537; zu weiteren Details des Experimentdesigns wie u.a. abhängige Variablen und Operationalisierung der Größe ‚Projektzielerreichung‘ vgl. Abdel-Hamid u.a. (1999), S. 535ff.; zur externen Validität der Ergebnisse vgl. Abdel-Hamid u.a. (1999), S. 545f.

⁷²⁸ Vgl. Abdel-Hamid u.a. (1999), S. 545f.

⁷²⁹ Vgl. Abdel-Hamid u.a. (1999), S. 548

⁷³⁰ Abdel-Hamid u.a. (1999), S. 548

ferenzen entsteht. Diese Transparenz gebe den Entscheidungsträgern eine Leitlinie vor und erhöhe somit die Entscheidungsqualität und Rationalität von Entscheidungen.⁷³¹

Allerdings muss an dieser Stelle betont werden, dass trotz der positiven Wirkung, die schon alleine durch die klare Vorgabe der übergeordneten Projektzielpräferenzen erreicht wird, auf wie auch immer detaillierte Vorgaben für den Projektablauf nicht verzichtet werden kann. Der Grund hierfür ist, dass für rationale Entscheidungen neben der Eindeutigkeit der Zielvorstellung auch ein gewisses Maß an Wissen und Transparenz über die zur Verfügung stehenden Handlungsalternativen und deren Wirkungen bei den Entscheidern vorhanden sein muss. Insgesamt gilt, dass je vollkommener die folgenden Voraussetzungen erfüllt sind, desto mehr entsprechen Organisationsentscheidungen den Anforderungen der normativen Entscheidungstheorie:⁷³²

- Eindeutigkeit der Zielvorstellung: Entscheider müssen die durch die übergeordneten Entscheidungsziele definierte Präferenzstruktur kennen. Wer keine Kenntnis darüber und über die Ziele an sich hat, kann schon aus logischen Gründen nicht entscheiden, denn zielentsprechend handeln setzt die Kenntnis des vollständigen Zielsystems voraus,⁷³³
- Wissen über die Situation und die Handlungsalternativen: Der Entscheider muss über die Kenntnis der Ausgangssituation und des verfügbaren organisatorischen Instrumentariums verfügen;
- Wissen über die zielbezogenen Wirkungen der Handlungsalternativen: Der Entscheider muss wissen, welche Auswirkungen die jeweils betrachtete organisatorische Handlungsalternative auf die Realisierung der Ziele hat;
- Wissen zur Beherrschung des Gestaltungsprozesses: Diese Voraussetzung stellt die Grundlage dar, um den „Prozess der Transformation der organisatorisch relevanten Ausgangssituation in die angestrebte Organisationsstruktur erfolgreich zu bewältigen.“⁷³⁴

3.4.2.2 Verbindlichkeitsgrad der Vorgaben für den Projektablauf

Im Unterschied zu konkreten Programmen stellen abstrakte Programme mit einem niedrigen Detaillierungsgrad der Vorgaben nur grobe Anleitungen dar, die für ihre Umsetzung erst interpretiert und konkretisiert werden müssen. Beim situativen Koordinationsansatz findet diese Interpretation und Konkretisierung über die projektspezifi-

⁷³¹ Experte ‚C.S.‘ zur Frage D1, Experte ‚J.W.‘ zur Frage D1, Experte ‚T.S.‘ zur Frage D1, Experte ‚O.S.‘ zur Frage D1, Experte ‚J.K.‘ zur Frage D1

⁷³² Vgl. Frese (2005), S. 71f.

⁷³³ Vgl. auch Fischer (1989), S. 99, mit zahlreichen weiteren Verweisen.

⁷³⁴ Frese (2005), S. 72

sche Adaption der Projekttypvorgaben statt. Durch den niedrigen Detaillierungsgrad der Vorgaben auf Projekttyp-Ebene ist hier der notwendige hohe Anpassungsbedarf bzw. Spielraum gegeben. Dieselbe Wirkung lässt sich aber auch erzielen, indem Vorgaben für den Projektablauf, unabhängig von ihrem Detaillierungsgrad, nicht ausnahmslos verpflichtend formuliert werden, sodass bei der projektspezifischen Umsetzung hiervon bei Bedarf abgewichen werden kann. Ein solcher niedriger Verbindlichkeitsgrad der Projektvorgaben ist nach Expertenansicht ein unabdingbarer Aspekt, der mit einem niedrigen Detaillierungsgrad der Projektvorgaben einhergehen muss. Es wird an dieser Stelle vom ‚Überschreiben der Empfehlungen‘, von ‚nicht bindenden Rahmenbedingungen‘ oder auch von ‚Vorschlägen statt Vorgaben‘ gesprochen.⁷³⁵

Auch in der umfangreichen Studie von Crawford u.a.⁷³⁶ wird der Verbindlichkeitsgrad von Vorgaben an zentraler Stelle angesprochen. Abb. 84 thematisiert die von den dort befragten Experten am häufigsten genannten potentiellen Problemkomplexe bei der Umsetzung von Ansätzen aller Art zur Projektkategorisierung. Als zweithäufigster Problemkomplex wird dabei die Rigidität bzw. Starrheit der Ansätze genannt. Weit über die Hälfte der Antwortenden berichten von derartigen Problemen, die in einer unerwünschten Bürokratisierung der Ansätze resultieren. In diesem Zusammenhang ziehen analog zur vorliegenden Arbeit auch Crawford u.a. die Schlussfolgerung: „Any [project categorization] system needs to be flexible enough to allow for exceptions“⁷³⁷

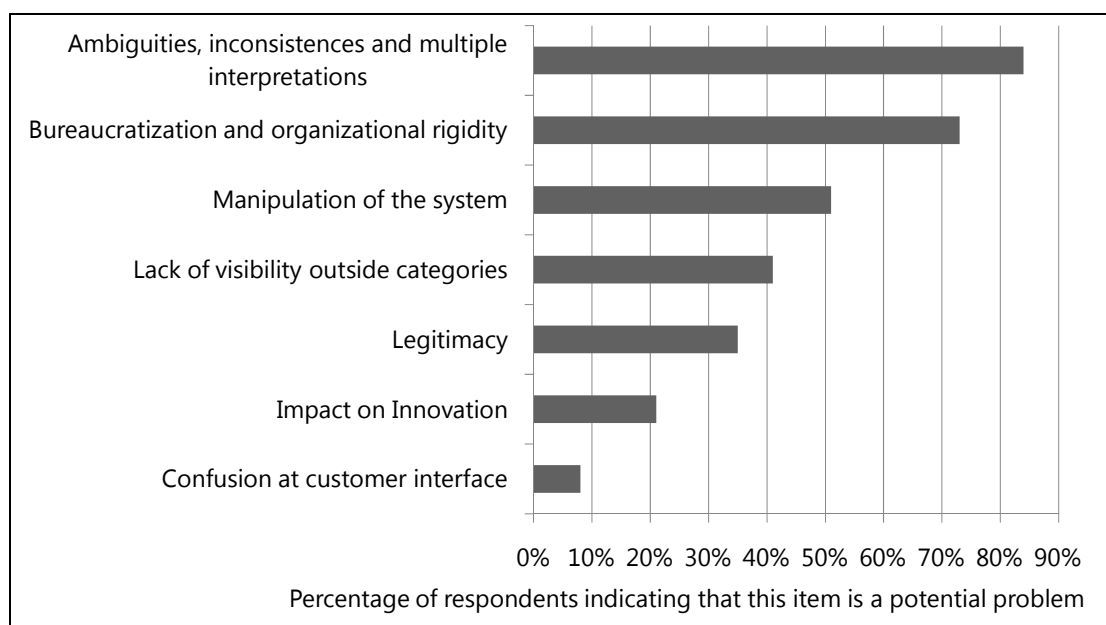


Abb. 84: Problemkomplexe bei der Umsetzung von Projektkategorisierungsansätzen⁷³⁸

⁷³⁵ Experte ‚T.M.‘ zur Frage C1, Experte ‚T.S.‘ zur Frage C1, Experte ‚A.D.‘ zur Frage C1, Experte ‚F.E.‘ zur Frage C1, Experte ‚L.G.‘ zur Frage C1, Experte ‚O.S.‘ zur Frage B2

⁷³⁶ Vgl. zu der Studie von Crawford u.a. (2005) auch Kap. 3.4.1.2

⁷³⁷ Crawford u.a. (2005), S. 40

⁷³⁸ Crawford u.a. (2005), S. 37

Die häufigsten potentiellen Probleme bei der Umsetzung von Projektkategorisierungsansätzen sind nach Crawford u.a. mangelnde Eindeutigkeit, Inkonsistenzen und mehrfache Auslegungsmöglichkeiten in Zusammenhang mit den Vorgaben für den Projektablauf. Die Autoren stellen fest, dass selbst in Unternehmungen mit etablierten und weit akzeptierten Projektkategorisierungsansätzen Vieldeutigkeiten bei wesentlichen Sachverhalten zu beobachten sind. Projektbeteiligte schilderten hier Fälle, in welchen ihnen die präzise Bedeutung von selbst zentralen Vorgaben der Projektkategorisierungsansätze in ihrer Unternehmung nicht bekannt seien.⁷³⁹

Bei der organisatorischen Umsetzung des situativen Koordinationsansatzes tut sich an dieser Stelle ein Konflikt auf: So sollen die Projektvorgaben auf der einen Seite nicht nur wenig detailliert, sondern zudem auch als nicht zwingend verbindliche Empfehlungen formuliert sein. Auf der anderen Seite müssen trotz des dadurch vorhandenen Spielraumes potentielle Doppeldeutigkeiten, Inkonsistenzen und mehrfache Auslegungsmöglichkeiten ausgeschlossen werden können. Dieser Konflikt kann aufgelöst werden, indem die Begrifflichkeiten, welche insbesondere den Verbindlichkeitsgrad der Vorgaben charakterisieren, in einer eingeschränkten und genau definierten Form verwendet werden. Eine solche Vorgehensweise ist bei Standards und Normen im Bereich der Softwareentwicklung gängig.⁷⁴⁰ Hierzu muss das Tailoring zwingend in einem formalen Rahmen stattfinden, den aber bereits ein pragmatisch ausgerichteter Workshop bieten kann. Auf beide Ansatzpunkte soll im Folgenden eingegangen werden.

Neben den in Kap. 3.2.3.1 angesprochenen Ansätzen zur organisatorischen Unabhängigkeit schlägt die Norm DIN EN 50128 zahlreiche weitere Techniken und Maßnahmen zur Anwendung in Abhängigkeit von der Software-Sicherheitsanforderungsstufe vor. Mit jeder Technik und Maßnahme ist eine Festlegung hinsichtlich ihres Verbindlichkeitsgrads für die jeweilige Software-Sicherheitsanforderungsstufe verbunden. Diese Festlegungen können wie folgt lauten:⁷⁴¹

- ‚M‘ (mandatory): Die Anwendung der in der Norm vorgeschlagenen Technik oder Maßnahme ist für die entsprechende Sicherheitsanforderungsstufe verbindlich;
- ‚HR‘ (highly recommended): Die Anwendung der in der Norm vorgeschlagenen Technik oder Maßnahme ist für die entsprechende Sicherheitsanforderungsstufe dringend empfohlen. Falls diese Technik oder Maßnahme entgegen der dringenden Empfehlung nicht angewendet wird, muss der Grund für die Nichtanwendung im

⁷³⁹ Vgl. Crawford u.a. (2005), S. 37f.

⁷⁴⁰ Vgl. Liggesmeyer, Rombach (2005), S. 53

⁷⁴¹ Vgl. DIN (Hrsg., 2009), S. 60

Qualitätsplan, oder in einem anderen vom Qualitätsplan referenzierten Dokument dargestellt werden;

- ‚R‘ (recommended): Die Anwendung der in der Norm vorgeschlagenen Technik oder Maßnahme ist für die entsprechende Sicherheitsanforderungsstufe empfohlen. Der Verbindlichkeitsgrad dieser Empfehlung ist jedoch geringer als der einer ‚HR‘-Empfehlung, sodass auch der Grund für die Nichtanwendung nicht dokumentiert werden muss;
- ‚NR‘ (not recommended): Die Anwendung der in der Norm vorgeschlagenen Technik oder Maßnahme ist für die entsprechende Sicherheitsanforderungsstufe definitiv nicht empfohlen. Analog zu der ‚HR‘-Empfehlung muss auch hier, falls diese Technik oder Maßnahme entgegen der Empfehlung angewendet wird, der Grund dafür im Qualitätsplan oder in einem anderen vom Qualitätsplan referenzierten Dokument dargestellt werden.

Etwas weniger ausdifferenziert als die Norm DIN EN 50128 verfährt das Prozessrahmenwerk CMMI bei seinen Festlegungen zum Verbindlichkeitsgrad der Vorgaben. Um anzuzeigen, wie die Modellkomponenten von CMMI in dieser Hinsicht zu interpretieren sind, sind diese in die drei Kategorien ‚erforderlich‘, ‚erwartet‘ und ‚informativ‘ eingeteilt:⁷⁴²

- Erforderliche CMMI-Modellkomponenten beschreiben, was eine softwareentwickelnde Organisation erreichen bzw. beherrschen muss, um ein Prozessgebiet zu erfüllen. Die erforderlichen Komponenten in CMMI sind die sog. spezifischen und generischen Ziele. CMMI empfiehlt also keine Methoden oder Werkzeuge verbindlich. Erforderlich ist vielmehr die Erfüllung von bestimmten Zielen, die durch die (Institutionalisierung der) Prozessgebiete verfolgt wird. Vorschläge, wie diese Prozessgebiete im Einzelnen ausgeführt werden können, gehören nicht zu den erforderlichen Komponenten von CMMI und damit nicht zu den verbindlichen Vorgaben. Erfüllt ein Prozessgebiet in der Umsetzung sichtbar seine Ziele, so gilt es als erreicht bzw. beherrscht;
- Zu den erwarteten CMMI-Modellkomponenten gehören die sog. spezifischen und generischen Praktiken, die den Prozessgebieten zugeordnet werden können. Werden diese Praktiken in einer softwareentwickelnden Organisation umgesetzt, so gelten die Ziele der entsprechenden Prozessgebiete als erreicht. Da es sich um erwartete und nicht um verbindliche Komponenten handelt, können Prozessgebiete aber ebenso durch alternative zielführende Praktiken erfüllt werden;

⁷⁴² Vgl. SEI (Hrsg., 2006), S. 16f.; zum Aufbau des CMMI vgl. auch Kap. 2.1.3.1

- Informative CMMI-Modellkomponenten gehen auf weiterführende Einzelheiten ein und sollen softwareentwickelnde Organisationen bei ihrer Umsetzung der erforderlichen und ggf. der erwarteten CMMI-Modellbestandteile unterstützen – sofern hierzu Bedarf besteht. Beispiele für informative Modellkomponenten sind typische Arbeitsergebnisse, Subpraktiken, Ausarbeitungen zu generischen Praktiken sowie diverse Anmerkungen und Querverweise.

Der Formalisierungsgrad von Tailoringansätzen, die auf der Projektkategorisierung basieren, ist in der Praxis sehr verschieden. Das Kontinuum reicht von formalen bis völlig informalen Ansätzen, die unternehmens- oder entwicklungsbereichsweit umgesetzt werden (vgl. Abb. 85). Dabei besteht ein Zusammenhang zwischen dem Differenzierungs- und dem Formalisierungsgrad: Je mehr Projekttypen definiert sind und je gefestigter das Projektkategorisierungssystem ist, desto formaler ist das Projekttailoring.⁷⁴³

	Frequency	Percentage
Formal system used throughout the organization	33	27
Formal system that varies between departments	25	21
Informal system used throughout the organization	19	16
Informal system that varies between departments	31	26
No system	11	10
Total	119	100

Abb. 85: Formalisierungsgrad von Projektkategorisierungsansätzen⁷⁴⁴

Auch Shenhar stellt fest, dass sich zwar in den meisten Organisationen implizit verschiedene Vorgehensweisen für verschiedene Projekte etabliert haben. In der Mehrzahl dieser Fälle findet aber kein formales Projekttypen-Tailoring statt. Seine Gestaltungsempfehlung, diesen Schritt zu formalisieren und in der Projektplanungsphase zu institutionalisieren, richtet Shenhar an alle softwareentwickelnden Organisationen, unabhängig von der Anzahl der (implizit) vorhandenen Projekttypen.⁷⁴⁵

Um die beiden zentralen Problemkomplexe bei der Umsetzung von Projektkategorisierungsansätzen zu umgehen, ist beim situativen Koordinationsansatz zusätzlich zu den präzisen Festlegungen des Verbindlichkeitsgrades der Vorgaben ein formales und in die Projektplanungsphase institutionalisiertes Projekttypen-Tailoring vorgesehen. Als formaler Rahmen dient dabei ein Projektprogrammstart-Workshop in Verbindung mit Projektstart-Workshops. In einem solchen Projekt- bzw. Projektprogrammstart-

⁷⁴³ Vgl. Crawford u.a. (2005), S. 24f.

⁷⁴⁴ Crawford u.a. (2005), S. 24

⁷⁴⁵ Vgl. Shenhar (2001a), S. 412

Workshop erarbeiten die Beteiligten ohnehin die wesentlichen Grundlagen für die späteren Phasen des Entwicklungsvorhabens. Zu den Tagesordnungspunkten gehört hier u.a.:⁷⁴⁶

- Rekapitulation des Projekt- bzw. Projektprogrammauftrages incl. einer Diskussion über die wichtigsten Rahmenbedingungen und die Projekt- bzw. Projektprogrammziele;
- Aufstellen von Team- und Entscheidungsregeln;
- Verteilung wichtiger Rollen;
- Festlegen des Informations- und Kommunikationssystems.

Oestereich und Weiss sprechen in diesem Zusammenhang von einem Präparationsworkshop. Dieser moderierte Workshop soll zur Vorbereitung einer bevorstehenden Phase des Entwicklungsvorhabens dienen, um u.a. Klarheit über deren Ziele und Herausforderungen zu erlangen. Zusätzlich zum Präparationsworkshop wird empfohlen, die offenen Fragen mit einer Reihe von weiteren, speziellen Workshops vertiefend zu bearbeiten. Genannt werden hier u.a. der Stakeholder-Workshop zur Identifikation und Priorisierung der projekt- bzw. projektprogrammrelevanten Interessenvertreter (Stakeholder) und der Auftragsklärungsworkshop zur Klärung des der Projekt- bzw. Projektprogrammziele.⁷⁴⁷

Im Zuge der o.g. Tagesordnungspunkte werden zur Umsetzung des situativen Koordinationsansatzes beim Projektprogrammstart-Workshop die zu diesem Zeitpunkt bereits gebildeten Projekte einem der acht definierten Projekttypen zugeordnet. Wie in Kap. 3.4.1.2 bereits erläutert geschieht dies mithilfe einer zweistufigen Gewichtungsskala für jede der vier Projektzieldimensionen und der in Kap. 3.2 dargestellten Kriterien.⁷⁴⁸ Anschließend werden auf Projektebene im Zuge der einzelnen Projektstart-Workshops im Projektprogramm die Vorgaben auf Projekttyp-Ebene projektspezifisch adaptiert. Der Projektstart-Workshop als formaler Rahmen für das Tailoring soll insbesondere gewährleisten, dass die Festlegungen zum Verbindlichkeitsgrad der Vorgaben auch tatsächlich umgesetzt werden (vgl. Abb. 86).

⁷⁴⁶ Vgl. Bea u.a. (2008), S. 109; vgl. Schelle u.a. (2005), S. 136f.

⁷⁴⁷ Vgl. Oestereich, Weiss (2007), S. 68f., 321

⁷⁴⁸ Zu diesem Zeitpunkt sind bereits die allgemeine Produktdefinition sowie wesentliche Zwischenergebnisse der Phase ‚Requirements Engineering und Analysis‘ abgeschlossen. Beides findet auf der Projektprogramm-Ebene statt und ist dem eigentlichen Entwicklungsvorhaben teilweise vorangestellt.

