



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Alexander Kaack

**Eine empirische Studie zum Stand von agiler
Software-Entwicklung in der Praxis**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Alexander Kaack

**Eine empirische Studie zum Stand von agiler
Software-Entwicklung in der Praxis**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachterin: Dr. Susanne Draheim

Eingereicht am: 06.06.2017

Alexander Kaack

Thema der Arbeit

Eine empirische Studie zum Stand von agiler Software-Entwicklung in der Praxis

Stichworte

Agile, Scrum, Scrum Master, Kanban, Transformation, Implementierung, Agilitätsanalyse, Softwareentwicklung, Empirische Untersuchung

Kurzzusammenfassung

Der Inhalt dieser Arbeit stellt die Untersuchung zum Stand agiler Softwareentwicklung dar. Hierzu werden sowohl die Abhängigkeiten der traditionellen Vorgehensmodelle sowie der agilen Methoden berücksichtigt. Im Zuge der weiteren Verbreitung von agilen Verfahren wird auf die Herausforderung der Implementierung sowie auf die weitere Adaption eingegangen. Zur Darstellung der aktuellen Lage bestehender Teams wird hierzu eine empirische Untersuchung präsentiert. Das Ergebnis stellt Konsequenzen zur Herangehensweise zur Verfolgung von agilen Verfahren dar.

Alexander Kaack

Title of the paper

Empirical study of a practical state in agile software development

Keywords

agile, scrum, scrum master, kanban, transformation, implementation, agile analysis, software development, empirical study

Abstract

This thesis presents a study of a practical state of the agile software development. In conclusion of that statement the dependency of the traditional as well as the agile models are required. Therefore, this document shows how a transition and continuous adaption of agile development can look like. To support the understanding of this challenges this work shows a practical study to the current state agile grade. The final result is the presentation of consequences in the agile method approach.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einleitung	1
1.1 Problemstellung	2
1.2 Lösungsansatz	3
1.3 Struktur dieser Arbeit	4
2 Traditionelle Vorgehensmodelle der Softwareentwicklung	6
2.1 Softwarekrise	6
2.2 Klassische Phasenmodelle	7
2.2.1 Wasserfallmodell	8
2.2.2 V-Modell	8
2.2.3 Spiralmodell	9
2.2.4 Evolutionäre Modelle	9
2.2.5 Rational Unified Process	10
2.3 Festhalten am Vorgehen	10
3 Motivation zur Veränderung	12
3.1 Veränderung der Softwareentwicklung	12
3.2 Vorteile agiler Vorgehensweisen	16
4 Agile Vorgehensmodelle	19
4.1 Agile Werte hinter dem agilen Manifest	19
4.2 Fokus agiler Modelle	21
4.2.1 Extreme Programming	22
4.2.2 Kanban	22
4.2.3 Scrum	23
5 Umwandlung des Entwicklungsprozesses	25
5.1 Unterschiede der Vorgehensmodelle	26
5.2 Vorgehen zur agilen Transition	40
5.2.1 Revolution	41
5.2.2 Evolution	42
5.2.3 Agile Potenzialanalyse	43
5.3 Durchführung einer Scrum Migration	46
5.3.1 Vom V-Modell zu Scrum	47

5.3.2	Vom Rational Unified Process zu Scrum	49
5.4	Fazit	52
6	Herausforderungen der agilen Softwareentwicklung	53
6.1	Stakeholder Identifikation	54
6.2	Product Owner im Fokus des Prozesses	55
6.3	Die Rolle des Scrum Masters	57
6.4	Impediments im Scrum Prozess	58
6.5	Produktive agile Teams	61
6.6	Agiles Projektmanagement	63
6.7	Agilitätsgrad	64
6.8	Fazit	65
7	Empirische Erhebung zur Scrum Implementierung	67
7.1	Ursprüngliche Untersuchung	67
7.1.1	Ziele	67
7.1.2	Aufbau	68
7.1.3	Interviewleitfaden	69
7.1.4	Durchführung der Untersuchung	69
7.1.5	Verarbeitung der Interviews	70
7.2	Neuauswertung der Untersuchung	72
7.2.1	Ziele	72
7.2.2	Aufbau	72
7.3	Auswertung der Einzelinterviews	74
7.3.1	Interview 1	75
7.3.2	Interview 2	77
7.3.3	Interview 4	78
7.3.4	Interview 5	80
7.4	Gesamtauswertung	82
7.4.1	Auswertungsverfahren	82
7.4.2	Auswertung	83
7.5	Fazit	87
8	Konsequenzen zur Herangehensweise	89
8.1	Vergangenheit der Softwareentwicklung	89
8.2	Identifikation des Ist-Zustands	92
8.3	Zielsetzung	93
8.4	Fazit	95
9	Schluss	96
9.1	Zusammenfassung	96
9.2	Ausblick	98

Literaturverzeichnis	99
Anhang	106
E-Mail Vorlage	106
Auswertungstabellen der ersten Auswertung	107
Weitere Punkte der ersten Auswertung	114
Interviewleitfaden	116
Kodierregeln	122
Einzelinterview 1	125
Einzelinterview 2	130
Einzelinterview 4	135
Einzelinterview 5	139

Abbildungsverzeichnis

1	Einleitung	1
2	Traditionelle Vorgehensmodelle der Softwareentwicklung	6
3	Motivation zur Veränderung	12
4	Agile Vorgehensmodelle	19
4.1	Vergleich der Vorschriften und Adaptionfähigkeit von Entwicklungsmodellen [32]	21
5	Umwandlung des Entwicklungsprozesses	25
5.1	Agile Softwareentwicklung - Beziehung zwischen den verschiedenen agilen Aspekten [49]	44
5.2	Agile Potenzialanalyse [49]	45
6	Herausforderungen der agilen Softwareentwicklung	53
6.1	Beeinflussung des Teams im Scrum Prozess	59
6.2	Impediment Impact Diagram mit extra Dimensionen für Risiko und Beseiti- gungsaufwand [45]	60
7	Empirische Erhebung zur Scrum Implementierung	67
7.1	Vergleich der Indizien - Interview 1	76
7.2	Vergleich der Indizien - Interview 2	78
7.3	Vergleich der Indizien - Interview 4	79
7.4	Vergleich der Indizien - Interview 5	81
7.5	Vergleich der traditionellen Indizien	84
7.6	Vergleich der agilen Indizien	85
7.7	Vergleich der Summe aller agil und traditionell ausgerichteten Punkte	87
8	Konsequenzen zur Herangehensweise	89
9	Schluss	96

1 Einleitung

Die Verfahren, wie Software entwickelt wird, haben sich über die letzten Jahre stark verändert. So sorgen Faktoren der Softwareentwicklungsbranche dafür, dass die Branche weiterhin wächst. Unter anderem geschieht dies durch die globale Wirtschaftsentwicklung [2]. Zudem ist innerhalb des IT-Softwareentwicklungsmarkts ein größerer Wettbewerb zu verzeichnen [2]. Parallel lässt sich auch eine Vergrößerung des Softwaremarkts darstellen [55, 56, 57], die den Trend des wachsenden Bedarfs an der Softwareentwicklung unterstützt. In diese Thematik gliedert sich auch der generelle Zuwachs der sogenannten *Wissensarbeiter* [1] ein, wo Mitarbeiter eines Unternehmens für die Anwendung ihres Wissens entlohnt werden.

Bereits die genannten Faktoren sorgen dafür, dass der Anteil an agilen Verfahren weiter steigt, wobei sich *Scrum* weiterhin an der Spitze hält und in mehr als der Hälfte aller agilen Softwareentwicklungsteams verwendet wird [57]. Dieser Trend wird weiterhin durch wachsende und komplexere Projekte unterstützt [49]. In diesen Projekten wirken die traditionellen Kontrollmechanismen über statistische Mittel nutzlos. Die Ungewissheit wird vielmehr als Bestandteil der Softwareentwicklung angesehen [14].

In Abhängigkeit dazu stellt die Veränderung der benötigten Entwicklungszeiträume von Jahren zu Monaten einen Faktor dar. Wöchentliche Veränderungen in den Branchen sorgen dafür, dass entsprechend schnell reagiert werden muss. Agile Methoden stellen hierzu eine Lösungsvariante dar, um zukünftige Märkte bewältigen zu können. Dabei verhalten sich die agilen Methoden als antifragile Systeme, indem durch Veränderungen des Kontextes und das Anerkennen von starken Planungsschwankungen, eine Verbesserung des Entwicklungsprozesses hervorgeht [37, 54].

Durch die Forderung von Veränderungen am Umgang mit Anforderungen [8, 47, 55] sowie die benötigte, schnellere und sofortige Lieferbereitschaft von Werten an den Kunden [2, 8], sind zusätzliche Herausforderungen gegeben. Diese lassen sich dabei nicht mit herkömmlichen Verfahren bewältigen. So muss und will der Kunde stärker involviert werden [55], sodass ein Raum geschaffen wird, indem zukünftige Innovationen überhaupt hervorgebracht werden können [2].

Ebenso wird ein verändertes Zeitmanagement benötigt. Hierbei gilt es, dass die Zeit zwischen Planung und Implementierung bewältigt und verbessert werden muss. Im Kontext dessen ist eine minimale Planung einer Aktion notwendig, um diese benötigte Geschwindigkeit realisieren zu können [15]. Es ist somit auch ein steigender Druck auf die Entwicklungsteams und Entwickler zu verzeichnen [2, 55]. Hierbei sind Prozesse notwendig, die das Lernen sowie die Kreativität fördern, um eine Unterstützungsmöglichkeit zu bieten [15].

Im Ergebnis dieser Punkte stellen sich besonders in Betrachtung der traditionellen Verfahren Projekte dar, die im Kontext der aktuellen Lage niemals hätten starten dürfen [47]. Es sind somit starke Veränderungen erkennbar, indem agile Verfahren notwendig sind, um zukünftigen und bestehenden Problemstellungen verbessert entgegenzuwirken. Diese Thematik soll in Bezug auf die im nächsten Abschnitt beschriebene Problemstellung innerhalb dieser Arbeit beleuchtet werden.

1.1 Problemstellung

Die Implementierung von agilen Vorgehensmodellen in der Softwareentwicklung stellt nach wie vor ein Problem für Unternehmen dar [33]. Im Allgemeinen lässt sich dies als fehlgeschlagene oder unvollständige Adaption bezeichnen. Aus dieser Aussage lassen sich zwei Fragestellungen ableiten. Wann ist eine Adaption komplett fehlgeschlagen und wann unvollständig realisiert.

Im Kern der Adaption von Scrum kann eigentlich nie von einer vollständigen Adaption gesprochen werden, da sich der Prozess kontinuierlich weiterentwickelt [54]. Jedoch kann eine Adaption stagnieren, sodass eine weitere nicht durchführbar ist. Zudem ist Voraussetzung, dass die Werte und Regeln des jeweiligen agilen Verfahrens umgesetzt wurden [61, 16]. Wird dieser Punkt nicht erreicht, so liegt ein Fehlschlag der Implementierung nahe. Allerdings können trotz einer fehlgeschlagenen Adaption von agilen Vorgehensmodellen positive Ergebnisse erzielt werden, auf die in dieser Arbeit nicht weiter eingegangen wird [8]. Es lässt sich somit ableiten, dass der Grad der Adaption den Punkt erreichen muss, dass zumindest die Werte und Regeln gelebt werden und ein Umfang gegeben ist, nachdem eine weitere Optimierung erreicht werden kann.

Die Problematik der Adaption lässt sich somit auf zwei Punkte im Zusammenhang mit Scrum reduzieren. Zum einen müssen die wesentliche Punkte (Werte, Rituale, Rollen, Artefakte) von Scrum eingeführt werden. Zum anderen muss erreicht werden, dass weitere Optimierungen ermöglicht werden [16, 20, 60]. Der Übergang von beiden Punkten ist dabei fließend, da z. B. die

wesentlichen Punkte in einem Team oder einer Abteilung erreicht werden können, aber nicht alle notwendigen Stakeholder involviert werden müssen [29, 44]. In diesem Fall spricht man von einer erfolgreichen Implementierung nach Scrum, die hinreichend erfüllt ist. So können je nach Unternehmen Abstufungen der Implementierung eher Sinn machen, als wenn eine komplette Implementierung erfolgt. Durch den gegebenen Kontext ist das Unternehmen unter Umständen nur in der Lage, Teile realistisch umzusetzen [49].

Fest steht, dass es verschiedenste Hindernisse gibt, die einer Adaption im Weg stehen. Häufig stehen hierbei generelle Verständnis- und Akzeptanzprobleme im Raum. Hierdurch werden falsche Erwartungen von außen generiert. Auch kann das Festhalten an alten Strukturen Adaptionsversuche blockieren. Insbesondere können Probleme auftreten, die sich durch die existierende Unternehmensstruktur ergeben. Zu unterscheiden ist hier, ob Scrum oder andere agile Methoden von Firmengründung an genutzt werden oder ob vorher mit anderen Verfahren wie dem *V-Modell* gearbeitet wurde.

In beiden Fällen stellt sich die Frage, wie z. B. eine Scrum Initialisierung realisiert werden kann. Zudem stellt die langfristige Adaption eine besondere Problematik dar.

1.2 Lösungsansatz

Zu der im vorherigen Abschnitt dargestellten Problemstellung gilt es zu verstehen, welche Hindernisse in dem Schritt der initialen Implementierung sowie in der Adaption von Scrum liegen. Hierzu fokussiert diese Arbeit mehrere Faktoren, die in der Bewertung dieser Situation hilfreich sind.

Ein Punkt stellt das Verständnis der bisher verfolgten Modelle dar sowie die Spezialisierung, die dabei vorgenommen wurde. Je nachdem ob bereits agile Methoden oder traditionelle Verfahren verwendet werden, ergeben sich hierdurch bereits Faktoren, die zu beachten sind. Im Zusammenhang dazu steht auch das Verständnis, warum sich die Entwicklung verändern soll. Durch die Verschiedenheiten der Verfahren muss verstanden werden, was sich wie verändert, wenn eine Migration angestoßen wird. In diesem Prozess sowie nach der initialen Implementierung, steht das Team und ebenfalls das Unternehmen verschiedenen generellen Herausforderungen entgegen. Diese können bei einer vorherigen Betrachtung, im Veränderungsprozess berücksichtigt werden [49]. Im Zusammenhang der kontinuierlichen sowie der initialen Adaption stellt die Rolle des *Scrum Masters* eine entscheidende Rolle. Hinter dieser Rolle steht die Idee, einen Blick auf den Prozess von einer Metaebene zu ermöglichen [35, 53]. In diesem Zusammenhang können insbesondere Erkenntnisse aus der Praxis relevante Problematiken darstellen. Der

Scrum Master bietet hierzu eine Identifikationsmöglichkeit, um Herausforderung des Prozesses in Teams und Unternehmen zu lokalisieren [13].

Die gemachten Erfahrungen stellen dabei keine globale Antwort sondern Möglichkeiten dar, ein besseres Gefühl und die Fähigkeit zu erlangen, Herausforderungen zu erkennen. In Abhängigkeit dazu können durch empirische Erhebungen, über die Erfahrung von Scrum Mastern, weitere Punkte aufgezeigt. Somit ist die Ausrichtung der Agilität sowie traditioneller Komponenten darstellbar. In Relation zur Summe aller Erkenntnisse treten somit Konsequenzen auf, die zur Herangehensweise berücksichtigt werden müssen.

1.3 Struktur dieser Arbeit

Um die Herangehensweise zur Problemstellung umsetzen zu können, ist diese Arbeit wie folgt aufgebaut. Zunächst wird im Rahmen der Betrachtung von traditionellen Vorgehensmodellen der Softwareentwicklung auf den Bezug zu verschiedenen relevanten Punkten im Kapitel 2 eingegangen. Hierzu stellt vor allem die Softwarekrise ein wichtiges Ereignis dar. Des Weiteren lassen sich die Modelle der klassischen Softwareentwicklung als Phasenmodelle beschreiben. Unterteilbar sind diese unter anderem in das bekannte Wasserfallmodell. Dieses Kapitel wird mit der Darstellung von Gründen abgeschlossen, warum Unternehmen an diesen Methoden festhalten.

Trotz des Festhaltens an traditionellen Verfahren existiert eine Vielzahl von Faktoren, warum der Schritt zu agilen Verfahren getätigt werden sollte. Hierzu stellt das Kapitel 3 die Veränderung der Softwarebranche im Zusammenhang mit den Vorteilen, die sich durch agilen Verfahren bieten, dar. Um agile Verfahren besser verstehen zu können, wird im Kapitel 4 auf diese eingegangen, wobei hier das agile Manifest und verschiedene agile Modelle aufgezeigt werden.

Die angestrebte Umwandlung des Entwicklungsprozesses beinhaltet bei der Umsetzung jedoch verschiedene Punkte, die berücksichtigt werden müssen. Hierzu dient das Kapitel 5, welches zunächst die Unterschiede zwischen traditionellen und agilen Verfahren darstellt, um dann auf die Ansätze zur Transition weiter einzugehen. Das Kapitel stellt in Abhängigkeit dazu Erfahrungen aus der Praxis anhand von zwei durchgeführten Migrationen vor.

Während der Umstellung und nach der initialen Implementierung existieren Herausforderungen, die bewältigt werden müssen. Hierzu wird im Kapitel 6 näher auf verschiedene und besondere Faktoren eingegangen, die bei einer langfristigen Verfolgung hilfreich sind.

In Abhängigkeit zu den Hindernissen stellt die empirische Untersuchung dieser Arbeit im Kapitel 7 aktuelle Firmen im Raum Hamburg dar. Hierzu wird dargelegt, inwieweit diese zum aktuellen Zeitpunkt traditionelle oder agile Ausrichtungen in den Entwicklungsabteilungen aufweisen. Inhalt dieser Untersuchung ist dabei die Untersuchung selbst sowie die Auswertung der Ergebnisse.

Das Kapitel 8 stellt im Zusammenhang die Ergebnisse zur Herangehensweise einer Scrum Implementierung dar. Ziel dieses Kapitels ist dabei, dass Fragestellungen aufgezeigt werden, die im Bezug zur Transition beachtet werden müssen. Diese ergeben hierzu kein generelles Muster zur Lösung dieser Herausforderung. Vielmehr werden generalisierte Herausforderungen als Resultat dieser Arbeit wiedergegeben.

2 Traditionelle Vorgehensmodelle der Softwareentwicklung

Heutzutage existieren eine Vielzahl unterschiedlicher Vorgehensweisen zur Softwareentwicklung. Bei der Betrachtung dieser Methoden lassen sich diese in zwei grundsätzliche Kategorien einteilen. Das liegt zum einen an den Methoden an sich, die sich dabei sehr unterschiedlich mit der Entwicklung von Software auseinandersetzen, zum anderen aber auch unterschiedliche Philosophien darstellen. Schon B. Böhm hat 2004 diese Debatte zwischen traditioneller und agiler Softwareentwicklung in dem Werk „Balancing agility and discipline : a guide for the perplexed“ näher betrachtet und verarbeitet. [8]

Um diese Debatte näher zu verstehen, ist es hilfreich, die Entstehungsgeschichte der Softwareentwicklung näher zu beleuchten. Hierzu wird in diesem Kapitel zunächst geklärt, wie die Vorgehensweisen zur Entwicklung von Software entstanden sind. Ein Meilenstein dieser Entstehung stellt die Softwarekrise dar, nach der die klassischen Phasenmodelle entwickelt wurden. Die klassischen Phasenmodelle beinhalten dabei insbesondere das Wasserfall-, V- und das Spiralmodell. Des Weiteren werden die evolutionäre Entwicklung, wie auch der *Rational Unified Process* in dieser Arbeit als klassische Phasenmodelle bezeichnet.

2.1 Softwarekrise

In der Geschichte der Softwareentwicklung stellt vor allem der Beginn der Debatte um das „Software Engineering“, einen Meilenstein dar. Diese Zeit steht dabei eng in Verbindung mit der Entstehung des Begriffs der Softwarekrise, welcher sich zeitlich in die 70er und 80er einordnen lässt. [27]

Die damals verfolgten Methodiken wie Software entwickelt wird, stellten mit den weiterentwickelten Anforderungen und der damit eingehenden Komplexität immer stärkere Herausforderungen für die Entwickler dar. Infolgedessen stiegen die Fehler in den Softwaresystemen enorm an. Daraus ergab sich, dass stärker als zuvor der Bedarf entstand, fehlerfreie Software

zu liefern, wofür die bestehenden Verfahren allerdings unzureichend waren. Insbesondere wurden hierzu neue Techniken entwickelt, die eine verbesserte Aufnahme von Spezifikationen ermöglichten. Um den Bedürfnissen der Kunden besser gerecht zu werden, wurden *Usability* Methoden sowie *Design Reviews* näher betrachtet. Durch die Fokussierung beschleunigte sich die Entstehung modernerer Programmiersprachen. [27]

Es wurde allerdings auch erkannt, dass die Bereitstellung von fehlerfreier Software nicht realisierbar ist und es eher um das Bestreben einer möglichst fehlerarmen Software geht. Maßgeblich dazu beigetragen hat unter anderem die Nato-Konferenz 1968 in Garmisch, die den Begriff „Software Engineering“ geprägt hat. [27, 40]

Nach der durchgeführten Konferenz hat sich vor allem die Ansicht, wie Softwareprojekte durchzuführen sind, verändert. In Folge dieses Sichtwechsels entstanden die ersten Vorgehensmodelle, die auch heutzutage noch aufzufinden sind [27]. Ebenfalls lassen sich Bestandteile dieser Modelle in aktuell verfolgten Modellen wiederfinden, auf die unter anderem in den nachfolgenden Abschnitten eingegangen wird.

2.2 Klassische Phasenmodelle

Unter den klassischen Phasenmodellen versteht man vor allem die ersten entwickelten Modelle, um Softwareprojekte zu verfolgen. Das erste konkrete Modell, was dazu regelmäßig in Verbindung gebracht wird, ist das Wasserfallmodell. In diesem Modell, wie allerdings auch in nachfolgenden Vorgehensmodellen, werden Phasen bestimmt, die typischerweise im Entwicklungsprozess berücksichtigt und durchlaufen werden. Dabei unterscheiden sich vor allem die Modelle entscheidend indem, wie diese Phasen durchlaufen und im Prozess berücksichtigt werden. [52]

Das V-Modell, welches heutzutage auch noch in Form des V-Modells XT in Deutschland häufig anzutreffen ist, ist in seiner ersten Veröffentlichung aufbauend auf dem Wasserfallmodell entwickelt worden. Das Spiralmodell knüpft dabei an dem damals stark verbreiteten Wasserfallmodell an und stellte eine neue Art der evolutionären Entwicklung dar. Das Rational Unified Process Verfahren (kurz RUP) wurde Ende der 90er vorgestellt und insbesondere durch das von IBM vorgeschlagene Modell zur Verfolgung von Softwareprojekten präsentiert. [52]

2.2.1 Wasserfallmodell

Das Wasserfallmodell wurde erstmals 1970 von Dr. Winstons W. Royce im Beitrag „Managing the development of large software systems“ vorgestellt und ist zugleich auch mit das bekannteste Modell. Studiert man diesen entworfenen Beitrag, so fällt dazu besonders das klassische Modell auf der zweiten Seite auf, welches man sich meistens in Relation zum Wasserfallmodell vorstellt. Interessant ist dabei, dass auf den nachfolgenden Seiten das erweiterte Wasserfallmodell folgt. Dieses beinhaltet Schritte von einer, zur nachfolgenden Phase zu wechseln sowie den Weg von einer zur vorherigen Phase. In diesem gesamten Beitrag ist beschrieben, was für ein Risiko die jeweilig dargestellten Modelle beinhalten und warum eine Erweiterung sinnvoll ist. Dabei wird auch auf Verfahren eingegangen, wie die iterative und inkrementelle Vorgehensweise. Zusätzlich wird die Relevanz des Kunden dargestellt, welcher im Entwicklungsprozess zu berücksichtigen ist. [48]

Trotz der Erwähnung von Risiken und Hinweisen, wurde zunächst das einfache und später das erweiterte Wasserfallmodell weitergetragen. Dabei verbindet man damit insbesondere, dass aus jeder Phase Dokumente hervorgehen und erst nach dem Abschließen einer Phase die nächste begonnen wird. Dies bedeutet wiederum, dass besonders früh Verpflichtungen eingegangen werden und ggf. ein späteres Reagieren nicht mehr ermöglicht wird. In der Praxis wird dies mehr oder weniger gelebt. Zwar wird dabei grundsätzlich nach dem Modell entwickelt, es finden aber besonders in der Implementierungsphase Iterationen statt, die dann aber zum gegebenen Zeitpunkt eingefroren werden. [48, 52]

2.2.2 V-Modell

Das V-Modell wurde von Barry W. Boehm 1979 in der Veröffentlichung „Guidelines for Verifying and Validating Software Requirements and Design Specifications“ vorgestellt. Dieses Modell stellt im eigentlichen Ablauf ebenfalls das Wasserfallmodell dar. Hinzu kommt, dass je Phase eine Validierung und eine Verifikation durch spätere, nachfolgende Phasen erfolgt. Wie auch beim Wasserfallmodell werden dabei entsprechende Dokumentationen erstellt, die das Ergebnis einer Phase darstellen. [6]

Dieses Vorgehensmodell hat sich von 1979 an stets weiterentwickelt. Seit 1992 stellt das V-Modell, bekannt als V-Modell 92, einen Entwicklungsstandard für IT-Systeme des Bundes (EstdIT) zur Softwareentwicklung dar. Dieser Standard entwickelte sich nachfolgend zu V-Modell 97 und letztendlich zu V-Modell XT. Das V-Modell XT stellt insbesondere für öffentliche Auftraggeber in Deutschland den Standard zur Verfolgung von Softwareprojekten dar. Das be-

deutet, dass Softwareprojekte, sofern der Auftraggeber als öffentlicher Auftraggeber deklariert ist, nach V-Modell XT verfolgt werden müssen. [19, 24]

Auch heutzutage ist das V-Modell in der Ursprungsform anzutreffen. So existieren Ideen, wie diese Art der Entwicklung in Umgebungen genutzt werden können, wo einzelne Personen Verantwortung und Weisungsbefugnisse schätzen. Dabei wird das Modell durch weitere Umstrukturierungsmaßnahmen raffiniert sowie die Teamkonstellation darauf abgestimmt. [25]

2.2.3 Spiralmodell

Das Spiralmodell, welches 1988 von Barry W. Boehm veröffentlicht wurde, stellt die bereits in der Veröffentlichung „Managing the development of large software systems“ [48] erwähnten iterativen und inkrementellen Bestandteile dar. Zudem werden iterativ vier Phasen durchlaufen, die sich nach jedem Durchlauf weiter spezifizieren. Eine Besonderheit in diesem Modell ist der Fokus auf Risiken im Prozess. Hierzu wird eine ganze Phase benötigt, die iterativ zur Realisierung der Prototypen durchlaufen wird. [7]

Im Spiralmodell finden sich bereits Teile wieder, die sich mit der agilen Entwicklung verbinden lassen. Zu diesen gehören unter anderem die inkrementelle Entwicklung von Prototypen. Diese Art der Softwareentwicklung wird als evolutionäres Vorgehensmodell bezeichnet. [7]

2.2.4 Evolutionäre Modelle

Die evolutionäre Softwareentwicklung versteht sich als inkrementelle und iterative Vorgehensweise. Eine Software entwickelt sich demnach Stück für Stück weiter. Das bedeutet insbesondere auch, dass das Wissen, wie die spätere Software auszusehen hat, sich im Kontext des zeitlichen Ablaufes weiterentwickelt und damit auch die Anforderungen an die weiteren Prototypen beeinflusst. Es findet somit eine explorative Softwareentwicklung statt, in der durch die gemachten Erfahrungen im Prozess sich das Endprodukt bildet. Die entwickelten Prototypen werden dabei als Wegwerf-Prototypen bezeichnet. [52]

Besonders für das Management ist diese Art der Entwicklung aus ihrer Sicht nicht optimal. Der Grund liegt darin, dass der Prozess an sich nicht so sichtbar gestaltet wird, wie z. B. im Wasserfall oder V-Modell. Unabhängig davon stellen die evolutionären Modelle Grundzüge der agilen Entwicklung dar. Dabei wird die Erfahrung, die während des Entstehungsprozesses aufgebaut wird, direkt inkludiert. Zudem ist die engere Kooperation mit dem Kunden ein Bestandteil des Vorgehens, um ein verbessertes Softwareprodukt hervorzubringen. [52]

2.2.5 Rational Unified Process

Der Rational Unified Process (Kurz RUP) stellt im Vergleich der vorherigen Vorgehensmodelle einen moderneren Ansatz dar. Veröffentlicht wurde diese Methode 1998 im Werk „The Rational Unified Process: An Introduction“ von Philippe Kruchten und besteht grundsätzlich aus Phasen, Arbeitsabläufen und Vorgehensweisen. [34]

Dieser Entwicklungsprozess enthält unter anderem eine evolutionäre Herangehensweise, die ein inkrementelles und iteratives Verfahren ermöglicht. Dies ist je Phase oder über alle Phasen gesammelt erlaubt. Zudem existieren auch statische Perspektiven, die die Prozessaktivitäten darstellen, auch Arbeitsabläufe genannt. Die Theorie ermöglicht, dass alle Arbeitsabläufe parallel verarbeitet werden. Die dritte Komponente stellt die empfohlenen Vorgehensweisen zum Software Engineering beim Einsatz der Systementwicklung dar. [52]

Auch dieser Ansatz beinhaltet Teile aus dem Wasserfall-Verfahren, wie auch aus dem Spiralmodell. Es existieren sequenzielle Abläufe, die besonders durch eine traditionelle Sichtweise der Softwareentwicklung bedeuten kann, dass eher nach dem Wasserfall-Prinzip entwickelt wird. Allerdings zeigt diese Methode auch, dass die inkrementellen und iterativen Bestandteile keine neuen Herangehensweisen sind, die erst in der agilen Entwicklung ihren Ursprung haben.