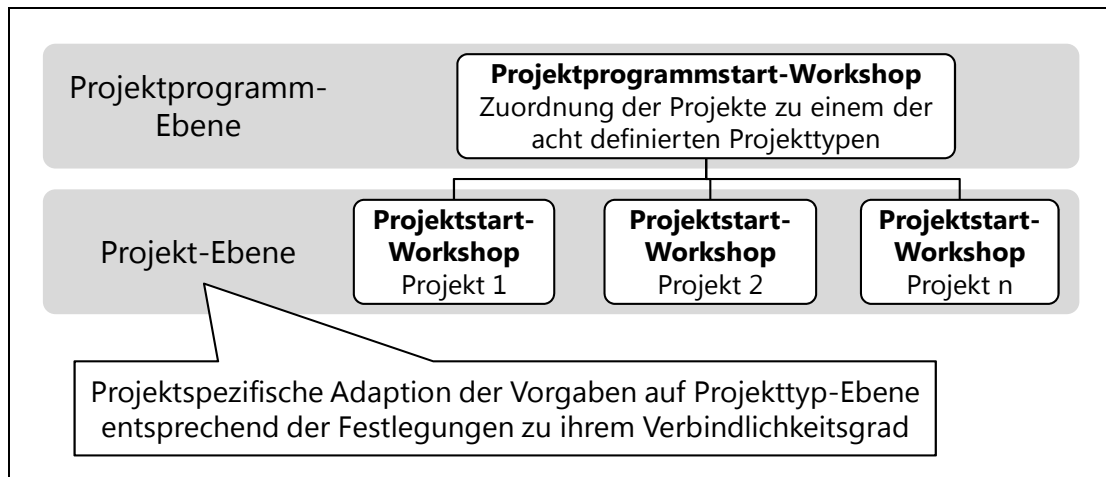


Abb. 86: Projektprogrammstart-Workshop in Verbindung mit Projektstart-Workshops als formaler Rahmen für das Tailoring beim situativen Koordinationsansatz

3.5 Umsetzung des situativen Koordinationsansatzes im Überblick

Im vorliegenden Kap. 3 der Arbeit wurde der Konzeptionsrahmen des situativen Koordinationsansatzes zum Entscheidungsrahmen weiterentwickelt (vgl. Abb. 87).

(3') Im Rahmen der Umsetzung des situativen Koordinationsansatzes wurden aufbauend auf dem Konzeptionsrahmen konkrete Gestaltungsmaßnahmen für die situative Ausrichtung der Koordinationsprozesse in Abhängigkeit von den übergeordneten Projektzielpräferenzen entwickelt. Die Gestaltungsmaßnahmen wurden dabei zu drei sog. Strukturdimensionen zusammengefasst: Artefaktspezifikation, Kompetenzverteilung, sowie ‚Personelle und organisatorische Unabhängigkeit‘.⁷⁴⁹

Die Strukturdimension Artefaktspezifikation beinhaltet alle entwickelten organisatorischen Regelungen zur Ausgestaltung der Informationsstruktur von Arbeitsergebnissen (Artefakten) der Koordinationsprozesse. Diese Regelungen gehen davon aus, dass die Informationsstruktur der Arbeitsergebnisse von Koordinationsprozessen in Abhängigkeit von den übergeordneten Projektzielpräferenzen gestaltet werden muss. Die Artefaktspezifikation bündelt damit alle konkreten Gestaltungsmaßnahmen, die nach dem Strukturierungskonzept von Laux über die Beeinflussung der Informationsstruktur und der Menge der erwogenen Alternativen Koordinationseffektivität sicherstellen sollen. Dabei wurden, wie bei allen drei Strukturdimensionen, ausgehend von den in der Fallstudienunternehmung identifizierten Gestaltungsmaßnahmen generalisierte Gestaltungsmaßnahmen für jeden der vier Koordinationsprozesse Projektsteuerung sowie Risiko-, Änderungs- und Qualitätsmanagement abgeleitet.⁷⁵⁰

⁷⁴⁹ Vgl. Kap. 3.2

⁷⁵⁰ Vgl. Kap. 3.2.1.1

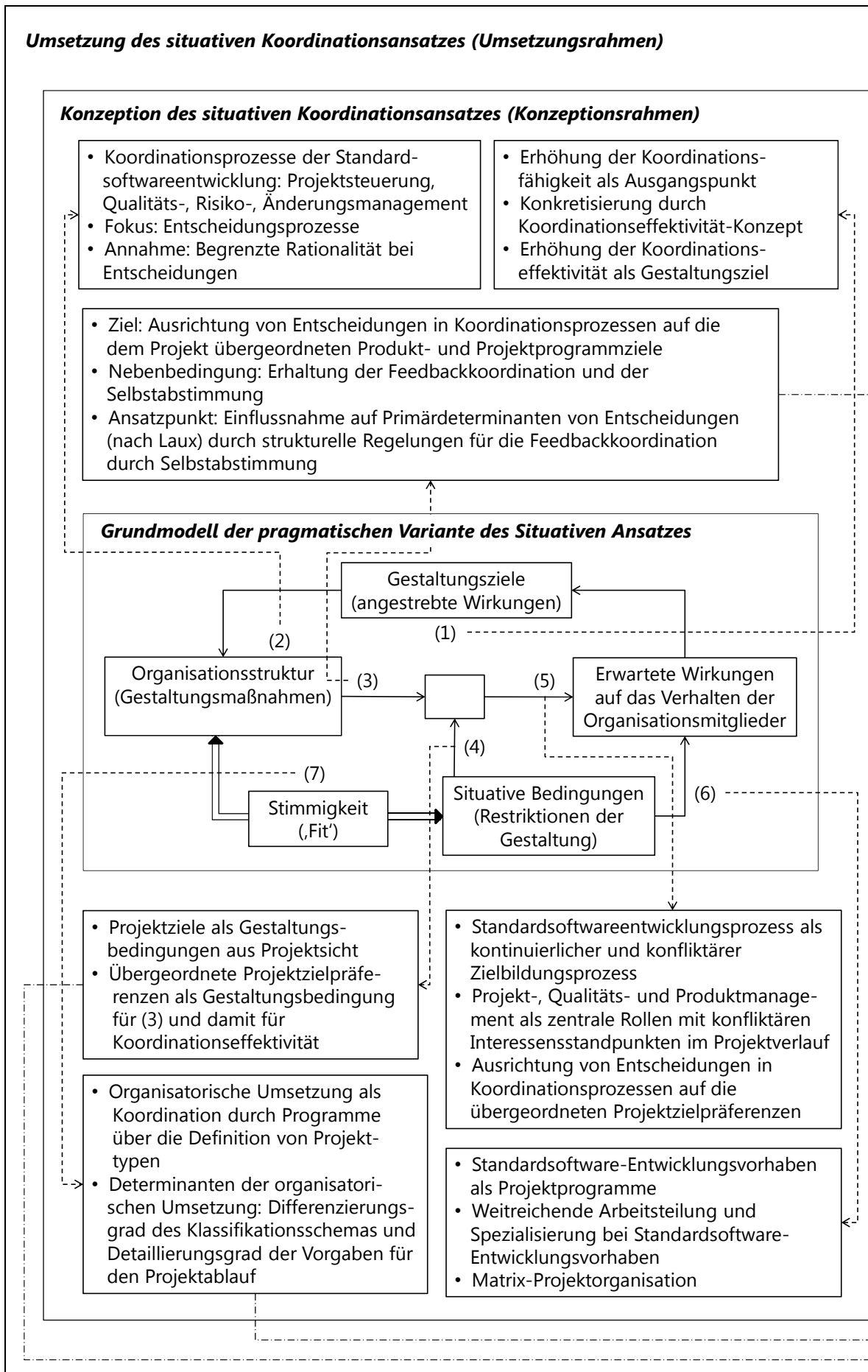


Abb. 87: Umsetzung des situativen Koordinationsansatzes im Überblick

(3')

- Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation
 - Regelungen zur situativen Ausgestaltung der Informationsstruktur von Arbeitsergebnissen der Koordinationsprozesse
 - Dadurch Beeinflussung der Informationsstruktur und der Menge der erwogenen Alternativen als Primärdeterminanten von Entscheidungen
- Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung
 - Regelungen zur situativen Kompetenzverteilung zwischen dem Projekt-, Qualitäts- und Produktmanagement bei Koordinationsprozessen
 - Dadurch Beeinflussung des Entscheidungsmodells als Primärdeterminante von Entscheidungen
- Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
 - Regelungen zur projektspezifischen Übernahme und Zuordnung von zentralen Projektrollen im Umfeld des Projekt-, Qualitäts- und Produktmanagements
 - Dadurch Beeinflussung der Zielfunktion als Primärdeterminante von Entscheidungen

- Differenzierungsgrad des Klassifikationsschemas
 - Wenig differenziertes Klassifikationsschema
 - Acht definierte Projekttypen zur Abbildung der möglichen übergeordneten Projektzielpräferenzen
- Detaillierungsgrad der Vorgaben für den Projektablauf
 - Wenig detaillierte Vorgaben für den Projektablauf
 - Präzise Festlegungen des Verbindlichkeitsgrades der Projektvorgaben
 - Formales Projekttypen-Tailoring im Rahmen eines Projektprogramm-Workshops und Projektstart-Workshops

(7')

- Konkrete Kriterien zur Bestimmung der übergeordneten Projektzielpräferenzen
 - Leistungsumfang: Anteil der hoch priorisierten funktionalen Anforderungen hinsichtlich des Priorisierungskriteriums ‚Schaden‘ am gesamten Projektumfang
 - Qualität: Anzahl der umzusetzenden Qualitätsanforderungen im Projekt und die dabei zu erreichenden Qualitätsstufen
 - Entwicklungszeit: Intensität bestimmter Abhängigkeitsbeziehungen zu anderen Projekten im Projektprogramm
 - Entwicklungskosten: Implizite Ableitung der Präferenz aus den drei anderen Projektzielpräferenzen

(4')

Dabei wurde insbesondere die Umsetzung der Artefaktspezifikation über Checklisten, welche in Abhängigkeit von den übergeordneten Projektzielpräferenzen gestaltet sind, diskutiert. Derartige Checklisten richten sich bei den vorgegebenen Prüfpunkten bzw. Fragen schwerpunktmäßig auf die kritische(n) Projektzieldimension(en) aus. Hierzu bieten sich jeweils zahlreiche Ansatzpunkte an, die aufgezeigt worden sind.⁷⁵¹

Die Gestaltungsmaßnahmen der Strukturdimension Kompetenzverteilung zielen auf die Verteilung von Kompetenzen auf die drei bei der Standardsoftwareentwicklung zentralen Rollen mit konfliktären Interessenstandpunkten zueinander: Projekt-, Qualitäts- und Produktmanagement müssen in Abhängigkeit von den übergeordneten Projektzielpräferenzen mit entsprechenden Kompetenzen bei den Koordinationsprozessen ausgestattet werden. Die Strukturdimension Kompetenzverteilung fasst damit alle konkreten Gestaltungsmaßnahmen zusammen, welche über die Beeinflussung des Entscheidungsmodells Koordinationseffektivität sicherstellen sollen. Hierzu wurde auf Kompetenzarten eingegangen, mit welchen sich die Kompetenzen des Qualitätsmanagements bei qualitätskritischen Projekten, des Projektmanagements bei zeit- und kostenkritischen Projekten, sowie des Produktmanagements bei leistungsumfangskritischen Projekten entsprechend dem Kongruenzprinzip jeweils in Relation zu den anderen Rollen ausgeprägter festlegen lassen.⁷⁵²

Dabei wird als eine generelle Gestaltungsmaßnahme vorgeschlagen, die Regelungen zur Kompetenzverteilung auf den Ausnahmefall i.S.d. ‚Management by Exception‘ zu beschränken. Mithilfe dieser Regelungen sollen komplexere, den Entscheidungsproblemen angemessene Eskalationsprozesse definiert werden können. Des Weiteren wird an dieser Stelle auf den Ansatzpunkt eingegangen, die Regelungen zur Kompetenzverteilung zwar (auch) auf den Normalfall zu beziehen, sich dabei aber nur auf die ‚schwachen‘ Kompetenzarten und/oder in einer Projektzieldimension besonders kritische Projekte zu beschränken.⁷⁵³

Die Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘ befasst sich mit dem Schritt der projektspezifischen Übernahme und Zuordnung von zentralen Projektrollen im Umfeld des Projekt-, Qualitäts- und Produktmanagements. Dessen Relevanz für das Gestaltungsziel Koordinationseffektivität ergibt sich aufgrund der konfliktären Interessensstandpunkte der Rollen beim kontinuierlichen Zielbildungsprozess in der Standardsoftwareentwicklung. Damit fasst die Struktur-

⁷⁵¹ Vgl. Kap. 3.2.1.2

⁷⁵² Vgl. Kap. 3.2.2.1

⁷⁵³ Vgl. Kap. 3.2.2.2

dimension ‚Personelle und organisatorische Unabhängigkeit‘ alle Gestaltungsmaßnahmen zusammen, die Koordinationseffektivität über die Beeinflussung der Zielfunktion sicherstellen sollen. Gestaltungsmaßnahmen zur personellen Unabhängigkeit gehen dabei davon aus, dass sich, in Analogie zur Kompetenzverteilung zwischen dem Projekt-, Qualitäts- und Produktmanagement, auch deren personelle Unabhängigkeit an den übergeordneten Projektzielpräferenzen orientieren muss. Aufgrund möglicher hierarchischer Über- und Unterordnungsverhältnisse ist durch die personelle Unabhängigkeit einer Rolle ihre Unabhängigkeit bei der Entscheidungsfindung insgesamt noch nicht garantiert. Diese kommt erst zustande, wenn auch die organisatorische Unabhängigkeit gegeben ist – d.h. wenn keine andere Rolle mit entgegengesetzten Interessensstandpunkten etwaige Fremdentscheidungs-, Weisungs- oder Kontrollkompetenzen gegenüber dem Entscheidungsträger ausüben kann. Hiermit befassen sich Gestaltungsmaßnahmen zur organisatorischen Unabhängigkeit der Rollen.⁷⁵⁴

Bei der kritischen Auseinandersetzung mit der Strukturdimension ‚Personelle und organisatorische Unabhängigkeit‘ wurde v.a. auf die Produktmanagement-Rolle Bezug genommen. In diesem Zusammenhang wurde die sog. ‚Product Owner‘-Rolle, die ein zentraler Bestandteil der agilen Methode Scrum ist, analysiert. Die ‚Product Owner‘-Rolle führt bei Scrum zu einer grundlegenden Veränderung des Zusammenspiels zwischen dem Produkt- und dem Projektmanagement. Diese mündet in der personell und organisatorisch sehr unabhängigen Position des Produktmanagements im gesamten Projektverlauf bei einer gleichzeitigen Aufgabe der personellen Unabhängigkeit des Projektmanagements.⁷⁵⁵

- (4') Als Weiterentwicklung zum Konzeptionsrahmen befasst sich der Entscheidungsrahmen mit der bei der Umsetzung des situativen Koordinationsansatzes zu klärenden Fragestellung, anhand welcher konkreten Kriterien die übergeordneten Projektzielpräferenzen als Gestaltungsbedingung bestimmt werden können. Dabei wird davon ausgegangen, dass bei der Standardsoftwareentwicklung Kostenzielverfehlungen eine relativ untergeordnete Rolle im Verhältnis zu Zielverfehlungen bei den anderen Projektzieldimensionen spielen.⁷⁵⁶

Für die Präferenzbestimmung der Projektzieldimension Leistungsumfang wird vorgeschlagen, sich am Anteil der hoch priorisierten funktionalen Anforderungen hinsichtlich des Priorisierungskriteriums ‚Schaden‘ am gesamten Projektumfang

⁷⁵⁴ Vgl. Kap. 3.2.3.1

⁷⁵⁵ Vgl. Kap. 3.2.3.2

⁷⁵⁶ Vgl. Kap. 3.3

auszurichten. Je höher dieser Anteil ist, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Leistungsumfang.⁷⁵⁷

Derartige Priorisierungsinformationen aus der Anforderungsanalyse sind in einem anderen Kontext entstanden und originär auch in einem anderen Kontext verwendet worden, als es beim situativen Koordinationsansatz der Fall sein soll. Daher wurde kritisch hinterfragt, unter welchen Umständen sie tatsächlich für die Ableitung der Kritikalität der Projektzieldimension Leistungsumfang herangezogen werden können.⁷⁵⁸

Bei der Standardsoftwareentwicklung macht der gewöhnlich große Umfang der Entwicklungsvorhaben eine Qualitätszielbestimmung je Teilprodukt und damit je Entwicklungsprojekt erforderlich. Denn die verschiedenen Teilbestandteile einer Standardsoftware erfordern hier verschiedene Kombinationen von Qualitätsmerkmalen. In diesem Zusammenhang wurden Qualitätsanforderungen und die dabei zu erreichenden Qualitätsstufen als Kriterium für die Präferenzbestimmung der Projektzieldimension Qualität identifiziert. Dabei wird angenommen, dass je mehr Qualitätsanforderungen in einem Projekt umzusetzen sind und je höher dabei die zu erreichenden Qualitätsstufen sind, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Qualität.⁷⁵⁹

Als diskussionsbedürftig erwies sich dabei die problematische Abgrenzung der Projektzieldimensionen Qualität und Leistungsumfang. Es konnte aber aufgezeigt werden, wie sich, zumindest aus der organisatorischen Perspektive auf die Standardsoftwareentwicklung, die Projektzieldimensionen Qualität und Leistungsumfang klar voneinander trennen lassen.⁷⁶⁰

Die identifizierten Kriterien für die Präferenzbestimmung der Projektzieldimension Zeit gehen davon aus, dass je intensiver die Interdependenzen zwischen dem betrachteten Projekt und den anderen Projekten im Projektprogramm sind, desto kritischer gestaltet sich die Zielerreichung in der Projektzieldimension Zeit. Diese Annahme kann mithilfe der beiden identifizierten Kriterien ‚Integrationsrisiko‘ sowie ‚Anforderungsabhängigkeiten vom Typ Vorbedingung/Benötigt‘ konkretisiert werden. Hinsichtlich ersterem kann davon ausgegangen werden, dass je höher der Anteil von hoch priorisierten Entwicklungsgegenständen hinsichtlich des Integrationsrisikos am gesamten Projektumfang ist, desto kritischer muss bei diesem Pro-

⁷⁵⁷ Vgl. Kap. 3.3.1.1

⁷⁵⁸ Vgl. Kap. 3.3.1.2

⁷⁵⁹ Vgl. Kap. 3.3.2.1

⁷⁶⁰ Vgl. Kap. 3.3.2.2

jekt die Zielerreichung in der Projektzieldimension Zeit angesehen werden. Derartige Abhängigkeitsbeziehungen zwischen Projekten oder Entwicklungsgegenständen können noch weiter herunter gebrochen werden – bis hin zu Abhängigkeiten zwischen den einzelnen umzusetzenden Anforderungen in den Entwicklungsprojekten innerhalb eines Projektprogramms. Dabei spielen bei der Präferenzbestimmung der Projektzieldimension Zeit Anforderungsabhängigkeiten vom Typ ‚Vorbedingung/Benötigt‘ die zentrale Rolle: Je mehr derartige Anforderungsabhängigkeiten auf das Projekt zurückgehen, desto kritischer ist bei diesem Projekt die Zielerreichung in der Projektzieldimension Zeit.⁷⁶¹

In diesem Zusammenhang wurde auch kritisch hinterfragt, inwiefern Anforderungsabhängigkeiten im Rahmen der Anforderungsnachvollziehbarkeit für die Präferenzbestimmung aller vier Projektzieldimensionen herangezogen werden können. Soll das der Fall sein, so sind alle Ausprägungen des Beziehungstyps ‚Requires‘, sowie der Beziehungstyp ‚Refined to‘ in einem Entwicklungsvorhaben nachzuverfolgen. Alternativ wird die Definition eines eigenen Beziehungstyps in Zusammenhang mit der Präferenzbestimmung der Projektzieldimensionen vorgeschlagen.⁷⁶²

(7') Für die Umsetzung des situativen Koordinationsansatzes mussten die beiden im Konzeptionsrahmen aufgeworfenen Fragestellungen zum Differenzierungsgrad des Klassifikationsschemas und zum Detaillierungsgrad der Vorgaben für die definierten Projekttypen geklärt werden.⁷⁶³

Bedenken im Hinblick auf eine nicht mehr handhabbare Komplexität der Prozessvariation sind der Grund für eine wenig differenzierte Umsetzung des situativen Koordinationsansatzes anhand von acht definierten Projekttypen. Die Anzahl ergibt sich durch die Festlegung einer zweistufigen Gewichtungsskala für jede der vier Projektzieldimensionen. Ihre Kritikalität wird jeweils mit Hilfe der abgeleiteten Kriterien als normal oder hoch bewertet, wobei die bei der Standardsoftwareentwicklung untergeordnete Kostenkritikalität implizit aus den drei anderen Zieldimensionen abgeleitet wird.⁷⁶⁴

Für die Umsetzung des situativen Koordinationsansatzes wird ein geringer Detaillierungsgrad der Vorgaben für den Projektablauf vorgeschlagen. Die Vorgaben müssen lediglich sicherstellen, dass bei den vier Koordinationsprozessen Projekt-

⁷⁶¹ Vgl. Kap. 3.3.3.1

⁷⁶² Vgl. Kap. 3.3.3.2

⁷⁶³ Vgl. Kap. 3.4

⁷⁶⁴ Vgl. Kap. 3.4.1

steuerung sowie Qualitäts-, Risiko- und Änderungsmanagement alle Gestaltungsmaßnahmen in den drei Strukturdimensionen Artefaktspezifikation, Kompetenzverteilung sowie ‚Personelle und organisatorische Unabhängigkeit‘ stets auf die Kritikalität der Projektziele und damit auf die übergeordneten Projektzielpräferenzen ausgerichtet sind. Die acht definierten Projekttypen bilden somit Konfigurationen im Sinne des Konfigurationsansatzes der Organisationsgestaltung.⁷⁶⁵

Um die beiden zentralen Problemkomplexe bei der Umsetzung von Projektkategorisierungsansätzen aller Art zu umgehen, ist beim situativen Koordinationsansatz zusätzlich zu präzisen Festlegungen des Verbindlichkeitsgrades der Vorgaben ein formales und in die Projektplanungsphase institutionalisiertes Projekttypen-Tailoring vorgesehen. Als formaler Rahmen dient dabei ein Projektprogrammstart-Workshop in Verbindung mit Projektstart-Workshops. Beim Projektprogrammstart-Workshop werden die zu diesem Zeitpunkt bereits gebildeten Projekte einem der acht definierten Projekttypen zugeordnet. Anschließend werden auf Projektebene im Zuge der einzelnen Projektstart-Workshops im Projektprogramm die Vorgaben auf Projekttyp-Ebene projektspezifisch adaptiert. Der Projektstart-Workshop als formaler Rahmen für das Tailoring soll insbesondere gewährleisten, dass die Festlegungen zum Verbindlichkeitsgrad der Vorgaben auch tatsächlich umgesetzt werden.⁷⁶⁶

⁷⁶⁵ Vgl. Kap. 3.4.2.1

⁷⁶⁶ Vgl. Kap. 3.4.2.2

4 Kritische Würdigung der Arbeit

Die organisatorische Gestaltung ist der zugleich mit den größten Risiken als auch mit den größten Verbesserungspotentialen behaftete Gestaltungsbereich der Softwareentwicklung. Ein wesentlicher organisatorischer Gestaltungsbereich ist dabei die Koordination, d.h. das Ausrichten von Einzelaktivitäten in einem arbeitsteiligen System auf ein übergeordnetes Gesamtziel. Koordinationsprobleme sind einer der Hauptgründe für die vielfach zitierte Softwarekrise und kommen insbesondere bei umfangreichen Softwareentwicklungsvorhaben zum Tragen. Solche Entwicklungsvorhaben sind im Bereich von Standardsoftware, dem Gegenstandsbereich der vorliegenden Arbeit, zahlreich vorzufinden.

Nach dem Verständnis der vorliegenden Arbeit kann es einen universellen und allseits optimalen Softwareentwicklungsprozess nicht geben. In diesem Sinn bestand das Ziel dieser Arbeit in der Entwicklung eines situativen Koordinationsansatzes für die Standardsoftwareentwicklung. Dies erforderte die Beantwortung von vier Forschungsfragen, die sich aus dem Situativen Ansatz der Organisationsgestaltung ableiten und mithilfe einer sachlich-analytischen Forschungsstrategie bei der Konzeption, sowie Methoden der qualitativen empirischen Forschung bei der Umsetzung des situativen Koordinationsansatzes angegangen wurden:

- (1) Die Frage, welches konkrete Ziel bei der Gestaltung der Koordination der Standardsoftwareentwicklung zu verfolgen ist, wurde mit der Entwicklung des Koordinationseffektivität-Konzeptes beantwortet. Die Koordinationseffektivität wird dabei als ein Stimmigkeitskonstrukt definiert.
- (2) Die Frage, durch welche Gestaltungsmaßnahmen das verfolgte Gestaltungsziel herbeigeführt werden kann, wurde ausgehend von der Betrachtung des Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess beantwortet. Es wurden organisatorische Regelungen zur situativen Ausgestaltung der Informationsstruktur von Arbeitsergebnissen, zur situativen Verteilung von Entscheidungskompetenzen zwischen dem Projekt-, Qualitäts- und Produktmanagement, sowie zur situativen Festlegung der personellen und organisatorischen Unabhängigkeit des Projekt-, Qualitäts- und Produktmanagements in einem Standardsoftware-Entwicklungsvorhaben entwickelt.
- (3) Die Frage, an welchen Gestaltungsbedingungen die Koordination der Standardsoftwareentwicklung ausgerichtet werden muss, führte zu der Erkenntnis, dass aufgrund der besonderen Ziel-Mittel-Beziehungen bei der Standardsoftwareentwicklung die Projektziele und dabei insbesondere die übergeordneten Projektzielpräfe-

renzen die zentrale Gestaltungsbedingung darstellen. Die Ausgestaltung der Koordinationsprozesse muss dazu führen, dass die einzugehenden Kompromisse immer auf diesen Projektvorgaben basieren. Daher wurden an dieser Stelle konkrete Kriterien zur Bestimmung der übergeordneten Projektzielpräferenzen abgeleitet.

- (4) Die Frage, wie die notwendige Stimmigkeit zwischen den Gestaltungsbedingungen und den Gestaltungsmaßnahmen hergestellt werden kann, führte zu einem Vorschlag zur organisatorischen Umsetzung des situativen Koordinationsansatzes in einer Standardsoftwareunternehmung. Die Umsetzung soll in Form der Koordination durch Programme und über acht definierte Projekttypen erfolgen.

Der theoretische Zugang zur Koordination der Standardsoftwareentwicklung als organisatorischer Gestaltungsbereich erfolgte in der vorliegenden Arbeit über zwei Theorien der Organisationslehre. Neben dem Situativen Ansatz handelt es sich dabei um den entscheidungsprozessorientierten Ansatz der Organisationsgestaltung. Um größeren Erkenntniszuwachs zu ermöglichen, wurden bei der Entwicklung des situativen Koordinationsansatzes beide Perspektiven soweit als möglich gleichberechtigt und gleichermaßen konsequent behandelt und umgesetzt. Dies soll auch bei der kritischen Würdigung der Arbeit fortgesetzt werden, die daher im Folgenden aus beiden Perspektiven vorgenommen wird.

4.1 Kritische Würdigung aus Sicht des Situativen Ansatzes der Organisationsgestaltung

Beim Situativen Ansatz der Organisationsgestaltung handelt es sich um eine Metatheorie, mit der ‚Phänomene‘ konzeptualisiert werden können. Es wird damit ein Rahmen bereitgestellt, der beim Forschungsprozess eine handlungsleitende Funktion übernehmen kann. In diesem Sinn wurde der Situative Ansatz auch in der vorliegenden Arbeit verwendet und führte zum situativen Koordinationsansatz für die Standardsoftwareentwicklung. Bei der vielfältigen Kritik am Situativen Ansatz der Organisationsgestaltung können die zwei Stoßrichtungen ‚Fundamentalkritik‘ und ‚Detailkritik‘ unterschieden werden. Es bietet sich im Folgenden an, diese als Leitfaden für die kritische Würdigung der Arbeit aus Sicht des Situativen Ansatzes zu nehmen.

Die Fundamentalkritik⁷⁶⁷ stellt die grundlegenden Annahmen und generellen Betrachtungsweisen des Situativen Ansatzes sowie die Aussagekraft der auf seiner Grundlage gewonnenen Ergebnisse in Frage. Im Rahmen der Fundamentalkritik wird darauf hingewiesen, dass⁷⁶⁸

⁷⁶⁷ Teilweise auch als konzeptionelle, ansatzimmanente oder exogene Kritik bezeichnet.

⁷⁶⁸ Vgl. Kieser, Walgenbach (2003), S. 44f., 470

- die Gestaltungsbedingungen nicht zwingend die Organisationsstruktur determinieren,
- es nicht nur jeweils eine kongruente Organisationsstruktur für bestimmte Gestaltungsbedingungen gibt, sondern dass Entscheidungen zur Organisationsgestaltung bestenfalls begrenzte Rationalität zugrundeliegt,
- der Situative Ansatz kein Konzept enthält, welches die Anpassung der Organisationsstruktur an die Gestaltungsbedingungen erklärt, sowie dass
- der Situative Ansatz die Ausübung von Herrschaft in Organisationen verschleiert und Organisationsstrukturen zudem als unabhängig von Wahrnehmungen, Intentionen und Handlungen der Organisationsmitglieder erfasst.⁷⁶⁹

Ein wesentlicher Punkt der Fundamentalkritik richtet sich darauf, dass der Situative Ansatz Gestaltungsbedingungen als gegeben und nicht beeinflussbar aus Sicht der Unternehmung betrachtet. Dies wird in Frage gestellt mit der Begründung, dass eine Unternehmung, zumindest in gewissen Grenzen, auch die Gestaltungsbedingungen verändern und gestalten kann.⁷⁷⁰

Auf diesen Punkt wurde bei der Ableitung der übergeordneten Projektzielpräferenzen als Gestaltungsbedingung für den situativen Koordinationsansatz eingegangen.⁷⁷¹ Demnach stellen die Ergebnisse strategischer Entscheidungen, bzw. die Zielformulierung auf den Ebenen des strategischen Projektmanagements und der Projektprogrammplanung, aus Sicht (der Gestaltung) von einzelnen Entwicklungsprojekten innerhalb eines Projektprogramms tatsächlich nicht beeinflussbare Gestaltungsbedingungen dar. Denn es sind übergeordnete, den Entwicklungsprojekten vorgegebene Projektziele, zu welchen die einzelnen Entwicklungsprojekte in einer Ziel-Mittel-Beziehung stehen. Daher muss auf der Projektebene von einer einseitigen Anpassung der Koordinationsmaßnahmen als Gestaltungsmaßnahmen an die übergeordneten Projektzielpräferenzen als Gestaltungsbedingungen ausgegangen werden. Dem ist nicht so

⁷⁶⁹ In der Literatur werden neben den voranstehenden fundamentalen Kritikpunkten und den nachfolgenden Punkten der Detailkritik weitere Kritikpunkte aufgeführt (sowohl Fundamental- als auch Detailkritik). Diese sind jedoch auf den situativen Koordinationsansatz nicht anwendbar, da sie sich entweder auf die analytische und nicht auf die pragmatische Variante des Situativen Ansatzes beziehen, von der Anwendung quantitativer Methoden der empirisch vergleichenden Organisationsforschung ausgehen, oder auf den Gegenstandsbereich dieser Arbeit (Koordination der Standardsoftwareentwicklung) nicht anwendbar sind. Es handelt sich dabei um Kritik bezüglich der verwendeten empirischen Maße und statistischen Verfahren, der mangelnden Repräsentativität und Vergleichbarkeit von Stichproben, der Propagierung einer konservativen Organisationsgestaltung, sowie der Vernachlässigung von kulturellen Unterschieden bei der Betrachtung der Regelmäßigkeiten in den Beziehungen zwischen Situation und Organisationsstruktur. Vgl. Kieser, Walgenbach (2003), S. 471; vgl. Jenner (2001), S. 82; vgl. Kieser (2006), S. 236ff.

⁷⁷⁰ Vgl. Kieser (2006), S. 233f.

⁷⁷¹ Vgl. Kap. 2.3.3.1

auf der Projektprogramm-Ebene, die allerdings nicht Gegenstand der vorliegenden Arbeit ist. Durch eine andere Aufteilung der Gesamtaufgabe in Teilprojekte, eine Veränderung bzw. Minimierung der Abhängigkeiten zwischen den Projekten, oder auch durch entsprechende Veränderungen an der Softwarearchitektur könnte hier Einfluss auf die Projektziele genommen werden.

Eine weitere fundamentale Kritik bemängelt, dass es entgegen den Annahmen des situativen Ansatzes nicht nur jeweils eine kongruente Organisationsstruktur für eine bestimmte Konstellation der Gestaltungsbedingungen gibt. Entscheidungen zur Organisationsgestaltung liegt vielmehr begrenzte Rationalität zugrunde, sodass eine Identifikation der optimalen Organisationsstruktur nicht möglich ist. Stattdessen muss von der Existenz unterschiedlicher Organisationsstrukturen, die für bestimmte Gestaltungssituationen eine gleich gute Lösung bieten, ausgegangen werden.⁷⁷²

Diese Kritik trifft auf den situativen Koordinationsansatz nicht zu. Im Gegenteil, durch die Zugrundelegung des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung als einer von zwei theoretischen Zugängen zu der Problemstellung wird hier die Ansicht der Kritiker explizit vertreten. Die heuristisch angelegten Konzeptionen der organisatorischen Gestaltung, wie u.a. der entscheidungsprozessorientierte Ansatz, erkennen die theoretischen und praktischen Grenzen einer zielführenden Organisationsgestaltung an und erheben daher auch nicht den (nicht einlösbaren) Anspruch einer optimalen Lösung für Organisationsprobleme.⁷⁷³ „Ihr Anliegen besteht vielmehr darin, auf Basis des jeweils verfügbaren organisatorischen Wissens Problemlösungen zu erarbeiten bzw. zu unterstützen, die im Rahmen des Möglichen vernünftig und vorteilhaft erscheinen.“⁷⁷⁴ Diese Auffassung und insbesondere das Konzept der begrenzten Rationalität⁷⁷⁵ bildeten den Ausgangspunkt des situativen Koordinationsansatzes und beeinflussten seine Konzeption und Umsetzung entscheidend.

Auch bei der nachfolgenden Fundamentalkritik am Situativen Ansatz zeigt sich die Vorteilhaftigkeit der Kombination des Situativen mit dem entscheidungsprozessorientierten Ansatz der Organisationsgestaltung. Hierdurch werden beim situativen Koordinationsansatz Organisationsstrukturen nicht als ausschließlich funktionale Erfordernisse der Situation interpretiert und dadurch Aspekte wie Wahrnehmungen, Intentionen und Machtausübung der Organisationsmitglieder vernachlässigt.⁷⁷⁶ Stattdessen werden Unternehmungen als Systeme betrachtet, in welchen Entscheidungen getroffen und

⁷⁷² Vgl. Kieser (2006), S. 233

⁷⁷³ Vgl. von Werder (1994), Sp. 1091

⁷⁷⁴ Von Werder (1994), Sp. 1091

⁷⁷⁵ Vgl. auch Kap. 2.2.1

⁷⁷⁶ Vgl. zu der Fundamentalkritik Kieser (2006), S. 236f.

koordiniert werden müssen.⁷⁷⁷ In diesem Zusammenhang wird der Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess angesehen.⁷⁷⁸ Entsprechend zielen die Gestaltungsmaßnahmen des situativen Koordinationsansatzes darauf ab, die Entscheidungsprozesse zielführend zu beeinflussen.⁷⁷⁹

Allerdings betrachtet der situative Koordinationsansatz das Motivations- als das dem Koordinationsproblem untergeordnete Problem, was zum weiteren Forschungsbedarf im Sinne der o.g. Fundamentalkritik führt. So sollten in weiteren Forschungsarbeiten Gestaltungsmaßnahmen in der Motivationsdimension entwickelt werden, welche den situativen Koordinationsansatz unterstützen. Auf der anderen Seite sind weitere Forschungsarbeiten denkbar, welche das Motivations- als das dem Koordinationsproblem gleichgestellte Problem ansehen. Aus dieser Sicht wäre es notwendig, die möglichen Wechselwirkungen zwischen den Gestaltungsmaßnahmen in der Motivations- und der Koordinationsdimension zu untersuchen und gegebenenfalls Anpassungen am situativen Koordinationsansatz vorzunehmen.

Nach Ansicht der Kritiker wird beim Situativen Ansatz nicht hinreichend untersucht und angegeben, wie Organisationsgestalter Unstimmigkeiten zwischen der Organisationsstruktur und den Gestaltungsbedingungen erkennen sollen. Die Fundamentalkritik lautet hier mit anderen Worten, dass der Situative Ansatz kein Konzept enthält, das die Anpassung der Organisationsstruktur an die Situation erklärt.⁷⁸⁰ Diese Fragestellung ist beim situativen Koordinationsansatz der Ableitung des Gestaltungsziels (Forschungsfrage 1), der Gestaltungsmaßnahmen (Forschungsfrage 2) und der Gestaltungsbedingungen (Forschungsfrage 3) gleichgestellt und bildet die vierte Forschungsfrage der Arbeit. Denn, wie die Kritiker zu Recht bemängeln, beantworten die drei erstgenannten, unmittelbar aus dem Situativen Ansatz abgeleiteten Forschungsfragen nicht, wie die notwendige Stimmigkeit zwischen den Gestaltungsbedingungen und der Struktur (Gestaltungsmaßnahmen) hergestellt werden kann. Hierauf zielt explizit die vierte Forschungsfrage der Arbeit hinsichtlich der organisatorischen Umsetzung des situativen Koordinationsansatzes in einer Standardsoftwareunternehmung ab.⁷⁸¹

Die zweite Stoßrichtung der Kritik am Situativen Ansatz der Organisationsgestaltung ist die Detailkritik⁷⁸², welche mit der grundsätzlichen Betrachtungsweise und dem generellen methodischen Vorgehen des Situativen Ansatzes zwar einverstanden ist, je-

⁷⁷⁷ Vgl. Kap. 2.2.1

⁷⁷⁸ Vgl. Kap. 2.3.2.1

⁷⁷⁹ Vgl. Kap. 2.3.2 und Kap. 3.2

⁷⁸⁰ Vgl. Kieser (2006), S. 235

⁷⁸¹ Vgl. Kap. 2.3.4 und Kap. 3.4

⁷⁸² Teilweise auch als methodische oder immanente Kritik bezeichnet.

doch Mängel im Detail herausstellt. Diese Mängel sind also kein generelles Charakteristikum des Situativen Ansatzes. So wird bemängelt, dass

- wichtige Gestaltungsbedingungen oder -maßnahmen nicht erfasst werden,
- die Vernachlässigung von möglichen Interdependenzen zwischen den Gestaltungsbedingungen zu einer Verzerrung der Zusammenhänge führt, sowie dass
- der Informationsgehalt der (empirisch gestützten) Ergebnisse des Situativen Ansatzes gering ist.