2.3 Festhalten am Vorgehen

Bei der Betrachtung der in den vorherigen Abschnitten beschriebenen Modellen stellt man fest, dass diese viele Bestandteile und Muster beinhalten, die auch in agilen Modellen wiederzufinden sind. Trotzdem unterscheiden sich diese sehr stark in verschiedenen Punkten. Hierzu gehören nicht nur die Regeln, wie das jeweilige Vorgehensmodell im Detail verfolgt werden muss. Vielmehr stehen Aspekte in einem Kontrast, die unter anderem die Werte widerspiegeln, wie Menschen produktiv zusammenarbeiten können und dabei auch noch motiviert arbeiten wollen. Diese Aspekte werden im Kapitel 5.1 näher beschrieben.

Obwohl die agile Softwareentwicklung weiterhin einen enormen Zuwachs genießt, worauf auch im Kapitel 3 eingegangen wird, werden weiterhin traditionelle Verfahren verwendet. Es stellt sich demnach die Frage, warum weiterhin traditionelle Vorgehensmodelle verfolgt werden und warum manche Projekte diese klassischen Verfahren nutzen.

Auch heute existieren eine Vielzahl von Vorurteilen, warum agile Methoden nicht verwendet werden sollten. Dies liegt zum einen darin, dass Wörter wie Disziplin in den unterschiedlichen Sichtweisen (traditionell oder agil) unterschiedlich ausgelegt werden. So gilt es in traditionellen

Verfahren, dass ein etablierter Prozess ausgeführt wird und in der agilen Welt, dass durch eine Teamverantwortung die Entwicklung im Fokus des Ablaufs steht. Die Vorurteile von Methoden sowie die fehlerhafte Verwendung von Methoden verstärken dabei das Missverständnis von Entscheidern, die deshalb eher auf klassische Methoden setzen. [8]

Hierdurch entstehen z. B. Aussagen wie, dass der agile Ansatz eher einem „hacking“ entspricht. Diese Übergeneralisierung findet dabei auf beiden Seiten (traditionell und agil) statt. Hierdurch wird unter anderem ausgedrückt, dass „agil“ immer mit Scrum gleichzusetzen ist und „traditionell“ auf ein Wasserfallmodell zurückzuführen ist. Dies beschleunigt die Vorverurteilung enorm. [8]

Die Ansprüche der Allgemeinheit, auf die beiden stark unterschiedlichen Ausrichtungen, ist dabei, dass die agilen Verfahren schneller auf Veränderungen reagieren können und die traditionellen die Projekte besser vorhersehen können. [8]

Festzuhalten ist hierbei, dass keine Herangehensweise einer automatischen „Silver Bullet“ entspricht, sodass eine sofort gültige Lösung bereitsteht. Wie bereits im Abschnitt 2.2.1 angedeutet, entstehen durch die zu schnelle Beurteilung von neuen Verfahren falsche Wege, die verfolgt werden. Anstatt sich detailliert mit dem Abwägen der Vor- und Nachteile zu beschäftigen resultiert dies in einem falschen Verständnis der dargestellten Erkenntnisse und unter Umständen in einem Methodenmissbrauch. Dies sorgt dann wieder für eine verstärkte Vorverurteilung, wie z. B. im Fall von Scrum. Hiernach stellt Scrum das Äquivalent zum vorher verfolgten Modell dar. Die ausschließliche Veränderung besteht darin, dass die Dokumentation vernachlässigt wird und sich sonst nichts ändert. Wiederum sagen Erfolgsgeschichten nichts über langfristige Erfolge aus. [8]

Ebenso ist das Initialisieren von agilen Verfahren ein Faktor, der die Entscheider insoweit auf klassische Verfahren zurückführt, als dass eine „Ganz oder Gar nicht“-Strategie nicht immer der besten Einführungsmethodik entspricht. Prozesse können nicht ohne aktive Unterstützung von Teilnehmern migriert oder adaptiert werden. Hierzu muss die Philosophie dafür offen sein und ggf. eine Plan-getriebene Herangehensweise verfolgt werden. Das Management muss dabei unterstützend tätig sein. Verwendete Standards, die über Jahre im Unternehmen etabliert wurden, widersprechen unter Umständen der neu verfolgten Art und Weise der Softwareentwicklung. Regularien von externen Behörden verstärken zudem die Blockade gegenüber der Migrationsbereitschaft. Es existieren somit verschiedenste Risiken, was das Wissen über Agilität bedeutet, in der initialen und weiteren Adaption der agilen Softwareentwicklung. [8]

3 Motivation zur Veränderung

Wie bereits im vorherigen Abschnitt beschrieben, existieren weiterhin eine Vielzahl von Gründen, warum traditionelle Vorgehensmodelle verwendet werden. Durch gegebene Abhängigkeiten kann dadurch z. B. zeitnah keine Transition durchgeführt werden.

Es ist allerdings festzustellen, dass bereits Veränderungen stattgefunden haben, die in der Vergangenheit nicht mit bestehenden Verfahren beherrschbar waren. Zudem stehen Unternehmen der Gegenwart und Zukunft neuen Herausforderungen gegenüber, die über die rein traditionellen Stichweise weiter erschwert werden. Dies kann auch bedeuten, dass z. B. der Kostenrahmen eingehalten wurde, allerdings der Veröffentlichungszeitpunkt nicht dem angestrebten Zeitpunkt entspricht. Auch ist diesbezüglich ein Projekt als gescheitert anzusehen, wenn das entwickelte Produkt nicht dem wirklich benötigten Produkt entspricht.

Die im Rahmen der Marktentwicklung gegebenen Umstände sollen in diesem Kapitel näher beschrieben werden, wonach sich Veränderungen ergeben haben, die als besondere Herausforderungen für Firmen zu kennzeichnen sind. Diese stellen Gründe dar, warum ein Umdenken der bestehenden Modelle notwendig ist. Demnach bieten besonders agile Vorgehensmodelle verschiedene Vorteile, die ebenfalls in diesem Kapitel beschrieben werden.

3.1 Veränderung der Softwareentwicklung

Wie bereits in der Einleitung beschrieben, stehen heutige Unternehmen verschiedensten Herausforderungen gegenüber. Dabei ist unerheblich, ob diese der Softwarebranche zugeschrieben sind oder nicht. Unter anderem stellt die Entwicklung der globalen Weltwirtschaft eine solche Herausforderung dar. Diese sorgt z. B. dafür, dass sich der Wettbewerb verändert und verstärkt hat. Demnach können selbst neue Firmen mit Firmen konkurrieren, die bereits längerfristig am Markt bestehen. Durch die Globalisierung ist dabei ebenfalls der internationale Markt im Fokus. [2]

Zudem ist der Bedarf an Software deutlich gestiegen. Dieser Bedarf spiegelt sich dabei in jeder Branche wieder. Unternehmen sind gezwungen umzudenken und ihm Rahmen der weltweiten Digitalisierung neue Wege einzuschlagen. Unterstützt wird dieser Trend durch die Umgestaltung mit geeigneten Softwareprodukten. Das Resultat ist, dass der Softwaremarkt wächst [55].

Parallel verändert sich auch die Softwareentwicklung an sich. Durch die bereits genannten Punkte sind Softwareentwicklungs-Firmen gezwungen, sich neu auszurichten, um die zukünftigen Herausforderungen bewältigen zu können. Im Kontext dessen verändern sich die Projekte dahingehend, dass sie größer und komplexer werden [49].

Die Komplexität lässt sich durch verschiedene Punkte beschreiben. Hierzu zählt, dass die Unklarheit von Anforderungen und Spezifikationen zum festen Bestandteil der Entwicklung geworden ist. Im Konkreten bedeutet dies, dass Lasten- und Pflichtenhefte immer unkonkreter werden. Des Weiteren scheitern Kontrollmechanismen, die über statistische und andere Mittel versuchen, den Prozess vorauszusehen. Hierzu zählen unter anderem Verfahren, die versuchen Kosten und Aufwände im Voraus zu schätzen. [14]

Die Geschwindigkeit der gesamten Softwareentwicklung muss sich verändern. In der Vergangenheit wurde in Jahren gedacht, nun müssen die Produkte in Monaten umgesetzt werden. Firmen konkurrieren um den zukünftigen Markt, wonach sich Trends stetig ändern und wöchentlich neu ausrichten. [47]

Durch die Notwendigkeit, sich schnell am Markt ausrichten zu können, entstehen neue Anforderungen an die Unternehmen. Diesbezüglich sind schnelle Lieferungen von Werten an den Kunden gefragt [2]. Unter den Werten versteht sich hierbei die Lieferung von Ergebnissen aus der Softwareentwicklung, die kurzfristig Unternehmensergebnisse für den Kunden erzielen können. Es entsteht somit das Bedürfnis sofortige Resultate vorzuweisen [8]. Das Ergebnis sind hoch frequentierte Versionsveröffentlichungen, die dem Kunden kontinuierlich zur Verfügung gestellt werden [2].

Um auf den zukünftigen Märkten im Wettbewerb mit anderen Unternehmen bestehen zu können, sind Innovationen ein wichtiger Bestandteil. So muss es dem Softwareunternehmen möglich sein, dass Innovationen überhaupt entwickelbar sind. Hierzu müssen im Entwicklungsunternehmen selbst innovative Ansätze verankert werden, um selbst derartige Produkte hervorbringen zu können. Hierzu müssen Prozesse implementiert werden, die ein Management von Ideen gewährleisten. [2]

Bei der Betrachtung der bereits genannten Punkte wird deutlich, dass die effiziente Nutzung der Zeit einen wichtigen Faktor spielt. Zudem wirken weitere Punkte der Effizienz entgegen,

die bewältigt werden müssen. Die Darstellung der Unklarheit von Anforderungen und Spezifikationen zeigt, dass Zeit für die Definition von Funktionalitäten verschwendet wird. Die Definition beinhaltet dabei z. B. das Schätzen von Bearbeitungszeit und Kosten, die für eine fest definierte Funktionalität voraussichtlich notwendig sind. Zudem kommt die Modellierung komplexer Pläne und Strategien, bei der ebenso die Relation zu weiteren Funktionalitäten vermerkt werden. So wird z. B. Zeit für eine Funktionalität vergeudet, welche wahrscheinlich niemals entwickelt wird. [47]

Eine Lösung zum Gegenlenken ist, dass man die Möglichkeit zum Gegenlenken als koexistent etabliert. Hierdurch lässt sich erreichen, dass Anforderungsänderungen als gegeben angenommen werden. Als Konsequenz beinhaltet die Planung einer Aktion nur noch wenige Details einer Implementierung. Das Resultat ist, dass durch diese Art der Herangehensweise dramatisch weniger Zeit zwischen Planung und Durchführung bereit steht. [15]

Zusätzlich stellen aber selbst im Fall der „gut“ definierten Anforderungen die Art und Weise der klassischen Herangehensweise den Entwicklern ein Hindernis dar. Die Entwickler stellen trotzdem fest, dass die detailliert definierten Anforderungen nicht gut genug für eine direkte Implementierung sind. Ebenfalls sind diese Anforderungen nicht nach den alten Modellen direkt umsetzbar, wonach zum Gegenlenken ein verbessertes *Change Management* notwendig ist. Zusätzlich zu dieser Problematik werden große Listen von Anforderungen auf einmal in die Entwicklungsabteilungen überstellt. Durch die mangelnde Absprache zwischen Anforderungsmanagement und der Entwicklung ist für die Umsetzung im Team bereits klar, dass ein Großteil nicht direkt umsetzbar ist und ggf. weggelassen wird. Dies bedeutet wiederum, dass eine sehr ineffiziente Produktentwicklung betrieben wird. Die Softwareentwicklung wird demnach von einem Wandel und Änderungen überschüttet. [47]

Der Druck über die Geschäfts-Stakeholder ist dabei allerdings sehr stark. Demnach müssen Änderungen direkt und unverzüglich umgesetzt werden. [8, 55]

Anforderungen wechseln dabei stetig und sind als unklar oder unbekannt gekennzeichnet. Diese Punkte finden zudem während der Projektdurchführung statt und müssen bewältigt werden. Zusätzlich können auch neue Anforderungen im laufenden Projekt auftauchen. [8, 47]

Stetige Änderungen im Projekt erfordern ein Umdenken der bisherigen Feedbackkultur. Es sind iterative Zyklen notwendig, nach denen Feedback für Lösungen eingeholt werden [47]. Demnach muss die Rückmeldungszeit der Implementierung stark reduziert werden, um auch ein Abarbeiten zu gewährleisten [55].

Hierzu ist es ebenfalls notwendig, dass eine verbesserte Transparenz des Entwicklungsprozesses geschaffen wird, um eine Evaluierung realisieren zu können [2]. Der Kunde hat sich im Laufe der Zeit weiterentwickelt, muss und will an der Entwicklung stärker beteiligt werden. Hierdurch kann letztendlich ein besserer Mechanismus etabliert werden, der einem Abdriften vom gewünschten Produkt entgegenwirkt, wodurch eine Verbesserung der Lieferung realisiert wird [55].

Die Entwicklung versucht dem Chaos entgegenzuwirken, indem mehr Struktur aufgebaut wird [47]. Durch die Vielzahl von neuen Ausrichtungspunkten stellt dies aber eine Herausforderung dar, welche unterstützt werden muss. Hiernach wird gefordert, dass die Phasen besser definiert und Personen als Kontrollinstanz etabliert werden, um den Fortschritt zu erkennen und abgearbeitete Punkte abhaken zu können [47]. Die Beseitigung von Hindernissen muss demnach auch in einer prorsierten Reihenfolge durchgeführt werden, da Hindernisse nicht mehr als Ausnahme zu betrachten sind. [2]

Entwickler werden im Resultat der Veränderungen innerhalb der Softwareentwicklung einem deutlich stärkeren Druck ausgesetzt, als in der Vergangenheit. Um diesem Druck entgegenzuwirken, sind Veränderungen unabdingbar. [2, 55]

Es ist notwendig, dass Lernprozesse besser ausgebaut werden, sodass das Team besser und effizienter Herausforderungen entgegentreten kann. Hiernach ist auch der Prozess der Kreativität relevant, um Lösungen und Verbesserungen nachhaltiger erarbeiten zu können. [15]

Im Rahmen der gesamten Softwareentwicklung starten Projekte, die durch die nun gegebenen Herausforderungen gar nicht hätten starten dürfen. Es fällt ggf. erst gar nicht auf, dass das Projekt frühzeitig zum Scheitern verurteilt ist und somit wird dies teilweise erst nach Jahren bemerkt. [47]

Diese Art der Projektarbeit stellt letztendlich ein fragiles System dar, welches fehlschlagende Projekte nach sich zieht. Durch die ständig sich verändernden Rahmenbedingungen wird ein System benötigt, welches als antifragil zu bezeichnen ist. Hiernach stärkt sich das System durch die vorhandene Belastung und sieht den Wechsel als einen ständig gegebenen Fakt dar. Notwendig ist hierbei allerdings ein kontinuierliches Lernen und Entwickeln der Systembestandteile. Antifragile Systeme lernen somit aus Veränderungen und bieten eine schnellere Reaktionsgeschwindigkeit sowie eine Reduzierung der Fehleranfälligkeit. Agile Methoden finden sich genau in dieser Beschreibung wieder. [37, 54]

3.2 Vorteile agiler Vorgehensweisen

Agilen Vorgehensmodellen werden zahlreiche Verbesserungen zugeschrieben. An oberster Stelle steht das verbesserte Resultat im Hinblick auf die Qualität. [33, 59]

Unterstützt wird die Qualität dabei von einer begleiteten Performanz des gesamten Vorgehens [33]. Demnach wird das Endprodukt in einer erheblich kürzeren Zeit vorzeigbar gestaltet. Dies wird vor allem durch die kurzen Entwicklungszyklen erreicht [12, 55]. Dies bedeutet auch, dass eine frequentierte Lieferung von fertigen Inkrementen zum Kunden implementiert werden kann [59]. Das Ergebnis sind schnellere Versionsveröffentlichungen für die Stakeholder [10]. Hiernach kann in kleinen Schritten eine verbesserte Produkteinführungszeit (*Time-to-Market*) aufgebaut werden [58, 59].

Parallel wird innerhalb der Entwicklung eine Transparenz des Prozesses aufgebaut [33], wonach es für die Entwicklung bessere Feedback-Möglichkeiten gibt [43]. Dieses Feedback wird vor allem durch die kleinen Iterationen und die Kundeninteraktion gewährleistet [43]. In Bezug auf den Kunden bzw. den relevanten Stakeholdern, wird eine verbesserte Orientierung aufgebaut [33]. Hiernach steht ein stärkeres Einbeziehen der Kundenrolle im Vordergrund. Das Resultat dieses Vorgehens ist eine höhere Kundenzufriedenheit [55] sowie die Zufriedenheit aller Stakeholder (Benutzer, Kunden, Geschäftsmanager, Projektmanager, Entwickler, ...) [59]. Hierdurch wird auch ein verbesserter Beitrag zum Kundenwert realisiert, indem auf der Ebene der Entwickler ein beiderseitiges Verständnis zum Kunden aufgebaut wird [58]. Dabei kann der Kunde auch direkt vor Ort involviert werden [43].

Diese Art des Feedbacks, welches sich auch von der Entwicklung zum Kunden orientiert [43], bietet dem Kunden eine gewisse Kontrolle über den Prozess und das Produkt, sodass bei einer falschen Entwicklung rechtzeitig gegengelenkt werden kann [43].

Die Veränderung des Vorgehens hat auch zur Folge, dass eine Effizienzsteigerung zu verzeichnen ist [33]. Ebenfalls wird die allgemeine Produktivität des Entwicklungsteams gesteigert [12, 59], was bemerkbare Auswirkungen auf den Erfolg des Produkts bedeutet [12].

Die gesteigerte Produktivität kommt auch zu Stande, da eine verbesserte Kostenbegrenzung vorgenommen wird [59]. Dies wird unter anderem durch die Minimierung von unnötiger Arbeit erreicht [14].

Der gesamte Softwareentwicklungsprozess wird vereinfacht [12]. Das bedeutet, dass dieser leichtgewichtiger wird und in Summe mehr fertiggestellte Aufgaben hervorbringt [58].

Ebenfalls werden Störfaktoren reduziert. Hierzu gehören z. B. ungenaue Projektanforderungen, die gegeben sind oder sich im Laufe des Projekts ergeben können [59]. Der Fokus der Anforderungen und des Designs wird dabei als ein kontinuierlicher Entdeckungsprozess verstanden, der als Herausforderung zu bewältigen ist [59]. Das Resultat hieraus ist auch, dass weniger Dokumentation erstellt wird [46]. Zu verstehen ist hierbei nicht der Verzicht von Dokumentation an sich, sondern die Reduzierung von unnötiger Dokumentation.

Die kontinuierliche Veränderung des Prozesses soll auch den Prozess des Lernens anregen [58]. Dessen Ergebnisse sollten auch auf andere Teammitglieder und Stakeholder im Rahmen eines Wissenstransfers übertragen werden [43]. Die Betrachtung des Wissenstransfers ermöglicht dem Team, dass Aufgaben nicht nur von einzelnen Personen bearbeitet werden können, sondern ebenfalls ein besseres Gesamtverständnis des Produkts erzeugt wird.

Durch die stetigen Veränderungen ist es notwendig, dass eine flexiblere Planung möglich ist, die über agile Verfahren erreicht wird [33]. Hierdurch kann auch ein flexibleres Design konstruiert werden [10]. [46]

Das Resultat ist eine Adpatierungsmöglichkeit in der Methode an sich [55]. Durch diesen Aufbau ist es möglich, dass Änderungen angenommen werden, die auch realistisch verarbeitbar sind [58]. Im Sinne der Marktdynamik werden somit unvermeidliche Änderungen beherrschbar [12], was auch eine Verbesserung des *IT Business Alignment* bedeutet [42]. Das *IT Business Alignment* beschreibt die zwingende Abstimmung der Geschäfts- sowie IT Bereiche, die beim Erstellen und Ausliefern von Softwareprodukten notwendig sind. Es entsteht eine Reaktionsfähigkeit zu angestrebten Geschäftsänderungen [42], was im Resultat sofortige und grundsätzliche Änderungen einschließt [26].

Dieser Sprung zwischen den Bereichen wird vor allem durch die effektivere und verbesserte Kommunikation hervorgerufen. Zu sehen ist die Kommunikation darüber hinaus auch zu weiteren Stakeholdern und die Kommunikation im Team. [10, 42, 43, 55]

Unter anderem wird durch die verbesserte Kommunikation auch die Zusammenarbeit in der Entwicklung positiv beeinflusst. Dies beinhaltet eine positive Auswirkung auf die Koordination im Team [10]. Das Team pflegt einen respektvolleren Umgang miteinander was auch ein gesteigertes Vertrauen hervorbringt [43]. Darüber hinaus wird bei der Betrachtung von tangierenden Bereichen auch eine optimierte Zusammenarbeit verzeichnet [46]. [14, 33, 42]

Diese gesammelten Punkte sorgen dafür, dass die Motivation der Mitarbeiter gesteigert wird [33] und eine erhöhte Moral festzustellen ist [59]. Insgesamt ist ein verbessertes Arbeitsleben zu verzeichnen [43], wodurch ein angemessenerer Entwicklungsprozess für die Entwickler geschaffen wird [10].

4 Agile Vorgehensmodelle

Die moderne Softwareentwicklung definiert sich heutzutage besonders durch den Begriff der agilen Softwareentwicklung. Der Grund hierfür liegt insbesondere auch in den im vorherigen Kapitel 3 beschriebenen Gründen in der Veränderung der Softwareentwicklung wie auch in den Vorteilen, die die agilen Methoden bieten.

Der Hintergrund des Verfahrens zur Entwicklung von Software lässt sich insbesondere auf das agile Manifest referenzieren, auf das in diesem Kapitel nachfolgend eingegangen wird. Dabei wird hierzu ein genereller Überblick dargestellt und folglich näher auf das Verständnis hinter den Werten eingegangen.

Im Anschluss soll ein Fokus auf verschiedenen Modelle gegeben werden, die besonders häufig mit der agilen Softwareentwicklung in Verbindung gebracht werden. Auch in diesem Kapitel gilt, dass ein Überblick gegeben wird. Dabei wird auch auf Verbindungen der Methoden untereinander eingegangen sowie einige wichtige Besonderheiten, die es zu verstehen gilt.

4.1 Agile Werte hinter dem agilen Manifest

Das agile Manifest oder auch „Manifesto for Agile Software Development“ stellt das Grundwerk der Definition zur agilen Softwareentwicklung dar. Dieses wurde 2001 von mehreren erfahrenen Autoren verfasst, die sich zusammen auf eine allgemeine Definition zur Agilität verständigt haben. Inhalte dieses Manifests sind unter anderem die vier Werte, nach der die agile Softwareentwicklung verfolgt werden sollte. [18]

- „Individuals and interactions over processes and tools“
- „Working software over comprehensive documentation“
- „Customer collaboration over contract negotiation“
- „Responding to change over following a plan“

[4]

Zusätzlich zu den dargestellten Werten existieren zwölf agile Prinzipien, die eine Verfeinerung der vier Werte repräsentieren. Zu dem Manifest zählen dabei vier Werte und zwölf Prinzipien auf denen die nachfolgenden agilen Modelle basieren. Die zwölf Prinzipien ermöglichen es, dass das Verständnis der Werte weiter ausgebaut wird. [5, 49]

Bei der Betrachtung der Gründe, warum agile Projekte fehlschlagen, stellt allerdings die Verletzung der agilen Kernwerte immer noch eine Hauptursache dar. Nun kann das daran liegen, dass entweder die Werte nicht verfolgbar sind, weil die Umgebung des Teams dies nicht zulässt oder die Erfahrung bzw. das Verständnis hierzu fehlt. Beide Punkte lassen sich ebenfalls in der Liste der Gründe für fehlschlagende agile Projekte wiederfinden. [56]

Eine der größten Gefahren liegt vor allem in dem falschen Verständnis der Werte. Trotz der weiteren Prinzipien führt eine falsche Vermittlung dazu, dass sich die Agilität verschlechtert und ebenfalls falsch verfestigt. Das bedeutet auch, dass sich, wie im Abschnitt 2.3 beschrieben wurde, Vorurteile ausbauen können, um je nach Standpunkt den Entwicklungsprozess passend auszulegen. Zu Verstehen ist, dass die Werte grundsätzlich streng zu verfolgen sind. Das heißt aber auch, dass der Gegenpunkt des Wertes nicht komplett ausgeblendet wird.

Ein Beispiel hierzu ist „Working software over comprehensive documentation“. Die funktionsfähige Software steht über einer umfassenden Dokumentation. Das Ziel ist, wie immer in der Softwareentwicklung, eine Software auszuliefern, die der Kunde angefordert hat. Es erscheint logisch, dass besonders die Dokumentation nicht wichtiger ist, als die Auslieferung. Es ist aber je nach Projekt notwendig, dass entsprechende Dokumentation an verschiedensten Stellen existieren, um z. B. die Wartung zu gewährleisten oder Abnahmen für externe Regulierungsstellen zu realisieren. Dokumentationen sollen hiernach bei Bedarf aufgebaut werden, sofern auch zu einem späteren Zeitpunkt ihre Daseinsberechtigung existiert.

Es gilt somit, dass das richtige Gleichgewicht jeder individuellen Situation anzupassen ist, aber nicht grundsätzlich eine Seite komplett vernachlässigt und als Argumentation zur agilen Softwareentwicklung missbraucht wird. Dabei gilt, dass alle Werte betrachtet und zu gegebenen Zeitpunkten überprüft werden. Der Transfer des korrekten Verständnisses stellt zudem eine essentielle Aufgabe dar, die besonders auch für Stakeholder aus den verschiedenen Managementebenen zu realisieren ist. Hierzu stellt das agile Manifest eine Denkweise dar, die auch für andere Projekte außerhalb der Softwareentwicklung nützlich sind. [1, 23]

4.2 Fokus agiler Modelle

Wie im vorherigen Abschnitt beschrieben, stellt das agile Manifest eine Denkweise und Definition der Agilität dar, worauf vor allem agile Entwicklungsmodelle aufbauen. Eine der bekanntesten Modelle stellen dabei Extreme Programming (kurz XP), Kanban und Scrum dar. Diese Verfahren zur Softwareentwicklung werden nachfolgend näher beschrieben. Dabei wird keine komplette Definition der Modelle dargelegt. Der Fokus liegt auf der Beschreibung der Idee und auf den Bezug zur agilen Softwareentwicklung.