Obwohl es gerade darauf ankomme, alle infrage kommenden Faktoren gleichzeitig in Ansatz zu bringen, werden bei zahlreichen Ansätzen und Untersuchungen wichtige Gestaltungsbedingungen oder -maßnahmen nicht erfasst. Die Kritik wird insofern relativiert, als dass anerkannt wird, dass einer solchen Vollständigkeit nicht nur Auffassungsunterschiede bei der Konzeptualisierung, sondern auch forschungstechnische Probleme entgegen stehen.⁷⁸³

Nach Ansicht des Autors muss dieser Kritikpunkt noch weiter relativiert werden. Es kommt nicht darauf an, alle möglichen Gestaltungsbedingungen und -maßnahmen zu erfassen. Dies wäre nicht zielführend, denn eine Forschungsarbeit muss sich stets auf einen Ausschnitt der Realität beschränken. Es geht bei einer Forschungsarbeit, die den Situativen Ansatz der Organisationsgestaltung verfolgt, vielmehr darum, die und auch nur die für das Gestaltungsziel der Arbeit bzw. des zu entwickelnden situativen Ansatzes relevanten Gestaltungsbedingungen und -maßnahmen zu erfassen. Daher hatte das Gestaltungsziel bei der Entwicklung des situativen Koordinationsansatzes eine den übrigen Größen vorgelagerte Stellung. Die anschließend abgeleiteten Gestaltungsmaßnahmen zielen darauf ab, ebendieses verfolgte Gestaltungsziel, hier Koordinationseffektivität, herbeizuführen. Ebenso konsequent im Hinblick auf das Gestaltungsziel ist nach den Gestaltungsbedingungen zu fragen. Somit sind für die vorliegende Arbeit genau die Gestaltungsbedingungen relevant, welche auf die (Gestaltung der) Koordination der Standardsoftwareentwicklung einwirken und an welchen sich diese ausrichten muss, um stets Koordinationseffektivität zu erreichen.

Die zweite hier anzusprechende Detailkritik bezieht sich auf die Vernachlässigung von möglichen Interdependenzen zwischen den Gestaltungsbedingungen. Diese können zu einer Verzerrung der Zusammenhänge und damit zu differierenden Empfehlungen führen. Viele organisatorische Fragestellungen sind aber gerade durch die hohe Komplexität und Vielzahl von Gestaltungsbedingungen gekennzeichnet. Fokussierung und Abgrenzung des Erkenntnisbereiches führe hier dazu, dass nicht die komplette Varia-

⁷⁸³ Vgl. Kieser, Walgenbach (2003), S. 470f.

bilität erklärt werden kann.⁷⁸⁴ Dieser Kritikpunkt trifft auch auf die pragmatische Variante des Situativen Ansatzes und damit auf den situativen Koordinationsansatz zu, wenngleich hier der Stellenwert der Kritik wesentlich geringer ist als bei den Ansätzen der analytischen Variante. Tatsächlich stellten auch beim situativen Koordinationsansatz Interdependenzen zwischen den Gestaltungsbedingungen eine nicht zu vernachlässigende Problemstellung dar. Darauf wurde allerdings ausführlich eingegangen und es wurden entsprechende Lösungsansätze erarbeitet, so bei der Abgrenzungsproblematik zwischen den Projektzieldimensionen Qualität und Leistungsumfang,⁷⁸⁵ sowie bei der Diskussion von Anforderungsabhängigkeiten als Grundlage für die Bestimmung von Projektzielpräferenzen.⁷⁸⁶

Die schwerwiegendste Detailkritik am Situativen Ansatz bezeichnet den Informationsgehalt seiner (empirisch gestützten) Ergebnisse als gering. Eine ‚typische‘ Ergebnisformulierung könne hier in etwa lauten, dass Unternehmungen mit einem hoch diversifizierten Produktionsprogramm mehr Gebrauch von dem Koordinationsinstrument Planung machen als Organisationen mit einem wenig diversifizierten Produktionsprogramm. Welche Arten von Plänen auf welche Weise dabei erstellt werden (sollen), bleibt unbeantwortet. Ein solcher Verzicht auf Informationen über den Inhalt der strukturellen Regelungen ist der Preis für die größere Allgemeingültigkeit der Aussagen. Diese wird beim Situativen Ansatz hergestellt, indem von wesentlichen inhaltlichen Details abstrahiert wird und stattdessen nur globale und abstrakte Beziehungen aufgezeigt werden. Entsprechend gering ist die Unterstützung, die der Situative Ansatz der bei der Organisationsgestaltung bietet.⁷⁸⁷

Die Kritik hinsichtlich des geringen Informationsgehalts von Ergebnissen trifft auf den situativen Koordinationsansatz nicht zu. Denn wie die Kritiker bemerken, ist „der geringe Informationsgehalt bisheriger Ergebnisse des Situativen Ansatzes .. kein irreparabler Mangel. ... Großzahlige Analysen können durch intensivere Fallstudien ergänzt werden.“⁷⁸⁸ In dieser Arbeit wurde aufgrund der verfolgten Explorationsstrategie auf eine großzahlige Untersuchung ganz verzichtet und stattdessen mit der Einzelfallstudie und in ihrem Rahmen mit der Dokumentenanalyse und den Experteninterviews auf Methoden der qualitativen empirischen Forschung zurückgegriffen. Auf diese Weise war es möglich, zu einer Umsetzung des situativen Koordinationsansatzes zu gelangen, welche nicht von den Inhalten möglicher struktureller Regelungen abstrahiert,

⁷⁸⁴ Vgl. Jenner (2001), S. 81-83; vgl. Sauerwald (2007), S. 74-77

⁷⁸⁵ Vgl. Kap. 3.3.2.2

⁷⁸⁶ Vgl. Kap. 3.3.3.2

⁷⁸⁷ Vgl. Kieser (2006), S. 232

⁷⁸⁸ Kieser (2006), S. 232f.

sondern im Gegenteil konkrete Gestaltungsmaßnahmen für die jeweiligen Gestaltungssituationen beinhaltet.⁷⁸⁹

Auf der anderen Seite verbleibt durch die in dieser Arbeit gewählte qualitativempirische Forschungsstrategie weiterer Forschungsbedarf. So sind weitere Fallstudien notwendig, die sich mit dem praktischen Einsatz des hier entwickelten situativen Koordinationsansatzes befassen. Des Weiteren sind mit den Ergebnissen der vorliegenden Arbeit die Grundlagen für weitere, darauf aufbauende und eine quantitativempirische Forschungsstrategie verfolgende, Forschungsarbeiten geschaffen.

4.2 Kritische Würdigung aus Sicht des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung

Anders als beim Situativen Ansatz mit den beiden Stoßrichtungen der Fundamental- und Detailkritik hat sich beim entscheidungsprozessorientierten Ansatz der Organisationsgestaltung kein derartiger ‚Kritikkonsens‘ herausgebildet. Die Gründe dafür sind nicht in einem Mangel an Kritik zu sehen. Der entscheidungsprozessorientierte Ansatz wird ebenso vielfältig kritisch hinterfragt wie der Situative Ansatz. Doch im Unterschied zu Letzterem stellt sich die dem entscheidungsprozessorientierten Ansatz zugrundeliegende Theorie – die Theorie organisatorischer Entscheidungen, nicht als ein relativ einheitlicher Block dar. Es haben sich unterschiedliche Strömungen entwickelt, die wegen unterschiedlicher Annahmen über den Entscheidungsprozess in Organisationen jeweils verschiedene Aspekte organisationaler Entscheidungen in den Vordergrund stellen. Vereinfachend können dabei drei Hauptmodelle unterscheiden werden: das Modell der organisatorischen Differenzierung⁷⁹⁰, des politischen Prozesses und der organisierten Anarchie.⁷⁹¹ Diese drei Modelle organisatorischer Entscheidungen sollen für die kritische Würdigung der Arbeit aus Sicht des entscheidungsprozessorientierten Ansatzes diskussionsleitend sein.

Der situative Koordinationsansatz orientiert sich am Modell der organisatorischen Differenzierung. Dieses Modell rückt das Merkmal des Organisiertseins in den Vordergrund der Betrachtung. Organisationen stellen sich aus dieser Sicht in erster Linie als komplexe Systeme dar - bestehend aus einer im Zuge der Arbeitsteilung und Spezialisierung ausdifferenzierten Aufbau- und Ablauforganisation, einem hierarchischen Koordinationssystem, sowie immer auch informalen Regeln, die neben den formalen Strukturen existieren. Die in Organisationen ablaufenden Entscheidungsprozesse werden bei Zugrundelegung des Differenzierungsmodells nicht organisationslos, sondern

⁷⁸⁹ Vgl. Kap. 3.2

⁷⁹⁰ Im Folgenden vereinfachend auch als Differenzierungsmodell bezeichnet.

⁷⁹¹ Vgl. Schreyögg (1992), Sp. 1748

immer vor dem Hintergrund der Unternehmensorganisation als das ordnungsstiftende Regelsystem analysiert. Es wird also angenommen, dass Organisationsmitglieder in ihren Entscheidungsverläufen und den dazu notwendigen Vorbereitungen von der Unternehmensorganisation beeinflusst werden.⁷⁹²

Die in Kap. 2.2 erfolgte Darstellung der Standardsoftwareentwicklung aus Sicht des entscheidungsprozessorientierten Ansatzes orientiert sich konsequent am Modell der organisatorischen Differenzierung. Die Spezialisierung und strukturelle Eingliederung von Entwicklungsprojekten in die Aufbauorganisation einer Standardsoftwareunternehmung sowie die Arbeitsteilung und Spezialisierung innerhalb von Entwicklungsprojekten sind aus dieser Sicht die wesentlichen Elemente einer Organisation mit Einfluss auf die ablaufenden Entscheidungsprozesse. Ebenso sind die Konzeption und die Umsetzung des situativen Koordinationsansatzes⁷⁹³ vor dem Hintergrund des Differenzierungsmodells zu sehen.

Alle Koordinationsmaßnahmen des situativen Koordinationsansatzes gehen von bestimmten Koordinationsphasen und -prozessen in der Standardsoftwareentwicklung aus.⁷⁹⁴ Ein solcher übergeordneter organisatorischer Rahmen ist unverzichtbar für die Institutionalisierung von Koordinationsprozessen⁷⁹⁵ als Ansatzpunkt für konkrete und in der Praxis umsetzbare Koordinationsmaßnahmen. Derartige, auf organisatorischen Regelungen beruhende Maßnahmen werden als strukturelle Koordinationsmaßnahmen bezeichnet und bilden schlussendlich selber einen Bestandteil der formalen Organisationsstruktur.⁷⁹⁶ Gerade das Fehlen solcher (struktureller) Maßnahmen, mit welchen der Koordinationsbedarf unter Berücksichtigung situativer Faktoren und real existierender Organisationsstrukturen gedeckt werden kann, wird in der Forschung zur Koordination von Innovationsprojekten im Allgemeinen als ein Defizit angesehen.⁷⁹⁷

Eine Forschungsarbeit muss sich stets auf einen Ausschnitt der Realität beschränken. Entsprechend beansprucht der situative Koordinationsansatz nicht die Lösung des gesamten Koordinationsproblems bei der Standardsoftwareentwicklung, sondern zielt auf die Erhöhung der Koordinationseffektivität ab. Ausgehend von der Konzeptualisierung der Koordinationseffektivität als Gestaltungsziel⁷⁹⁸ wurden die hierfür relevanten Koordinationsobjekte abgegrenzt. Um Koordinationseffektivität zu erhöhen, adressiert der situative Koordinationsansatz demnach die Ausrichtung von Entscheidungen in

⁷⁹² Vgl. Schreyögg (1992), Sp. 1748

⁷⁹³ Vgl. Kap. 2.3 und Kap. 3

⁷⁹⁴ Vgl. Kap. 2.3.2.2

⁷⁹⁵ Vgl. Kap. 2.3.2.3

⁷⁹⁶ Vgl. Kieser, Kubicek (1992), S. 117f.

⁷⁹⁷ Vgl. Lühring (2006), S. 96

⁷⁹⁸ Vgl. Kap. 2.3.1

den Koordinationsprozessen Projektsteuerung, Qualitäts-, Änderungs- und Risikomanagement auf die übergeordneten Projektzielpräferenzen und damit auf die übergeordneten Projektprogramm- und Produktziele. Andere, d.h. in der vorliegenden Arbeit nicht tangierte Koordinationsobjekte und -ebenen liegen bspw. innerhalb der primären fachlich-technischen Prozesse der Softwareentwicklung. V.a. die Koordination von Entwicklungsaufgaben und -teams sei an dieser Stelle genannt.⁷⁹⁹ Diese notwendige, präzise Abgrenzung des Gegenstands der Arbeit von den zahlreichen weiteren Koordinationsebenen und -objekten wird wesentlich durch die Zugrundelegung des Differenzierungsmodells begünstigt.

Ein weiterer zentraler Aspekt des situativen Koordinationsansatzes, welcher vor dem Hintergrund des Differenzierungsmodells zu bewerten ist, ist die Erhöhung von Koordinationseffektivität über die Einflussnahme auf Primärdeterminanten von Entscheidungen der zentralen Projektrollen.⁸⁰⁰ Alle abgeleiteten Koordinationsmaßnahmen zielen schlussendlich darauf ab, hierüber auf die eigentlichen Entscheidungen in Koordinationsprozessen Einfluss zu nehmen.⁸⁰¹ Ein solcher Ansatz folgt der Vorstellung, dass die eigentlichen Entscheidungen begrenzt rational fallen, d.h. dass Organisationsentscheidungen zwar nicht optimierbar sind, aber bei diesen durchaus noch der entscheidungslogischen Vorstellung folgend bewusst zwischen Alternativen gewählt wird.⁸⁰² Für die Entscheidungsträger besteht dabei ein Handlungs- und Entscheidungsspielraum, der v.a. von der Organisationsstruktur beeinflusst wird. Insbesondere das Modell der organisierten Anarchie, aber auch das Modell des politischen Prozesses hat sich zu sehr vom präskriptiven Modell der Entscheidung entfernt, als dass man bei ihrer Zugrundelegung diese Basisannahme vertreten könnte.⁸⁰³

Das Modell des politischen Prozesses setzt bei der Betrachtung von Entscheidungen in Organisationen zwar wie das Differenzierungsmodell an der Arbeitsteilung und Spezialisierung an, nimmt dabei aber eine ganz spezielle Perspektive ein, nämlich die auf organisatorische Aspekte der Macht. „Es sieht organisatorische Entscheidungen als das Resultat äußerst verwickelter, auf mehreren Ebenen ausgetragener ‚politischer Spiele‘ an.“⁸⁰⁴ Den übergeordneten organisatorischen Rahmen bildet hierfür die Organisationsstruktur. Es wird angenommen, dass die Interessensposition, die ein ‚Spieler‘ bzw.

⁷⁹⁹ Vgl. bspw. Kraut, Streeter (1995), S. 71ff.

⁸⁰⁰ Beeinflussung des Entscheidungsmodelltyps, der Menge der erwogenen Handlungsalternativen, der Informationsstruktur, der Prognosefunktion, der Zielfunktion sowie der Ergebnisse der Alternativen; vgl. Kap 2.3.2.3.

⁸⁰¹ Vgl. einleitend in Kap 3.2

⁸⁰² Vgl. Bea, Göbel (2006), S. 136

⁸⁰³ Vgl. Schreyögg (1992), Sp. 1752; vgl. Wolf (2008), S. 253

⁸⁰⁴ Schreyögg (1992), Sp. 1751

Entscheidungsträger im Entscheidungsprozess vertritt, wesentlich durch seine Zugehörigkeit zu einem organisatorischen Subsystem geprägt wird. Diese Zugehörigkeit legt schlussendlich die Erwartungen, Verpflichtungen, Verhaltensmuster und Entscheidungskompetenzen der ‚Spieler‘, sowie ihre Aufmerksamkeit auf bestimmte Probleme, Umweltausschnitte und Informationen fest. In diesem organisatorischen Rahmen umgreift ‚politisches Verhalten‘ in Entscheidungsprozessen die gezielte Mobilisierung und den kalkulierten Einsatz von Macht. Die potentielle Durchsetzbarkeit von Präferenzen und Interessenspositionen gründet sich beim Modell des politischen Prozesses auf verschiedenen Machtquellen, die in mehr oder minder konfliktären Entscheidungsprozessen eingesetzt werden.⁸⁰⁵

Zusammengefasst charakterisieren gemeinsame Basisannahmen, aber verschiedene Perspektiven auf Entscheidungsprozesse den Zusammenhang zwischen den Modellen der organisatorischen Differenzierung und des politischen Prozesses. Dieser Zusammenhang spiegelt sich auch in der vorliegenden Arbeit wieder: Beim situativen Koordinationsansatz bildet die Organisationsstruktur einer Standardsoftwareunternehmung den übergeordneten Rahmen, wobei die Interessenspositionen der zentralen Projektrollen durch ihre Zugehörigkeit zu den Subsystemen Qualitäts-, Produkt-, und Projektmanagement bestimmt werden. Beim Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess wird offensichtlich von konfliktären Entscheidungsprozessen ausgegangen.⁸⁰⁶ Diesen gemeinsamen Basisannahmen stehen dagegen Unterschiede in der eingenommenen Perspektive auf Entscheidungsprozesse gegenüber. Zwar werden die Steuerungsmaßnahmen in der Strukturdimension Kompetenzverteilung und die Steuerungsmaßnahmen hinsichtlich organisatorischer Unabhängigkeit teilweise mithilfe des Macht-Konzeptes begründet,⁸⁰⁷ doch die organisatorischen Aspekte der Macht bestimmen nicht durchgängig die Perspektive der Arbeit. Dies führt dazu, dass aus Sicht des Modells des politischen Prozesses weiterer Forschungsbedarf zum situativen Koordinationsansatz besteht, allen voran hinsichtlich seiner organisatorischen Umsetzung.

Als formaler Rahmen für die organisatorische Umsetzung des situativen Koordinationsansatzes⁸⁰⁸ dient ein Projektprogrammstart-Workshop, bei welchem die zu diesem Zeitpunkt bereits gebildeten Projekte anhand festgelegter Kriterien⁸⁰⁹ einem der acht definierten Projekttypen zugeordnet werden. Im zweiten Schritt werden im Zuge der

⁸⁰⁵ Vgl. Schreyögg (1992), Sp. 1751f.; vgl. Scholl (1992), Sp. 1995

⁸⁰⁶ Vgl. Kap. 2.3.2.1

⁸⁰⁷ Vgl. Kap. 3.2.2 und Kap. 3.2.3

⁸⁰⁸ Vgl. Kap. 3.4

⁸⁰⁹ Vgl. Kap. 3.3

einzelnen Projektstart-Workshops die Vorgaben auf Projekttyp-Ebene projektspezifisch adaptiert. Dieser formale Rahmen soll insbesondere die Einhaltung der Festlegungen zum Verbindlichkeitsgrad der Vorgaben gewährleisten. Weiterführende, ggf. auch fallspezifische Vorgaben für das Tailoring im Rahmen der Startworkshops stellen bei Zugrundelegung des Differenzierungsmodells Randaspekte dar, deren Festlegung im Einzelfall der Praxis überlassen werden kann. Werden organisatorische Entscheidungen jedoch nach dem Modell des politischen Prozesses betrachtet, so ist dies anders zu bewerten.

Die Zuordnung eines Projektes zu einem der definierten Projekttypen und die weitere projektspezifische Adaption der Vorgaben auf Projekttyp-Ebene stellen Vorgänge dar, durch welche Macht-, sowie mittelbar auch Ressourcenverteilungen bestimmt werden. Entsprechend der hohen Konflikthaftigkeit ist in derartigen Situationen eine weitgehende ‚Politisierung‘ der Entscheidungen zu erwarten.⁸¹⁰ Desweiteren sind durch den niedrigen Detaillierungs- und Verbindlichkeitsgrad der Vorgaben für den Projektablauf relativ große Spielräume bei dieser Entscheidungssituation gegeben. In diesem Zusammenhang wird davon ausgegangen, dass je größer die Spielräume in einer Entscheidungssituation sind, umso politischer werden unter sonst gleichen Umständen die Entscheidungsprozesse.⁸¹¹

Insgesamt lassen sich die mit der organisatorischen Umsetzung des situativen Koordinationsansatzes zusammenhängenden Entscheidungssituationen als hoch politisch charakterisieren. Obgleich das Ergebnis von politischen Entscheidungen zumeist Kompromisslösungen sind, werden sie als Nullsummenspiele behandelt – d.h. es gibt immer Gewinner und Verlierer.⁸¹² Je nach Zuordnung eines Projektes zu einem der acht definierten Projekttypen kann das mit dem Projekt betraute Projekt-, Produkt- oder Qualitätsmanagement in diesem Sinn als Gewinner oder Verlierer aus dieser Machtverteilung hervorgehen. Daher wirft die Reduzierung der Entscheidungsprozesse an dieser Stelle auf eine ‚unpolitische‘ Erfassung der übergeordneten Projektzielpräferenzen bei Zugrundelegung des Modells des politischen Prozesses weiteren Forschungsbedarf auf. Dabei muss davon ausgegangen werden, dass nicht primär die durch möglichst objektive Kriterien erfassten übergeordneten Projektzielpräferenzen,⁸¹³ sondern andere, „partikuläre Interessen die Zielvorschläge für die Organisation bestimmen, die dann in ... konfliktären Entscheidungsprozessen zu autorisierten Zielen der Organisa-

⁸¹⁰ Vgl. Scholl (1992), Sp. 1998

⁸¹¹ Vgl. Schreyögg (2008), S. 350

⁸¹² Vgl. Schreyögg (1992), Sp. 1752

⁸¹³ Vgl. Kap. 3.3

tion führen.⁸¹⁴ Das Tailoring des situativen Koordinationsansatzes muss daher um Mechanismen erweitert werden, welche die dysfunktionalen Wirkungen von politischen Prozessen, bspw. Ablenkung von organisatorischen Zielen, ausschließen.⁸¹⁵

Beim Umgang mit der ‚Politisierung‘ organisatorischer Prozesse scheiden sowohl eine volle Legitimierung und damit de facto die Aufgabe der Unternehmensorganisation als das ordnungsstiftende Regelsystem, als auch eine in der Realität nicht mögliche vollständige Unterbindung der ‚Politisierung‘ aus.⁸¹⁶ Daher kommt es darauf an, „politische Prozesse zu kanalisieren, die Organisation vor politischen Exzessen zu schützen und Alternativen aufzuzeigen.“⁸¹⁷ Politische Prozesse im Umfeld der Koordination umfangreicher Standardsoftwareentwicklung sind bei Zugrundelegung des organisatorischen Differenzierungsmodells in ihrer Gesamtheit und Komplexität jedoch nicht zugänglich. Sie stellen gewissermaßen eine ‚zweite Sphäre‘ in Organisationen dar, die sich abseits von der formalen Organisationsstruktur abspielt.⁸¹⁸ Um diese zu adressieren, muss der Koordination der Standardsoftwareentwicklung das Modell des politischen Prozesses zugrundegelegt werden. Zu untersuchen wäre dann bspw., ob und in wie weit verdeckte Machtstrategien, die Bildung von Koalitionen, sowie diverse weitere machtpolitische Handlungsalternativen⁸¹⁹ der zentralen Projektrollen zu berücksichtigen sind.

Das dritte hier zu diskutierende Hauptmodell organisatorischer Entscheidungen, das Modell der organisierten Anarchie, begreift Unternehmen als schlecht strukturierte Systeme mit deutlichen Tendenzen in Richtung einer ausgeprägten Unordnung. Die Unternehmensorganisation ist aus dieser Sicht zwischen Hierarchie und Anarchie positioniert – sie stellt eine organisierte Anarchie dar.⁸²⁰ Entscheidungen in organisierten Anarchien können nur unzureichend nach dem klassischen Entscheidungsmodell strukturiert werden. Das Modell der organisierten Anarchie schlägt daher ein alternatives Konzept organisationaler Entscheidungsprozesse vor. Es vergleicht diese mit ‚Sammelbecken‘ bzw. ‚Mülleimern‘, in welche Organisationsmitglieder unregelmäßig Probleme und Lösungen ‚hineinwerfen‘ (sog. ‚Mülleimer-Modell‘⁸²¹). Ein Entscheidungsprozess wird in diesem Sinn als das Zusammentreffen der vier relativ unabhängig voneinander fließenden ‚Ströme‘ von Problemen, Entscheidungsgelegenheiten,

⁸¹⁴ Scholl (1992), Sp. 1995

⁸¹⁵ Zur (Dys-)Funktionalität von Machtprozessen vgl. Scholl (1992), Sp. 2000

⁸¹⁶ Vgl. Schreyögg (2008), S. 360

⁸¹⁷ Schreyögg (2008), S. 360

⁸¹⁸ Vgl. Schreyögg (2008), S. 349f.

⁸¹⁹ Vgl. Scholl (1992), Sp. 1993; vgl. Schreyögg (1992), Sp. 1752

⁸²⁰ Vgl. Wolf (2008), S. 253

⁸²¹ Vgl. erstveröffentlicht in Cohen u.a. (1972)

Entscheidungsteilnehmern und Lösungen charakterisiert. Ihre Gleichzeitigkeit führt zu einem Verteilungsproblem aus Sicht der Organisationsmitglieder. Sie müssen bei einem gegebenen Zeitbudget ihre Aufmerksamkeit auf verschiedene Probleme und Ereignisse verteilen. Empirische Ergebnisse deuten dabei auf eine fluktuierende und instabile Aufmerksamkeitssituation hin, unabhängig vom Inhalt der Entscheidung.⁸²²

Der Standardsoftwareentwicklungsprozess als ein kontinuierlicher und konfliktärer Zielbildungsprozess orientiert sich am Modell der organisatorischen Differenzierung. Dabei wird zwar von einer hohen Unsicherheit und Konflikthaftigkeit bei den Entscheidungsprozessen ausgegangen, doch diese spielen sich im Rahmen einer geordneten Unternehmensorganisation ab.⁸²³ Dem Modell der organisierten Anarchie folgend ist an dieser Stelle dagegen von Entscheidungsprozessen auszugehen, die sich weitaus unorganisierter und unklarer gestalten. D.h. Entscheidungsprozessen und -situationen im Umfeld der Koordination der Standardsoftwareentwicklung müssen Merkmale von Entscheidungsprozessen und -situationen in organisierten Anarchien unterstellt werden.⁸²⁴

Bei der organisierten Anarchie wird vom beschränkten Wissen über die Umwelt und die entscheidungsrelevanten Kausalbeziehungen ausgegangen. Es sind weder eindeutige Maßnahmen bekannt, durch deren Einsatz ein gegebenes Ziel sicher erreicht werden kann, noch lassen sich Ereignisse und Ergebnisse ex post eindeutig den von der Organisation ergriffenen Maßnahmen zurechnen. Analog sind bei der Softwareentwicklung im Allgemeinen die (Interdependenz-)Beziehungen zwischen den Projektzieldimensionen noch nicht ausreichend verstanden.⁸²⁵ Es wird zudem davon ausgegangen, dass sie sich je nach Reifegrad der Prozesse in einer Softwareunternehmung anders gestalten.⁸²⁶ Kausalbeziehungen zwischen ergriffenen Maßnahmen und Projektzieldimensionen sind damit kaum herzustellen.

Der Verlauf von Entscheidungsprozessen ist bei organisierten Anarchien durch schlecht definierte, inkonsistente und instabile Ziele gekennzeichnet. Ziele können sich hier in unvorhersehbarer Weise, oder auch durch die Konfrontation mit neuem Wissen, ändern. D.h. mit neuem Wissen können auch neue oder geänderte Ziele einhergehen. Dieser Verlauf von Entscheidungsprozessen entspricht sinngemäß dem Verständnis des Standardsoftwareentwicklungsprozesses als ein kontinuierlicher und

⁸²² Vgl. Schreyögg (1992), Sp. 1753f.

⁸²³ Vgl. Kap. 2.2.2 und Kap. 2.3.2.1

⁸²⁴ Zu den Merkmalen von Entscheidungsprozessen und -situationen in organisierten Anarchien im Folgenden vgl. Berger, Bernhard-Mehlich (2006), S. 186f., mit zahlreichen weiteren Verweisen.

⁸²⁵ Vgl. Yourdon (1995), S. 81

⁸²⁶ Vgl. McConnell (1996), S. 20

konfliktärer Zielbildungsprozess. Auch hier verläuft der Zielbildungsprozess parallel zum Problemlösungsprozess. Die sich daraus ergebenden Ziele sind häufig nur unscharf formuliert und nicht konstant, sondern variieren im Zeitablauf.⁸²⁷

Die Entscheidungsträger in organisierten Anarchien sind gleichzeitig an mehreren Entscheidungsprozessen beteiligt und widmen diesen unterschiedliche Aufmerksamkeit. D.h. sowohl die Teilnehmer an Entscheidungsprozessen, als auch ihre Aufmerksamkeit für diese sind nicht stabil, sondern fluktuieren in unvorhersehbarer Weise. Durch die weite Verbreitung von mehrdimensionalen Organisationsformen und dabei insbesondere der Matrix-Projektorganisation kann bei Standardsoftwareunternehmen durchaus von der hier beschriebenen wechselnden Teilnehmer- und Aufmerksamkeits-situation bei Entscheidungsprozessen ausgegangen werden. So können zentrale Rollen bei verschiedenen Projekten gleichzeitig durch dieselben Personen wahrgenommen werden.⁸²⁸ Diese partizipieren somit nicht nur parallel an mehreren Entscheidungen in verschiedenen Koordinationsprozessen eines Projektes, sondern mehrerer Projekte gleichzeitig.

Wird bei der Koordination der Standardsoftwareentwicklung das Modell der organisierten Anarchie zugrundegelegt, so ergibt sich daraus wesentlicher Forschungsbedarf. Denn neben den bisher angesprochenen, zum situativen Koordinationsansatz konformen Merkmalen von Entscheidungsprozessen und -situationen in organisierten Anarchien gibt es auf der anderen Seite auch fundamentale Unterschiede bei den Basisannahmen. Sie resultieren daraus, dass das Modell der organisierten Anarchie die in den beiden zuvor diskutierten Modellen vorausgesetzte Ordnung der Entscheidungselemente aufgibt. Entscheidungsgelegenheiten, Probleme, Lösungen und Teilnehmer sind hier nicht eindeutig einander zugeordnet. Welche Art von Problemen bei welchen Entscheidungsgelegenheiten durch welche Lösungen von welchen Teilnehmern gelöst werden, hängt vielmehr vom jeweiligen Kontext des Entscheidungsprozesses ab.⁸²⁹

Mit der Aufgabe der festen Ordnung von Entscheidungselementen entfernt sich das Modell der organisierten Anarchie weit vom klassischen Modell des Entscheidens. Es gibt selbst die Vorstellung auf, dass bewusst zwischen Alternativen gewählt wird.⁸³⁰ Der situative Koordinationsansatz geht dagegen, dem Modell der organisatorischen Differenzierung folgend, davon aus, dass dem Projekt-, Qualitäts- und Produktmana-

⁸²⁷ Vgl. Kap. 2.3.2.1

⁸²⁸ Vgl. hierzu auch in Kap. 3.2.3

⁸²⁹ D.h. welche anderen Probleme und Entscheidungsgelegenheiten sich gleichzeitig ergeben, welche Lösungen sich dabei anbieten, wie die Teilnehmer ihre Aufmerksamkeit darauf verteilen und wie viel Zeit für Entscheidungen zur Verfügung steht. Vgl. Berger, Bernhard-Mehlich (2006), S. 187

⁸³⁰ Vgl. Schreyögg (1992), Sp. 1752; vgl. Wolf (2008), S. 253; vgl. Bea, Göbel (2006), S. 136

gement als zentrale Rollen im Projektverlauf zwar konfliktäre, jedoch eindeutige Interessensstandpunkte bzw. Präferenzen fest zugeordnet werden können. Mit der Institutionalisierung von Koordinationsprozessen als Ansatzpunkt für Koordinationsmaßnahmen, den übergeordneten Projektzielpräferenzen als Gestaltungsbedingung für Koordinationseffektivität und der organisatorischen Umsetzung des situativen Koordinationsansatzes über die Definition von Projekttypen (Koordination durch Programme) sind hier Entscheidungsgelegenheiten, Probleme, Lösungen und Teilnehmer relativ eindeutig einander zugeordnet. Die Einflussnahme auf Primärdeterminanten von Entscheidungen der o.g. zentralen Projektrollen, um Koordinationseffektivität sicherzustellen, setzt schließlich voraus, dass diese bewusst zwischen Handlungsalternativen wählen.⁸³¹

Da das Modell der organisierten Anarchie ein weitergehend neues Verständnis von Entscheidungen in Organisationen entwickelt, steht es zu wesentlichen Teilen in Konkurrenz zu den anderen beiden Hauptmodellen organisatorischer Entscheidungen. Eine gleichzeitige Zugrundelegung dieser Modelle müsste erst die durch ihre verschiedenen eingenommenen Perspektiven auf Entscheidungsprozesse begründeten Widersprüche lösen. Hierzu bedarf es zuallererst eines entsprechenden theoretischen Bezugsrahmens.⁸³² Zwei Aspekte mit Bedarf zur weiteren Forschung, die ein solcher Bezugsrahmen für die Koordination der Standardsoftwareentwicklung adressieren sollte, sind nicht-strukturelle Koordinationsmaßnahmen und dabei insbesondere die Koordination durch Unternehmenskultur, sowie organisatorisches Lernen.

Bei organisierten Anarchien wird davon ausgegangen, dass die Organisationsstruktur den Lauf der vier ‚Ströme‘ von Problemen, Entscheidungsgelegenheiten, Entscheidungsteilnehmern und Lösungen nur bis zu einem gewissen Grad ‚kanalisiert‘.⁸³³ Da strukturelle Koordinationsmaßnahmen auf organisatorischen Regelungen basieren, verlieren sie in diesem Sinn an Bedeutung zugunsten von nicht-strukturellen Koordinationsmaßnahmen. Letztere stützen sich gar nicht oder nur zu einem geringen Teil auf organisatorische Regelungen.⁸³⁴ So basiert das (nicht-strukturelle) Koordinationsinstrument der Organisationskultur darauf, dass Organisationsmitglieder zentrale handlungssteuernde Normen und Werte, die in einer Organisation vorherrschen, verinnerlichen und sich entsprechend verhalten.⁸³⁵ Koordination durch Unternehmenskultur ist

⁸³¹ Vgl. Kap. 2.3

⁸³² Vgl. Schreyögg (1992), Sp. 1755f.

⁸³³ Vgl. Schreyögg (1992), Sp. 1754

⁸³⁴ Vgl. Kieser, Kubicek (1992), S. 117f.

⁸³⁵ Vgl. Kieser, Walgenbach (2003), S. 109, 129

also vor allem dort von großer Bedeutung, wo strukturelle Vorgaben nur bedingt handlungssteuernd wirken können.⁸³⁶

Ein durch das Modell der organisierten Anarchie geleiteter Bezugsrahmen für die Koordination der Standardsoftwareentwicklung muss bei Berücksichtigung der Koordination durch Unternehmenskultur zunächst wesentliche und bisher nicht hinreichend geklärte Beziehungszusammenhänge aufschlüsseln. So lässt sich die Organisationskultur nicht ohne Weiteres von den anderen Koordinationsmaßnahmen trennen. Denn Regelungen in Form von Programmen, Plänen oder anderen strukturellen Koordinationsmaßnahmen sind immer auch Ausdruck von Werten und Einstellungen in einer Organisation.⁸³⁷ Unklar ist dabei, in wie weit die Organisationskultur und strukturelle Koordinationsmaßnahmen als funktionale Äquivalente bei der Lösung des Koordinationsproblems betrachtet werden können. Da selbst in organisierten Anarchien ‚Kultur ohne Struktur‘ kaum vorstellbar ist, kann hier von graduellen Substitutionen, keinesfalls aber von einer Vollsubstitution ausgegangen werden.⁸³⁸ Insbesondere kann Koordination durch Unternehmenskultur die situationsspezifische, entwicklungsprozessbegleitende Koordination mit Bedarf an direkter Wirkungsweise nicht ersetzen.⁸³⁹

Das Modell der organisierten Anarchie vertraut im Hinblick auf die Frage, welche Problemlösungen und Entscheidungsgelegenheiten sich langfristig durchsetzen, sehr stark auf die Kraft der Evolution. Man geht also davon aus, dass sich im Laufe der Zeit nur die erfolgsträchtigen Praktiken, Entschlüsse, Verhaltensweisen, etc. durchsetzen werden. Organisationen, die nicht in der Lage sind, solche organisatorischen Lernprozesse erfolgreich zu vollziehen, werden nicht bestehen können.⁸⁴⁰ Ganz in diesem Sinn konzentrieren sich Weiterentwicklungen des entscheidungsprozessorientierten Ansatzes der Organisationsgestaltung zum Thema organisatorisches Lernen auf die Grenzen des Erfahrungslernens und der adaptiven Rationalität von Organisationen. Das Interesse der Forschung gilt hier insbesondere möglichen Verzerrungen der Erfahrungsbasis und den Folgen dieser Defizite für organisatorisches Lernen und Umweltanpassung.⁸⁴¹

Ein mit dem organisatorischen Lernen einhergehender Strang von Weiterentwicklungen des entscheidungsprozessorientierten Ansatzes mündet im Forschungsumfeld des evolutionstheoretischen Ansatzes der Organisationsgestaltung.⁸⁴² Dieser stellt neben der Bedeutung von Kompetenzen (organisatorisches Lernen) gleichermaßen auch die

⁸³⁶ Vgl. Schreyögg (2008), S. 386; vgl. Kieser, Walgenbach (2003), S. 133

⁸³⁷ Vgl. Kieser, Walgenbach (2003), S. 135f.

⁸³⁸ Vgl. Schreyögg (2008), S. 168

⁸³⁹ Vgl. Billing (2003), S. 54

⁸⁴⁰ Vgl. Schreyögg (1992), Sp. 1754

⁸⁴¹ Vgl. Berger, Bernhard-Mehlich (2006), S. 192

⁸⁴² Vgl. Argote, Greve (2007), S. 338

Bedeutung von Prozessen des ungeplanten Wandels in einer Unternehmung für ihre Überlebensfähigkeit heraus.⁸⁴³ So geht bspw. der Koevolutionsansatz von diversen, nicht genau bestimmten Feedbackschleifen zwischen der Unternehmung und ihrer Umwelt aus. Diese und andere Annahmen führen schlussendlich dazu, dass sich Unternehmungen koevolutiv mit ihrer Unternehmensumwelt entwickeln.⁸⁴⁴

Ein konkreter Forschungsbedarf zum organisatorischen Lernen in Zusammenhang mit der Koordination der Standardsoftwareentwicklung sei an dieser Stelle beispielhaft herausgegriffen. So machen Entscheider in Organisationen tendenziell zu wenig Erfahrungen mit ‚Unbekanntem‘ und zu viele überflüssige Erfahrungen mit ‚Altbekanntem‘. Dieser systematischen Verzerrung der Erfahrungsbasis liegt eine gestörte Balance zwischen der ‚Ausbeutung‘ von bekannten Praktiken und Wissensbeständen (sog. ‚Exploitation‘) einerseits und dem Wissenserwerb durch Experimentieren mit neuen Praktiken (sog. ‚Exploration‘) auf der anderen Seite zugrunde.⁸⁴⁵

Derartige Verletzungen der Balance zwischen ‚Exploration‘ und ‚Exploitation‘ wirken sich grundsätzlich auch bei der Koordination der Standardsoftwareentwicklung dysfunktional aus. Sie können sowohl durch strukturelle, als auch durch nicht-strukturelle Koordinationsmaßnahmen begünstigt werden und so die Anpassungsfähigkeit der Softwareunternehmung an die Umwelt gefährden. So kann durchaus angenommen werden, dass die Bildung von Projekttypen bzw. die Koordination durch Programme tendenziell ‚Exploitation‘ fördert, wobei diese Wirkung durch einen niedrigen Detaillierungs- und Verbindlichkeitsgrad der Vorgaben für den Projektablauf gemindert wird.⁸⁴⁶ Dem Modell der organisierten Anarchie und der Forschung zum organisatorischen Lernen folgend ist dagegen grundsätzlich eine ambidextre Ausrichtung von Organisationen anzustreben.