Bei dieser Betrachtung stellt sich in Bezug auf die Herangehensweise zur Implementierung dieser Methoden die Frage der Adaptionsfähigkeit. Je nach Auswahl der Methode existieren bereits Vorschriften, die verfolgt werden müssen, sofern eine Implementierung nach der Definition der Vorgehensweisen erfolgen soll. Die Abbildung 4.1 stellt die Modelle RUP, XP, Scrum, Kanban und bei der Verfolgung keines Modells, in ein Verhältnis. Betrachtet werden dabei die vorhandenen Regeln im Modell (in den Klammern dargestellt) und im Gegensatz dazu die Adaptionsmöglichkeiten. So ist ein Vorgehensmodell flexibler, um so weniger Regeln und Vorschriften existieren. Dies sagt allerdings nichts darüber aus, wie erfolgversprechend die jeweilige Methode ist.

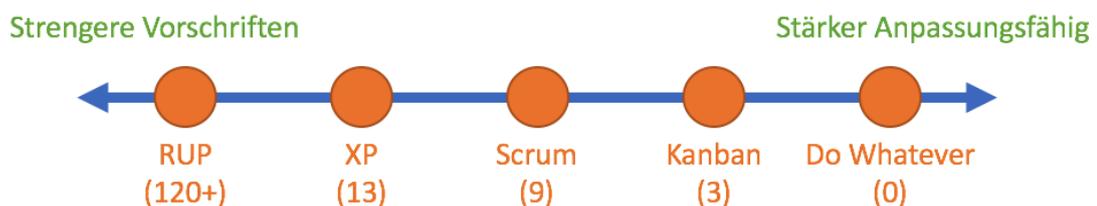


Abbildung 4.1: Vergleich der Vorschriften und Adaptionsfähigkeit von Entwicklungsmodellen [32]

Wie zuerkennen ist, stellt RUP dabei eine Vielzahl von vordefinierten Regeln. Interessant ist auch der Unterschied zwischen XP, Scrum und Kanban. So stellt Kanban bessere Fähigkeiten dar, sofern weitere Regeln im Kontext der Entwicklungsmethodik vorhanden sind.

Die nachfolgenden Abschnitte sollen dabei auch auf diesen Punkt der Adaptionsfähigkeit eingehen.

Im Zuge dieser Arbeit wurde im Kapitel 2 bereits auf die klassischen Modelle eingegangen. Der Ansatz des RUP stellt dabei ein Modell dar, welches sich durch die Anzahl an Vorschriften besonders hervorheben lässt. Trotzdem stellt dieses Modell bereits einige Praktiken dar, die sich auch in der agilen Welt wiederfinden lassen. Im Vergleich zu XP ist hier ein deutlicher

Sprung zu erkennen. Die agile Methode XP stellt auch eine Besonderheit in der agilen Welt dar, auf die im nächsten Abschnitt eingegangen werden soll.

4.2.1 Extreme Programming

Das Vorgehensmodell Extreme Programming (kurz XP) stellt eines der ersten agilen Methoden der neueren Generation der Softwareentwicklung dar. Entwickelt wurde dies dabei im Rahmen eines realen Projekts (Chrysler C3) von Kent Beck, der damit 1996 startete. Ken Beck gehört zudem zu den Unterstützern und Entwicklern des Agilen Manifests. Er erkannte damals bereits die Schwierigkeiten in Projekten, die sich schnell ändernden Spezifikationen sowie die Bedeutung der Nähe zu den Stakeholdern in der Entwicklungsphase. [22]

Eine weitere Besonderheit dieses Modells stellt die Einteilung in Werte und Prinzipien dar. Insgesamt werden fünf Werte und vierzehn Prinzipien benannt, die den Prozess beschreiben. Es fällt auf, dass besonders diese Einteilung Ähnlichkeiten zum agilen Manifest aufweist. So ist auch beim Analysieren dieser Werte und Prinzipien zu erkennen, dass es mehr Überschneidungen gibt, die auf die Entwicklung der agilen Werte und Prinzipien schließen lassen. [22]

Dieses Vorgehensmodell beinhaltet zudem weitere Punkte, die auch in anderen agilen Methoden wiederzufinden sind. Hierzu gehört unter anderem, dass versucht wird, den Kunden möglichst eng in die Entwicklung zu involvieren. Neben diesen Merkmalen bietet XP eine Reihe von Praktiken, die in verschiedenen Phasen behilflich sein sollen. Zu diesen gehört z. B. das sogenannte *Pair-Programming*, indem in zweier Teams versucht wird, gemeinsam den Programmcode zu entwickeln. Die Aneinanderreihung der Phasen stellt dabei eher Wasserfall-Ähnlichkeiten dar. Die Praktiken allerdings bieten Verfahren und Richtlinien die im Einzelnen betrachtet für konkrete Vorschläge zur Umsetzung einer verbesserten Softwareentwicklung hilfreich sein können. Diese können dabei auch losgelöst von XP genutzt werden, was in der Praxis agiler Softwareentwicklung auch geschieht. Durch ihre isolierte Einsetzbarkeit stellen diese besonders in agilen Methoden eine Ergänzung dar, die keinen Widerspruch zu den Regeln des jeweiligen Prozesses darstellen. [22]

4.2.2 Kanban

Kanban wurde um 1947 im Rahmen eines Optimierungsprozesses bei Toyota entwickelt und stellt besonders im Aspekt der Adaptionfähigkeit ein interessantes Entwicklungsmodell dar. Dies wird dadurch erreicht, dass wenige Richtlinien vorhanden sind. Hierdurch wird es

möglich, dass Kanban wesentlich besser in eine bestehende Entwicklung integriert werden kann. Darüber hinaus kann eine Anwendung sogar in weiteren Umgebungen stattfinden, z. B. in der Abwicklung von Support-Anfragen innerhalb des IT-Betriebs. [17, 32]

Die Grundlage von Kanban stellt das sogenannte „Taskboard“ dar, welches zur Visualisierung des Arbeitsprozesses dient. Diese Visualisierung stellt die erste existierende Regel dar. Insgesamt besteht Kanban aus drei bzw. sechs Prinzipien. In der Ursprungsversion von Kanban sind folgende Regeln vorzufinden:

- „Visualize the workflow“
- „Limit Work In Progress (WIP)“
- „Manage Flow“

Im Rahmen der Veröffentlichungen von David J. Anderson, der als Mitgründer von Kanban in der IT-Welt gilt, wurden zwei weitere Prinzipien vorgestellt:

- „Make Process Policies Explicit“
- „Improve Collaboratively (using Models and the Scientific Method)“

In gewisser Weise stellen die zwei weiteren Regeln eine Erweiterung der dritten Regel („Manage Flow“) dar. Diese wurde entsprechend weiter spezifiziert und mit klareren Anweisungen erweitert. [17, 32]

Zu den vorgestellten Praktiken stellt Kanban auch den Anspruch an sich, eine evolutionäre sowie inkrementelle Veränderung des Prozesses zu ermöglichen. Eine weitere Besonderheit, die sich auch in anderen agilen Methoden wiederfindet, stellt das *Pull*-Verfahren dar. Hierbei werden Aufgaben von Mitarbeitern selbstständig zur Bearbeitung herangezogen und nicht, wie häufig in anderen Vorgehensmodellen anzutreffen, delegiert (*Push*-Verfahren). Diese Betrachtung der Arbeitsweise sowie weitere Aspekte von Kanban, finden sich ebenfalls im Scrum Modell wieder, auf das nachfolgend eingegangen wird. [17, 32]

4.2.3 Scrum

Das Vorgehensmodell Scrum stellt aktuell eines der erfolgreichsten Vorgehensmodelle dar. Laut einer Studie liegt Scrum im Vergleich agiler Methoden bei einer Abdeckung von 58 % [57]. Entwickelt wurde Scrum in den frühen 1990er Jahren von Ken Schwaber und Jeff Sutherland, wobei die ersten Grundgedanken erst 2004 niedergeschrieben worden. Die Idee basierte dabei auf den sogenannten *Lean Prinzipien*. [22]

Wie bereits im Kapitel 3 beschrieben, gib es eine Reihe von Gründen, warum agile Methoden, vor allem Scrum, bevorzugt werden. Scrum vereint dabei viele Grundsätze der agilen Prinzipien. Hierzu gehören die Transparenz, Inspektion und Anpassung des Vorgehens. Zudem existieren gewisse Werte, die innerhalb des Scrum Modells verfolgt werden, wie das Fokussieren der Tätigkeit oder die gemeinsam vereinbarte Verpflichtung (*Commitment*). Die Offenheit, wie auch der Respekt und die Courage der Teammitglieder stellen einen gemeinsamen Wertepool zur Entwicklung dar. [21, 51]

Das Gerüst dieses Vorhensmodells besteht aus drei essenziellen Unterscheidungen. Hierzu zählen Rollen, Rituale und Artefakte. Die Rollen beschreiben exklusiv die Teammitglieder und nicht die Stakeholder außerhalb des sogenannten *Scrum Teams*. Rituale beschreiben Treffen, um unter anderem eine Planung von sogenannten *Sprints* zu ermöglichen. Ein Artefakt stellt im Scrum Kontext z. B. ein Backlog dar, welches die Verwaltung und Abarbeitung von Aufgaben ermöglicht. Diese Unterscheidungen stellen im konkreten Zusammenhang mit grundsätzlichen Werten und Begründungen die Richtlinien dieses Modells dar. [51]

Scrum selbst stellt dem Entwicklungsprozess im Vergleich zu Kanban zwar mehr Leitlinien zur Verfügung, bietet aber in der Umsetzung der Aufgaben auch weiterhin viele Freiheiten. So können Verfahren verwendet werden, wie sie im Rahmen des Extreme Programmings angeboten werden oder im Konkreten ein Taskboard, wie es in Kanban existiert. Diese Freiheit bedeutet dabei auch, dass Grundsätze des Vorgehensmodells ihre Daseinsberechtigung besitzen und vollkommen verstanden und analysiert werden sollte, warum man sich gegen einen Leitsatz stellt. Insbesondere können hierdurch eine Reihe von Adaptionsbremsen im Prozess entstehen, die es zu vermeiden gilt. Eine detailliertere Behandlung dieser Thematik wird im Kapitel 6 vorgenommen.

Die Umstellung zu agilen Methoden wie Scrum erfordert eine Reihe von Aktionen, die in erster Linie auch eine Analyse von verschiedenen Ansätzen beinhalten. Durch diese Analyse kann erst verstanden werden, wie eine Implementierung von agilen Ansätzen stattfinden kann. Einige grundsätzliche Herangehensweisen zur Umwandlung und Eingliederung von agilen Vorgehensmodellen werden im nachfolgenden Kapitel 5 näher beleuchtet.

5 Umwandlung des Entwicklungsprozesses

Ist die Motivation im Unternehmen stark genug, dass der Bereich der Softwareentwicklung sich verändern und sich das Verfahren in eine agile Richtung transformieren soll, so steht dies verschiedensten Herausforderungen gegenüber. Einen wichtigen Punkt stellen hierzu die Gründe dieses Willen dar, welche unter anderem im Abschnitt 3.1 beschrieben werden.

Zusätzlich zur generellen Veränderung der Gegebenheiten der Softwareentwicklung, die Veränderungen der eher klassischen Methoden bedeuten, existieren die agilen Modelle, die eine Reihe von Vorteilen bieten. Diese Vorteile, die sich mit der agilen Softwareentwicklung ermöglichen lassen, werden im Abschnitt 3.2 ausführlicher erläutert.

Wie bereits in den vorherigen Kapitel beschrieben, stellt das Verständnis der Vorgehensmodelle einen wichtigen Aspekt dar. So sollte vor der Entscheidung der Migration bekannt sein, was sich alles verändern wird. Hierzu ist es hilfreich, die bisher verfolgte Methode sowie die angestrebte Vorgehensweise zu verstehen. Dabei entstehen in der generellen Betrachtung der traditionellen und agilen Vorgehensmodelle zusätzlich Gegensätze, die in diesem Kapitel näher beleuchtet werden.

Nach dieser generellen Gegenüberstellung gilt es zu verstehen, welche Ansätze es gibt, die verfolgt werden können, um diese Transition anzustoßen. Hierzu wird im Abschnitt 5.2 auf zwei generelle Richtungen eingegangen. Zusätzlich dazu wird ein Verfahren zur Realisierung einer kontextabhängigen Transition vorgestellt, indem die Abhängigkeiten des Unternehmens mit den angestrebten Zielen verglichen werden.

Der darauf folgende Abschnitt stellt zwei Fallstudien vor, die die Transition von klassischen Modellen zu Scrum begleitet und beschrieben haben. Hierbei wird jedoch nicht auf eine detaillierte Beschreibung jedes Schritts eingegangen und eine universelle Lösung des jeweiligen Modells dargestellt. Vielmehr werden die Herausforderungen innerhalb dieser Migration verdeutlicht und die Veränderungen, die nach dieser Umstellung sichtbar gemacht wurden.

Im letzten Abschnitt dieses Kapitels steht das Fazit. Dieses Fazit fasst die Ergebnisse nochmals zusammen.

5.1 Unterschiede der Vorgehensmodelle

Untersucht man die Unterschiede zwischen traditionellen und agilen Entwicklungsprozessen, so gibt es verschiedene Aspekte die zu betrachten sind. So kann in der Methodik gegenübergestellt werden, ob das Modell ein evolutionäres Vorgehen besitzt und auf einen inkrementellen sowie iterativen Ansatz zielt. Ebenfalls lassen sich Elemente, wie Rollen vergleichen, wodurch ein bestimmter Fokus erkennbar ist. Rituale bieten Auskunft darüber, welche Meetingstrukturen vorherrschen und inkludieren bestimmte Stakeholder-Gruppierungen. Damit lässt sich wiederum aussagen, wie mit Wissen, Verantwortung und Befugnissen umgegangen wird. Spezielle Artefakte lassen sich ebenfalls vergleichen und geben Fokussierungen des Ablaufes wieder.

Bei einer Gegenüberstellung von unterschiedlichen Modellen bieten sich somit verschiedenste Punkte, die verwendet werden können. Dabei können bereits erlangte Erkenntnisse aus bestehenden Arbeiten verwendet und kombiniert werden, um eine detailliertere Untersuchung zu ermöglichen. Diese Arbeiten verwenden dabei unterschiedliche Fokussierungen bei der Erarbeitung von Vergleichsfaktoren. Grundsätzlich lässt sich hier zwischen Theorie- und Praxis-geleiteter Quellenart unterscheiden.

Die im Folgenden dargestellten Faktoren beziehen sich auf einen generellen Unterschied zwischen traditionellen und agilen Methoden. Diese beziehen sich z. B. auf die Verfolgung der Dokumentation. Die Gegenüberstellung dieser Faktoren wird zusammengefasst in der Tabelle 5.1 präsentiert. Erarbeitet wurden diese Faktoren maßgeblich über die Arbeit von Conboy u.a. [11], welche sich wiederum auf die Arbeiten von Nerur u.a. [41] und Schuh [50] bezieht. Dabei verweisen alle Werke wiederum auf das Werk von Boehm und Turner [8]. Die Quellen können dabei als Theorie betrachtet werden, die durch eine geleitete Praxisbetrachtung entworfen wurde. So wird insbesondere im Werk von Boehm und Turner [8] auf Abweichungen zwischen traditionellen und agilen Verfahren eingegangen, die sich auf die Realität beziehen. Insgesamt kann zu den Quellen zusätzlich ausgesagt werden, dass die Faktoren zur Darstellung der Unterschiede aus unterschiedlichen Erkenntnissen gewonnen wurden, wobei sich auch auf das agile Manifest [5] bezogen wird.

Die nachfolgenden Faktoren in Tabelle 5.1 beziehen sich im Gegensatz zu den bereits angedeuteten Unterscheidungsmöglichkeiten in den Kapiteln 2, 3 und 4 auf umfassendere und konkret gegenübergestellte Punkte. Diese Tabelle sowie die nachfolgende nähere Beschreibung der einzelnen Ausrichtungen wurde aus den eben vorgestellten Werken und Arbeiten abgeleitet

und weiter verfeinert. Unter anderem lassen sich so Diskrepanzen in der Rollenbesetzung oder den ausgerichteten Ritualen beschreiben.

Die Projektkomponente umfasst dabei den Faktor, der jeweils traditionell oder agil auslegbar ist. Die Tabelle fasst zusätzlich zusammen, wie sich jeweils eine Projektkomponente traditionell bzw. agil beschreiben lässt.

Tabelle 5.1: Unterschiede zwischen traditionellen und agilen Methoden [8, 11, 41, 50]

Projektkomponente	Traditionell	Agile
Entwicklungsmodell	Lebenszyklus Modell (Wasserfall, Spiral, oder Variationen)	Evolutionäres Bereitstellungsmodell
Projektzyklus	Bestimmend durch Aufgaben und Aktivitäten	Bestimmend durch Produktfunktionalitäten
Projektplanung	Im Voraus	Kontinuierlich
Managementstil / -kultur	Kommando und Kontrolle	Leader und Zusammenarbeit sowie responsiv
Rollenzuordnung	Individuen - Favorisierte Spezialisierung	Selbst organisiertes Team - Begünstigt Rollenaustauschbarkeit untereinander
Anstrebende Organisationsform oder -struktur	Mechanistisch (Bürokratie mit hohen Formalismus)	Organisch (Flexibel und teilnehmend, begünstigt kooperative soziale Interaktionen)
Steuerung	Prozess-zentriert	Menschen-zentriert
Dokumentation	Substantiell	Minimal
Wissensmanagement	Explizit (Zwingende Dokumentation)	Taktisch
Kommunikation	Formal und nur wenn notwendig	Informell und Kontinuierlich
Teamnähe	Vorwiegend verteilt	Vorwiegend zusammenhängend
Teamgröße	Oft größer als 10	Normalerweise weniger als 10
Teambeteiligung	Nicht vorgeschrieben	Notwendig
Kundenbeteiligung	Normalerweise nur während der Projektanalyse wichtig	Kritisch und Kontinuierlich
Feedbackmechanismus	Nicht einfach beschaffbar	Gewöhnlicherweise sind mehrere Mechanismen vorhanden
kontinuierliches Lernen	Nicht regelmäßig aufgefördert	Fester Bestandteil

Die erfasste Tabelle 5.1 stellt nur eine Kurzfassung der Punkte dar. Diese Faktoren werden nun nacheinander erläutert. Dabei wird zunächst nur die Projektkomponente beschrieben. Danach folgt jeweils die Betrachtung zur traditionellen Sichtweise und dann die der agilen Betrachtung. Die Beschreibung fällt bei der jeweiligen Komponente und der jeweiligen Sichtweise unterschiedlich lang aus, da die Punkte an sich unterschiedlich komplexe Tiefen besitzen.

Entwicklungsmodell

Das Entwicklungsmodell stellt grundsätzlich nur die Auswahl des gewählten Vorgehensmodells dar. Die Wahl des Modells kann dabei bedeuten, dass sich das Team oder Unternehmen einfach nur für ein Verfahren entschieden hat. Es bedeutet nicht, dass dieses korrekt oder vollständig verfolgt und verstanden wurde. [41]

Traditionell

Traditionell werden insbesondere Modelle verwendet, die auf das Wasserfall Modell reduziert werden können. Des Weiteren finden sich ebenfalls Modelle wieder, die dem Spiral Modell zuzuordnen sind oder Variationen von beiden Modellen entsprechen. Besonders auffällig sind in diesen Modellen oder Varianten die festdefinierten Phasen, die nacheinander oder wiederholend mit unterschiedlicher Granularität verfolgt werden. [41]

Agil

Agile Vorgehensweisen verstehen sich als evolutionäre Modelle. Hierunter ist zu verstehen, dass das Ziel erst über die Zeit geschärft wird und zunächst nicht in einer hohen Detailtiefe bestimmt wird, was als Nächstes zu tun ist. Um das zu ermöglichen werden Funktionalitäten oder Aufgaben in kleine Stücke zerteilt und somit inkrementell und iterativ zur Funktionalität bzw. zum Softwareprodukt zusammengesetzt. Man spricht meist von evolutionären Bereitstellungsmodellen. [41]

Projektzyklus

Der Projektzyklus beschreibt den Prozess, nachdem das Softwareprojekt realisiert werden soll. Unter dem Prozess versteht sich dabei nicht das Vorgehensmodell an sich. Die verwendeten Verfahren um Aufgaben zu verwalten und abzuarbeiten lassen sich allerdings Vorgehensmodellen zuordnen. [41]

Traditionell

Bei der Betrachtung von traditionellen Entwicklungsumgebungen wird der Projektzyklus durch Aufgaben und Aktivitäten bestimmt. Dabei werden diese für die jeweilige Phase definiert sowie das entsprechende Ergebnis spezifiziert. [41]

Agil

Agile Entwicklungsmodelle zeichnen sich vor allem dadurch aus, dass sie von Produktfunktionalitäten getrieben werden. Dies bedeutet, dass das nächste Inkrement beispielsweise bestimmte Funktionalitäten besitzt, die in der Iteration zu diesem Inkrement umgesetzt werden sollen. [41]

Projektplanung

Unter der Projektplanung ist zu verstehen, dass es um die Art und Weise geht, wie Aufgaben definiert werden. Hierzu gehört z. B. der Detailgrad in dem Anforderungen sowie deren Abnahmekriterien formuliert werden. Ebenfalls ist der Zeitpunkt der Erfassung als besonderer Punkt zu erwähnen. Dieser kann entweder versucht werden, lange im Voraus bestimmt zu werden oder im extremen Fall gar nicht oder kurz vor der Ausführung erfolgen. [8, 41]

Traditionell

Innerhalb der traditionellen Projektplanung wird vor allem im Voraus geplant. Pläne werden dabei genutzt, um Prozesse zu verfestigen und ein weites Spektrum an Kommunikation zu ermöglichen, was wiederum große Mengen an Dokumentationen benötigt. Hierzu gehören Zeitpläne, Meilensteine und Prozeduren. Das bedeutet während des Prozesses, dass kontinuierlich angepasst und neu geplant werden muss, falls sich die vorher vorausgesetzte Situation verändert. [8]

Agil

Agile Methoden gehen mit der Prämisse zur Planung voran, dass alles ungewiss ist. Das bedeutet wiederum, dass eine kontinuierliche, minimal kritische Planung zur Entwicklung durchgeführt werden muss. Daraus ergibt sich, dass eine schnellere und flexiblere Entwicklung realisierbar ist. [41]

Managementstil und -kultur

Das Management, welches sich je nach Unternehmen, Projekt oder Team unterschiedlich ansiedeln lässt, unterliegt bei der Betrachtung der Softwareentwicklung verschiedenen

Stilen und lässt sich einer eigenen Kultur in der Durchführung ihrer Aufgaben zuordnen. Bei dieser Betrachtung sind vor allem Managementstrukturen im Fokus, die einen direkten Einfluss auf Softwareentwicklungsteams besitzen. Dabei spielt insbesondere die Isolierung der Teams vom Rest des Unternehmens eine Rolle, falls diese sich in kultureller Hinsicht nicht miteinander vereinen lassen. [8, 41]

Traditionell

Der Managementstil wird in traditionellen Modellen dadurch beeinflusst, dass man versucht, hoch optimierte und wiederverwendbare Prozesse zu konstruieren. Hierbei ist eine Planung und Kontrolle notwendig, die durch einen Kommando- und Kontroll-Stil erreicht werden soll. Aufgaben werden verteilt und zugewiesen. Es ist vorgesehen, dass Projekte kontrolliert werden. Die Arbeit der Entwickler wird über strikte Arbeitspakete abgedeckt, sodass der Entwickler genau weiß, was er zu tun hat. Dies ermöglicht dem Management, dass die Arbeiten genau kontrolliert und ggf. Arbeiten an andere Entwickler delegiert werden. Nach dieser Art der Projektführung dienen vor allem hierarchische Strukturen zum Delegieren der Arbeitspakete. Informationen werden nur bei Bedarf weitergeleitet. Es gelten feste Richtlinien, wer wie was zu tun hat. [8, 41]

Agil

Die Managementkultur kann als responsiv dargestellt werden. Hierunter versteht sich, dass das Management auf gegebene Situationen reagiert und versucht die Umgebung für das Team zu verbessern, sodass die Arbeitsbedingungen optimiert werden. Es ist dabei nicht vorgesehen, dass eine Kontrolle oder Zuweisungen von Arbeiten durchgeführt werden. Das Management regt dabei zur Zusammenarbeit an und ist als *Leader* zu sehen. Leader verstehen sich dabei nicht als Führungsposition im Sinne des Vorgeben von Tätigkeiten, sondern als Vertreter zur Schaffung besser Arbeitsbedingungen für das Team. [8, 41]

Rollenzuordnung

Unter Rollenzuordnung versteht sich die Besetzung von Rollen im Team und die Besetzung im Kontext des Softwareentwicklungsteam. Das bedeutet, dass auch Rollen relevant sind die sich durch ihre Tätigkeit als direkten Bezugspunkt zum Team verstehen. Die Belegung der Rollen kann durch Unternehmensrichtlinien bestimmt worden sein, kann aber auch durch die Ausführung einer Rolle als andere Rolle interpretiert werden. Ein Beispiel hierzu wäre, dass ein Scrum Master in seiner Tätigkeit als Projektleiter

wahrgenommen wird. Auch kann ein Projektleiter durch sein Verständnis, ein agiles Projektmanagement beherrschen, aber durch das Unternehmen daran gehindert werden. Ebenso finden sich je nach Sichtweise ähnliche Rollen wieder, die eher der traditionellen oder agilen Softwareentwicklung zugeordnet werden können. [41]

Traditionell

Die generelle Rollenzuordnung wird in traditionellen Modellen Individuen zugeordnet. Darunter ist zu verstehen, dass Personen Rollen zugesprochen werden, welche sich mit den entsprechenden Aufgaben hinter dieser Rolle identifizieren lassen. Dabei werden klassische Rollenbezeichnungen verwendet, wie Systemanalyst, Programmierer und Architekt. [41]

Agil

In agilen Vorgehensweisen gilt als einer der obersten Gesetze, dass sich das Team selbst organisiert. Dies bedeutet auch, dass besonders außerhalb der Scrum Rollen (Scrum Master, Product Owner, Entwickler) ein Entwickler keine weitere Rollen besitzt. Durch die Selbstorganisation können sich weitere Rollen innerhalb des Teams bilden. Insbesondere wird hierdurch eine Begünstigung erreicht, die die Austauschbarkeit von bestimmten Rollen und eine Redundanz durch die Verteilung von Wissen auf mehrere Personen bedeutet. [41]

Anstrebende Organisationsform oder -struktur

Die angestrebte Organisationsform oder -struktur bezeichnet die Orientierung eines Unternehmens, in die sich dieses von seiner Struktur bewegt. Besonders bei großen Unternehmensstrukturen finden sich diesbezüglich Standardisierungen wieder, wie auch Richtlinien, nach denen gewissen Prozesse durchgeführt werden müssen. Zu betonen ist hier das Wort „müssen“ wonach Vorschriften befolgt werden müssen. Dieses Regelwerk führen Unternehmen ein, damit unter anderem eine Messbarkeit und Kontrolle ermöglicht wird. Im Zuge dessen dienen hierarchische Strukturen zum Filtern von Informationen und einer besseren Delegation sowie Kontrollmöglichkeit. Das komplette Gegenteil bieten Unternehmen mit flachen Strukturen. Dieser Punkt beeinflusst insbesondere auch die Kommunikationsmöglichkeit sowie die Flexibilität von Unternehmen. [41]

Traditionell

Bei der Betrachtung von anstrebenden Organisationsformen oder Strukturen wird sich bei traditionellen Modellen auf mechanistisch Formalismen fokussiert. Darunter ist vor allem eine stark bürokratische Struktur mit einem hohen Formalismus zu verstehen. [41]

Agil

Die anstrebende Organisationsform kann als autonom und kooperativ beschrieben werden. Insbesondere bedeutet dies eine starke Reduzierung und ein Verzicht auf Hierarchien. Die Vorteile bestehen vor allem in der steigenden Flexibilität und Verantwortung einzelner. [41]

Steuerung

Als Steuerung versteht sich die Delegation von Aufgaben und deren Abarbeitung. Das bedeutet, dass Personen z. B. Aufgaben zugeteilt werden, die vorher definiert wurden. Die Steuerung muss allerdings nicht einzelnen Vorgesetzten obliegen. Sie kann auch einer Personengruppe übertragen werden. Auch ist es möglich, dass die Steuerung komplett an die jeweiligen Teams übergeben wird, die sich dann selbst steuern können. [41]

Traditionell

Die traditionellen Vorgehensmodelle sind als Prozesszentriert zu betrachten. Im konkreten versteht sich dabei die Definition und Abarbeitung von Aufgaben nach einem definierten Ablauf. Der Grund liegt in der Annahme, dass man die Ursache aller Abweichungen als identifizierbar bestimmt. Erreicht werden soll dies durch ein kontinuierliches Messen und Präzisieren. [41]

Agil

Die wirkliche Steuerung der Arbeit wird innerhalb agiler Methoden auf das Team übertragen. So werden Entscheidungen nicht mehr von Managern dem Team zugetragen, welches dann diese umsetzen muss. Der Gedanke liegt nun vielmehr darin, dass die Entscheidung von allen zusammen getragen wird und das Team entscheidet. Das gesamte Team gibt somit den Weg vor, welches diesen selbstständig und gemeinsam erarbeitet. So stellen nicht einzelne Personen oder feste Prozesse fest, wie gearbeitet wird. [41]

Dokumentation

Im Vergleich der traditionellen und agilen Vorgehensmodelle könnte der Punkt der Dokumentation nicht unterschiedlicher sein. Unter der Dokumentation versteht man dabei verschiedene Aspekte, die den Entwicklungsprozess beeinflussen. Darunter fällt z. B. die Dokumentationspflicht, bei der Aufgaben detailliert beschrieben werden, sodass jeder versteht, worum es sich bei der Aufgabe handelt. Zudem müssen Fortschritte weitergehend dokumentiert werden, sodass jede Aktion nachvollzogen werden kann. Die Dokumentation kann somit auch ein Medium darstellen, um Entscheidungen und Vorgehen zu verstehen bzw. um dieses Wissen zu transportieren. Die Detailstufe sowie die Anzahl an Dokumentationen in verschiedenen Prozessschritten ist je nach Durchführung gewählt. [8]

Traditionell

Die Dokumentation ist in traditionellen Vorgehen unabdingbar. Ohne eine ausreichende Dokumentation greifen viele Prozesse nicht mehr wie gewünscht. Auch die Planung setzt einen hohen Grad an Dokumentationen voraus. Die Kommunikation wie auch das Wissensmanagement bauen auf dieser auf. [8]

Agil

Das Ziel ist hier, dass ein aktuell entwickeltes Inkrement bzw. das Produkt über der zu schreibenden Dokumentation steht. Das bedeutet dabei nicht, dass komplett auf Dokumentation verzichtet wird. Vielmehr steht eine Reduzierung der Dokumentationspflichten im Vordergrund. Dies lässt sich vor allem dadurch erreichen, dass andere Mechanismen greifen, wofür sonst Dokumentationen zwingend Voraussetzung waren. Somit werden Dokumentationen in sinnvollen Fällen erstellt, die hierdurch auch eher aktuell bleiben und die Wirklichkeit widerspiegeln. Dies bedeutet z. B., dass Entwickler selbständig APIs dokumentieren, um eine bessere Kommunikationsmöglichkeit mit abhängigen Teams zu realisieren. [8]

Wissensmanagement

Das Wissensmanagement stellt einen der zentralen Punkte einer Firma dar. Das *Knowhow*, welches jeder Mitarbeiter im Laufe der Zeit in einem Unternehmen aufbaut, stellt Wissen dar, in das das Unternehmen sowie der Mitarbeiter investiert hat. Je nach Umgang mit dem Wissen, ist es grundsätzlich möglich, dass das Wissen weitergetragen wird bzw. sich das Wissen auf mehrere Personen verteilen und duplizieren lässt. Bleibt das Wissen auf wenige Personen verteilt, so können sogenannte Wissensmonolithen entstehen, die ein

hohes Risiko für das Unternehmen darstellen. In Bezug auf die Softwareentwicklung gibt es verschiedenste Praktiken um dieses Wissen transparent und übertragbar zu gestalten. [41]

Traditionell

Das Wissen des zu entwickelnden Produkts verteilt sich auf einzelne Köpfe und ist nicht zusammenhängend und übergreifend verfügbar. Dabei können Einzelpersonen zu Wissensmonopolen heranreifen. Das Wissen wird zudem über detaillierte und umfangreiche Dokumentationen erfasst, die fest reguliert sind. [41]

Agil

Das Wissensmanagement wird über das sogenannte *Lean thinking* betrieben. Zu verstehen ist dies so, dass taktisch versucht wird, das Wissen über die Teammitglieder zu verteilen. Es ist vorgesehen, dass Lerneffekte sowie Erfahrungen aktiv geteilt werden. Hierdurch wird auch Wissen übertragen, was in herkömmlichen Verfahren erst gar nicht bei den Teams angekommen wäre, da dies z. B. Managern vorbehalten ist. [41]

Kommunikation

Betrachtet man die Kommunikation, so gibt es verschiedene Aspekte, die sich referenzieren lassen. So existiert grundsätzlich der Unterschied zwischen formeller und informeller Kommunikation. Im Kontext der Softwareentwicklung gibt es eine Vielzahl von Situationen, in denen die Kommunikation entweder als formell oder informell gekennzeichnet werden kann. So bieten informelle Kommunikationsformen die Möglichkeiten, formelle Räume zur Kommunikation zu reduzieren und stellen darüber hinaus weitere Vorteile in der Bindung zwischen Kommunikationspartnern. So wird eine andere Art der Beziehung gefördert, als wenn man sich nur über strikte Kommunikationspunkte und nur über die bestimmten Gesprächsziele unterhält. Das spontane und unbefangene Gespräch, welches trotzdem produktive Inhalte besitzt, kann somit auch einen Faktor zum Wissensmanagement darstellen. [41]

Traditionell

Bei der Betrachtung von traditionellen Vorgehensmodellen wird die Kommunikation formal bezeichnet. Dies bedeutet insbesondere, dass zur Kommunikation große Mengen von Dokumentationen benötigt werden, die wiederum zwischen Projektteilnehmern als Schnittstelle dienen. Vernachlässigt wird dabei eher der

direkte Kontakt zwischen Personen. Der direkte Kontakt erfolgt wenn nur über geplante Meetings und bei bestimmter Notwendigkeit. [41]

Agil

Die Kommunikation kann zum größten Teil innerhalb der agilen Softwareentwicklung als informell betrachtet werden. Zwar gibt es auch formale Aspekte in Meetings wie dem *Daily Scrum*, wiederum begünstigen die Methodiken und Rituale einen verstärkten informellen Austausch außerhalb regulatorischer Vorgaben. [26, 41]

Teamnähe

Unter der Teamnähe versteht sich grundsätzlich die Nähe zwischen einzelnen Teammitgliedern. Das bedeutet, dass ein Team eine besonders starke Teamnähe besitzt, wenn sich das gesamte Team an einem Tisch befindet. Im Fokus dieser Betrachtung liegt somit die rein räumliche Nähe. Die Abstufungen bedeuten dann, dass sich die Teamnähe „verringert“, sofern sich die Teammitglieder voneinander entfernen. In der nächsten Stufe würde sich z. B. das Team auf unterschiedliche Tische verteilen, auf verschiedene Räume oder auf verschiedene Gebäude. Im extremeren Fall kann eine Verteilung auch bedeuten, dass sich das Team auf unterschiedliche Orte, Länder und weiter auf Zeitzonen verteilt. Je nach Grad der Verteilung verändert sich das Setup für ein einzelnes Team so, dass sich besonders in Bezug auf die Kommunikation verschiedenste Herausforderungen ergeben. Der „Team“ Begriff, kann sich auch im Bezug der Softwareentwicklung auf mehrere Teams beziehen. Dabei besteht ein Team aus beliebig vielen Subteams, die gemeinsam ein Ziel verfolgen. Daraus ergibt sich im Fokus der Teamnähe eine mannigfaltige Betrachtungsmöglichkeit. [8]

Traditionell

Besonders im Kontext verteilter Teams stellen die traditionellen Methoden durch ihren strukturellen Aufbau und der Kommunikation über Dokumente eine Möglichkeit dar. Die Teamnähe ist in diesen Methoden kein definierter Diskussionspunkt. [8]

Agil

Bezüglich der Ortlichkeit werden bei agilen Vorgehensmethoden dicht beieinander arbeitende Teammitglieder angestrebt. Ebenfalls ist die Nähe vom Kunden zum

Team mit einer geringen Entfernung bevorzugt, um diesen gut einbeziehen zu können. [8]

Teamgröße

Die Teamgröße stellt in der Softwareentwicklung ebenfalls einen wichtigen Identifikator dar. Je größer das Team wird, um so komplexer wird die Interaktion im Team zwischen den Teammitgliedern. Die Dynamik wie auch der Bedarf der angestrebten Interaktionen stellen somit Faktoren zur Konstruktion der Teams in Bezug auf die Teamgröße dar. [8]

Traditionell

Innerhalb traditioneller Vorgehensmethoden sind häufig auch mehr als 10 Personen in einen Team anzutreffen. Durch die feste Struktur der Pläne und Dokumentationen wird hier eine bessere Kommunikation für größere Teams zur Verfügung gestellt. Hierbei ist die Bürokratie der Organisationen zusätzlich hilfreich. [8]

Agil

Für agile Teams ist es empfehlenswert, dass Teams klein gehalten werden. Die Anzahl der Teammitglieder sollte in der Regel nicht mehr als 10 Personen überschreiten. Die kleinen Teamgrößen sind notwendig, damit auch die agilen Praktiken effizient durchführbar sind. [8]

Teambeteiligung

Im Fokus der Teambeteiligung versteht man die angestrebte Dynamik, Verantwortung und Entscheidungsmöglichkeit des Teams bzw. von Teammitgliedern. Das beutet insbesondere inwieweit das Team im Prozess zur Anforderungsdefinition, Abarbeitung, Kundenfeedback und Auslieferung einbezogen wird. Auch die Transparenz und das Involvieren in Managementprozesse stellen einen Punkt zur Teambeteiligung dar. [11]

Traditionell

Bei traditionellen Methoden ist es nicht vorgesehen, dass eine Teambeteiligung stattfindet. Einzelne Personen treffen Entscheidungen und beteiligen nur weitere einzelne Personen bzw. einen fest definierten Kreis an Personen. [11]

Agil

Agile Entwicklungsvorgehen sind im Gegensatz zu herkömmlichen Methoden auf eine starke Teambeteiligung angewiesen. Informationen müssen verbreitet, ausge-

tauscht werden und das Team muss zusammen Entscheidungen und Konsequenzen tragen. [11]

Kundenbeteiligung

Im Unterschied zur Teambeteiligung steht die Kundenbeteiligung im Hauptfokus auf das Involvieren des Kunden in den Entwicklungsprozess. Hierzu dient insbesondere die Transparenz der Softwareentwicklung. Diese Transparenz bedeutet z. B. dem Kunden einen kontinuierlichen Einblick in die Softwareentwicklung zu gewähren und Feedback-Iterationen zur Verfügung zu stellen, um das Erfüllen der Kundenanforderungen zu prüfen. Die Kundenbeteiligung bedeutet auch, dass gegenüber verschiedener Stakeholder geprüft wird, ob diese in diesem Sinne als Kunde darzustellen sind. Die unterschiedlichen Stakeholder besitzen im Kontext der Entwicklung verschiedenartige Einflussmöglichkeiten und somit auch Bedeutungen für das Softwareprojekt. Die Tiefe beim Einbeziehen des Kunden unterscheidet sich jedoch je nach Entwicklungsmethode und nach der verfolgten Strategie im Unternehmen. Grundsätzlich stellen die Vorgehensmodelle aber deutliche Unterschiede in der Formulierung der Prozessbeschreibung dar. [41]

Traditionell

Das Einbeziehen des Kunden genießt auch in traditionellen Methoden eine wichtige Rolle. Hauptsächlich findet ein intensiverer Kontakt während der Entwicklung der Spezifikation bzw. der Analyse-Phase statt. Dabei beschränkt sich allerdings der Kontakt meist auf ein Minimum und ist außerhalb dieser Phase selten bis gar nicht vorgesehen. Somit findet hierdurch wenig Kontakt zur Entwicklungsabteilung während den Implementierungsarbeiten statt. [41]

Agil

Die Kundebeteiligung ist als fester Bestandteil der Entwicklung eingeplant. Der regelmäßige Austausch sowie das benötigte Feedback wird explizit bei der Verfolgung der agilen Modelle benötigt, um z. B. Zwischenstände zu präsentieren, gezeulernen und Anforderungen neu zu bewerten. Die Beteiligung von Stakeholdern bzw. insbesondere des Kunden sind als kritisch und notwendig zu betrachten. [41]

Feedback-Mechanismus

Der Feedback-Mechanismus steht in einer direkten Relation zur Kundenbeteiligung sowie der Teambeteiligung. Eine Feedback-Kultur ermöglicht es, dass ein besseres Lernen

ermöglicht wird. Das bedeutet insbesondere auch, dass Fehler nicht wiederholt bzw. Fehler getätigt werden. So können Hindernisse frühzeitiger aufgedeckt werden. Auch wird durch das Feedback ein Kanal erschaffen, der bei der richtigen Platzierung im Prozess ein rechtzeitiges Gegenlenken bei einem Fehlkurs ermöglicht. Vorausgesetzt ist, dass Feedback aufgenommen werden kann und das Team, Stakeholder und das Unternehmen, offen für konstruktive Kritik und Kursänderungen sind. [8]

Traditionell

Es ist in der Regel kein fester Feedbackmechanismus eingeplant. Die Evaluierung wird im Rahmen von Abnahmen und nicht als kontinuierlicher Prozess verstanden. Es können dabei allerdings sporadische Mechanismen existieren, die z. B. in Rahmen von Meilensteinen oder Statusreports durchgeführt werden. Sie sind aber nicht wie in agilen Verfahren zu betrachten. [8]

Agil

Innerhalb der agiler Verfahren existieren gewöhnlicherweise mehrere Mechanismen, die Feedbackmöglichkeiten beinhalten. Zurückzuführen sind diese auf verankerte Rituale innerhalb des Prozesses, die den Austausch von Hindernissen und Evaluierungen von Arbeiten festsetzen. Hierzu können unter anderem agile Praktiken wie *Code Reviews* beitragen. Auch dienen Rituale wie die Retrospektive auf der Prozessebene dafür, dass Verbesserungspotentiale über das Feedback des Teams aufgenommen werden. [8]

kontinuierliches Lernen

Die Betrachtung des kontinuierlichen Lernens stellt in der Softwareentwicklung und der gesamten IT-Branche einen nicht umgänglichen Faktor dar, der verfolgt werden muss. Nur durch das Lernen wird ermöglicht, dass aktuelle und zukünftige Änderungen in der Branche verfolgt oder sogar Trends erzeugt werden können. Das kontinuierliche Lernen stellt im Entwicklungsprozess selbst einen wichtigen Punkt dar. Das bedeutet, dass während der Projektdurchführung konstruktive Kritik bzw. Feedback als Lernbestandteil aufgenommen werden. Es wird somit innerhalb eines gewissen Zeitraums eine Möglichkeit präsentiert, über die Optimierungspotentiale aufgedeckt werden. Im Konkreten bedeutet das, dass z. B. nach einer aktiven Entwicklungsphase von zwei Wochen rückwirkend rekapituliert wird, was gut und was nicht so gut funktioniert hat. Daraus lassen sich Maßnahmen ableiten, woraus sich wiederum eine Beschleunigung

des Prozesses erreichen lässt. Zusätzlich zur Betrachtung des Entwicklungsprozesses steht der Lernprozess und die Weiterentwicklung einzelner Mitarbeiter im Fokus. [8]

Traditionell

Der Punkt des Lernens ist nicht als fester Bestandteil eingeplant. Dies heißt aber nicht, dass dies nicht möglich ist. Innerhalb der traditionellen Entwicklungsmodelle stellt dieses kontinuierliche Lernen nur keinen Fokus in der Projektdurchführung dar. [8]

Agil

Das kontinuierliche Lernen ist ein fester Bestandteil von agilen Methoden und besitzt in der Regel ein eigenes Ritual oder eine Phase, in der Maßnahmen zur Verbesserung im Entwicklungszyklus erarbeitet werden. [8]

5.2 Vorgehen zur agilen Transition

In diesem Abschnitt wird näher beschrieben, über welche Ansätze agile Methoden in Unternehmen oder Teams eine Implementierung stattfinden kann. Dabei existieren grundsätzlich zwei verschiedene Ansätze, auf die sich die Implementierung reduzieren lässt. Zum einen kann ein revolutionärer oder zum anderen ein evolutionärer Ansatz verfolgt werden. Welcher Ansatz in welcher Detailstufe verfolgt werden kann, ist dabei, als nicht trivial einzustufen und benötigt zur Auswahl eine geeignete Voranalyse zur Bestimmung. Ein Ansatz stellt die sogenannte *Agile Potentialanalyse* dar, die das Ende dieses Unterabschnittes beinhaltet.

Wie bereits erörtert wurde, stellt die Implementierung eines agilen Prozesses eine enorme Herausforderung dar. Eine Studie hat bei dieser Fragestellung herausgefunden, dass dabei 63 % aller befragten Unternehmen auf individuell selektierte Aktivitäten setzen. Die simultane Einführung agiler Prozesse in allen Feldern wird mit 18 % beziffert. Nur 11 % setzten dabei zunächst auf fokussierte Aktivitäten durch das Testen. [33]

Bei der Einführung der agilen Vorgehensmodelle stellen besonders unterstützende Maßnahmen einen elementaren Bestandteil einer Implementierungsstrategie dar. Innerhalb der Studie bewerteten 68 % der Befragten, dass dabei die Initiative von individuellen Angestellten einen Auslöser darstellen. Das Training des Personals stellt mit 60 % auch einen hohen Anteil der Antworten der Befragten dar. In der Beantwortung stellten 50 % fest, dass externe Berater bei der Implementierung gut unterstützten und immerhin 19 % bezeichneten spezialisierte Abteilungen als Kompetenzzentrum. [33]

Je nach strategischer Ausrichtung der Softwareentwicklung gibt es jedoch generelle Unterschiede, die bei einer Integration von agilen Vorgehensweisen eine Rolle spielen. Im Fokus dieser Arbeit werden dabei drei Konzepte verfolgt. Diese sind das Projektgeschäft, das Produktgeschäft sowie die interne Entwicklung von Software.

Das Projektgeschäft stellt eine klassische Variante dar. Hierbei wird in einem mehr oder weniger festen Zeitraum ein Projekt abgewickelt. Das bedeutet im Konkreten, dass ein abgestecktes Projekt für einen Kunden verfolgt wird.

Beim Produktgeschäft wird ein Softwareprodukt kontinuierlich weiterentwickelt. Dabei kann es intern zwar Projekte zur Umsetzung verschiedener Teilkomponenten geben, gegenüber den Kunden werden aber ein oder mehrere Produkte angeboten.

Die interne Entwicklung ist die dritte Unterscheidung die innerhalb dieser Arbeit betrachtet wird. Teil dieser Ausrichtung ist die Softwareentwicklung innerhalb des Unternehmens. Das Unternehmen verfolgt dabei nicht das Hauptgeschäftsziel, dass ein Softwareprodukt gegenüber den Kunden dargestellt wird. Vielmehr stehen z. B. Finanzaktivitäten, Versicherungsdienstleistungen oder der Handel mit Waren im Mittelpunkt. Die Entwicklung stellt somit Komponenten, um ein Geschäftsmodell zu verfolgen. Dabei kann die Betrachtung im Inneren des Unternehmens durchaus einen Produkt- oder Projektcharakter darstellen. Gegenüber dem Kunden sind diese allerdings nicht unbedingt erkennbar, wenn z. B. ein internes Verrechnungssystem für die Unternehmenszwecke kontinuierlich entwickelt wird.

Die strategische Ausrichtung des Unternehmens spielt bei der Integration agiler Prozesse und Vorgehensmodellen eine Rolle, wie auch die anderen bereits vorgestellten Faktoren. Zur Implementierung dieser Modelle existieren dabei auch im Kontext der bestehenden Umgebung verschiedenste Verfahren. Unterscheiden kann man dabei zwei grundsätzliche Ansätze auf die nachfolgend weiter eingegangen wird.

5.2.1 Revolution

Eine Möglichkeit zur Implementierung agiler Vorgehensmodelle stellt der revolutionäre Ansatz dar. Das Ziel dieser Einführungsstrategie ist es, dass alle Aspekte des ausgewählten Entwicklungsmodells in einer Phase verankert werden. Häufig wird dieser Ansatz auch als sogenannte „Big Bang“ bezeichnet. [49]

Wählt man in diesem Fall z. B. das Entwicklungsmodell Scrum, so gilt es zum einen die Leitlinien des Modells selbst, in Form von Ritualen, Rollen und Artefakten zu implementieren. Zum anderen beinhaltet dies auch, dass in einer Phase der Großteil des kompletten Verständnisses

von Scrum in die Unternehmensstruktur eingegliedert wird. Andernfalls besteht die Gefahr, dass eine weitere Adaption des Prozesses verhindert wird. [49]

Besonders diese Strategie beinhaltet ein hohes Risiko des Scheiterns. Es müssen viele Annahmen über das Unternehmen zu Beginn getroffen werden, damit eine Einführung eine reale Chance besitzt. Dabei entstehen einige Herausforderungen der Implementierung erst im Prozess der Einführung selbst. Hierzu gehört unter anderem die Reaktion der Stakeholder und ob diese realistische Erwartungen an die neue Art der Entwicklung gestellt haben. [49]

Insbesondere das kritische Einbeziehen des Unternehmenskontextes bietet eine Angriffsfläche dieses Ansatzes. Mitarbeiter sind ggf. alte Methoden gewohnt und verfallen bei einem unvollständigen Verständnis in alte Muster zurück. Im Gegensatz hierzu steht die evolutionäre Einführungsstrategie, die im nachfolgenden Abschnitt näher beschrieben wird.

5.2.2 Evolution

Das Vorgehen der evolutionären Implementierung einer agilen Methode fokussiert sich sehr stark auf die bereits ausübende Strategie der Softwareentwicklung. So ist das Ziel innerhalb dieses Ansatzes, dass eine schrittweise Integration der Bestandteile einer agilen Methode durchgeführt werden. [49]

So wird z. B. bei einer Einführung von Kanban darauf geachtet, dass nicht direkt alle Teile umgesetzt werden und dass das dort vorhandene Taskboard schrittweise aufgebaut wird. Im Fokus stehen somit eher der Wechsel auf die Entwicklung von Funktionalitäten und lauffähige Inkremente mit minimalem Funktionsumfang. Weitere Bestandteile, wie die Richtlinie der WIPs, können dabei später erfolgen. Durch dieses Verfahren wird es dem Unternehmen ermöglicht, dass Schritt für Schritt alte Teile der Entwicklung auslaufen können und neue implementiert werden. [49]

Ein besonderer Vorteil wird dem Unternehmen dadurch gegeben, dass dem Unternehmenskontext eine angemessenere Reaktionsmöglichkeit angeboten wird. Regulatorische Prozesse, die sich auch von außerhalb des Unternehmens auf dieses beziehen können, können hierdurch schrittweise und erfolgreicher umgestellt werden. [49]

Eine Herausforderung dieses Verfahrens stellt dabei der zeitliche Rahmen und die Vollständigkeit der Implementierung dar. Insbesondere die Vollständigkeit kann im konkreten Fall bedeuten, dass auf einen langen Zeitraum gemessen nur ein Teil implementiert wird und wichtige Bestandteile wie das tiefere Verständnis der agilen Werte außer Acht gelassen werden. Zu erwähnen ist, dass durch die evolutionäre Transition eine Vermischung von traditionellen

und agilen Aspekten erfolgt. Es ist somit auf wechselseitige Abhängigkeiten zu achten. Dies kann dann auch bedeuten, dass sich fehlerhafte Erwartungen ergeben, die sich auf weitere Teams vererben lassen. Die strategische Ausrichtung auf die Vollständigkeit und Korrektheit über einen angemessenen Zeitraum muss somit gewährleistet werden.

Dabei spielt auch hier die Analyse und die Bestimmung des Unternehmenskontextes und die sich daraus ergebene mögliche Tiefe des initialen agilen Grads eine entscheidende Rolle. Auf die Bestimmung des auszuwählenden Grades wird im nächsten Abschnitt weiter eingegangen.

5.2.3 Agile Potenzialanalyse

Die agile Potenzialanalyse stellt ein Verfahren dar, welches von einer Forschungsgruppe des Fraunhofer Institutes publiziert wurde. Das Ziel dieses Verfahrens stellt die Definition des richtigen Grads der Agilität für ein Team oder ein Unternehmen dar, wobei der Kontext nicht außer Acht zu lassen ist. [49]

Realisiert wird dies vor allem durch die Auswahl geeigneter agiler Praktiken, die die Grundvoraussetzung zur evolutionär getriebenen Umstellung bietet. Der Grad der Agilität lässt sich hiernach über die Anzahl und Korrektheit der verfolgten agilen Praktiken beschreiben. Die Einordnung agiler Praktiken zu weiteren elementaren Bestandteilen kann der Abbildung 5.1 entnommen werden. [49]

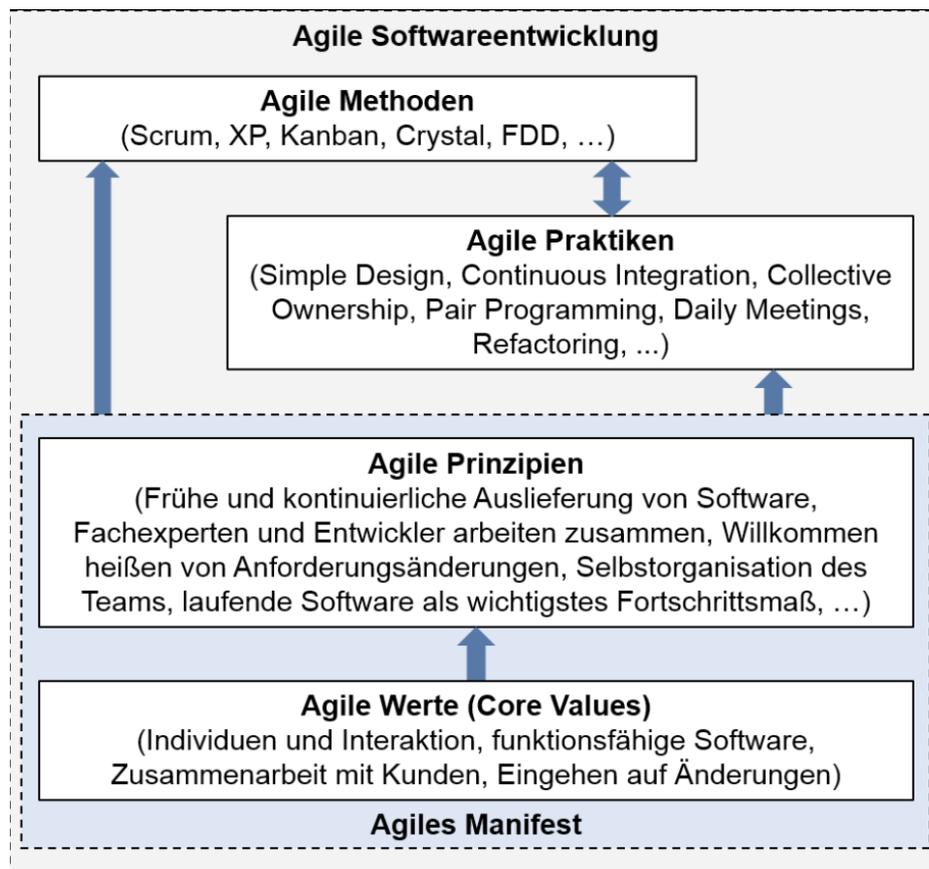


Abbildung 5.1: Agile Softwareentwicklung - Beziehung zwischen den verschiedenen agilen Aspekten [49]

Wie die Grafik 5.1 darstellt und auch im Abschnitt 4.1 beschrieben wurde, beinhaltet das agile Manifest die agilen Werte sowie die agilen Prinzipien. Das Agile Manifest lässt sich dabei als Grundstein der agile Softwareentwicklung betrachten. Das agile Manifest referenziert auf agile Methoden wie z. B. Scrum und Kanban, aber auch auf agile Praktiken wie Pair Programming. Die agile Praktiken stehen wiederum in Verbindung mit agilen Methoden. Hierzu wurde im Abschnitt 4.2 dargestellt, dass zwischen den agilen Methoden und den agilen Praktiken eine wechselseitige Beziehung existiert. Agile Methoden können dabei verschiedenartige agile Praktiken verwenden, wobei agile Praktiken in verschiedenen Modellen bereits definiert sind. Beide Bereiche, die agilen Methoden und Praktiken sind in diesem Kontext Bestandteil der agilen Softwareentwicklung. Die Einordnung der verschiedenen Aspekte der agile Entwicklung ermöglicht es, ein besseres Verständnis der Abhängigkeiten und Bestandteile dieser Verfahren wahrzunehmen, welche im Bezug auf die agile Potenzialanalyse notwendig ist. [49]

Wie eingangs beschrieben, stellt das oberste Ziel dieser Analyse die Erfassung des richtigen Grads an Agilität für das Unternehmen zu einen gegebenen Kontext dar. Benötigt werden hierzu neben dem Kontext, der vor allem Abhängigkeiten definiert, auch die konkreten, priorisierten Ziele zur Verbesserung, die die Organisation oder das Projekt anstrebt. [49]

Die agile Potenzialanalyse kann im Kern als Einflussmodell bezeichnet werden, welches in Abbildung 5.2 dargestellt ist.