⁸⁴³ Vgl. Bea, Göbel (2006), S. 174ff.

⁸⁴⁴ Vgl. Lewin, Volberda (1999), S. 526f.

⁸⁴⁵ Vgl. Berger, Bernhard-Mehlich (2006), S. 196

⁸⁴⁶ Vgl. Kap. 3.4

Literaturverzeichnis

Abdel-Hamid, T.K., Sengupta, K., Swett, K. (1999), The impact of goals on software project management: an experimental investigation, in: *MIS Quarterly*, 23, 1999, 4, S. 531-555

Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J. (2003), New Directions on Agile Methods: A Comparative Analysis, in: *IEEE Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon USA 2003, S. 244-254

Agile Alliance (Hrsg., 2001a), Manifesto for Agile Software Development, <http://www.agilemanifesto.org/>, Stand: 2001

Agile Alliance (Hrsg., 2001b), Principles behind the Agile Manifesto, <http://www.agilemanifesto.org/principles.html>, Stand: 2001

Ahn, H., Dyckhoff, H. (1997), Organisatorische Effektivität und Effizienz, in: *WiSt*, 26, 1997, 1, S. 2-6

Al-Kilidar, H., Cox, K., Kitchenham, B. (2005), The Use and Usefulness of the ISO/IEC 9126 Quality Standard, in: *IEEE Proceedings of the 2005 International Symposium on Empirical Software Engineering*, Noosa Heads, Queensland, Australia 2005, S. 126-132

Albach, H. (1967), Die Koordination der Planung im Großunternehmen, in: *Schriften des Vereins für Sozialpolitik*, Band 45: Rationale Wirtschaftspolitik und Planung in der Wirtschaft von heute, Berlin 1967, S. 332-438

Alewell, D. (2004), Arbeitsteilung und Spezialisierung, in: *Schreyögg*, von Werder (Hrsg., 2004), Sp. 37-45

Ambler, S. (1998), *Process Patterns*, Cambridge 1998

Andres, H.P., Zmud, R.W. (2002), A Contingency Approach to Software Project Coordination, in: *Journal of Management Information Systems*, 18, 2002, 3, S. 41-70

Angermeyer, G. (2005), *Projektmanagement-Lexikon*, München 2005

Argote, L., Greve, H. (2007), A Behavioral Theory of the Firm—40 Years and Counting: Introduction and Impact, in: *Organization Science*, 18, 2007, 3, S. 337-349

Arinze, B. (1991), A Contingency Model of DSS Development Methodology, in: *Journal of Management Information Systems*, 8, 1991, 1, S. 149-166

- Atteslander, P. (2003), Methoden der empirischen Sozialforschung, 10. Aufl., Berlin und New York 2003
- Aurum, A., Wohlin, C. (Hrsg., 2005), Engineering and Managing Software Requirements, Berlin und Heidelberg 2005
- Aurum, A., Wohlin, C., Porter, A. (2006), Aligning Software Project Decisions: A Case Study, in: International Journal of Software Engineering and Knowledge Engineering, 16, 2006, 6, S. 795-818
- Austin, R.D. (2001), The Effects of Time Pressure on Quality in Software Development: An Agency Model, in: Information Systems Research, 12, 2001, 2, S. 195-207
- Balzert, H. (2001), Lehrbuch der Softwaretechnik, Bd. 1, Software-Entwicklung, 2. Aufl., Heidelberg 2001
- Balzert, H. (2008), Lehrbuch der Softwaretechnik, Bd. 2, Softwaremanagement, 2. Aufl., Heidelberg 2008
- Barki, H., Rivard, S., Talbot, J. (2001), An Integrative Contingency Model of Software Project Risk Management, in: Journal of Management Information Systems, 17, 2001, 4, S. 37-69
- Bass, L., Clements, P., Kazman, R. (1998), Software Architecture in Practice, 2. Aufl., Reading, Harlow u.a. 1998
- Bea, F.X. (2004), Ziele und Zielkonflikte, in: Schreyögg, von Werder (Hrsg., 2004), Sp. 1674-1680
- Bea, F.X., Göbel, E. (2006), Organisation - Theorie und Gestaltung, 3. Aufl., Stuttgart 2006
- Bea, F.X., Scheurer, S., Hesselmann, S. (2008), Projektmanagement, Stuttgart 2008
- Beck, T. (1996), Die Projektorganisation und ihre Gestaltung, Berlin 1996
- Berander, P., Andrews, A. (2005), Requirements Prioritization, in: Aurum, Wohlin (Hrsg., 2005), S. 69-94
- Berger, U., Bernhard-Mehlich, I. (2006), Die Verhaltenswissenschaftliche Entscheidungstheorie, in: Kieser, Ebers (Hrsg., 2006), S. 169-214
- Biffel, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (Hrsg., 2006), Value-Based Software Engineering, Berlin 2006

- Biffel, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (2006), Preface, in: Biffel u.a. (Hrsg., 2006), S. IX-XIV
- Billing, F. (2003), Koordination in radikalen Innovationsvorhaben, Wiesbaden 2003
- Bitz, M. (1977), Die Strukturierung ökonomischer Entscheidungsmodelle, Wiesbaden 1977
- Bleicher, K. (1980), Kompetenz, in: Grochla, E. (Hrsg., 1980), Sp. 1056-1064
- Boehm, B. (1981), Software Engineering Economics, Englewood Cliffs, NJ 1981
- Boehm, B. (1986), A Spiral Model of Software Development and Enhancement, in: ACM SIGSOFT Software Engineering Notes, 11, 1986, 4, S. 14-24
- Boehm, B. (2006), Value-Based Software Engineering: Overview and Agenda, in: Biffel u.a. (Hrsg., 2006), S. 3-14
- Boehm, B., Brown, W., Huang, L.-G., Port, D. (2000), Inverting the Software Process: Schedule, Cost, and Quality as Independent Variables, in: IEEE Proceedings of the 22rd International Conference on Software Engineering, ICSE 2002, Orlando, USA 2002
- Boehm, B., Jain, A. (2006), An Initial Theory of Value-Based Software Engineering, in: Biffel u.a. (Hrsg., 2006), S. 15-38
- Boehm, B., Port, D., Huang, L.-G., Brown W. (2002), Using the Spiral Model and MBASE to Generate new Acquisition Process Models: SAIV, CAIV, and SCQAIV, in: CrossTalk - The Journal of Defense Software Engineering, 2002, 1, S. 20-25
- Boehm, B., Ross, R. (1989), Theory-W Software Project Management Principles and Examples, in: IEEE Transactions on Software Engineering, 15, 1989, 7, S. 902-916
- Boehm, B., Turner, R. (2004), Balancing Agility and Discipline, Boston u.a. 2004
- Bogner, A., Littig, B., Menz, W. (Hrsg., 2002), Das Experteninterview - Theorie, Methode, Anwendung, Opladen 2002
- Bogner, A., Menz, W. (2002), Das theoriegenerierende Experteninterview: Erkenntnisse, Wissensformen, Interaktion, in: Bogner u.a. (Hrsg., 2002), S. 33-70
- Bomarius, F., Komi-Sirviö, S. (Hrsg., 2005), PROFES 2005 – Proceedings of the 6th International Conference on Product Focused Software Process Improvement, Berlin 2005

- Bortz, J., Döring, N. (2006), *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*, 4. Aufl., Heidelberg 2006
- Bozheva, T., Gallo, M.E. (2005), *Framework of Agile Patterns*, in: Richardson u.a. (Hrsg., 2005), S. 4-15
- Bruhn, M., Strauss, B. (Hrsg., 2007), *Wertschöpfungsprozesse bei Dienstleistungen*, Wiesbaden 2007
- Brykczynski, B. (1999), *A Survey of Software Inspection Checklists*, in: *Software Engineering Notes*, 24, 1999, 1, S. 82-89
- Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R. (2007), *Situational Method Engineering: On the Differentiation of "Context" and "Project Type"*, in: Ralyte u.a. (Hrsg., 2007), S. 33-48
- Bühler-Niederberger, D. (1991), *Analytische Induktion*, in: Flick u.a. (Hrsg., 1991), S. 446-450
- Bünting, H.F. (1995), *Organisatorische Effektivität von Unternehmungen - Ein zielorientierter Ansatz*, Wiesbaden 1995
- Bund (Hrsg., 2008), *V-Modell XT*, 2008
- Bund (Hrsg., 2009), *Häufig gestellte Fragen zum V-Modell XT*, auf den Seiten des Bundesministeriums des Inneren, http://www.cio.bund.de/DE/IT-Methoden/V-Modell_XT/V-Modell_XT_FAQ/v-modell_xt_faq_node.html, Stand: 04.11.2009
- Buxmann, P., Diefenbach, H., Hess, T. (2008), *Die Softwareindustrie*, Berlin und Heidelberg 2008
- Buxmann, P., Hahn, M. (2010), *Individualsoftware, Markt für*, in: Kurbel u.a. (Hrsg., 2010)
- Cameron, J. (2002), *Configurable Development Processes - Keeping the focus on what is being produced*, in: *Communications of the ACM*, 45, 2002, 3, S. 72-77
- Campbell, J. P. (1977), *On the Nature of Organizational Effectiveness*, in: Goodman, Jennings (Hrsg., 1977), S. 13-55
- Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J. (2001), *An Industrial Survey of Requirements Interdependencies in Software Product Release Planning*, in: *IEEE Proceedings of the 5th International Symposium on Requirements Engineering*, 2001, S. 84-93

- Carmel, E. (1995), Cycle Time in Packaged Software Firms, in: Journal of Product Innovation Management, 12, 1995, 2, S. 110-123
- Carr, M.J., Konda, S.L., Monarch, I., Ulrich, F.C., Walker, C.F. (1993), Taxonomy-Based Risk Identification, SEI Technical Report CMU/SEI-93-TR-6, 1993
- Cockburn, A. (2000), Selecting a Project's Methodology, in: IEEE Software, 17, 2000, 4, S. 64-71
- Cockburn, A. (2005), Crystal Clear - Agile Software-Entwicklung für kleine Teams, Bonn 2005
- Cockburn, A. (2008), Using Both Incremental and Iterative Development, in: Cross-Talk: The Journal of Defense Software Engineering, 2008, S. 27-30
- Cohen, M., March, J., Olsen, J. (1972), A Garbage Can Model of Organizational Choice, in: Administrative Science Quarterly, 17, 1972, 1, S. 1-25
- Conway, M. (1968), How Do Committees Invent?, in: Datamation, 14, 1968, 4, S. 28-31
- Corsten, H., Reiß, M. (Hrsg., 2008), Betriebswirtschaftslehre: Bd. 1 Grundlagen, Internes Rechnungswesen, Externes Rechnungswesen, Beschaffung, 4. Aufl., München 2008
- Crawford, L., Hobbs, J.B., Turner, J.R. (2005), Project Categorization Systems: Aligning Capability with Strategy for Better Results, Newton Square 2005
- Crozier, M., Friedberg, E. (1993), Die Zwänge kollektiven Handelns: über Macht und Organisation, Frankfurt/Main 1993
- Cusumano, M.A., Selby, R.W. (1996), Die Microsoft Methode - Sieben Prinzipien, wie man ein Unternehmen an die Weltspitze bringt, Freiburg i. Br. 1996
- Dahlstedt, A., Persson, A. (2005), Requirements Interdependencies: State of the Art and Future, in: Aurum, Wohlin (Hrsg., 2005), S. 95-116
- Denger, C., Olsson, T. (2005), Quality Assurance in Requirements Engineering, in: Aurum, Wohlin (Hrsg., 2005), S. 163-186
- Denger, C., Shull, F. (2007), A Practical Approach for Quality-Driven Inspections, in: IEEE Software, 24, 2007, 1, S. 79-86
- DIN (Hrsg., 1997), DIN 69901-5: Projektmanagement - Projektmanagementsysteme - Teil 5: Begriffe, Berlin 1997

- DIN (Hrsg., 2001), DIN EN 61508-3: Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme - Teil 3: Anforderungen an Software, Berlin 2001
- DIN (Hrsg., 2009), DIN EN 50128: Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Software für Eisenbahnsteuerungs- und Überwachungssysteme, Berlin 2009
- Drews, G., Hillebrand, N. (2007), Lexikon der Projektmanagement-Methoden, Freiburg u.a. 2007
- Ebert, C. (2008), Systematisches Requirements Engineering und Management - Anforderungen ermitteln, spezifizieren und verwalten, 2. Aufl., Heidelberg 2008
- Emery, J. (1969), Organizational Planning and Control: Theory and Technology, London 1969
- Ewert, R., Wagenhofer, A. (2008), Interne Unternehmensrechnung, 7. Aufl., Berlin und Heidelberg 2008
- Fiedler, R. (2010), Controlling von Projekten, 5. Aufl., Wiesbaden, 2010
- Fieten, R. (1977), Die Gestaltung der Koordination betrieblicher Entscheidungssysteme, Frankfurt/Main u.a. 1977
- Fischer, J. (1989), Qualitative Ziele in der Unternehmensplanung - Konzepte zur Verbesserung betriebswirtschaftlicher Problemlösungstechniken, Berlin 1989
- Fitzgerald, B., Russo, N.L., O’Kane T. (2003), Software Development Method Tailoring at Motorola, in: Communications of the ACM, 46, 2003, 4, S. 65-70
- Flick, U. (2004), Triangulation - Eine Einführung, Wiesbaden 2004
- Flick, U. (Hrsg., 2006), Qualitative Evaluationsforschung, Hamburg 2006
- Flick, U. (2006), Interviews in der qualitativen Evaluationsforschung, in: Flick (Hrsg., 2006), S. 214-232
- Flick, U., von Kardoff, E., Keupp, H., von Rosenstiel, L., Wolff, S. (Hrsg., 1991), Handbuch Qualitative Sozialforschung, München 1991
- Flick, U., von Kardoff, E., Steinke, I. (Hrsg., 2004), Qualitative Forschung, 3. Aufl., Hamburg 2004
- Frank, U. (2006), Ein Vorschlag zur Konfiguration von Forschungsmethoden in der Wirtschaftsinformatik, in: Lehner, F., Zelewski, S. (Hrsg., 2006), S. 155-184

- Freimut, B., Hartkopf, S., Kontio, J., Kobitzsch, W. (2001), An Industrial Case Study of Implementing Software Risk Management, in: ACM SIGSOFT Proceedings of the 8th European software engineering conference, Vienna, Austria 2001, S. 277-287
- Frese, E. (2005), Grundlagen der Organisation - Entscheidungsorientiertes Konzept der Organisationsgestaltung, 9. Aufl., Wiesbaden 2005
- Frese, E. (Hrsg., 1992), Handwörterbuch der Organisation, 3. Aufl., Stuttgart 1992
- Friedrich, J., Hammerschall, U., Kuhrmann, M., Sihling, M. (2008), Das V-Modell XT, Berlin und Heidelberg 2008
- Froschauer, U., Lueger, M. (2003), Das qualitative Interview, Wien 2003
- Gerhardt, T. (1992), Strategie und Struktur in der deutschen Softwareindustrie - eine industrieökonomische Untersuchung der Unternehmensentwicklung in der Softwarebranche, München 1992
- GfK Marktforschung, Fraunhofer-Institut für Experimentelles Software Engineering IESE, Fraunhofer-Institut für Systemtechnik und Innovationsforschung ISI (Hrsg., 2000), Analyse und Evaluation der Softwareentwicklung in Deutschland - Eine Studie für das Bundesministerium für Bildung und Forschung, o.O. 2000
- Goodman, P. S., Jennings, J. M. (Hrsg., 1977), New Perspectives on Organizational Effectiveness, San Francisco u.a. 1977
- Grochla, E. (1976), Praxeologische Organisationstheorie durch sachliche und methodische Integration, in: Zeitschrift für betriebswirtschaftliche Forschung, 28, 1976, S. 617-637
- Grochla, E. (1978), Einführung in die Organisationstheorie, Stuttgart 1978
- Grochla, E. (Hrsg., 1980), Handwörterbuch der Organisation, 2. Aufl., Stuttgart 1980
- Grünbacher, P., Seyff, N. (2005), Requirements Negotiation, in: Aurum, Wohlin (Hrsg., 2005), S. 143-162
- Hamel, W., Gemünden, H.G. (Hrsg., 2001), Außergewöhnliche Entscheidungen, München 2001
- Hanssen, G.K., Westerheim, H., Bjørnson, F.O. (2005), Tailoring RUP to a Defined Project Type: A Case Study, in: Bomarius, Komi-Sirviö (Hrsg., 2005), S. 314-327
- Hauer, R. (1996), Total Quality Management in der Softwareproduktion, Frankfurt/Main u.a. 1996

- Hauschildt, J. (1977), Entscheidungsziele - Zielbildung in innovativen Entscheidungsprozessen, Tübingen 1977
- Hauschildt, J. (1980), Zielsysteme, in: Grochla (Hrsg., 1980), Sp. 2420-2430
- Hauschildt, J. (1981), „Ziel-Klarheit“ oder „kontrollierte Ziel-Unklarheit“ in Entscheidungen?, in: Witte (Hrsg., 1981), S. 305-322
- Hauschildt, J. (2004), Innovationsmanagement, 3. Aufl., München 2004
- Heindl, M., Biffel, S. (2005), A Case Study on Value-based Requirements Tracing, in Wermelinger, Gall (Hrsg., 2005), S. 60-69
- Heinrich, L., Heinzl, A., Roithmayr, F. (2007), Wirtschaftsinformatik – Einführung und Grundlegung, 3. Aufl., München und Wien 2007
- Henrich, A. (2002), Management von Softwareprojekten, München und Wien 2002
- Henselek, H.F. (1996), Das Management von Unternehmenskonfigurationen, Wiesbaden 1996
- Herzwurm, G. (2000), Kundenorientierte Softwareproduktentwicklung, Stuttgart u.a. 2000
- Herzwurm, G., Mikusz, M. (Hrsg., 2008), Industrialisierung des Software-Managements – Tagungsband zur GI-Fachtagung Software-Management 2008, Bonn 2008
- Herzwurm, G., Mikusz, M. (2010a), Qualitätsmerkmale von Software, in: Kurbel u.a. (Hrsg., 2010)
- Herzwurm, G., Mikusz, M. (2010b), Software-Qualitätsmanagement, in: Kurbel u.a. (Hrsg., 2010)
- Herzwurm, G., Pietsch, W. (2009), Management von IT-Produkten - Geschäftsmodelle, Leitlinien und Werkzeugkasten für softwareintensive Systeme und Dienstleistungen, Heidelberg 2009
- Herzwurm, G., Schockert, S., Mellis, W. (1997), Qualitätssoftware durch Kundenorientierung - Die Methode Quality Function Deployment (QFD), Braunschweig und Wiesbaden 1997
- Herzwurm, G., Stelzer, D. (2008), Wirtschaftsinformatik versus Information Systems - Eine Gegenüberstellung, Stuttgarter Schriften zur Unternehmenssoftware - Arbeitsbericht Nr. 2, Stuttgart 2008