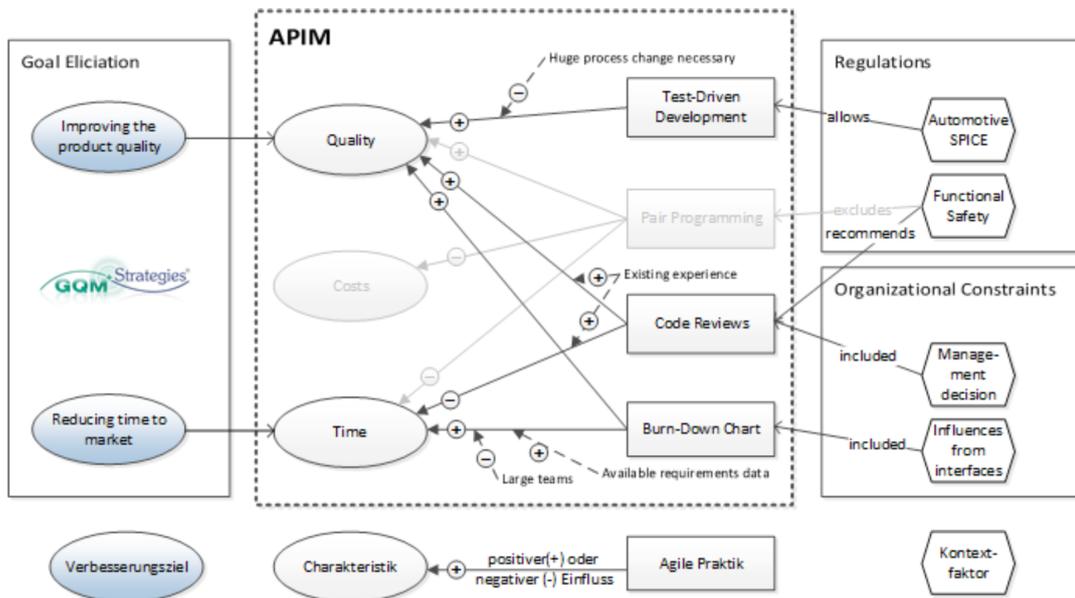


Abbildung 5.2: Agile Potenzialanalyse [49]

Diese Methode besteht aus drei generellen Schritten, die ebenfalls in der Grafik 5.2 zu erkennen sind. Zunächst werden auf Basis der individuellen Verbesserungsziele (links dargestellt) diese als Eingangsgröße und Anforderung aufgenommen. Hieraus ergeben sich jeweils Charakteristiken. [49]

Im nächsten Schritt werden eine Reihe von agilen Praktiken ausgewählt, die verfolgt werden könnten. Diese Praktiken werden dabei in Relation zu den definierten Charakteristiken gestellt. Ebenfalls erfolgt die Aufnahme von Beschränkungen, die sich z. B. aus dem Unternehmenskontext ergeben sowie Regularien, die sich durch den Industriezweig des Unternehmens ergeben können, die allerdings von außerhalb des Unternehmens die Softwareentwicklung beeinflussen. Diese Beschränkungen können dabei bestimmte Praktiken bereits verbieten, vorschreiben oder evtl. empfehlen. [49]

Durch die vorher durchgeführten Schritte ergibt sich somit ein Bild, wie sich agile Praktiken im Zusammenspiel mit dem Unternehmenskontext und den angestrebten Zielen verhalten. Im letzten Schritt erfolgt nun eine Reduzierung von sinnvoll genutzten Praktiken, wodurch das Bild weiter geschärft wird. Hieraus lässt sich nun ableiten, inwieweit gewisse agile Methoden sogar ganz wegfallen können. Vor allem bietet dieses Verfahren aber eine Möglichkeit, wie eine kontinuierliche agile Adaption des Unternehmens in realistischer Weise durchführbar ist. [49]

Das Level der Detailstufe kann innerhalb dieses Modells frei gewählt werden und somit auch tiefer liegende Problematiken darstellen. Ebenfalls kann hierdurch eine schrittweise Transition verbildlicht werden, um z. B. eine Migration zu einem anderen agilen Modell näher zu beschreiben. Der nächste Abschnitt dieses Hauptkapitels geht auf den Punkt der Migration im konkreten Fall zum agilen Vorgehensmodell Scrum weiter ein und stellt hierzu weitere verwendete Strategien zur Transitionen aus der Praxis dar.

5.3 Durchführung einer Scrum Migration

Die Transition einer klassischen Entwicklungsmethode hin zu einer agilen Softwareentwicklung hat eine Vielzahl von Schwierigkeiten, die bereits in den vorherigen Abschnitten beschrieben wurden. Je nach Unternehmenskontext ergeben sich ganz individuelle Themen, die bei der Migration der Methoden, beachtet werden müssen.

In den nachfolgenden Unterabschnitten werden in Bezug auf die Umwandlung eines traditionellen Vorgehensmodell auf Scrum, zwei Studien vorgestellt, die diesen Prozess begleitet haben. Hierzu ist nochmal zu betonen, dass diese kein generelles Muster darstellen, nach der eine solche Umwandlung verfolgbar ist. Hierbei wurden Entscheidungen getroffen, die sich auf den jeweiligen Unternehmenskontext beziehen. Zunächst wird zu dieser Thematik eine Migration näher dargestellt, wo das V-Modell auf Scrum umgestellt wurde. Im Anschluss darauf folgt eine Studie, die den Rational Unified Process als Ausgangssituation verwendet. Zu beiden Studien wird zunächst der Kontext kurz beschrieben. Danach folgt eine Beschreibung der Herausforderungen und der Veränderungen des Prozesses von den jeweiligen Entwicklungsverfahren hin zu Scrum.

5.3.1 Vom V-Modell zu Scrum

Das V-Modell stellt ein sehr klassisches Modell dar, welches bereits im Abschnitt 2.2.2 näher beschrieben wurde. Die hier vorgestellte Studie stellt die Transition dieses Modells auf die agile Vorgehensweise Scrum dar. Hierzu wird eine Zusammenfassung zu den Punkten Herausforderungen und den Veränderungen gegeben, nachdem eingangs eine Beschreibung des Kontextes zur Migration gegeben wurde.

Die Firma, die den Schritt vom V-Modell auf Scrum wagt, lässt sich dem Bereich der Softwareentwicklung für *Embedded Systems* (deutsch eingebettete Systeme) zuordnen. Dieses Unternehmen behandelt dabei die Thematik der Qualitätssicherung von Systemen für sogenannte *System on Chips* (kurz. SoC), welche z. B. für mobile Endgeräte wie Mobiltelefone und Tablets verwendet werden. Getragen wurde die Entscheidung der Migration vor allem durch die Organisation selbst. Grund für diesen Schritt waren die Erwartungen von Stakeholdern. Der Markt hat sich für dieses Unternehmen in die Richtung bewegt, dass Stakeholder es immer mehr gewohnt sind, dass sie stärker in die Softwareentwicklung eingebunden werden. Die Entwicklung selbst verteilt sich in diesem Unternehmen auf mehrere Zeitzonen und verfolgt zusätzlich eine verteilte Entwicklung. [28]

Herausforderungen

Die Herausforderungen dieser Migration verteilen sich auf verschiedene Punkte. Allerdings stellt die verteilte Entwicklung wohl das größte Problem dar. Durch die Verteilung der Entwicklungsteams über zwei geografische Zeitzonen entstehen besonders durch die angestrebten Rituale, wie die *Standup Meetings*, Hindernisse für die Transition. So fällt es schwer, dass der Projektstatus an alle Teams gerecht transportiert wird. Hinzukommt, dass unterschiedliche Sprachen im Projekt existieren, wozu z. B. Madarin und Englisch gehören. [28]

Trotzdem muss das Team enger zusammenarbeiten und sich zusammen synchronisieren. Realisiert wird dies über das Stattfinden von gemeinsamen Meetings, die synchronisiert werden. Innerhalb dieser existieren verschiedene Subteams, die eigene Sprints verfolgen. Diese Meetings werden genutzt, um eine Wissensverteilung und *Brainstormings* zu unterschiedlichen Themen zu ermöglichen. Auch sollen dabei wichtige Fragen gestellt werden und die Stakeholder am Informationsverteilungsprozess teilnehmen. [28]

Eine zusätzliche Herausforderung ist die Aufgabenverteilung von komplexeren Projekten. Diese müssen verfolgt und auch hier wieder in den Kontext der verteilten Entwicklung gestellt werden. Ermöglicht werden kann dies nur über geeignete Tools, wie z. B. innerhalb dieser

Studie mit dem Projekt- und Vorgangsnachverfolgungs-Tool *JIRA*¹. Dies erfordert wiederum eine Disziplin, damit Abhängigkeiten zu anderen Kollegen korrekt dargestellt werden können und auch entsprechende Updates für diese zur Verfügung stehen. Auch stellt dieses Tool für das obere Management eine Möglichkeit dar, dass Fortschritte dargestellt und verfolgt werden können. Um diese Disziplin der korrekten Abarbeitung der Aufgaben zu ermöglichen, ist der Scrum Master in dieser Studie eine Art Projektmanager, welcher die korrekte Verwendung des Tools prüft. In Betrachtung der agilen Grundsätze sowie der Bedeutung des Scrum Masters ist dies eher kritisch zu betrachten. Allerdings ist besonders in der Betrachtung der verteilten Entwicklung zu berücksichtigen, dass das Zerteilen der Anwendung und Aufgaben wichtig ist. Hierbei gilt es, möglichst überschneidungsfreie Aufgaben zu erzeugen, sodass die Aufgaben sich bestmöglich verteilen. [28]

Die schnelle und regelmäßige Auslieferung von Software, die über die agile Softwareentwicklung erreicht wird, stellt auch eine Problemstellung dar. Verantwortlich sind dafür Abhängigkeiten zu anderen externen Projekten wie auch zum eher klassischen Projektmanagement, die mit der neuen Geschwindigkeit der Entwicklung synchronisiert werden müssen. Hierzu sind entsprechende Schlüsselfiguren im Management zu überzeugen und die Erwartungen entsprechend zu schulen. [28]

Durch die Neustrukturierung der Teams und das stärkere Erzeugen von funktionsübergreifenden Teams stellen sich bei der Initialisierung und auch über den Projektverlauf Teams bereit, die entsprechend ihrer Strukturierung eher gewisse Fachlichkeiten abbilden. Das führt in dieser Studie dazu, dass Teams oder auch einzelne Personen eher neuen Teams oder Projekten zugeordnet werden. Hierdurch entsteht ein regelmäßiges Problem, dass das Wissen, das Team oder Projekt verlässt. Als Reduzierung dieser Problematik wird hierzu versucht, über Wissensportale das Fachwissen zu archivieren und anderen Kollegen bei Notwendigkeit zur Verfügung zu stellen. Das bedeutet wiederum, dass wieder eine stärkere Dokumentation erzeugt wird. Diese verhält sich dabei allerdings nicht im Sinne von Dokumentationen für formale Bedingungen des Entwicklungsprozesses, sondern eher auf einem technischeren und direkteren Kanal. So werden nur Thematiken dokumentiert, die nicht selbsterklärend sind. Die Schwierigkeit besteht in diesem Fall darin, wie detailliert dokumentiert wird und wie sich die Dokumentation aktuell hält. [28]

¹<https://de.atlassian.com/software/jira> - Abruf am 05.06.2017

Veränderungen

Unabhängig vom reinen Entwicklungsprozess haben sich innerhalb dieser Fallstudie Veränderungen ergeben. Somit hat sich z. B. die Unabhängigkeit der Qualitätssicherungsbereiche verändert. Durch die Umstellung ist eine dichtere Zusammenarbeit entstanden, die sich je Standort sowie standortübergreifend über asynchrone und synchrone Kommunikationsmittel realisieren ließ. Wissen wird hierdurch stärker zwischen den unterschiedlichen Abteilungen geteilt und es existieren Feedbackmechanismen. [28]

Die Veränderung des Produktmanagements ermöglicht es dem Entwicklungsteam, dass Anforderungen direkter von den Stakeholdern vorgetragen werden. Die aufgeführten Akzeptanzkriterien helfen der Qualitätssicherung entsprechend, dass die wirklichen Bedürfnisse der Stakeholder besser verstanden und geprüft werden können. Dies bedeutet, dass eine höhere Qualität der veröffentlichten Versionen erzeugt wird. [28]

Ebenfalls hat sich die Organisation der Teams verbessert. Durch das Splitten der großen Teams in kleinere Teams wird eine höhere Flexibilität geboten. Die Spaltung bezieht sich dabei auch auf die Geschäftsseite, die ihre Teams ebenfalls spaltet. Die Teams auf beiden Seiten werden so kleineren Themen zugeordnet, die im Einzelnen betrachtet, besser verfolgbar sind. Hierzu werden auf der technischen Seite vor allem die funktionsübergreifenden Teams benötigt. Die jeweiligen Teams können somit auch besser verschiedenen Kunden zugeordnet werden. Die Fähigkeit funktionsübergreifend zu agieren sowie das bessere Verständnis von allen Teilnehmern, wird innerhalb der Technik durch agile Praktiken erreicht. Hierzu dienen Code Reviews und das Pairprogramming um einen Wissensaustausch zu ermöglichen. [28]

5.3.2 Vom Rational Unified Process zu Scrum

Der Rational Unified Process (kurz RUP) lässt sich vom verfolgten Verfahren eher den klassischen Modellen der Softwareentwicklung zuordnen und wurde bereits im Abschnitt 2.2.5 näher beschrieben. Dabei wurden die auch mit agilen und anderen traditionellen Modellen verglichen. Innerhalb der in diesem Abschnitt beschriebenen Studie wurde eine Transition vom Rational Unified Process zu Scrum realisiert. Hierzu wird nachfolgend der Kontext dieser Studie beschrieben und wie beim vorherigen Abschnitt auf die Punkte Herausforderungen und Veränderungen der Migration eingegangen.

Die hier zusammengefasste Studie behandelt die Migration hinzu Scrum. In diesem Zusammenhang bewegten sich drei Firmen, die vorher die Softwareentwicklung über RUP verfolgt haben. Diese Firmen verteilen sich auf zwei Länder. Zum einen befinden sich zwei Unternehmen in

den Niederlanden und eins in Indien, wobei die Hauptimplementierung der Anwendungen in Indien erfolgt und nur ein eingeschränktes Testen der Anwendung von dort ermöglicht wird. Verfolgt werden hierbei zwei Softwareprojekte, die für die Finanzbranche umgesetzt werden. Die komplette Migration erfolgt durch die Unterstützung von einer weiteren externen Firma. [42]

Herausforderungen

Als Herausforderung der Transition zum agilen Modell Scrum wird vor allem die global verteilte Softwareentwicklung bezeichnet. Die Zeitverschiebung erschwert neben der generellen Distanz die Problematik der Synchronisierung von Meetings. Hierzu gibt es im Bezug auf die Finanzbranche zusätzliche Hindernisse. Hierzu stellt die Europäische Union Vorschriften, die den Zugriff auf Daten innerhalb der EU von außerhalb schützen sollen. Dies verhindert im Entwicklungsprozess, dass das Testen nur eingeschränkt außerhalb der EU möglich ist. Laut der Studie standen hierdurch keine realistischen Testdaten zur Verfügung. [42]

Wie bereits angedeutet, verhindert diese räumliche Distanz auch die optimale Zusammensetzung der Teams. Fachlichkeiten wie auch Teamkollegen innerhalb eines Teams verteilen sich. Die angestrebte Kommunikation, die als besonderer Vorteil der agilen Softwareentwicklung bereitgestellt wird, wird nicht annähernd erreicht. [42]

Aus diesen tiefen Strukturen erschwert sich auch die Organisation der Aufgaben. Da die Hauptleistung zur Softwareentwicklung in Indien ist, werden anderweitige Rollen über die Niederlande realisiert. Dies bedeutet auch, dass wichtige Rollen wie Product Owner und wichtige Stakeholder nicht vor Ort sind. Hieraus lassen sich bereits zukünftige Herausforderungen, wie auch Optimierungspotentiale ableiten. [42]

Die eher evolutionäre Herangehensweise der Transition hat auch zur Folge, dass die Migration weiterhin viele traditionelle Bestandteile besitzt. So wird zwar über die Erweiterung des Scrum Modells gesprochen, doch wird bei der engeren Betrachtung der Erweiterungen bewusst, dass es sich um Aspekte des RUP Verfahrens dreht. Hierzu zählt, dass alte Rollen parallel bestehen bleiben und weitere Phasen zur Abnahme „eingeführt“ werden. Diese Phase dient dabei der Abnahme und Prüfung von Dokumenten und des Backlogs. Je nach Stärke der Verfolgung dieser Ansätze und Nicht-Beachten der agilen Prinzipien und Werte bedeutet dies, dass der weitere Adaptionprozesse behindert werden. [42]

Auch wird in Bezug auf die Herausforderung zur Überzeugung der Stakeholder klar, dass generelle Verständnisprobleme auch nach der Migration noch existieren. So stellen die Stakeholder

fest, dass das neue Vorgehensweisen zwar Vorteile mit sich bringen, wichtige Punkte aber zu stark vernachlässigt werden. Hierzu zählen vor allem das Risikomanagement und die Sicherheit der erzeugten Anwendungen. Zusätzlich hierzu wird festgestellt, dass die Dokumentationsstruktur ungenügend ist und das empfohlen wird, hier wieder auf traditionelle Verfahren zu setzen, was ebenfalls auf Kommunikations- und Umsetzungsschwierigkeiten schließen lässt. [42]

Als weiterer Punkt wird innerhalb dieser Migration die Synchronisierung der Auslieferungstände genannt. So entsprechen Pläne zur Synchronisierung von Projekten nicht dem aktuellen Entwicklungsstand. Die realisierten Sprint Ergebnisse werden hiernach asynchron hervorgebracht und veraltet in Berichterstattungen über der Entwicklungsebene thematisiert. Zusätzlich hierzu stehen Prozesse zur Erzeugung von veröffentlichten Versionen. Diese erzeugen bürokratische und zugleich schlecht synchronisierte Schnittstellen zur Entwicklung, die ebenfalls Aufwände bedeuten. [42]

Veränderungen

Die Migration erzeugt auch in dieser Studie einige Veränderungen, die die Entwicklung vorantreiben. So stellt das Involvieren von Kunden in den Prozess besonders für die Entwicklung einen positiven Aspekt dar. Hierdurch werden die Entwickler weg von reinen Entwicklungsaufgaben getragen und es wird ihnen ermöglicht, dass Fachdomänenwissen erzeugt wird. Hierdurch wird ein tieferes Verständnis dafür geschaffen, was wirklich erreicht werden soll. Das Ergebnis ist auch die Erzeugung eines besseren Verständnis für die Nutzeraktivitäten. [42]

Das bedeutet ebenfalls, dass sich auch Anforderungen anders als bisher zur Entwicklung bewegen. Es sind weniger Prozesse bei der Transformation der Anforderungen involviert und die Formulierung ist aus Sicht des Geschäfts und der Technik besser dargestellt. [42]

Durch die verfolgten Rituale des Scrum Vorgehens wird es ermöglicht, dass die Kommunikation auch trotz der verteilten Entwicklung, erheblich verbessert wird. Es wird mehr direkte Kommunikation eingesetzt. Hierdurch entstehen wiederum weniger Missverständnisse, die Softwareentstehungsprozesse behindern. Auch hier entsteht trotz der schwierigen Synchronisierung eine bessere Synchronisierung wie zuvor aus Sicht der Entwicklungsteams. [42]

Die erzeugten Veränderungen resultieren in den jeweiligen Unternehmen auch in einer Kostenveränderung, wodurch jede Firma einen Vorteil besitzt. Die effizientere Kommunikation reduziert die bisher notwendigen langen Meetings, wodurch Zeit eingespart wird. Es werden weniger Überstunden erzeugt und es kann flexibler auf Anpassungen reagiert werden, oh-

ne dass eine falsche Entscheidung längerfristig durchgezogen wird, was wiederum auch in Geldwerten bemessen werden kann. [42]

Durch die Umstrukturierung der Teams werden kleinere funktionsübergreifende Teams aufgebaut. Die Aufgabenverfolgung sorgt in Verbindung dafür, dass keine Aufgaben mehr in ein anderes Projekt getrieben werden und danach keinerlei Rückkommunikation mehr stattfindet. Vielmehr entsteht ein Fluss, indem Aufgaben fließen und eine kontinuierliche Performanzsteigerung bedeuten. Das Resultat ist auch, dass die Entwicklungszeit sich reduziert und eine höhere Motivation der Mitarbeiter erzeugt wird. [42]

5.4 Fazit

Dieses Kapitel ist näher auf die Problemstellung der Umwandlung des Entwicklungsprozesses von traditionellen zu agilen Verfahren eingegangen. Hierzu wurden eingangs die generelle Unterschiede zwischen traditionellen und agilen Vorgehensmodellen beschrieben. Zur Darstellung der Herangehensweisen zur Transition wurden zwei generelle Ansätze erläutert, nach der eine Migration eher revolutionär oder evolutionär geprägt sein kann. Zusätzlich dazu diente der Ansatz der agilen Potenzialanalyse, in der eine Analyse der angestrebten Transition dargestellt werden kann und somit vor der eigentlichen Migration ein zu erreichender Agilitätsgrad geprüft wird.

Nach der eher theoretischen Ausrichtung dieses Kapitels folgte die Darstellung von zwei Fallstudien, nach der eine Migration von klassischen Modellen hin zu Scrum verfolgt wurde. Dabei wurden vor allem die Herausforderungen dargestellt, die die jeweiligen Fallstudien darstellten. Zusätzlich dazu wurden Veränderungen beschrieben, die nach der Migration sichtbar wurden. Auch in diesen Fallstudien ist festzustellen, dass keine Transition nach einem globalen Muster verfolgt wurde. Auch ist die erreichte Vorgehensmethode Scrum unterschiedlich stark zur Theorie vergleichbar. Zum Teil ist dies nachvollziehbar und zum Teil werden dabei Verletzungen der agilen Grundsätze sichtbar. Diese Fallstudien, wie auch die vorherigen Kapitel haben dargestellt, dass bei der Migration eine Reihe von Faktoren zu beachten sind. Dabei stehen zum Teil starke Kontraste gegenüber, die erhebliche Herausforderungen in der initialen Einführung von agilen Verfahren sowie in der späteren Adaption bedeuten. Eine weitere Betrachtung von unterschiedlichen Herausforderungen wird im nächsten Kapitel 6 dargelegt.

6 Herausforderungen der agilen Softwareentwicklung

Die agile Softwareentwicklung stellt, wie bereits in den vorherigen Kapiteln und Abschnitten beschrieben, eine ganze Reihe von Vorteilen dar und verspricht dabei häufig eine leichtgewichtige Methode anzubieten. Auch wurde besonders im vorherigen Kapitel auf die Schwierigkeit der Implementierung eingegangen. Ist die erste Implementierung erfolgt, gilt es eine kontinuierliche Adaption des Prozesses zu gewährleisten, um neuen Situationen und Veränderungen der Umwelt, wie z. B. dem Unternehmen, stand zu halten. Des Weiteren existieren eine Reihe von speziellen Herausforderungen, die Inhalt dieses Kapitels sind.

Zunächst soll in diesem Kapitel auf das Thema der Stakeholderidentifikation eingegangen werden. Dabei soll die Problematik hinter diesem Thema verdeutlicht werden sowie Verantwortungen im Scrum Prozess, die im Kontext der Identifikation stehen. Im Anschluss daran folgt eine nähere Auseinandersetzung mit dem Product Owner, welcher Teil des Scrum Modells ist. Im Kontext verschiedener Unternehmensanforderungen und -Prozesse kann die Interpretation der Rolle stark variieren, worauf näher eingegangen wird. Die Rolle des Scrum Masters stellt mit die wichtigste Rolle innerhalb Scrums dar. Auch hier soll auf die Bedeutung der Rolle und die Hindernisse bei der Implementierung sowie das Ausleben der Rolle eingegangen werden. Eine der Aufgaben des Scrum Masters stellt die Identifikation und Verwaltung von Impediments im Prozess dar. Hierzu soll auf die Schwierigkeit der Identifikation nachfolgend eingegangen werden. Im Anschluss folgt die Auseinandersetzung mit der Fragestellung von produktiven agilen Teams. Inhalt dieses Abschnitts ist es, auf die Teamzusammensetzung und Einflussfaktoren einzugehen, die bei dem Diskurs zu produktiven Teams zu führen sind. Besonders aus der klassischen Sichtweise auf Entwicklungsteams, aber auch auf die IT-Abteilung von der Geschäftssicht, wird das Management von Personal, als eine Art Voraussetzung zur Existenz angesehen [15]. Im Bezug auf die agile Softwareentwicklung kann diese Sichtweise zu erheblichen Problemen führen. Vielmehr verändern sich die Aufgaben und Sichtweisen des Managements hierbei und müssen differenziert betrachtet werden. Im Anschluss soll auf die Bestimmung eines Agilitätsgrades weiter eingegangen werden. Inhalt diesen Abschnitts ist

dabei die Beschreibung dieses Begriffs sowie dessen Auswirkungen. Das Ende dieses Kapitels bildet das Fazit, welches eine Zusammenfassung aller Abschnitte darstellt.

6.1 Stakeholder Identifikation

Innerhalb agiler Softwareprojekte besitzen die sogenannten Stakeholder eine etwas andere Bedeutung als in herkömmlichen Softwareprojekten. So stellen diese einen essentiellen Bestandteil der Entwicklung dar und werden dabei an verschiedenen Punkten zum Austausch benötigt. Hierzu ist eine verbesserte Kommunikation notwendig, wie es in nicht agilen Methoden der Fall ist, um z. B. den Fortschritt zu sehen und gegebenenfalls Unklarheiten sowie Priorisierungen von Anforderungen verarbeiten zu können. Es wird dabei innerhalb der Vorgehensmodelle eine kontinuierliche Kommunikation mit den Stakeholdern angestrebt. Die Berichterstattung reduziert sich innerhalb dieser auf die bekannten Punkte: Qualität, Zeit und Budget. [29, 44]

Eine Schwierigkeit, die sich in diesem Kontext ergibt, stellt die Identifikation der Stakeholder dar. Es muss z. B. aus Sicht des Product Owners geklärt werden, wer in den Feedbackprozess und die Aufstellung von Stories involviert wird. Der Begriff des Stakeholders stellt hierzu eine sehr umfangreiche Beschreibung dar und kann je nach Projektgeschäft, Produktgeschäft und einer internen Entwicklung variieren. Hierzu ist die Definition des Kunden selbst variable. Dabei kann sich diese Rolle auf verschiedene Personen verteilen und im Fall des Produktgeschäfts sich auf eine hohe Zahl von Personen referenzieren. In der Betrachtung der internen Entwicklung bildet sich diese Rolle über verschiedenste Managementebenen von Fachbereichen des eigenen Unternehmens ab. [29, 44]

Im Kontext der agilen Softwareentwicklung mit Scrum existieren vor allem zwei Rollen die bei der Identifikation und dem Aufbau der Kommunikation die Verantwortung tragen. Hierzu gehört unter anderem der Scrum Master. Seine Aufgabe besteht zum Teil darin, die Meetings mit zu organisieren. Hierzu ist es wichtig, dass im Fall des Sprint Reviews Personen anwesend sind, die im Sinne von Scrum als Stakeholder gelten und somit darüber entscheiden, ob das Produkt den angedachten Anforderungen entspricht. Unter Umständen gehören hierzu auch Personen, die über Qualität und Geschwindigkeit sowie den zeitlichen Aspekt Stellung nehmen. Als weitere Aufgabe im Bezug auf Stakeholder steht die Implementierung des Prozesses an sich und der Transport des Wissens, wie Scrum funktioniert und zu verstehen ist. Dies ist besonders auch im Bezug auf die Organisationsstrukturen essentiell, um eine kontinuierliche Adaption des Prozesses vornehmen zu können. [29, 44]

Eine weitere wichtige Rollen, die einen sehr starken Bezug zu den Stakeholdern besitzt, hat der Product Owner im Scrum Prozess. Für diese Tätigkeit steht die Kommunikation mit den Stakeholdern besonders im Fokus. Anforderungen, die an das Projekt gestellt werden, werden dabei der Softwareentwicklung übermittelt. Dabei fungiert der Produkt Owner als Transformator der Anforderungen und stellt diese in Rücksprache mit den Stakeholdern dem Team zu Verfügung. Er stellt somit die Verbindung zwischen Stakeholdern und Entwicklern bereit. Dabei wird der Informationsfluss entsprechend gefiltert und auf die für das Team benötigten Fakten reduziert. Um dieser Tätigkeit nachzukommen, gilt es hierbei, mit den richtigen Personen zu kommunizieren und ggf. unterschiedliche Prioritäten der Stakeholder zu regeln. Die Rücktransformation der technischen Begründungen, warum z. B. eine Funktionalität in der gewünschten Form nicht umgesetzt werden kann sowie die Bedeutung und Lösungszeit aus Entwicklersicht, gehören ebenfalls zu seinem Aufgabengebiet.

6.2 Product Owner im Fokus des Prozesses

Die Rolle des Product Owners, ebenso wie des Scrum Masters, stellt im Scrum Prozess eine sehr wichtige und gleichzeitig schwierige dar. Der Product Owner trägt eine große Verantwortung, die über den Erfolg des Projekts entscheidet. Insbesondere fokussiert sich seine Tätigkeit auf den finanziellen Aspekt der Entwicklung. Hierzu gehört der sogenannte „Return on Investment“ (kurz ROI), wodurch ausgesagt wird, ob ein entsprechendes Ergebnis höher ist, als der Einsatz, der investiert wird. Diese Aufgabe obliegt dieser Rolle im kompletten Lebenszyklus der Entwicklung. Realisiert wird dies durch die Verwaltung der Anforderungen, die durch die Entwicklung abgearbeitet werden. In der Funktion des Product Owners trägt dieser die Entscheidung, ob eine Anforderung umgesetzt wird oder nicht. Dazu muss dieser entsprechend eine Rücksprache mit der Entwicklungsabteilung halten, um diese Entscheidung abwägen zu können. So werden Anforderungen zu verschiedenen Zeitpunkten von Seiten der Entwicklung sowie der Stakeholder angepasst und über den Product Owner gefiltert an die jeweiligen Stellen weitergetragen. [35, 53]

Die Kooperation mit dem Team muss entsprechend eng aufgebaut werden, damit eine Verwaltung der Anforderungen möglich ist. Hierzu wird das sogenannte *Product Backlog* in der Regel von ihm verwaltet. Trotz der Entscheidungsgewalt durch die Vertretung der Stakeholder in der Entwicklung bzgl. der Anforderungen besitzt dieser keine Befugnisse im Team selbst. Durch die Vertretung der Stakeholder bedeutet dies auch, dass diese eine entsprechend gute

Verbindung zu dem Product Owner besitzen und ihn mit den Möglichkeiten der Verwaltung ihrer Wünsche und Ideen ausstatten. [35, 53]

In der Betrachtung der Praxis steht der Product Owner verschiedenen Herausforderungen gegenüber. Wie bereits in den vorherigen Absätzen verdeutlicht wurde, liegen viele Koordinierungstätigkeiten und Verantwortungen auf der Rolle des Product Owners. Nach der theoretischen Auslegung dieser Rolle ist diese generell als einzelne Person bestimmt, was wiederum bedeutet, dass all diese Verantwortungen und Aufgaben auf eine Person delegiert werden. Es steht kein Team von Product Owner hiernach zur Verfügung, um ein Feedback oder einen Austausch unter Product Ownern zu ermöglichen. [35, 53]

Die Herausforderung der Verwaltung von Anforderungen wird bei der Betrachtung des Anforderungsflusses im Projekt deutlicher. So muss dieser die Anforderung aus Sicht der Fachabteilung und Stakeholder voll und ganz verstehen, um diese entsprechend an die Entwicklungsabteilung transportieren zu können. Bei diesem Transport muss dieser zusätzlich die Anforderungen der technischen Abteilung verständlich erläutern können. Hierbei muss ein gewisses Maß an technischen Fachwissen vorhanden sein, um einen Kommunikationsaufbau effizient führen zu können. Insbesondere bei Rücksprachen zu den Stakeholdern, wo ggf. Aufgaben begründet abgelehnt werden müssen, stellt die Rücktransformation der Anforderungen mit technischer Begründung, dieses Wissen als Voraussetzung, genau wie die aus Sicht der Fachdomäne. So erfordert diese Rolle einen hohen Wissensstand von beiden Perspektiven. In der Praxis versucht man diese Herausforderungen zu verringern, indem der Product Owner in zwei reale Personen geteilt wird, sodass eine die technische Sicht übernimmt und eine die Sicht des Unternehmens. Die technische Tiefe des technischen Product Owners variiert dabei stark. [35, 53]

Eine Problematik, die sich auch aus der Betrachtung der Rolle in einer Person ergibt, stellt die hohe Abhängigkeit zu einer Person dar. So wird das Risiko eingegangen, dass ein Wissensmonopol über einer Person erzeugt wird. Diese trägt die Vision der Stakeholder und leitet das Projekt. Auch trägt diese eine Person die volle Verantwortung nach der Definition. Um dies auch nur in einem gewissen Maßen realisieren zu können, verlangt diese Rolle, dass sie mit einem hohen Maß an Rechten und Möglichkeiten ausgestattet wird. Hiernach erlangt diese eine hohe Macht innerhalb des Projekts und zu dem gegebenen Kontext. Es ist somit essentiell, dass diese Rolle durch ihre Möglichkeiten sich im Sinne der agilen Entwicklung verhält und diese nicht zu ihrem persönlichen Zweck nutzt. [35, 53]

Zwar steht nach wie vor die Kooperation mit dem Team und dem Scrum Master in der Abarbeitung im Fokus, doch die Ausstattung an Möglichkeiten schafft auch weitere Problematiken. So wird die getragene Motivation des Produktes an das Team übertragen, was wiederum das Team

negativ aber auch positiv beeinflussen kann. Im Zuge der Durchsetzung der Anforderungen erzeugt dieser aber auch Druck auf Manager-Ebenen, indem Anforderungen ggf. klarer formuliert werden müssen oder ein Widerspruch der Anforderungen einfach akzeptiert werden muss. [35, 53]

Diese Rolle stellt wie hier beschrieben eine sehr mächtige und schwer realisierbare Aufgabe dar. In diesem Prozess sind die Entwicklung und auch die Stakeholder involviert. Das bedeutet auch, dass darüber hinaus die Organisationskultur einen erheblichen Einfluss auf diese Rolle haben kann. So wird bei einer falschen Ausstattung an Möglichkeiten sowie einer falschen Besetzung dieser Rolle ein erhebliches Risiko für die Entwicklung eingegangen. Eine Möglichkeit, das Risiko zu reduzieren, stellt bei einer entsprechenden Größe der Entwicklung die Aufteilung in mehrere Personen dar, wodurch auch eine Trennung der Fachlichkeit und ein Austausch untereinander ermöglicht wird. Allerdings ergeben sich hieraus wiederum Herausforderungen, die zu berücksichtigen sind. So müssen die Ansprechpartner in Abhängigkeit dazu definiert werden sowie die Abhängigkeiten unter den Product Ownern. [35, 53]

6.3 Die Rolle des Scrum Masters

Die Rolle des Scrum Masters stellt innerhalb des agilen Vorgehensmodelle Scrum eine der wichtigsten Rollen für eine erfolgreiche Implementierung dar. Seine Tätigkeiten werden dabei besonders in der Praxisbetrachtung nicht immer korrekt verstanden und ausgeübt. Somit ist die Ausübung der Rolle selbst bereits eine Herausforderung. Zudem erwarten den Scrum Master bei der Durchführung seiner Aufgaben weitere Hindernisse. [61, 16, 20, 60]

So wird unter anderem durch ein fehlerhaftes Verständnis die Rolle auch mit anderen Rollen im Scrum Prozess und darüber hinaus vermischt. Ein Scrum Master ist somit parallel ein Entwickler, Produkt Owner, Projektleiter oder Personalverantwortlicher. Diese Liste kann dabei beliebig weitergeführt werden. Die Begründung zur Vermischung liegt dabei häufig in der Auslastung der Tätigkeit oder im benötigten Domänenwissen, um einen wirklichen Überblick zu haben. Zusätzlich ist die Rolle bezogen auf das Ergebnis in Geldwerten schwer messbar. Solche Entscheidungen zur Rollenvermischung werden generell vom Management auf verschiedensten Ebenen gefällt und behindern die eigentlichen Tätigkeit des Scrum Masters. [61, 16, 20, 60]

Ein Grund für Probleme durch Fehlentscheidungen stellt das mangelnde Verständnis von Scrum zwischen der Entwicklung und dem IT-Management dar. Dieses sorgt dafür, dass Fehlinterpretationen entstehen. Die generelle Offenheit der Organisation spielt dabei ebenfalls eine Schlüs-

selbste. Ist die generelle Implementierung bewältigt, so warten weitere Herausforderungen, wie z. B. bei der kontinuierlichen Schaffung von Awareness der agilen Softwareentwicklung gegenüber Stakeholdern und dem Unternehmen an sich. [61, 16, 20, 60]

Im aktiven Prozess können dabei immer wieder Angriffe auf die Entwicklungsteams gestartet werden, um das klassische Projektmanagement wieder stärker zu betrachten oder eine unternehmensweite Struktur von IT-Abteilungen durchzusetzen. So entstehen hierdurch auch Abhängigkeiten zu Standard-IT-Prozessen, die im Entwicklungsteam keine Beachtung finden, indem sich als Beispiel *Sprintdurchläufe* mit *Meilensteinplanungen* überschneiden. Der korrekte Wissenstransfer zu den Stakeholdern stellt somit innerhalb des Unternehmens sowie ggf. zu den Kunden einen wichtigen Punkt dar. Zur Wissensthematik gehören dabei auch der Transfer von Wissen im und außerhalb des Teams, um eine nachhaltige Softwareentwicklung zu ermöglichen. Dabei bezieht sich das Wissen sowohl auf den Prozess selbst, als auch auf die Unterstützung des Transfers von Fachwissen. [61, 16, 20, 60]

Die kontinuierliche Adaption stellt für den Scrum Master eine Herausforderung dar, die sich durch die bereits genannten Schwierigkeiten abbilden lässt. Hierzu benötigt dieser auch die Möglichkeit, sich durch seinen Charakter als starke Persönlichkeit mit Konfliktbereitschaft durch das Unternehmen zu bewegen. Die Fähigkeit, den Prozess auf einer Metaebene zu betrachten, ermöglicht es ihm, Probleme abstrakt zu erkennen und zu beseitigen. Um dies zu erreichen, stellt sich die Rollenvermischung sowie starke Abhängigkeit zur direkten Umsetzung der Projekte und Produkte, ebenfalls als weitere Adaptionbremse dar. Diese Probleme und Herausforderungen werden hierzu über Impediments erfasst und versucht zu bewältigen. Die Betrachtung der Impediments wird im nachfolgenden Abschnitt näher erläutert.

6.4 Impediments im Scrum Prozess

Innerhalb des Scrum Prozesses ist es notwendig, dass Herausforderungen und Problemstellungen des Teams beseitigt werden. Hierzu wird insbesondere der Begriff der Impediments verwendet, um diese zu erfassen. Die Aufgabe der Lösung und der Aufstellung dieser Impediments obliegt dem Scrum Master. [3]

Eine Verwaltung dieser Impediments kann dabei über verschiedene Arten realisiert werden. Eine Möglichkeit stellt das sogenannte Impediment Backlog dar. Welches hierzu Ähnlichkeiten bzgl. der Eigenschaften zum Produkt und Sprint Backlog aufweisen kann. Dieses ist dabei *DEEP* (Detailed Appropriately - Estimated - Emergent - Prioritized) aufzubauen und von oben nach unten abzarbeiten. Hierunter versteht sich unter dem ersten „D“, dass Impediments

detaillierter werden, umso weiter sie in der Reihenfolge nach oben wandern. Das eine „E“ steht für die Abschätzung, also wie viel Aufwand benötigt wird, um das Impediment zu beseitigen. Unter dem weiteren „E“ ist zu verstehen, dass das Backlog nicht starr ist. Es entwickelt und verändert sich über die Zeit. Die Reihenfolge der Impediments, nach der diese abgearbeitet werden, kann sich somit durch einen wechselnden Kontext ergeben. Der letzte Buchstabe „P“ steht für das, wonach eine Priorisierung der Impediments zu verstehen ist. Diese Priorisierung wird über die Reihenfolge der Impediments im Backlog dargestellt. Die wichtigsten Punkte sind somit oben im Backlog und dann absteigend erfasst. Eine weitere Möglichkeit stellt die Vermischung von Impediments und dem generellen Sprint Backlog dar. Dabei können Impediments auch dort verfolgt werden. Nach einer Studie von 2014 [33] setzten weniger als 50 % Impediments Backlogs ein. Eine weitere Beschreibung, wie diese Backlogs aussehen, verwaltet und abgearbeitet werden, wird dabei nicht getätigt.

Wie bereits in dem vorherigen Kapitel angedeutet, existiert eine große Vielfalt an unterschiedlichen Impediments, die während der Entwicklung das Team beeinflussen können. Die Auflösung dieser stellt somit eine wichtige Aufgabe des Scrum Masters dar, damit eine weitere Adaptierung wie auch Produktivitätssteigerung realisiert werden kann. Der Ursprung dieser Beeinflussung ist dabei ebenfalls als vielfältig zu kennzeichnen. Die Abbildung 6.1 stellt dies beispielhaft dar.

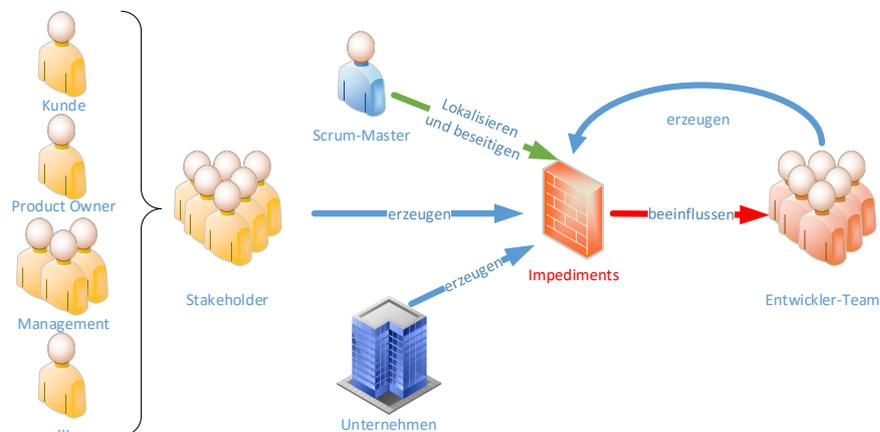


Abbildung 6.1: Beeinflussung des Teams im Scrum Prozess

Wie in der Grafik zu erkennen, stellt diese als Erzeuger der Impediments drei Gruppen dar. Hierzu gehören die unterschiedlichsten Stakeholder, sie sich wiederum auf verschiedene Untergruppierungen referenzieren lassen. Diese lassen sich ebenfalls als Herausforderung

des Prozesses definieren und wurden im Abschnitt 6.1 näher beschrieben. Das Unternehmen selbst kann z. B. über die Unternehmenskultur und den Entwicklungskontext, losgelöst von Personen, die den Stakeholdern zuzuordnen sind, die Entwicklung im Zyklus stören. Die letzte Gruppierung der Abbildung stellt das Team selbst dar. Das Team selbst besitzt die Macht, durch persönliche Differenzen mit anderen Kollegen oder durch eine mangelnde Bereitschaft zur Verfolgung der agilen Softwareentwicklung, den Ablauf stören.

Zusätzlich zu den hier nicht vollständig genannten Impediments lassen sich auch weitere Bereiche finden, die eine Beeinflussung des Entwicklungsvorgehens vornehmen können. Die Erfassung der Impediments stellt dabei genau wie die Verwaltung dieser ebenfalls eine Herausforderung dar. Diese besitzt zusätzlich unterschiedliche Prioritäten, die sich in Kosten oder auch *Impact* (engl. Auswirkung) und Beseitigungsaufwand aufschlüsseln lassen.

Eine Möglichkeit zur Messung von Impediments stellt das sogenannte Impediment Impact Diagramm dar [45]. Hierzu gilt es den Impact auf das Team zu bestimmen sowie die benötigte Zeit zur Beseitigung. Die Abbildung 6.2 stellt dieses mit extra Dimensionen dar.

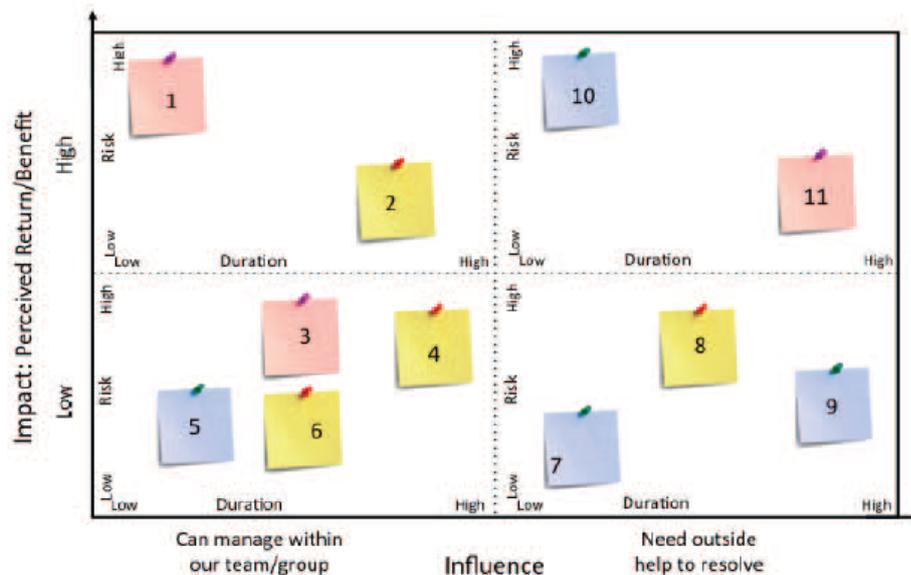


Abbildung 6.2: Impediment Impact Diagramm mit extra Dimensionen für Risiko und Beseitigungsaufwand [45]

Das Diagramm dient dabei der Erkennung von Einfluss eines Impediments, welches im Verhältnis zum Vorteil der Beseitigung steht. Zur Bestimmung dieses Diagramms wird dabei Aufwand an verschiedenen Stellen generiert, um die Informationen zu erlangen. Wie in der Grafik

dargestellt, lässt sich auch eine feinere Unterteilung realisieren. So kann je Quadrant wiederum die Zeit zur Beseitigung sowie das Risiko der jeweiligen Impediments zur Priorisierung und Sortierung beitragen.

Es müssen Stakeholder bestimmt werden, die eine Relation zu den Impediments besitzen und Personen, die eine Einschätzung zum Vorteil der Beseitigung vertreten können. Somit lässt sich ableiten, dass der Scrum Master nicht mehr alleine diese Aufgabe übernehmen kann und die Auswahl der Impediments zur Analyse entsprechend sorgfältig getätigt werden muss.

Der Vorteil stellt eine verbesserte Beseitigungsmöglichkeit dar. Es werden hierzu entsprechende Argumente zusammengetragen, die auch bei der Durchsetzung von Veränderungen an unterschiedlichen Stellen im Unternehmen hilfreich sind. Das Ergebnis bietet dem Team daraufhin eine gesteigerte Adaptionfähigkeit sowie eine parallele Produktivitätssteigerung. Zur näheren Befassung von produktiven Teams dient der nachfolgende Abschnitt.

6.5 Produktive agile Teams

Eines der Vorteile, die den agilen Vorgehensmodellen zugeschrieben wird, stellt die Steigerung der Produktivität von Entwicklungsteams dar. Hiernach wird im Zusammenspiel mit der ausgewählten agilen Methode und den sich daraus resultierenden Veränderungen im Unternehmen dieser Vorteil erreicht.

Ein wichtiger Faktor, der hier eine Rolle spielt, ist die Fähigkeit des Teams, sich auf diese neue Ausrichtung der Entwicklungsabteilung einzustellen. Hierzu ist es wichtig, dass eine entsprechende Teamstruktur aufgebaut werden kann. Durch die intensivere Kommunikation und Transparenz der Teammitglieder wird erwartet, dass sich auf diese starken Veränderungen eingelassen wird. So ist hierdurch ein viel größeres Vertrauen unter allen Beteiligten notwendig und es bauen sich geteilte mentale Modelle auf, nach dem sich vergleichbare Werte und Ziele der Projektausrichtungen darstellen. Im Zuge dessen liefert die Teamzusammensetzung sowie weite Einflussfaktoren einen wichtigen Punkt in der initialen Implementierung und in der weiteren Adaption der agilen Methoden. [12, 39]

Nach dieser Aussage stellt die Hervorhebung und Betrachtung des Teams eine essentielle Aufgabe der Analyse zur Realisierung von agilen Methoden dar. So wird auch klar, dass die soziologischen Faktoren innerhalb der Teams eine entscheidende Rolle spielen. Anders als mit traditionellen Vorgehensmodellen muss verstanden werden, wie sich Teams aufbauen

und weiterentwickeln können. Nur so lassen sich zukünftige Herausforderungen im Team bewältigen und das Team gemeinsam voranbringen. [39]

Ein zusätzlich wichtiger Punkt ist die Berücksichtigung von Einflussfaktoren, die das Team bzgl. ihrer Produktivität stören. Das bedeutet, wie bereits beschrieben, dass diese auch generelle Impediments im Fluss der Entwicklung darstellen und sich zusätzlich negativ auf die bisherige Produktivität auswirken. Hierzu sind vor allem Abhängigkeiten zu alten Vorgehensmodellen zu beachten. Im Konkreten bedeutet dies, dass Teammitglieder parallel Arbeiten in traditionell und agil ausgerichteten Teams durchführen müssen. Durch die sehr unterschiedlichen Herangehensweisen der Modelle sind diese, nicht nur von der Terminierung her, schlecht zu synchronisieren. So wird bei dieser Verfolgung ein zusätzliches Risiko eingegangen, da die Beeinflussung der Teammitglieder auch Auswirkungen auf die Performance der agilen Teams hat. Ein weiteres Thema stellen Kommando- und Kontrollstrukturen dar. Diese sind generell schlecht mit den agilen Werten vereinbar und zerstören Praktiken wie das Pull statt Push Verfahren. Bei der Betrachtung des Unternehmens, welches sich im Kontext zur Entwicklung befindet, existieren weitere generelle Produktivitätshindernisse. So stellen insbesondere Hierarchien sowie Unternehmenskulturen und -philosophien Risiken dar, sofern sich diese kontraproduktiv zur Softwareentwicklung abbilden lassen. [12, 39]

In der Analyse dieser Ausrichtung von Hindernissen des Prozesses sind, wie in den vorherigen beschriebenen Kapiteln dargestellt, die Betrachtung von praxisnahen Untersuchungen unabdingbar. Rein theoretische Modelle, die sich auf veraltete Gegebenheiten beziehen, können hier nur bedingt Vorschläge liefern. Vergleichbare Situationen aus praxisnahen Forschungen lassen hierzu mehr Spielraum der möglichen Ausrichtung, wobei diese auch nur optionale Wege darstellen. Die Gegebenheiten der eigenen Firma bzw. der realen Situation lassen sich nicht über globale Muster erklären und verbessern, da jedes Unternehmen, jedes Team, jede Person individuelle Werte, Vorstellungen und Erfahrungen besitzt. So ist zum einen das Verständnis der einzelnen Methoden zu schulen. Zum anderen gilt es, ein Verständnis dafür zu schaffen, dass Wissen sich im Team nicht auf einzelne Personen referenzieren, sondern sich immer über mehrere Personen rückbilden lässt. [12, 39]

Die Schulung und das *Coaching* muss dabei für alle Beteiligten erfolgen. In der Betrachtung des Managements ist hier insbesondere ein Umdenken erforderlich, damit agile Prozesse realisiert werden können. Hierzu stellt die Umstellung des Projektmanagements auf das agile Projektmanagement eine Herausforderung dar, auf das im nachfolgenden Abschnitt näher eingegangen wird.

6.6 Agiles Projektmanagement

Das Projektmanagement stellt aus Sicht des Management eine äußerst wichtige Aufgabe dar. Durch das Verfolgen eines generalisierten Projektmanagements wird es auch auf oberen Managementebenen ermöglicht, einen Überblick über laufende Projekte und deren Projektstatus zu erhalten. Zusätzlich bietet ein vereinheitlichtes Projektmanagement die Möglichkeit, eine Synchronisierung zu ermöglichen. Die Verfolgung dieser Projektführungsstrategie ist aus der agilen Sicht eher klassischen Verfahren zuzuordnen. Die angestrebte Vorausplanung versucht vor Beginn des Projekts möglichst viele Ereignisse und Aktivitäten festzusetzen. Diese Art der Verfolgung erinnert dabei stark an die klassischen Modelle der Softwareentwicklung.