- Hill, W., Fehlbaum, R., Ulrich, P. (1994), *Organisationslehre 1 - Ziele, Instrumente und Bedingungen der Organisation sozialer Systeme*, 5. Aufl., Bern, Stuttgart, u.a. 1994
- Huang, L., Hu, H., Ge, J., Boehm, B., Lü, J. (2006), *Taylor the Value-Based Software Quality Achievement Process to Project Business Cases*, in: Wang u.a. (Hrsg., 2006), S. 56-63
- Hull, E., Jackson, K., Dick, J. (2005), *Requirements Engineering*, 2. Aufl., London u.a. 2005
- IEEE (Hrsg., 1988), *IEEE Std 1058.1-1987: IEEE Standard for Software Project Management Plans*, New York 1988
- IEEE (Hrsg., 1998a), *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*, New York 1998
- IEEE (Hrsg., 1998b), *1233-1998, IEEE Guide for Developing System Requirements Specifications*, New York 1998
- ISO/IEC (Hrsg., 2001), *ISO/IEC 9126-1: Software engineering — Product quality — Part 1: Quality model*, Geneva 2001
- Jacobs, S. (2010), *Projektplanung, -steuerung und -kontrolle*, in: Kurbel u.a. (Hrsg., 2010)
- Jenner, T. (2001), *Zur Verwendung des Kontingenzansatzes in der betriebswirtschaftlichen Forschung*, in *WISU*, 30, 2001, 1, S. 79-84
- Jenny, B. (1995), *Projektmanagement in der Wirtschaftsinformatik*, 5. Aufl., Zürich 1995
- Jones, C. (1996), *Software estimating rules of thumb*, in: *Computer*, 29, 1996, 3, S. 116-118
- Jones, C. (2003), *Variations in Software Development Practices*, in: *IEEE Software*, 20, 2003, 6, S. 22-27
- Jönsson, P., Lindvall, M. (2005), *Impact Analysis*, in: Aurum, Wohlin (Hrsg., 2005), S. 117-142
- Jost, P.-J. (2008), *Organisation und Motivation - Eine ökonomisch-psychologische Einführung*, Wiesbaden 2008

- Jost, P.-J. (2009), *Organisation und Koordination - Eine ökonomische Einführung*, Wiesbaden 2009
- Kano, N., Nobuhiku, S., Fumio, T., Shinichi, T. (1984), Attractive quality and must-be quality, in: *Journal of the Japanese Society for Quality Control* 14, 1984, 2, S. 39-48
- Keil, M., Li, L., Mathiassen, L., Zheng, G. (2006): The Influence of Checklists and Roles on Software Practitioner Risk Perception and Decision-Making, in: *IEEE Proceedings of the 39th Hawaii International Conference on System Sciences*, S. 1-12
- Kerzner, H. (2003), *Projektmanagement - Ein systemorientierter Ansatz zur Planung und Steuerung*, Bonn 2003
- Kieser, A. (2006), Der Situative Ansatz, in: Kieser, Ebers (Hrsg., 2006), S. 215-246
- Kieser, A., Ebers, M. (Hrsg., 2006), *Organisationstheorien*, 6. Aufl., Stuttgart 2006
- Kieser, A., Kubicek, A. (1992), *Organisation*, 3. Aufl., Stuttgart 1992
- Kieser, A., Walgenbach, P. (2003), *Organisation*, 4. Aufl., Stuttgart 2003
- Kittlaus, H.-B., Rau, C., Schulz, J. (2004), *Software-Produkt-Management*, Berlin 2004
- Köhler, R. (Hrsg., 1977), *Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre*, Stuttgart 1977
- Köhler, P. (2006), *PRINCE2*, Berlin und Heidelberg 2006
- König, W., Heinzl, A., von Poblitzki, A. (1995), Die zentralen Forschungsgegenstände der Wirtschaftsinformatik in den nächsten zehn Jahren, in: *Wirtschaftsinformatik*, 37, 1995, 6, S. 558-569
- Kontio, J. (2001), *Software Engineering Risk Management: A Method, Improvement Framework and Empirical Evaluation*, Helsinki 2001
- Kraut, R. E., Streeter, L. A. (1995), Coordination in Software Development, in: *Communications of the ACM*, 38, 1995, 3, S. 69-81
- Krcmar, H. (2010), Softwaremarkt, in: Kurbel u.a. (Hrsg., 2010)
- Krüger, W. (1992), Aufgabenanalyse und -synthese, in: Frese (Hrsg., 1992), Sp. 221-236
- Kühl, S., Strodtholz, P. (Hrsg., 2002), *Methoden der Organisationsführung - Ein Handbuch*, Hamburg 2002

- Kubicek, H. (1977), Heuristische Bezugsrahmen und heuristisch angelegte Forschungsdesigns als Elemente einer Konstruktionsstrategie empirischer Forschung, in: Köhler (Hrsg., 1977), S. 3-36
- Kubicek, H., Kieser, A. (1980), Organisationsforschung, vergleichende, in: Grochla (Hrsg., 1980), Sp. 1533-1557
- Kubicek, H., Wollnik, M., Kieser, A. (1981), Wege zur praxisorientierten Erfassung der formalen Organisationsstruktur, in: Witte (Hrsg., 1981), S. 79-114
- Kurbel, K., Becker, J., Gronau, N., Sinz, E., Suhl, L. (Hrsg., 2010), Enzyklopädie der Wirtschaftsinformatik - Online-Lexikon, 4. Aufl., München 2010, auf den Seiten der Enzyklopädie der Wirtschaftsinformatik, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de>, Stand: 6.10.2010
- Lamnek, S. (2005), Qualitative Sozialforschung, 4. Aufl., Weinheim, Basel 2005
- Lang, C. (2004), Organisation der Software-Entwicklung, Wiesbaden 2004
- Laux, H., Liermann, F. (2003), Grundlagen der Organisation, 5. Aufl., Berlin 2003
- Lee, K., Boehm, B. (2005), Empirical Results from an Experiment on Value-Based-Review (VBR) Processes, in: Proceedings of the 2005 International Symposium on Empirical Software Engineering ISESE 2005, Noosa Heads, Australia, S. 3-12
- Lehner, F., Zelewski, S. (Hrsg., 2006), Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik, Berlin 2006
- Lehtola, L., Kauppinen, M. (2006), Suitability of Requirements Prioritization Methods for Market-driven Software Product Development, in: Software Improvement and Practice, 11, 2006, S. 7-19
- Lewin, A., Volberda, H. (1999), Prolegomena on Coevolution: A Framework for Research on Strategy and New Organizational Forms, in: Organization Science, 10, 1999, 5, S. 519-534
- Liebold, R., Trinczek, R. (2002), Experteninterview, in: Kühl, Strodtholz (Hrsg., 2002), S. 33-70
- Liggesmeyer, P. (1992), Software-Qualität - Testen, Analysieren und Verifizieren von Software, Heidelberg und Berlin 1992
- Liggesmeyer, P., Rombach, D. (2005), Software Engineering eingebetteter Systeme, München 2005

- Lindstädt, H. (2005), Stimmigkeit und ihre Anwendung als Kriterium der Entscheidungsorganisation, in: Spengler, Lindstädt (Hrsg., 2005), S. 23-38
- Ludewig, J., Lichter, H. (2006), Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, Heidelberg 2006
- Lühring, N. (2006), Koordination von Innovationsprojekten, Wiesbaden 2006
- Marr, R., Steiner K. (2004), Projektmanagement, in: Schreyögg, von Werder (Hrsg., 2004), Sp. 1196-1208
- Marschak, J. (1963), The Payoff – Relevant Description of States and Acts, in: *Econometrica*, 31, 1963, S. 719-725
- Maxwell, K., Van Wassenhove, L., Dutta, S. (1996), Software Development Productivity of European Space, Military, and Industrial Applications, in: *IEEE Transactions on Software Engineering*, 22, 1996, 10, S. 706-718
- Mayring, P. (2002), Einführung in die qualitative Sozialforschung, 5. Aufl., Weinheim und Basel 2002
- McConnell, S. (1996), Rapid Development - Taming Wild Software Schedules, Redmond 1996
- McConnell, S. (2006), Aufwandschätzung für Softwareprojekte, Unterschleißheim 2006
- Mellis, W. (2004), Projektmanagement der SW-Entwicklung - Eine umfassende und fundierte Einführung, Wiesbaden 2004
- Miebach, B. (2007), Organisationstheorie: Problemstellung – Modelle – Entwicklung, Wiesbaden 2007
- Mikusz, M., Herzwurm, G. (2008), Adaption, Umsetzung, Grenzen und Nutzen von Six Sigma in der Softwareentwicklung, in: Herzwurm, Mikusz (Hrsg., 2008), S. 45-56
- Mintzberg, H. (1992), Die Mintzberg-Struktur, Landsberg/Lech 1992
- Müller-Böling, D. (1992), Organisationsforschung, Methodik der empirischen, in: *Fresse* (Hrsg., 1992), Sp. 1491-1505
- Münch, J. (2002), Muster-basierte Erstellung von Software-Projektplänen, Stuttgart 2002
- Münch, J., Heidrich, J. (2004), Software project control centers: concepts and approaches, in: *The Journal of Systems and Software*, 70, 2004, S. 3-19

- Myrach, T. (2010), Lastenheft, in: Kurbel u.a. (Hrsg., 2010)
- Nanda, V. (2001), On tailoring an organizational standard software development process for specific projects, in: ASQ Proceedings of the 11th International Conference on Software Quality, Pittsburgh, PA 2001, S. 1-13
- Neubauer, W., Rosemann, B. (2006), Führung, Macht und Vertrauen in Organisationen, Stuttgart 2006
- Nguyen, V.P., Henderson-Sellers, B. (2003), OPENPC: A tool to automate aspects of method engineering, in: Proceedings of the ICSSEA 2003 - 16th International Conference on Software and Systems Engineering and their Applications, Paris, vol. 5, S. 7ff.
- Nidumolu, S. R. (1996), A Comparison of the Structural Contingency and Risk-Based Perspectives on Coordination in Software-Development Projects, in: Journal of Management Information Systems, 13, 1996, 2, S. 77-113
- Nidumolu, S. R., Subramani, M. (2003), The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development, in: Journal of Management Information Systems, 20, 2003, 3, S. 159-196
- Oestereich, B., Weiss, C. (2007), APM – Agiles Projektmanagement, Heidelberg 2007
- Omagbemi, R. (1994), Die Messung und Beurteilung der Effizienz von Projekten der angewandten Forschung und Entwicklung, Berlin 1994
- Paech, B., Rolland, C. (Hrsg., 2008), Requirements Engineering: Foundation for Software Quality, Proceedings of the 14th International Working Conference, REFSQ 2008, Montpellier, LNCS 5025, Berlin 2008
- Patton, N., Shechet, A. (2007), Earned Value Management: Are Expectations Too High?, in: Crosstalk - The Journal of Defense Software Engineering, 2007, S. 10-15
- Patzak, G., Rattay (2004), Projektmanagement, Leitfaden zum Management von Projekten, Projektportfolios und projektorientierten Unternehmen, 4. Aufl., Wien 2004
- Pedreira, O., Piattini, M., Luaces, M. R., Brisaboa, N. R. (2007), A Systematic Review of Software Process Tailoring, in: ACM SIGSOFT Software Engineering Notes, 32, 2007, 3, S. 1-6
- Pichler, R. (2008a), Scrum - Agiles Projektmanagement erfolgreich einsetzen, Heidelberg 2008

- Pichler, R. (2008b), Erfolgsfaktor „Product Owner“, in: OBJEKTSpektrum, 2008, 3, S. 14-15
- Pichler, R. (2010), Product Owners Should Care About Quality, in: Better Software, 2010, 5, S. 35-36
- Picot, A., Dietl, H., Franck, E. (2005), Organisation - Eine ökonomische Perspektive, 4., Aufl., Stuttgart 2005
- Pietsch, W. (1992), Methodik des betrieblichen Software-Projektmanagements - Grundlagen, Begründung und Konzeption eines evolutionären Ansatzes, Berlin und New York 1992
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still J. (2008), The impact of agile practices on communication in software development, in: Empirical Software Engineering, 13, 2008, S. 303-337
- Pohl, K. (2007), Requirements Engineering - Grundlagen, Prinzipien, Techniken, Heidelberg 2007
- Project Management Institute (Hrsg., 2006), The Standard for Program Management, Newton Square 2006
- Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (Hrsg., 2007), Situational Method Engineering: Fundamentals and Experiences - Proceedings of the IFIP WG 8.1 Working Conference, Geneva, Berlin 2007
- Ramesh, B., Cao, L., Mohan, K., Xu, P. (2006), Can Distributed Software Development be Agile?, in: Communications of the ACM, 49, 2006, 10, S. 41-46
- Regnell, B., Berntsson Svensson, R., Wnuk, K. (2008), Can We Beat the Complexity of Very Large-Scale Requirements Engineering?, in: Paech, Rolland (Hrsg., 2008), S. 123-128
- Richardson, I., Abrahamsson, P., Messnarz, R. (Hrsg., 2005), Proceedings of the 12th European Conference Software Process Improvement, LNCS 3792, Berlin und Heidelberg 2005
- Royce, W. (2005), Successful Software Management Style: Steering and Balance, in: IEEE Software, 22, 2005, 5, S. 40-47
- Sauerwald, K. (2007), Effektivität und Effizienz: Zielbeziehungen organisationaler Entscheidungen, München, Mering 2007

- Schanz, G. (1994), Organisationsgestaltung - Management von Arbeitsteilung und Koordination, 2. Aufl., München 1994
- Schelle, H., Ottmann, R., Pfeiffer, A. (2005), Projektmanager, Nürnberg 2005
- Schmelzer, H. J. (1991), Organisation und Controlling der Entwicklung von Serienprodukten, Karlsruhe 1991
- Scholl, W. (1992), Politische Prozesse in Organisationen in: Frese (Hrsg., 1992), Sp. 1993-2004
- Scholz, C. (1992), Effektivität und Effizienz, organisatorische, in: Frese (Hrsg., 1992), Sp. 533-585
- Scholz, P. (2005), Softwareentwicklung eingebetteter Systeme - Grundlagen, Modellierung, Qualitätssicherung, Berlin und Heidelberg 2005
- Schonberger, R.J. (1980), MIS Design - A Contingency Approach, in: MIS Quarterly, 4, 1980, 1, S. 13-20
- Schrader, S., Göpfert, J. (2001), Zielklarheit und Zieloffenheit, in: Hamel, Gemünden (Hrsg., 2001), S. 75-100
- Schramke, A. (2002), Organisatorische Gestaltung von Softwareentwicklungsprojekten, Unterföhring 2002
- Schreyögg, G. (1992), Organisationstheorie, entscheidungsprozessorientierte, in: Frese (Hrsg., 1992), Sp. 1746-1757
- Schreyögg, G. (2008), Organisation – Grundlagen moderner Organisationsgestaltung, 5. Aufl., Wiesbaden, 2008
- Schreyögg, G., von Werder, A. (Hrsg., 2004), Handwörterbuch Unternehmensführung und Organisation, 4. Aufl., Stuttgart 2004
- Schulte-Zurhausen, M. (2002), Organisation, 3. Aufl., München 2002
- Schwaber, K., Sutherland, J. (2010), The Scrum Guide, auf den Seiten des Scrum.org, <http://www.scrum.org/scrumguides/>, Stand: Februar 2010
- SEI (Hrsg., 2006), CMMI for Development Ver. 1.2, Technical Report CMU/SEI-2006-TR-008, Software Engineering Institute, Pittsburgh, USA 2006
- SEI (Hrsg., 2010), Process Maturity Profile: CMMI For Development, SCAMPI Class A Appraisal Results, 2009 End-Year Update, March 2010, auf den Seiten des SEI,

<http://www.sei.cmu.edu/cmami/casestudies/profiles/pdfs/upload/2010MarCMMI.pdf>,

Stand: März 2010

Shenhar, A. J. (2001a), One Size Does Not Fit All Projects: Exploring Classical Contingency Domains, in: *Management Science*, 47, 2001, 3, S. 394-414

Shenhar, A. J. (2001b), Contingent management in temporary, dynamic organizations: The comparative analysis of projects, in: *Journal of High Technology Management Research*, 12, 2001, 2, S. 239-271

Shenhar, A., Dvir, D., Milosevic, D., Mullenburg, J., Patanakul, P., Reilly, R., Ryan, M., Sage, A., Sauser, B., Srivannaboon, S. Stefanovic, J., Thamhain, H. (2005), Toward a NASA-Specific Project Management Framework, in: *Engineering Management Journal*, 17, 2005, 4, S. 8-15

Simon, H.A. (1955): A Behavioral Model of Rational Choice, in: *Quarterly Journal of Economics*, 69, 1955, S. 99-118

Spengler, T. (1999), *Grundlagen und Ansätze der strategischen Personalplanung mit vagen Informationen*, München und Mering 1999

Spengler, T., Lindstädt, H. (Hrsg., 2005), *Strukturelle Stimmigkeit in der Betriebswirtschaftslehre*, München und Mering 2005

Staehe, W. (1977), Empirische Analyse von Handlungssituationen, in: Köhler (Hrsg., 1977), S. 103-116

Standish Group International (Hrsg., 1994), *The Chaos Report*, West Yarmouth 1994

Steinke, I. (2004), Gütekriterien qualitativer Forschung, in: Flick u.a. (Hrsg., 2004), S. 319-331

Stelzer, D. (1998), *Möglichkeiten und Grenzen des prozessorientierten Software-Qualitätsmanagements*, Köln 1998

Strodtholz, P., Kühl, S. (2002), Qualitative Methoden der Organisationsforschung – Ein Überblick, in: Kühl, Strodtholz (Hrsg., 2002), S. 11-31

Szyperski, N., Möller-Böling, D. (1981), Zur technologischen Orientierung der empirischen Forschung, in: Witte (Hrsg., 1981), S. 159-188

Tasharofi, S., Ramsin, R. (2007), Process Patterns for Agile Methodologies, in: Ralyte u.a. (Hrsg., 2007), S. 222-237

- Thelin, T., Runeson, P., Wohlin, C. (2003), An Experimental Comparison of Usage-Based and Checklist-Based Reading, in: IEEE Transactions on Software Engineering, 29, 2003, 8, S. 687-704
- Thompson, J.D. (1967), Organizations in action: social science bases of administrative theory, New York 1967
- Van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L. (2006), On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support, in: IEEE Proceedings of the First International Workshop on Software Product Management, in conjunction with the 14th IEEE International Requirements Engineering Conference, Minneapolis/St. Paul, Minnesota, USA 2006, S. 7-16
- Von Werder, A. (2004), Organisatorische Gestaltung (Organization Design), in: Schreyögg, von Werder (Hrsg., 2004), Sp. 1088-1101
- Wallace, L., Keil, M. (2004), Software project risks and their effect on outcomes, in: Communications of the ACM, 47, 2004, 4, S. 68-73
- Wallmüller, E. (2004), Risikomanagement für IT- und Software-Projekte - Ein Leitfa- den für die Umsetzung in der Praxis, München und Wien 2004
- Wang, Q., Pfahl, D., Raffo, D., Wernick, P. (Hrsg., 2006), SPW/ProSim 2006 - Pro- ceedings of the Software Process Change / International Software Process Workshop and International Workshop on Software Process Simulation and Modeling, Berlin 2006
- Weber, M. (1972), Wirtschaft und Gesellschaft – Grundriss der verstehenden Soziolo- gie, Tübingen 1972
- Weill, P., Olson, M.H. (1989), An Assessment of the Contingency Theory of Man- agement Information Systems, in: Journal of Management Information Systems, 6, 1989, 1, S. 59-85
- Weltz, F., Ortmann, R. (1992), Das Softwareprojekt – Projektmanagement in der Pra- xis, Frankfurt/Main und New York 1992
- Wermelinger, M., Gall, H. (Hrsg., 2005), Proceedings of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT International Sympo- sium on Foundations of Software Engineering, Lisbon, Portugal 2005
- Wicher, H. (1993), Kongruenzprinzip der Organisation, in: WISU 22, 1993, 7, S. 579- 580

Wieggers, K. E. (2005), Software Requirements, Unterschleißheim 2005

Wildemann, H., Kersten, W. (2000), Kundenorientiertes Management der Software-Entwicklung, in: ZfB, 70, 2000, 11, S. 1215-1239

Witte, E. (Hrsg., 1981), Der praktische Nutzen empirischer Forschung, Tübingen 1981

Wolf, J. (2008), Organisation, Management, Unternehmensführung - Theorien und Kritik, 3. Aufl., Wiesbaden 2008

Wollnik, M. (1977), Die explorative Verwendung systematischen Erfahrungswissens - Plädoyer für einen aufgeklärten Empirismus in der Betriebswirtschaftslehre, in: Köhler (Hrsg., 1977), S. 37-64

Wong, B. (2005), Understanding Stakeholder Values as a Means of Dealing with Stakeholder Conflicts, in: Software Quality Control, 13, 2005, 4, S. 429-445

Xu, P., Ramesh, B. (2007), Software Process Tailoring: An Empirical Investigation, in: Journal of Management Information Systems, 24, 2007, 2, S. 293-328

Yourdon, E. (1995), When good enough Software is best, in: IEEE Software, 12, 1995, 3, S. 79-81

Zelewski, S. (2008), Grundlagen, in: Corsten, Reiß (Hrsg., 2008), S. 1-98

Anhang: Beispielhafte Konfigurationen zu den definierten Projekttypen

Projekttyp 1: Übergeordnete Projektzielpräferenz für Entwicklungskosten (LU₀; Q₀; EZ₀; EK₁)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Entwicklungskosten und Rollenorientierung auf Projektmanagement-Rollen; bspw. Methoden: EVM, ATA	Projektmanagement verfügt wie üblich über Richtlinienkompetenz und in diesem Rahmen über weitgehende Kompetenzen bei der Projektsteuerung Projektmanagement verfügt über Vetokompetenz beim Eskalationsprozess für Abweichungen von den Entwicklungskosten	Projektmanagement-Rolle darf weder unbesetzt noch vom Qualitäts- oder Produktmanagement wahrgenommen werden Besonders kritische Kostenaspekte werden durch eigenständige und personell unabhängige besetzte Rollen vertreten (bspw. Kosten-Controller, der einem Projekt vollständig zugeordnet ist)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - Perspektiven: insb. 'Entwicklungsprozessaspekte'	Mitentscheidungs- bis hin zur Fachentscheidungskompetenz des Projektmanagements bei Produktreviews, v.a. bei Perspektive 'Entwicklungsprozessaspekte' Fachentscheidungs- bis hin zur Vetokompetenz des Projektmanagements beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen	Projektmanagement ist dem Produkt- und Qualitätsmanagement (Rolle) gegenüber weisungsbefugt, d.h. keine organisatorische Unabhängigkeit des Produkt- und Qualitätsmanagements
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Entwicklungskosten	Projektmanagement kann situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Projektmanagement hat Richtlinienkompetenz bei der Risiko- steuerung	
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Entwicklungskosten ('value-based requirements tracing'; Checklisten)	Informationskompetenz des Projektmanagements bei allen Anforderungsanträgen Richtlinienkompetenz des Projektmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen	

Projekttyp 2: Übergeordnete Projektzielpräferenz für Leistungsumfang, Qualität und Entwicklungszeit (LU₁; Q₁; EZ₁; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Leistungsumfang, Qualität und Entwicklungszeit sowie gleichmäßiger Rollenorientierung	Relative Abschwächung von Kompetenzen des Projektmanagements zugunsten des Produkt- und Qualitätsmanagements über eine teilweise Umsetzung der bei Projekttyp 6 aufgeführten Maßnahmen	Qualitäts-, Produkt- und Projektmanagement-Rollen dürfen jeweils weder unbesetzt noch organisatorisch abhängig sein Besonders kritische Qualitätsteilmerkmale, Terminaspekte und Aspekte des Leistungsumfangs werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. ‚Product standard owner‘-Rollen, Zeit-Controller, der einem Projekt vollständig zugeordnet ist, oder Verantwortlicher für ein bestimmtes Anwendungsszenario)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - mit gleichmäßiger Fokussierung auf die Perspektiven	Relative Abschwächung von Kompetenzen der Qualitätsmanagement-Rolle zugunsten des Produkt- und Projektmanagements über eine teilweise Umsetzung der bei Projekttyp 7 aufgeführten Maßnahmen	
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Leistungsumfang, Qualität und Entwicklungszeit	Gleichmäßige Kompetenzverteilung beim Risikomanagement	
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Leistungsumfang, Qualität und Entwicklungszeit ('value-based requirements tracing'; Checklisten)	Gleichmäßige Kompetenzverteilung bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen	

Projekttyp 3: Übergeordnete Projektzielpräferenz für Leistungsumfang (LU₁; Q₀; EZ₀; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefakt-spezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf den Leistungsumfang und Rollenorientierung auf Produkt- und Projektmanagement-Rollen	Produktmanagement kann situationsbedingt zusätzliche Projektreviews bzw. Statussitzungen einberufen (Antragskompetenz) Ausgedehnte Mitentscheidungs- bzw. Mitberatungskompetenzen des Produktmanagements (durch Beteiligung) bei Lenkungs- und Beratungsausschüssen Produktmanagement hat Veto-kompetenz im Rahmen von Ausnahmeplanungen (Projektplan)	Produktmanagement-Rolle darf weder unbesetzt noch vom Qualitäts- oder Projektmanagement wahrgenommen werden Besonders kritische Aspekte des Leistungsumfangs werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. Verantwortlicher für ein bestimmtes Anwendungsszenario)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - Perspektiven: insb. 'Kunde'	Mitentscheidungs- bis hin zur Fachentscheidungskompetenz des Produktmanagements bei Produktreviews, v.a. bei Perspektive 'Kunde' Fachentscheidungs- bis hin zur Vetokompetenz des Produktmanagements beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen	Produktmanagement ist dem Projektmanagement gegenüber weisungsbefugt und leitet somit das Projekt de facto (bspw. Ausgestaltung der Produktmanagement-Rolle im Sinne des 'Product Owners' bei 'Scrum')
Risiko-management	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Leistungsumfang	Produktmanagement kann situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Produktmanagement hat Richtlinienkompetenz bei der Risiko-steuerung	
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Leistungsumfang ('value-based requirements tracing'; Checklisten)	Informationskompetenz des Produktmanagements bei allen Anforderungsanträgen Richtlinienkompetenz des Produktmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen	

Projekttyp 4: Übergeordnete Projektzielpräferenz für Qualität (LU₀; Q₁; EZ₀; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Qualität und Rollenorientierung auf Qualitäts- und Projektmanagement-Rollen	Qualitätsmanagement kann situationsbedingt zusätzliche Projektreviews bzw. Statussitzungen einberufen (Antragskompetenz) Ausgedehnte Mitentscheidungs- bzw. Mitberatungskompetenzen des Qualitätsmanagements (durch Beteiligung) bei Lenkungs- und Beratungsausschüssen Qualitätsmanagement hat Veto-kompetenz im Rahmen von Ausnahmeplanungen (Projektplan)	Qualitätsmanagement-Rolle darf weder unbesetzt noch vom Projekt- oder Produktmanagement wahrgenommen werden Besonders kritische Qualitätsteilmerkmale werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. 'Product standard owner'-Rollen)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - Perspektiven: insb. 'Softwarearchitekt' und 'Tester'	Qualitätsmanagement (Rolle) verfügt wie üblich über Richtlinienkompetenz und in diesem Rahmen über weitgehende Kompetenzen beim Qualitätsmanagementprozess Qualitätsmanagement (Rolle) verfügt über Vetokompetenz beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen	Qualitätsmanagement (Rolle) ist dem Projektmanagement gegenüber weisungsbe-fugt und leitet somit das Projekt de facto, Projektmanagement nimmt hier lediglich eine Projektcontroller-Rolle wahr
Risiko-management	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Qualität	Qualitätsmanagement kann situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Qualitätsmanagement hat Richtlinienkompetenz bei der Risiko-steuerung	
Änderungs-management	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Qualität ('value-based requirements tracing'; Checklisten)	Informationskompetenz des Qualitätsmanagements bei allen Anforderungsanträgen Richtlinienkompetenz des Qualitätsmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen	

Projekttyp 5: Übergeordnete Projektzielpräferenz für Entwicklungszeit (LU₀; Q₀; EZ₁; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefakt-spezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Entwicklungszeit und Rollenorientierung auf Projektmanagement-Rollen; bspw. Methoden: EVM, MTA, Projektpuffer-Verfahren	Projektmanagement verfügt wie üblich über Richtlinienkompetenz und in diesem Rahmen über weitgehende Kompetenzen bei der Projektsteuerung Projektmanagement verfügt über Vetokompetenz beim Eskalationsprozess für Abweichungen von der Entwicklungszeit	Projektmanagement-Rolle darf weder unbesetzt noch vom Qualitäts- oder Produktmanagement wahrgenommen werden Besonders kritische Terminaspekte werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. Zeit-Controller, der einem Projekt vollständig zugeordnet ist)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtiC'-Ansatz) - Perspektiven: insb. 'Entwicklungsprozessaspekte'	Mitentscheidungs- bis hin zur Fachentscheidungskompetenz des Projektmanagements bei Produktreviews, v.a. bei Perspektive 'Entwicklungsprozessaspekte' Fachentscheidungs- bis hin zur Vetokompetenz des Projektmanagements beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen	Projektmanagement ist dem Produkt- und Qualitätsmanagement (Rolle) gegenüber weisungsbefugt, d.h. keine organisatorische Unabhängigkeit des Produkt- und Qualitätsmanagements
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Entwicklungszeit	Projektmanagement kann situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Projektmanagement hat Richtlinienkompetenz bei der Risikosteuerung	
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Entwicklungszeit ('value-based requirements tracing'; Checklisten)	Informationskompetenz des Projektmanagements bei allen Anforderungsanträgen Richtlinienkompetenz des Projektmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen	

Projekttyp 6: Übergeordnete Projektzielpräferenz für Leistungsumfang und Qualität (LU₁; Q₁; EZ₀; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefakt-spezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Leistungsumfang und Qualität sowie "gleichmäßiger" Rollenorientierung	Produkt- und Qualitätsmanagement können situationsbedingt zusätzliche Projektreviews bzw. Statussitzen einberufen (Antragskompetenz) Ausgedehnte Fachentscheidungs- bzw. Beratungskompetenzen des Produkt- und Qualitätsmanagements (durch Beteiligung) bei Lenkungs- und Beratungsausschüssen Produkt- und Qualitätsmanagement haben Vetokompetenz im Rahmen von Ausnahmeplanungen (Projektplan)	Qualitäts- und Produktmanagement-Rollen dürfen weder unbesetzt noch vom Projektmanagement wahrgenommen werden Besonders kritische Qualitätsteilmerkmale und Aspekte des Leistungsumfangs werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. 'Product standard owner'-Rollen oder Verantwortlicher für ein bestimmtes Anwendungsszenario)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - Perspektiven: insb. 'Kunde', 'Softwarearchitekt' und 'Tester'	Relative Abschwächung von Kompetenzen des Qualitätsmanagements (Rolle) zugunsten des Produktmanagements über eine teilweise Umsetzung der bei Projekttyp 3 aufgeführten Maßnahmen	
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Leistungsumfang und Qualität	Produkt- und Qualitätsmanagement können situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Fachentscheidungskompetenzen des Produkt- und Qualitätsmanagements bei Risikosteuerung hinsichtlich Leistungsumfang bzw. Qualität	Projektmanagement kann vom Produkt- oder Qualitätsmanagement wahrgenommen werden Projektmanagement/-controlling steht gleichzeitig mehreren Projekten zur Verfügung
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Leistungsumfang und Qualität ('value-based requirements tracing'; Checklisten)	Informationskompetenzen des Qualitäts- und Produktmanagements bei allen Anforderungsanträgen Weitreichende Mitentscheidungskompetenzen des Qualitäts- und Produktmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen (bspw. Vorbringung von Einwänden, Genehmigung der Änderung etc.)	

Projekttyp 7: Übergeordnete Projektzielpräferenz für Leistungsumfang und Entwicklungszeit (LU₁; Q₀; EZ₁; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Leistungsumfang und Entwicklungszeit sowie Rollenorientierung auf Produkt- und Projektmanagement-Rollen	Relative Abschwächung von Kompetenzen des Projektmanagements zugunsten des Produktmanagements über eine teilweise Umsetzung der bei Projekttyp 3 aufgeführten Maßnahmen	Projekt- und Produktmanagement-Rollen dürfen weder unbesetzt noch vom Qualitätsmanagement wahrgenommen werden Besonders kritische Terminaspekte und Aspekte des Leistungsumfangs werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. Zeit-Controller, der einem Projekt vollständig zugeordnet ist, oder Verantwortlicher für ein bestimmtes Anwendungsszenario)
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQtIC'-Ansatz) - Perspektiven: insb. 'Kunde' und 'Entwicklungsprozessaspekte'	Mitentscheidungs- bis hin zu Fachentscheidungskompetenzen des Projekt- und Produktmanagements bei Produktreviews, v.a. bei den Perspektiven 'Entwicklungsprozessaspekte' und 'Kunde' Fachentscheidungs- bis hin zu Vetokompetenzen des Projekt- und Produktmanagements beim Eskalationsprozess für Abweichungen von den geplanten Qualitätszielen	Qualitätsmanagement-Rolle kann vom Projekt- oder Produktmanagement wahrgenommen werden
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Leistungsumfang und Entwicklungszeit	Projekt- und Produktmanagement können situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Fachentscheidungskompetenzen des Projekt- und Produktmanagements bei Risikosteuerung hinsichtlich Entwicklungszeit bzw. Leistungsumfang	Qualitätsmanagement-Rolle steht gleichzeitig mehreren Projekten zur Verfügung
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Leistungsumfang und Entwicklungszeit ('value-based requirements tracing'; Checklisten)	Informationskompetenzen des Produkt- und Projektmanagements bei allen Anforderungsanträgen Weitreichende Mitentscheidungskompetenzen des Produkt- und Projektmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen (bspw. Vorbringung von Einwänden, Genehmigung der Änderung etc.)	

Projekttyp 8: Übergeordnete Projektzielpräferenz für Qualität und Entwicklungszeit (LU₀; Q₁; EZ₁; EK₀)			
Koordinationsprozess	Gestaltungsmaßnahmen in der Strukturdimension Artefaktspezifikation	Gestaltungsmaßnahmen in der Strukturdimension Kompetenzverteilung	Gestaltungsmaßnahmen in der Strukturdimension 'Personelle und organisatorische Unabhängigkeit'
Projektsteuerung	'Software project control center' mit Zielorientierung auf Qualität und Entwicklungszeit sowie Rollenorientierung auf Qualitäts- und Projektmanagement-Rollen	Relative Abschwächung von Kompetenzen des Projektmanagements zugunsten des Qualitätsmanagements über eine teilweise Umsetzung der bei Projekttyp 4 aufgeführten Maßnahmen	Qualitäts- und Projektmanagement-Rollen dürfen weder unbesetzt noch vom Produktmanagement wahrgenommen werden
Qualitätsmanagement	Perspektivenbasiertes Lesen mithilfe von Produktreview-Checklisten (bspw. 'TAQ ₁ IC'-Ansatz) - Perspektiven: insb. 'Entwicklungsprozessaspekte', 'Softwarearchitekt' und 'Tester'	Relative Abschwächung von Kompetenzen des Qualitätsmanagements (Rolle) zugunsten des Projektmanagements über eine teilweise Umsetzung der bei Projekttyp 5 aufgeführten Maßnahmen	Besonders kritische Qualitätsteilmerkmale und Terminaspekte werden durch eigenständige und personell unabhängig besetzte Rollen vertreten (bspw. 'Product standard owner'-Rollen oder Zeit-Controller, der einem Projekt vollständig zugeordnet ist)
Risikomanagement	Risikoidentifikation und -analyse mithilfe von Risikochecklisten (bspw. 'TBQ') oder Risikoszenarien ('RiskIT') insb. hinsichtlich Qualität und Entwicklungszeit	Projekt- und Qualitätsmanagement können situationsbedingt Risikoworkshops oder Sofortmaßnahmen einberufen (Antragskompetenzen) Fachentscheidungskompetenzen des Projekt- und Qualitätsmanagements bei Risikosteuerung hinsichtlich Entwicklungszeit bzw. Qualität	Produktmanagement-Rolle kann vom Projekt- oder Qualitätsmanagement wahrgenommen werden
Änderungsmanagement	Aufzeichnungsstrategie von Nachvollziehbarkeitsinformationen und Dokumentation der Einflussanalyse im Änderungsantrag insb. hinsichtlich Qualität und Entwicklungszeit ('value-based requirements tracing'; Checklisten)	Informationskompetenzen des Qualitäts- und Projektmanagements bei allen Anforderungsanträgen Weitreichende Mitentscheidungskompetenzen des Qualitäts- und Projektmanagements bei der Freigabe, Steuerung und Überwachung von Anforderungsänderungen (bspw. Vorbringung von Einwänden, Genehmigung der Änderung etc.)	Produktmanagement-Rolle steht gleichzeitig mehreren Projekten zur Verfügung

WIRTSCHAFTSINFORMATIK

Herausgegeben von Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Hans-Georg Kemper, Stuttgart, Prof. Dr. Georg Herzwurm, Stuttgart, Prof. Dr. Dirk Stelzer, Ilmenau, und Prof. Dr. Detlef Schoder, Köln

Band 69

Ludwig Fuchs

Methodology for Hybrid Role Development

Lohmar – Köln 2010 ♦ 272 S. ♦ € 58,- (D) ♦ ISBN 978-3-89936-978-6

Band 70

Andreas Helferich

Software Mass Customization

Lohmar – Köln 2010 ♦ 380 S. ♦ € 65,- (D) ♦ ISBN 978-3-8441-0006-8

Band 71

Tyge-F. Kummer

Akzeptanz von Ambient Intelligence in Krankenhäusern – Ein Ländervergleich zwischen Deutschland und Australien am Beispiel der Medikationsunterstützung

Lohmar – Köln 2010 ♦ 320 S. ♦ € 62,- (D) ♦ ISBN 978-3-8441-0008-2

Band 72

Stephan Wildner

Problemorientiertes Wissensmanagement – Eine Neukonzeption des Wissensmanagements aus konstruktivistischer Sicht

Lohmar – Köln 2011 ♦ 384 S. ♦ € 65,- (D) ♦ ISBN 978-3-8441-0043-3

Band 73

Rolf Schillinger

Semantic Service Oriented Architectures in Research and Practice

Lohmar – Köln 2011 ♦ 304 S. ♦ € 59,- (D) ♦ ISBN 978-3-8441-0062-4

Band 74

Martin Mikusz

Koordination der Standardsoftwareentwicklung – Ein situativer Ansatz

Lohmar – Köln 2011 ♦ 296 S. ♦ € 59,- (D) ♦ ISBN 978-3-8441-0073-0