Besonders beim Übergang des Projektmanagements in die agile Softwareentwicklung ergeben sich durch die klassische Verfolgung von Projekten Hindernisse im Entwicklungsprozess. So lässt sich die Meilensteinplanung des Projektmanagements mit einer *Sprintplanung* schwieriger synchronisieren, als mit einer langfristigen Vorausplanung wie bei klassischen Modellen. [15]

Durch die starken Abweichungen, die in agilen Softwareentwicklungsverfahren existieren, ist eine grundsätzlich andere Verfolgung des Projektmanagements in diesen agilen Projekten notwendig. Das Durchsetzen des klassischen Projektmanagements führt hingegen zu einer Reduzierung der Adaptionsgeschwindigkeit und kann diese im Extremfall komplett verhindern. [15]

So wird zusätzlich zur generellen Planung von Projekten eine andere Art der „Führung“ verlangt. Die traditionelle Art der Kommando- und Kontrollstrukturen entfällt dabei. Vielmehr agiert ein agiler Projektmanager als „Leader“, sodass das Team unterstützt wird, statt zu delegieren und zu prüfen. [15]

Aus der Verfolgung der agilen Projekte geht zusätzlich hervor, dass Aktivitäten nicht mehr wie bisher klar und detailliert beschrieben werden. Durch kurzfristig auftretende Veränderungen im Verlauf des Projekts wird versucht, dass der Fluss der Aufgaben optimiert wird. Das bedeutet, dass die Zeit zwischen Planung und Abarbeitung deutlich reduziert wird. Der veränderte zeitliche Aspekt liegt auch daran, dass Projekte immer komplexer und unklarer zu Anfang sind. Eine zu detaillierte Vorabplanung würde bedeuten, dass ein erheblicher Aufwands- und Kostenfaktor entsteht. Agile Methoden versuchen besonders bei der Betrachtung der Aktivitäten anders vorzugehen. So wird eine minimal notwendige Planung der Aktionen angestrebt. Durch den Verlauf des Projekts wird diese dann entweder bei Bedarf weiter spezifiziert oder in der Abarbeitung geklärt, ohne dass eine möglichst vollständig, detaillierte, initiale Beschreibung erfolgt. [15]

Die veränderte Bedeutung der Teams im agilen Kontext fordert ebenfalls andere Sichtweisen wie bisher. Das Thema „Kreativität“ und „Lernen“ stellt eine Voraussetzung zur Verfolgung von agilen Projekten aus Sicht des Managements dar. In der Betrachtung komplexer, sich verändernder Anforderungen und zukünftigen Herausforderung, kann so die Verfolgung eher sichergestellt werden, als beim Ignorieren dieser Faktoren. So wird in Abhängigkeit hierzu versucht, dass eine Redundanz von Wissen und Entscheidung berücksichtigt wird, damit keine Wissensmonolithen entstehen können und im Risikofall die Entwicklung annähernd handlungsunfähig wird. Um diese Strategie zu unterstützen, sind intensivere Feedback- und Lernprozesse zu implementieren, damit ein Austausch und eine kontinuierliche Optimierung der agilen Prozesse erreicht werden kann. [15]

Das Projektmanagement steht sich angesichts dieser aufgeführten Punkte, einem kompletten Wandel gegenüber. So verändern sich insbesondere die Softwareprojekte hin zu einem geteilten Entscheidungsmanagement, wo ein Austausch und eine weitere Verfolgung der Tätigkeiten gemeinsam entschieden werden. Insgesamt wird dabei angestrebt, dass die Teams in ein Selbstmanagement übergehen. Das Team mit seinen Fähigkeiten stellt in der Zusammensetzung und in dem zugesprochenen Vertrauen einen ganz andere Fokus, als es bisher der Fall war. In diesem Sinne müssen Teamstrukturen besser verstanden, das Team besser kennengelernt und verstanden werden, damit ein agiles Projektmanagement realisierbar ist. Ohne die Betrachtung dieser Punkte ist das Erreichen eines höheren agilen Grads nicht möglich, worauf im nächsten Abschnitt weiter eingegangen wird. [15]

6.7 Agilitätsgrad

Wie bereits in diesem Kapitel beleuchtet wurde, stellt die Implementierung von agilen Prozessen eine erhebliche Herausforderung für die Unternehmen und Teams dar. Ebenso wurde im Kapitel 5 beschrieben, dass die Transition über verschiedene Ansätze erfolgen und eine vorherige Analyse der Gegebenheiten die Erwartungshaltung in Bezug auf die Realisierungsmöglichkeiten realistisch darstellen kann. Insbesondere wurde hierzu auf die agile Potenzialanalyse im Abschnitt 5.2.3 näher eingegangen.

Wie bereits dort angedeutet, stellt sich bei der Implementierung von Scrum und anderen agilen Verfahren die Frage, wie viele Aspekte wirklich umsetzbar sind. Durch diese Analysearbeit lässt sich häufig bereits darstellen, dass eine evolutionäre Einführung einer revolutionären Implementierung zu bevorzugen ist. Je nach Unternehmenskontext ist hiernach nicht jeder Agilitätsgrad geeignet. [49]

Nach der Definition der Fraunhofer Forschungsgruppe IESE (Fraunhofer-Institut für Experimentelles Software Engineering) definiert sich der Grad an Agilität an der Anzahl und Art der verwendeten agilen Praktiken. Wonach sich der maximale Grad nicht absolut definieren lässt, sondern sich im Kontext des Unternehmens bewegt. Das bedeutet insbesondere auch, dass ein hoher Grad im Vergleich zu Unternehmen nicht dadurch erreicht werden kann, dass alle existierenden agilen Praktiken umgesetzt werden. Eher geht es um die Analyse und die sich daraus ergebenden Möglichkeiten für das Unternehmen. Zusätzlich hierzu steht dann die Detailstufe der agilen Praktiken, inwieweit diese verstanden und korrekt verfolgt werden. Die ausschließliche Definition über die Anzahl der agilen Praktiken ist kritisch zu betrachten. [49]

Nach diesen Erkenntnissen lässt sich auch darstellen, dass Scrum nicht nach einem global gültigen Muster für alle Unternehmen implementierbar ist. Der Scrum Guide ist demnach eher als Richtlinie zu verstehen und macht Problemstellungen deutlich sowie Potenziale, die erreicht werden können. Identifikatoren für korrekte Adaptionen sind demnach eher, dass das Team entscheidet, wie etwas besser im Prozess zu verwenden ist. Implementierung die alleine über die Potentialanalyse angestoßen werden, ohne entsprechenden Rückhalt im Team zu haben, werden demnach ebenfalls scheitern. Zu dieser Art der Durchsetzung stehen auch die generellen Entscheidungen, die eher Hierarchie-getrieben sind. [13, 35, 49, 53]

Der Agilitätsgrad beschreibt somit eher eine Darstellungsmöglichkeit eines Unternehmens, nach der sich dieses agiler fortbewegen möchte. So wird dieser Begriff eher verwendet, um kontinuierlich zu verstehen, wie sich das Team in Kooperation verbessern kann. So kann dieser wiederum bedeuten, dass die Grenzen des Unternehmens klarer definiert werden müssen. Eine Betrachtung der Ist-Situation ist dafür unabdingbar. Nur durch eine praktische Analyse und einem Diskurs kann geklärt werden, wie eine Implementierung und spätere Adaption realisierbar ist.

6.8 Fazit

Inhalt dieses Kapitels war die Darstellung von verschiedenen Herausforderungen, die bei der Implementierung sowie der weiteren Adaption von agilen Verfahren zu berücksichtigen sind. Hierzu wurde vor allem ein Fokus auf die agile Softwareentwicklung mithilfe von Scrum gesetzt. Die Auseinandersetzung der Stakeholderidentifikation hat deutlich gemacht, dass die Identifikation eine wichtige und zugleich nicht triviale Aufgabe darstellt. Es muss geklärt werden, welche Stakeholder stark in den Prozess involviert werden und welche Rollen bei

der Identifikation unterstützen. Je nach Rolle ergeben sich verschiedenste Bedingungen und Abhängigkeiten.

Im Bezug zum Product Owner wurde darauf aufbauend die Aufgabe dieser Rolle spezifiziert und auf die Vielfältigkeit durch den starken Bezug zum Domänenwissen und den technischen Abhängigkeiten hingewiesen. Zusätzlich wurde auf die generellen Herausforderungen im Prozess zu dieser Rolle eingegangen. Dadurch wurde ersichtlich, dass eine große Verantwortung und Abhängigkeit nach der Theorie auf einer Person liegt.

Eine weitere Rolle die danach vorgestellt wurde, war die Rolle des Scrum Masters. Auch hier wurde weniger auf eine generelle Beschreibung der Rolle gesetzt und ein stärkerer Fokus auf die Herausforderungen im Scrum Prozess gelegt. Dabei wurde auf die Problematik der Rollenvermischung und die sich daraus ergebenden Probleme eingegangen. Des Weiteren ist im Bezug auf die Aufgaben auch auf die Herausforderung der agilen Wissensimplementierung hingewiesen worden. In Abhängigkeit zu den Aufgaben der Rolle wurden darauffolgend mögliche Impediments im Prozess, in Abgrenzung zu technischen Impediments, diskutiert. Als besondere Problematik ist dabei auf die Identifikation und Verarbeitung eingegangen worden. Die Relevanz der Impediments in Bezug auf produktive agile Teams wurde darauf aufbauend beschrieben. Hierzu sind weitere Faktoren dargestellt worden, die die Optimierung von Teams beeinflussen können. Diesbezüglich stellt auch das Management zur Projektverfolgung einen Einflussfaktor dar. Durch die generellen Veränderungen im Prozess hinzu agilen Verfahren unterliegt auch der Bereich des Projektmanagements einem großen Wandel. Dieser wurde im Anschluss erläutert.

Die Aufgaben des Managements beinhaltet in dieser Betrachtung eher unterstützende Maßnahmen, um eine kontinuierliche Adaption und Optimierung zu ermöglichen. Im Zuge der Weiterentwicklungsmöglichkeiten wurde nachfolgend auf den Agilitätsgrad eingegangen und dieser Begriff näher spezifiziert. Hiernach sind vor allem die Kontextinformationen sowie die Abhängigkeiten durch das Unternehmen, zu erfassen, um realistische Ziele aufstellen zu können und Argumentationsmöglichkeiten bei größeren Veränderungen im Unternehmen über die Entwicklungsabteilungen zu erhalten. Die notwendigen Analysen, die für die Bestimmung des Agilitätsgrades notwendig sind zeigen unter anderem, dass die Realität und Praxis bestimmt, inwieweit theoretische Praktiken angewendet werden können. Insgesamt hat dieses Kapitel mehrmals auf die Herausforderung und Bedeutung aufmerksam gemacht, die sich vor allem durch Erkenntnisse von Untersuchungen aus der Praxis agiler Softwareentwicklung ergeben. In Abhängigkeit hierzu wird im nachfolgenden Kapitel eine empirische Erhebung zur Scrum Implementierung dargestellt, um zusätzliche Informationen zum aktuellen Stand darzulegen.

7 Empirische Erhebung zur Scrum Implementierung

Inhalt dieses Kapitels ist die Erhebung von Informationen aus der Praxis agiler Softwareentwicklung, die zur Scrum Implementierung beitragen. Im Rahmen eines vorherigen Projekts wurden bereits Experteninterviews durchgeführt, die in der Arbeit für die Verfolgung anderer Ziele neu verarbeitet wurden.

Zunächst wird die ursprüngliche Untersuchung näher beschrieben. Zu erwähnen ist, dass die Ergebnisse der bereits durchgeführten Interviews nicht Teil dieser Arbeit sind. Darauf aufbauend wird im nachfolgenden Abschnitt die Neubetrachtung der durchgeführten Untersuchung erläutert. Jedes Interview wird hierzu nachfolgend einzeln beleuchtet und anschließend in einer Gesamtauswertung dargestellt. Das Fazit dieses Kapitels fasst das gesamte Kapitel nochmals zusammen.

7.1 Ursprüngliche Untersuchung

Im Rahmen eines Projekts zur Analyse der Impediments von Scrum Mastern [31] wurde eine Untersuchung durchgeführt, die nachfolgend beschrieben wird. Die Herausforderung der Handhabung von Impediments wurde im Abschnitt 6.4 bereits beleuchtet.

7.1.1 Ziele

Ziel der Untersuchung war die Identifikation von Impediments des Scrum Masters. Generell erfasst der Scrum Master Impediments, die sich im Team ergeben. Durch die Untersuchung sollte hierdurch eine Reflexion der Rolle durchgeführt werden. Im Detail bedeutet dies, dass beschrieben werden sollte, mit welchen Herausforderungen der Scrum Master in seiner Tätigkeit konfrontiert wird.

Anschließend sollte es durch die Ergebnisse möglich sein, die Impediments einzustufen. Die Einstufung sollte es ermöglichen, dass eine Generalisierung der Impediments über alle Interviews stattfindet.

Das weitere Ziel stellte die Rückführung der Erkenntnisse auf Unternehmensstrukturen dar. Hierunter ist zu verstehen, dass die Ergebnisse ggf. in typisch hierarchischen Unternehmensstrukturen wiederzufinden sind.

7.1.2 Aufbau

Nachfolgend wird der Aufbau der durchgeführten Untersuchung beschrieben. Zunächst wird die Methode erläutert, nach der vorgegangen wurde. Darauf aufbauend wird die Auswahl der Probanden vorgestellt, die an der Untersuchung teilnehmen sollten. Anschließend wird die Verarbeitung der Interviews näher beschrieben.

Methode

Die Untersuchung wurde mittels Experteninterviews durchgeführt. Beim Experteninterview handelt es sich um eine Methode der qualitativen Sozialforschung. Diese dient vor allem der Quantifizierung der Repräsentativität. Hierdurch wird es möglich, dass subjektive Deutungen und Interpretationen rekonstruiert werden. Die Alternative wäre, dass man versucht, Fakten darzulegen. [9]

Zusätzlich wurde das Experteninterview über einen Interviewleitfaden [30] gestützt, welcher im Abschnitt 7.1.3 näher beschrieben wird.

Auswahl der Probanden

Bei dieser Untersuchung waren durch die Zielvorgaben bestimmte Voraussetzungen gesetzt. Es sollten ausschließlich Scrum Master als Probanden gefunden werden. Dadurch musste ein Unternehmen gefunden werden, welches sich an der Vorgehensmethode Scrum orientiert. Im Vorhinein war allerdings bewusst, dass Firmen gewisse Abweichungen zur Theorie pflegen. Dadurch lag der letztendliche Fokus auf Probanden, die eine Rolle besetzten, die dem Scrum Master zuzuordnen sind.

Auf der Suche nach geeigneten Firmen wurden vor allem Kontakte genutzt, die eine Verbindung zur Hochschule besitzen. Diese Kontakte wiederum stellten durch die Vorgabe der Scrum Master

angelehnten Rolle einen Kontakt zu möglichen Probanden her. Hierüber ließen sich mögliche Teilnehmer der Untersuchung finden. Die weitere Korrespondenz verlief zunächst über E-Mail. Neben eines Einverständnisses sowie einer Terminabstimmung lautete der Inhalt dieser Korrespondenz wie folgt:

1. Kurzbeschreibung meiner Person
2. Kurzbeschreibung des Untersuchungsrahmens
3. Das Hauptziel der Untersuchung
4. Ablauf des Interviews

Die verwendete Vorlage der E-Mail kann dem Anhang [A](#) entnommen werden.

7.1.3 Interviewleitfaden

Der verwendete Interviewleitfaden wurde im Rahmen einer vorherigen Arbeit [30] ausgearbeitet. Das Ziel dieses Leitfadens war dabei, dass Probleme und Herausforderungen von Scrum Masters besser eingestuft werden können. Um die weiteren Ziele, die im Abschnitt 7.1.1 zu finden sind, zu erreichen, wurde der folgende Ablauf aufgebaut:

1. Vorstellung des Projekts
2. Befragung zum professionellen Kontext des Scrum Masters
3. Befragung bzgl. Hindernissen (Impediments) in Projekten
4. Befragung zu nützlichen Tools bzw. Hilfsmitteln

Der komplette Leitfaden mit allen Fragen kann dem Anhang [D](#) entnommen werden.

7.1.4 Durchführung der Untersuchung

Zur Untersuchung sind, wie bereits genannt worden, die Experteninterviews verwendet worden. Insgesamt wurden in der durchgeführten Untersuchung fünf verschiedene Probanden befragt. Diese Personen waren, wie bereits in Abschnitt 7.1.2 beschrieben, entweder ein Scrum Master oder belegten Rollen, die von der Tätigkeit her der Rolle des Scrum Masters zuzuordnen sind. Zudem sind die Personen in unterschiedlichen Firmen beschäftigt. Insgesamt haben vier verschiedene Unternehmen an der Untersuchung teilgenommen. Das bedeutet, dass in einer Firma zwei Probanden befragt wurden. Die Unternehmen sind dabei als Mittel- oder Großunternehmen zu bezeichnen. Zuzuordnen sind diese den Branchen der IT, dem Online- und

Versandhandel sowie Internet und Onlinemedien. Die Unternehmen besitzen jeweils einen Standort in Hamburg, wo auch die Befragung durchgeführt wurde.

Die Interviews wurden in der Regel in den Unternehmen direkt durchgeführt. Hierzu hatten die Probanden jeweils nach dem Vorschlag zugesagt, in den jeweiligen Unternehmen das Interview durchzuführen. Die Begründung hierzu lag auf der besseren Planungsmöglichkeit. Das bedeutet wiederum, dass die Hemmschwelle für eine Zusage zur Untersuchung gesenkt wurde, da sich diese besser in den Arbeitsalltag integrieren ließ. Durch getätigte Aussagen aus den Interviews lässt sich auch vermuten, dass eine bessere und direktere Verbindung zu den Fragen erreicht wurde. Bestimmte Herausforderungen stellten aktuelle Situationen dar, die bewältigt werden mussten.

Die einzige Ausnahme bei der Wahl der Örtlichkeit zur Untersuchung stellte eine Befragung dar. Hier wurde ein Proband in einem Café befragt. Zudem wurde die Befragung außerhalb der Arbeitszeit durchgeführt.

Bei der Betrachtung der zeitlichen Komponente wurden insgesamt acht Stunden und vierzig Minuten für die Befragungen aufgewendet. Das bedeutet, dass je Interview ca. eine Stunde und vierundvierzig Minuten benötigt wurden. Der Zeitraum der Befragung erstreckt sich dabei von Anfang Januar bis Anfang März 2016.

Zur Erfassung der Ergebnisse wurden grundsätzlich zwei Verfahren gewählt. Zum einen diente der konstruierte Interviewleitfaden, welcher im Abschnitt 7.1.3 erläutert wird, zur direkten Aufnahme von Notizen. So wurden je Beantwortung einer Frage oder bei Diskussionen, die sich im Kontext einer Frage bewegten, Informationen direkt an der Frage niedergeschrieben. Zum anderen wurden die Untersuchungen über eine Sprachaufzeichnung aufgenommen. Hierzu diente die Aufnahmefunktion eines Smartphones. Diese Aufzeichnungen wurden mit Ausnahme eines Interviews aufgenommen. Dabei ist im Vorhinein das Einverständnis der Probanden eingeholt und darauf verwiesen worden, dass die Ergebnisse und Details der Befragung nicht auf einzelne Personen oder die Firma zurückverfolgbar ist bzw. eine Anonymisierung bei der Auswertung stattfindet. Die Sprachaufzeichnungen werden ebenfalls nicht veröffentlicht.

7.1.5 Verarbeitung der Interviews

Die Interviews wurden im Rahmen eines vorherigen Projektes wie folgt ausgewertet. Zunächst wurde jedes Interview einzeln betrachtet. Hierzu ist im ersten Schritt, direkt nach dem Interview,

der Fokus auf den jeweils mit Notizen versehenen Interviewleitfaden gesetzt worden. Die Notizen wurden ergänzt und näher beschrieben.

Im zweiten Schritt ist die aufgezeichneten Tonspuren verwendet worden, um die einzelnen Fragen ein zweites Mal zu prüfen. Hierbei wurden die Notizen ergänzt und bestehende Notizen begutachtet. Das Ergebnis jedes Interviews ergab somit eine Anzahl von Stichpunkten je Frage.

Bei der Analyse der aufgezeichneten Tonspuren wurden bereits weitere Informationen aufgenommen, die zwar nicht in diesem Fokus waren, aber trotzdem weitere Erkenntnisse präsentieren. So stellten diese Ergebnisse Punkte dar, die für weitere Untersuchungen hilfreich sein können. Diese Daten sind im Anhang C abgelegt.

Innerhalb der nächsten Auswertungsphase wurden die Ergebnisse in Relation zueinander gesetzt. Hierfür dienten entworfene Tabellen, die je Fragenbereich gegenübergestellt wurden. Die Fragenbereiche lauten dabei:

- Rahmeninformationen (Firma, Ort des Interviews, Dauer, ...)
- Professioneller Kontext (Seit wann Scrum Master, parallele Rollen, ...)
- Impediments (Auswirkungen, Beseitigungsaufwand, ...)
- Tools und Hilfsmittel (Erfassung der Impediments, Archivierung, ...)

Diese Tabellen können dem Anhang B entnommen werden. Um einen konkreteren Bericht für das Projekt zu erstellen, wurden die Daten im Fokus der Auswirkungen (engl. Impact) zusammengefasst. Diese Daten sind ebenfalls im Anhang C zu finden.

Im letzten Schritt wurden die erfassten Impediments kategorisiert und näher beleuchtet. Die Kategorien entstanden durch die bereits referenzierten Daten. Diese enthalten Umschreibungen oder konkrete Informationen, die sich auf die nachfolgenden Kategorien beziehen. Hieraus ergab sich zum einen die Kategorie „Impediments im Fokus des Teams“, welche die folgenden Subkategorien besitzt:

- Technische Impediments
- Impediments von außen
- Impediments durch das Team

Die zweite Hauptkategorie „Impediments im Fokus des Scrum Masters“ besitzt die folgenden Subkategorien:

- Organisatorische Impediments
- Impediments von außen

- Impediments durch Rollenkonflikte

Eine detaillierte Beschreibung der Kategorien und weiteren Erkenntnissen kann dem Projektbericht entnommen werden [31].

Der Fokus der Befragung lag auf der Erfassung von Impediments im Kontext der Rolle des Scrum Masters. Dabei wurde, unabhängig von der Anzahl an Projekten, zwischen dem Produkt und der Art der Softwareentwicklung unterschieden. Es sollten möglichst Informationen in breiter Form sowie einer höheren Detailstufe zu den jeweiligen Impediments erfasst werden.

7.2 Neuauswertung der Untersuchung

Durch die bereits durchgeführten Interviews, welche neu ausgewertet werden, ergeben sich direkte Konsequenzen zur Ergebnisverwertung. So ist das Gespräch mit dem Probanden durch den Fokus auf Impediments auf Kritik am eigenen Prozess ausgelegt. In der Ergebnisverwertung fallen somit deutlich mehr kritische Punkte ins Gewicht, die sich im Kontext der agilen Softwareentwicklung wahrscheinlich auf traditionelle Punkte widerspiegeln werden. Eine balancierte Erstattung, in der durch die Befragung nach positiven und negativen Aspekten gefragt wird, ist hierdurch nicht möglich. Das Resultat zeigt somit Punkte auf, die generell als Hindernisse oder kontraproduktiv zur agilen Softwareentwicklung stehen. Ergebnisse oder Erwähnungen, die gut funktionieren und in die Interviews aufgenommen werden, sind somit als besonders und herausstehend zu bewerten.

7.2.1 Ziele

Das Ziel dieser Untersuchung ist es, die Ausprägung der Gespräche in Richtung agilen und traditionellen Unterschieden zu erfassen. Das Ergebnis kann je Interview einen Vergleich zwischen traditionellen und agilen Ausprägungen bereitstellen. Ebenfalls kann das Ergebnis in einer Gesamtauswertung eventuelle Gemeinsamkeiten oder ähnliche Ausprägungen einzelner Attribute repräsentieren.

7.2.2 Aufbau

In den folgenden Abschnitten wird der Aufbau der Neuauswertung beschrieben. Hierzu wird zunächst die Methode fokussiert, nach der vorgegangen wurde. Anschließend wird die Verarbei-

tung der Interviews näher erläutert, die für die Gewinnung der Ergebnisse der Einzelinterviews und der Erlangung der Erkenntnisse zur Gesamtauswertung verwendet wurde.

Methode

Um eine weitere Verarbeitung der Audiospuren durchzuführen, wurde sich einer Technik aus den Sozialwissenschaften bedient. Dabei handelt es sich um Verfahren der qualitativen Inhaltsanalyse, die als eine Standardmethode der Textanalyse bezeichnet wird. Dabei wird das zu analysierende Material in ein Kommunikationsmodell eingegliedert. Das Ziel ist die Schlussfolgerung auf Teile eines Modells. Realisiert wird dies innerhalb dieser Auswertung über ein Kategoriensystem, welches die Auswertungspunkte darstellt. Auf diese wird dann im Ziel dieser Analyse geschlussfolgert. Zudem werden Kodierregeln entworfen, die beim Identifizieren von Textstellen behilflich sein sollen. Ein Ankerbeispiel kann dabei die Analyse zusätzlich unterstützen. Neben der Untersuchung von Transkripten können zur Verarbeitung auch Tonaufnahmen ohne Transkription verwendet werden. [38]

Durch die Vielzahl von unterschiedlichen Kategorien, die sich im Kontext dieser Untersuchung ergeben, ist eine Ergebniserfassung notwendig gewesen, die einen Vergleich aller Ausprägungen zulässt. Hierzu eignen sich für eine grafische Darstellung vor allem Netzdiagramme. Jedes Kriterium kann so auf einer Linie, ausgehend vom Mittelpunkt, nach einander repräsentiert werden. Bei einer Auswertung werden die entsprechenden Punkte gemäß der Wertung, z. B. Häufigkeit vom Mittelpunkt aus auf der Linie positioniert. Ist ein Kriterium nicht erfüllt und die Häufigkeit z. B. mit Null zu werten, so verbleibt die Markierung im Mittelpunkt. So wird in einem Kreis laufend jedes Kriterium betrachtet und bewertet. Anschließend werden die gesetzten Punkte von Markierung zu Markierung miteinander verbunden. Bezogen auf die in der vorherigen Methode vorgestellten Kategorien können diese als Kriterium betrachtet werden und so eine Auswertung ermöglichen. [36]

Verarbeitung der Interviews

Die vorgestellte Methode benötigt zur Auswertung sogenannte Kodierregeln. Diese Kodierregeln sind im Anhang E zu finden und wurden bei der Zuordnung auf Kategorien verwendet.

Bei der Auswertung ist bei jedem Interview wie folgt vorgegangen worden. Zunächst wurde eine Tabelle entworfen, die die Spalten „Kodierregel-Nr.“, „Zeit“ und „Zitat“ besitzt. Die Spalte „Kodierregel-Nr.“ dient der Referenzierung auf die entsprechende Kodierregel. In der letzten Spalte „Zitat“ wird das Zitat erfasst, welches zur Darstellung der Kodierregel dient. Hierbei

können auch mehrere Kodierregeln zu einem Eintrag verwendet werden. Ganze Sätze sowie Teilsätze mit Unterbrechung können zusätzlich aufgenommen werden. Sensible Informationen, die z. B. auf das Unternehmen schließen, werden ausgelassen. Das Zitat dient im Kontext des Gesprächs und der entsprechenden Situation zur Zuordnung. Die Nachvollziehbarkeit ist an dieser Stelle zum Teil nur über das Hören der kompletten Tonspur und der Teilnahme am realen Interview verfolgbar. Die „Zeit“ stellt den Zeitpunkt dar, wo in der Tonspur das Zitat beginnt und die Zuordnung auf die Kodierregel vorgenommen wird.

Die Darstellung der Ergebnisse erfolgt neben der beschriebenen Tabelle zusätzlich über eine Grafik. Die Grafik stellt ein Netzdiagramm dar und bietet, wie bereits beschrieben, die Möglichkeit, die Kriterien auszurichten. Die Häufigkeit des Kriteriums wird hierzu je Kriterium vom Mittelpunkt ausgehend erfasst. Dies wird sowohl für die traditionellen sowie die agilen Erwähnungen des Interviews vorgenommen. Die Grafiken der Auswertungen, die in den nachfolgenden Abschnitten dargestellt werden, stellen die jeweiligen traditionellen und agilen Kategorien in einer Abbildung gegenüber. Die Unterscheidung wird dabei über zwei unterschiedliche Farben sichtbar.

Die Ergebnisse spiegeln insgesamt subjektive Wahrnehmungen der durchgeführten Experteninterviews wieder und die subjektive Auswertung über die entworfenen Kodierregeln. Anhand dessen lässt sich somit nicht eindeutig festlegen, wie generell das Unternehmen, Team oder einzelne Wahrnehmungen von Personen zu bewerten sind. Allerdings kann ein relatives Verhältnis der Interviews auch in Bezug auf die Gesamtauswertung zur Ausprägung von agilen und traditionellen Unterschieden dargestellt werden. Ebenso stellen die festgelegten Kodierregeln und das Verfahren zur Präsentation der Ergebnisse eine Möglichkeit dar, wie eine Ist-Analyse unterstützt werden kann.

7.3 Auswertung der Einzelinterviews

Insgesamt wurden fünf Interviews durchgeführt, wobei vier davon als Sprachaufzeichnung aufgenommen wurden. In den nachfolgenden Unterabschnitten sind die Ergebnisse von vier Einzelinterviews dargestellt. Dabei handelt es sich um die Interviews eins und zwei sowie das vierte und fünfte Interview. Das dritte Interview fehlt, da keine Sprachaufzeichnung existiert. In der Auswertung wird nicht im Detail auf die Rahmeninformationen, wie Ort der Durchführung, Start des Interviews, Dauer des Interviews, etc. eingegangen. Diese Informationen können bei Bedarf den Kurzberichten im Anhang entnommen werden. Die Referenz zum Anhang ist in der nachfolgenden Auswertung jedes Interviews dokumentiert.

Jedes der nachfolgenden Interviews ist nach einem bestimmten Schema aufgebaut. Als erstes wird eine Kurzbeschreibung zum Unternehmen dargelegt, in der z. B. auf den Kontext der Entwicklungsabteilung zum Unternehmen eingegangen wird. Danach wird näher auf den Probanden der Untersuchung eingegangen. Dabei wird die Rolle sowie die Ausrichtung der Person kurz beschrieben. Der dritte Absatz stellt den Verweis auf die folgende Abbildung dar, in der die Grafik kurz bzgl. der Kurven beschrieben wird. Wie bereits erwähnt, folgt danach die Grafik selbst. Anschließend wird die Kurve der traditionellen Ausrichtung beschrieben, dabei wird absteigend der Stärke einer Ausrichtung vorgegangen. Hierzu wird eine Interpretation der traditionellen Ausrichtung dargelegt. Erneut zu erwähnen ist, dass vor allem Probleme und Herausforderungen innerhalb aller Experteninterviews angesprochen wurden. Hiernach lässt sich, bezogen auf die Ergebnisse, interpretieren, dass vor allem die dargestellten Kriterien die Agilität mit beeinflussen. Im Anschluss folgt die Beschreibung der Grafik im Fokus der agilen Ausrichtung mit einer nachfolgenden Interpretation der Ausrichtung. Auch hier wird bei der Beschreibung der Grafik absteigend von der Häufigkeit bzgl. der Ausrichtung vorgegangen. Im letzten Auswertungsteil der Einzelinterviews wird auf den Vergleich zwischen traditioneller und agiler Ausrichtung eingegangen.

7.3.1 Interview 1

Das erste Interview ist dem Projektgeschäft zuzuordnen. Dabei handelt es sich im Detail um ein Unternehmen, welches verschiedenste Softwareentwicklungsprojekte für Kunden durchführt. Dies beinhaltet die Entwicklung an sich sowie die Weiterentwicklung und Wartung von Softwareprojekten. Das Unternehmen kann als „junges“ Unternehmen eingestuft werden. Ebenso stellt es sich als Unternehmen mit „StartUp“-Mentalität in Hamburg dar. Weitere Details finden sich im Anhang **F** wieder.

Zur intervieweten Person ist zu ergänzen, dass es sich um einen Scrum Master handelt der ebenfalls als Entwickler tätig ist. Im Experteninterview zeigte sich dadurch auch, dass es zum Teil Diskussionen mit einem technischen Fokus gab.

Die nachfolgende Grafik **7.1** stellt die Auswertung des Interviews mit den entwickelten Kodierregeln dar. Wie bereits im Abschnitt **7.2.2** beschrieben, werden diese Kodierregeln in der Grafik gegenübergestellt. Die blaue Kurve stellt die agile und die orange Kurve die traditionelle Ausrichtung des Gesprächs dar.

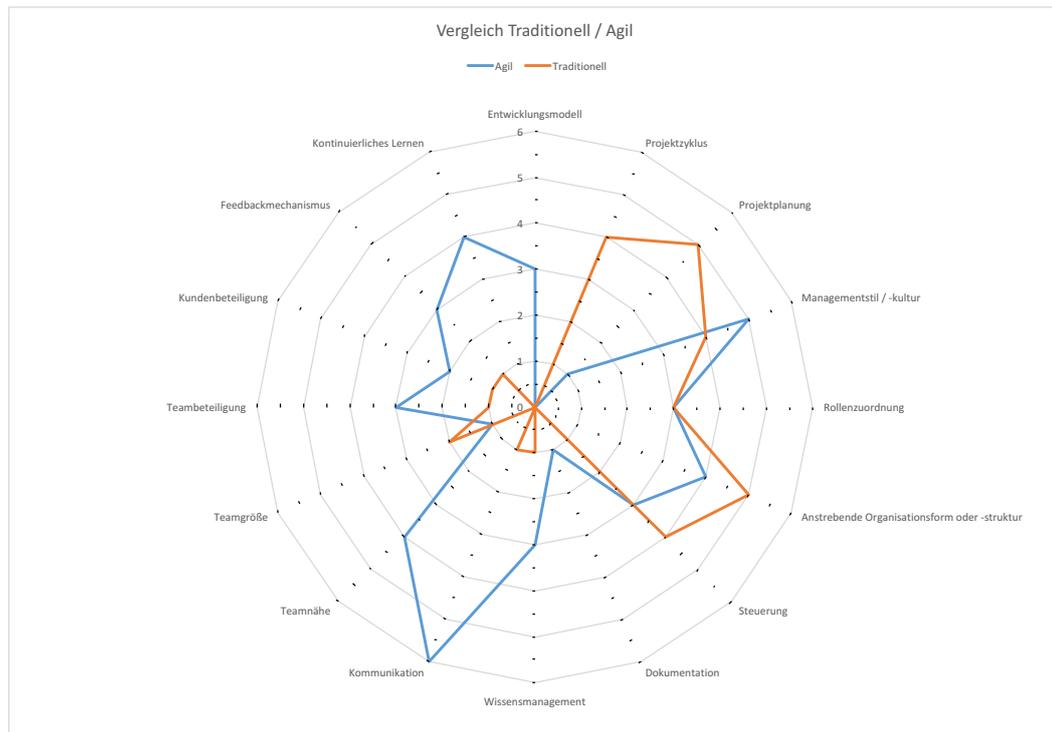


Abbildung 7.1: Vergleich der Indizien - Interview 1

In Bezug auf die traditionelle Ausrichtung ist zu erkennen, dass insbesondere der Punkt „Projektplanung“ sowie die „anstrebende Organisation“ hervorstechen. Danach folgen die Punkte „Steuerung“, „Managementstil / -kultur“ und „Projektzyklus“.

Die Betrachtung der traditionellen Konzentration bezieht sich vor allem auf die Schnittstellen, die das Unternehmen dem Entwicklungsteam bietet. Dabei stellt diese Schnittstelle insbesondere traditionelle Züge im Prozess dar. Dies scheint sich auch auf die Projektabwicklung auszuwirken, ebenso wie die Rollenzuordnung. Auffällig ist hierbei die Projektabwicklung, da das Unternehmen sich im Projektgeschäft befindet.

Betrachtet man die Gewichtung der Kriterien zur Agilität, so fallen die Punkte der „Kommunikation“, „Managementstil / -kultur“, „anstrebende Organisation“, „Teamnähe“ und „Kontinuierliches Lernen“ besonders auf. In der nächsten Abstufung finden sich „Steuerung“, „Wissensmanagement“, „Teambeteiligung“, „Feedbackmechanismus“, „Entwicklungsmodell“ und „Rollenzuordnung“ wieder.

Im Fokus der Agilität werden besonders die Stärken der Kommunikation und des Managements sichtbar. Insgesamt sorgt die Anzahl an Kriterien mit einer gleichen oder fast gleichen Anhäufung dafür, dass ein deutlich runderes Bild dargestellt wird als bei den traditionellen Punkten. Interessant ist auch, wie bezüglich des Managements, der Organisation und der Steuerung der Unterschied zur traditionellen Ausrichtung gering ist. Die Rollenzuordnung wird genau gedeckt.

Betrachtet man die extremen Unterschiede zwischen den agilen und traditionellen Kriterien, so fällt insbesondere auf, dass die Projektdurchführung tendenziell traditionell gehalten wird. Agil betrachtet werden allerdings deutlich mehr Punkte, wie die Kommunikation, im Vordergrund präsentiert. Von daher stellt sich trotz des Fokus auf Impediments im Gespräch ein eher agiles Verhalten als traditionelles Verhalten im Bezug auf das Vorgehensschema dar.

7.3.2 Interview 2

Bei dem zweiten durchgeführten Experteninterview steht eine interne Softwareentwicklung im Fokus der Entwicklungsabteilung. Dabei werden verschiedenste interne Projekte und Produkte entwickelt und weiterentwickelt. Das Unternehmen selbst ist als traditionelles Unternehmen in Hamburg zu betrachten und ist der Versicherungsbranche zuzuordnen. Weitere Details finden sich im Anhang G wieder.

Die interviewte Person besetzt die Rolle des Scrum Masters und ist ebenfalls als Entwickler tätig. Genau wie beim ersten Interview sind technische Diskussionen entstanden.

In der nachfolgenden Abbildung 7.2 ist die Auswertung über die Kodierregeln dargestellt. Wie in dem vorherigen Experteninterview zeigt die orange Kurve die traditionellen Ausrichtungen und die blauen, die agilen.

In der Grafik ist besonders die starke Ausrichtung von „Managementstil / -kultur“ sowie der „Rollenzuordnung“ zu erkennen. Danach folgen sowohl die „angestrebte Organisationsform oder -struktur“ und die „Kommunikation“. Auffällig sind die Punkte „Teambeteiligung“ und abtufend die „Teamnähe“ und „Steuerung“.

Auch in diesem Interview ist eine starke traditionelle Ausrichtung zu erkennen, die auf Schnittstellenproblematiken zwischen der Entwicklungsabteilung mit dem Management schließen lässt. Durch die verschiedenen Ausrichtungen lässt sich deuten, dass dieser Stand noch über einen längeren Zeitraum bestehen bleibt. Zu lesen ist dies in Kombination mit der „angestrebten Organisationsform oder -struktur“ sowie „Rollenzuordnung“, „Kommunikation“ und der „Teambeteiligung“, die essentielle Unterschiede zur agilen Ausprägung darstellen.

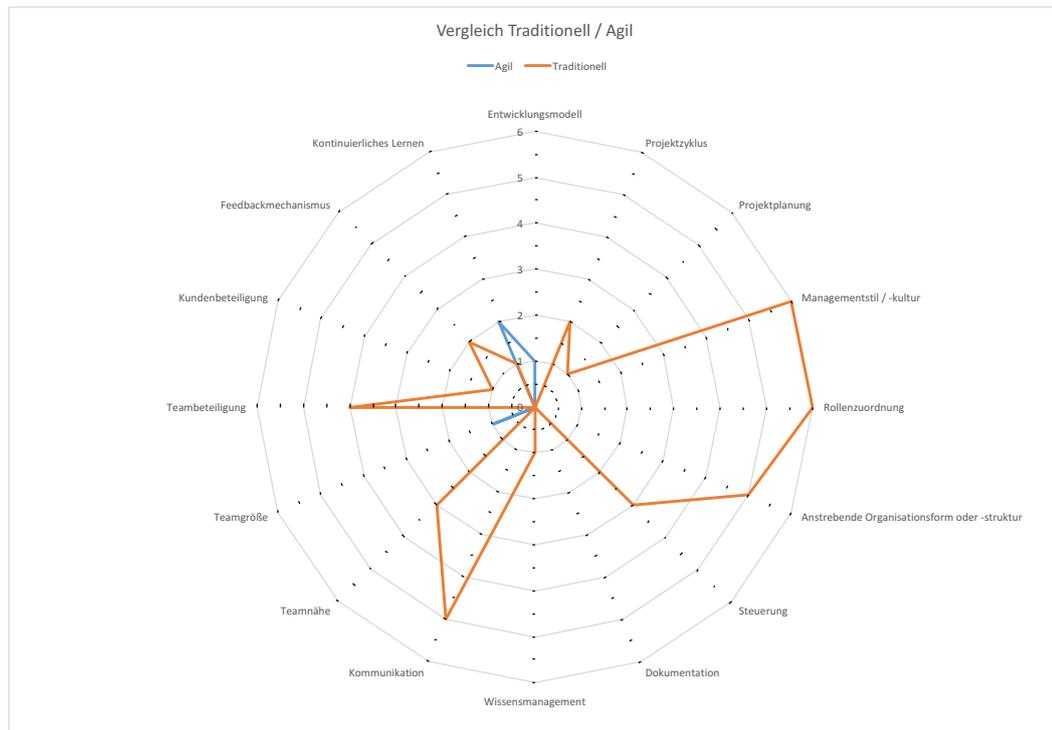


Abbildung 7.2: Vergleich der Indizien - Interview 2

Im Kontext der agilen Ausprägung fallen nur minimale Ausrichtungen auf. Hierzu zählen „Kontinuierliches Lernen“ sowie das dargestellte „Entwicklungsmodell“ und die „Teamgröße“. In der Interpretation der agilen Faktoren kann keine eindeutige Aussage getätigt werden. Es kann lediglich bestätigt werden, dass es Ansätze gibt, die keine starken Ausprägungen besitzen. Vielmehr wird angedeutet, dass eine agile Richtung mittels eines Modells angegangen wird. Im Vergleich der traditionellen und der agilen Ausrichtung fällt vor allem auf, dass die agilen Aspekte nicht mit den traditionellen vergleichbar sind. Diese sind so minimal dargestellt, dass die traditionellen um ein Vielfaches überwiegen. Es gibt somit eine ganz klare Ausrichtung der Entwicklungsumgebung in Richtung traditioneller Softwareentwicklung.

7.3.3 Interview 4

Das dritte, wie auch das vierte Interview wurden in einem Hamburger Unternehmen durchgeführt, welches als traditionelles Unternehmen, angesehen werden kann. Zudem ist dieses

Unternehmen dem Online- und Versandhandel zuzuordnen. Dieses stellt eine Plattform zur Verfügung, welches die Hauptschnittstelle zum Kunden über einen Webshop bereitstellt. Nach der bereits gegebenen Beschreibung, ist die Softwareentwicklung des Interviewpartners der internen Entwicklung zuzuordnen. Hierbei wird intern ein Produkt kontinuierlich weiterentwickelt. Wie bereits erwähnt, wird hier das vierte Interview ausgewertet und nicht das dritte. Weitere Details finden sich im Anhang H wieder.

In diesem Interview war, im Unterschied zu den vorherigen Gesprächen, keine Person mit einem technischen Hintergrund anwesend. In der Rolle ist die Tätigkeit aber auch der eines Scrum Masters zuzuordnen. Der Fokus des Gesprächs verlief verstärkt in Richtung agiler Prozesse und Praktiken sowie deren Hindernissen und Probleme in dem Unternehmen.

Die folgende Abbildung 7.3 stellt die Ergebnisse dar, wie sie auch in den vorherigen Interviewauswertungen verwendet wurde. Ebenfalls wird eine blaue Kurve für die agile Ausrichtung verwendet und eine orange Kurve für die traditionelle Kategorisierung.

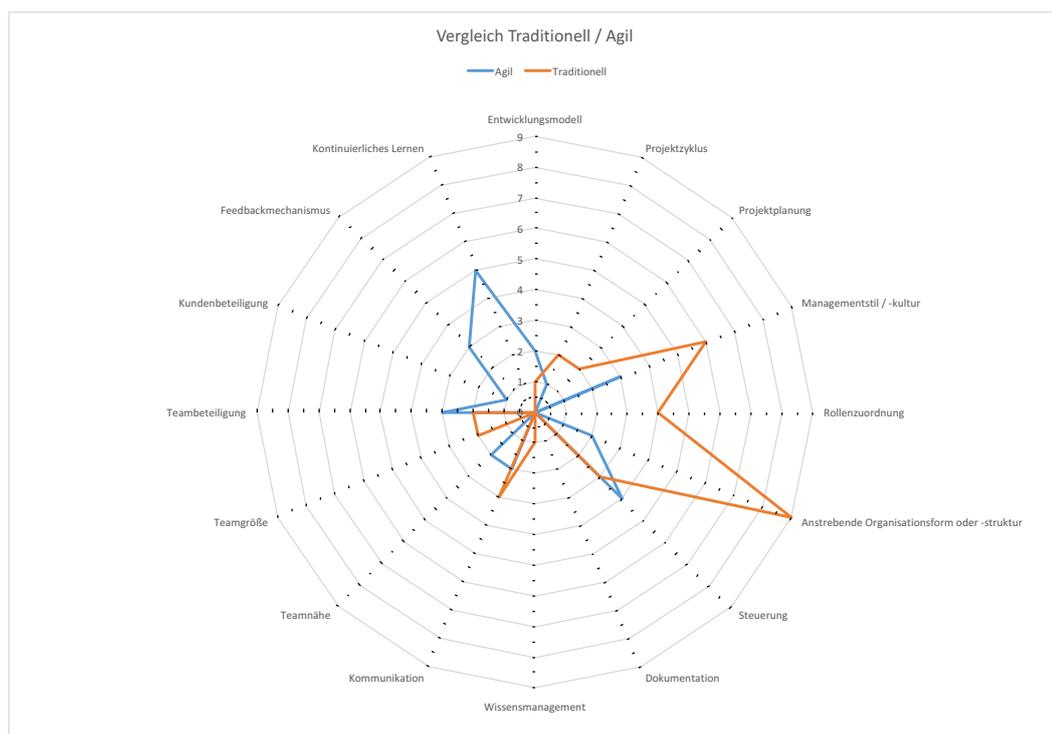


Abbildung 7.3: Vergleich der Indizien - Interview 4

Die traditionelle Ausrichtung zielt vor allem auf „anstrebende Organisationsform oder -struktur“ ab. Danach folgen in abgestufter Reihenfolge „Managementstil / -kultur“, „Rollenzuordnung“ sowie „Steuerung“ und „Kommunikation“.

In der Interpretation dieser Ausprägung fällt vor allem auf, dass die Organisation enorme Auswirkungen auf die traditionelle Ausrichtung besitzt. Hinzu kommt das Management, welches sich in Kombination zu den Vorgaben auf den Punkt „Rollenzuordnung“ auswirkt. Interessant ist allerdings die Häufigkeit der nachfolgenden traditionellen Aspekte. Zwar ist hier die „Steuerung“ und die „Kommunikation“ in der Bewertung den häufigsten Ausprägungen zuzuordnen, diese sind allerdings vergleichsweise gering darzustellen, da die zugehörigen agilen Werte diese abschwächen.

Betrachtet man die agilen Faktoren, so stellt „Kontinuierliches Lernen“ den stärksten Punkt dar. Danach kommt die „Steuerung“ und darauf folgend die „Teambeteiligung“, „Feedbackmechanismus“ wie auch „Managementstil / -kultur“.

Die agile Ausrichtung wird im Interview besonders durch agile Funktionalitäten, wie das Lernen oder die Feedbackmechanismen bestimmt. Zudem fallen vereinzelt weitere agile Ausprägungen auf.

Betrachtet man die agile und traditionelle Fokussierung, so stellt, wie bei der traditionellen Interpretation erwähnt, die traditionelle Ausrichtung eine deutliche Ausprägung dar. Allerdings existieren bereits starke agile Aspekte. Besonders interessant ist, dass zum Teil die traditionellen Punkte durch die gleichen agilen Punkte angegangen werden. So zeigt sich in der „Steuerung“, dass die agile Fokussierung hier stärker ist. Im Gesamtbild ist die traditionelle Zuordnung zwar stärker als die agile, diese ist allerdings nicht allzu weit entfernt und nimmt, wie eben beschrieben, bereits Punkte in Angriff.

7.3.4 Interview 5

Im letzten durchgeführten Interview handelt es sich ebenfalls um ein Unternehmen mit Sitz in Hamburg, welches den Onlinemedien angehört. Die verfolgte Strategie der Softwareentwicklung ist dem Produktgeschäft zuzuordnen. Es wird eine Plattform bereitgestellt, die über Mobile-Apps und weitere Geschäftsinhalte ausgebaut wird. Aus Sicht des Kunden steht diesem nur eine Plattform gegenüber. Weitere Details finden sich im Anhang I wieder.

Wie im Interview 4 ist auch innerhalb dieses Interviews eine Person ohne technischen Fokus als Proband vernommen worden. In ihrer Rolle stellte er den Scrum Master für mehrere Teams

dar. Auch hier ist der Fokus von agilen Prozessen, Methodiken sowie Praktiken aufgefallen, in Bezug auf die Anwendbarkeit im Unternehmen.

Auch in diesem Ergebnisbericht stellt die nachfolgende Abbildung 7.4 die Auswertung über Kategorien dar. Blau steht für die agile Ausrichtung und orange für die traditionelle Kategorisierung.

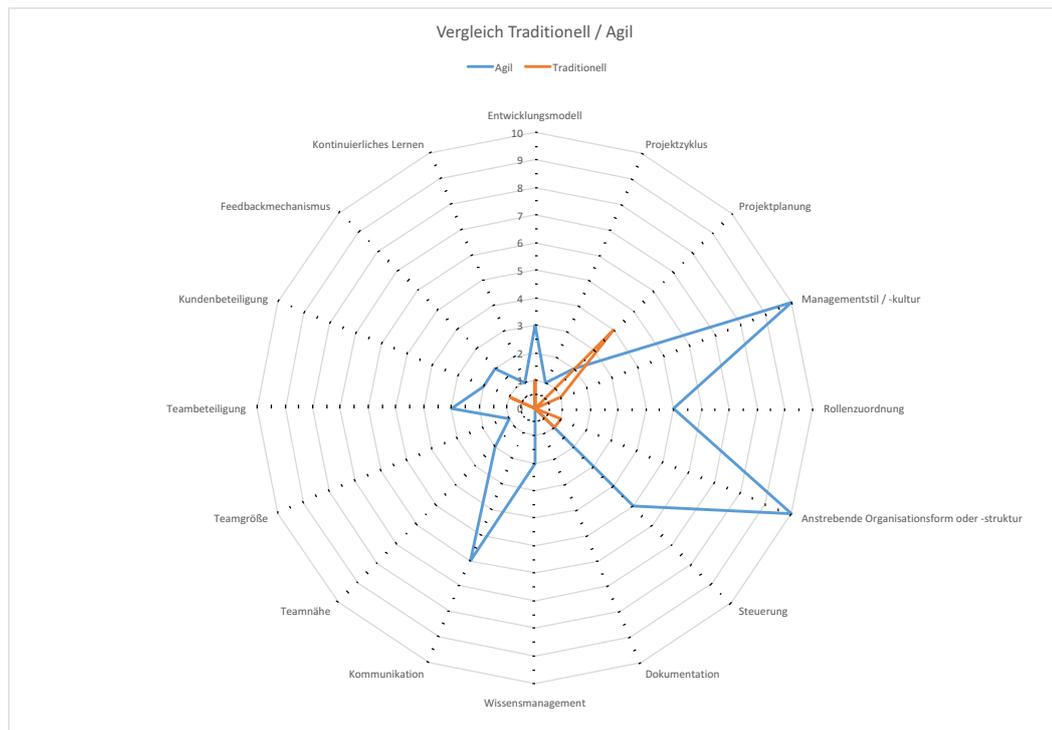


Abbildung 7.4: Vergleich der Indizien - Interview 5

In der traditionellen Betrachtung fällt vor allem der Punkt „Projektplanung“ ins Gewicht. Alle weiteren Punkte sind so minimal erfasst worden, sodass sie nicht weiter benannt werden.

Auch die Interpretation der traditionellen Ausrichtung lässt nur die Erkenntnis zu, dass eine fast nicht zu erwähnende traditionelle Projektplanung vorhanden ist. In Kombination mit dem Gespräch ist zu ergänzen, dass eine traditionelle Projektplanung vor allem als eine höher gelagerte Instanz zu sehen ist, die die Skalierung von mehreren Teams innerhalb eines Projekts ermöglicht. Die restlichen Punkte können, wie im vorherigen Absatz erwähnt, ignoriert werden.

Die agile Ausprägung stellt mehrere größere Punkte dar. Hierzu gehören vor allem die „anstrebende Organisationsform oder -struktur“ sowie „Managementstil / -kultur“. Darauf folgen die „Kommunikation“ und danach die „Steuerung“ sowie die „Rollenzuordnung“.

Die starke Ausrichtung zu Management- und Organisationsthemen lässt sich auch auf den eher nicht technischen Hintergrund der befragten Person zurückführen. Dies wird verstärkt durch die Sicht des agilen Projektmanagements. Zudem stellt auch die Kommunikation innerhalb des Interviews einen wichtigen Faktor dar, bei dem die Ausdrucksweise einer Formulierung durch agile Aspekte Auswirkungen auf die Bewertung hat. Hierzu sind viele weitere agile Punkte vorhanden, die ein stimmiges, agiles Bild abgeben.

Im Vergleich der agilen und traditionellen Ausprägung kann von einem sehr starken agilen Unternehmen gesprochen werden. Es ist ein hohes Maß an agiler Erfahrung und Wissen vorhanden. Nur minimale, traditionelle Spuren sind erkennbar gewesen. Vor allem die Schnittstellenproblematik zwischen dem Unternehmen und der Entwicklung, die häufig anzutreffen ist, wird innerhalb dieses Interviews sehr agil vorgestellt.

7.4 Gesamtauswertung

Inhalt dieses Kapitels ist die Darstellung der Interviewergebnisse im Vergleich zueinander. Hierzu wird zunächst das Verfahren vorgestellt, worauf die Auswertung aufbaut. Im Anschluss folgt die Auswertung aller Interviews.

7.4.1 Auswertungsverfahren

Das Auswertungsverfahren beschreibt das Verfahren, wonach eine Rangfolge der Interviews bestimmt werden kann. Dabei wird zunächst je Vergleich ein Ranking aufgebaut. Verwendet wird hierzu die Häufigkeit je Kategorie. Zunächst wird die Summe der Häufigkeit je Kategorie festgestellt. Danach wird definiert, nach welcher Häufigkeit ein Einbeziehen der Werte stattfindet. Dabei ist die Anzahl an Interviews zu berücksichtigen, die in der Summe der Häufigkeiten auftritt. Im nächsten Schritt findet ein Ranking je Kategorie statt. Dabei werden alle Interviews berücksichtigt. Ist die Häufigkeit einer Kategorie über mehrere Interviews identisch, so teilen sich die Interviews einen Rang. Dies wirkt sich wiederum auf die nachfolgenden Ränge aus. Sind z. B. zwei Interviews auf Platz eins, so ist der Platz zwei nicht belegt und danach folgt erst wieder der dritte Platz. Ist keine Häufigkeit der Kategorie in einem Interview vorhanden, so wird diese immer an den letzten Platz gestellt. Gibt es mehrere Interviews mit einer Häufigkeit

Tabelle 7.1: Ranking Matrix

	Platz 1	Platz 2	Platz 3	Platz 4
Interview 1	2			
Interview 2				
Interview 4				
Interview 5				

von Null, so wird das vorherige Platzierungsverfahren angewendet. Das bedeutet, dass Platz vier geteilt wird und Platz drei nicht belegt wird.

Sind je Kategorie die Rankings konstruiert, so kann eine neue Matrix erstellt werden. Dabei wird jedem Interview die Summe auf die jeweiligen Plätze eins bis vier eingetragen. Die Tabelle 7.1 stellt diese Matrix beispielhaft dar. Hat im konkreten Fall das erste Interview nur in zwei Kategorien den Platz eins gehabt, so wird für „Platz 1“ „Interview 1“ eine „2“ eingetragen.

Ist die Tabelle vollständig ausgefüllt, so sollte je Zeile danach geprüft werden, ob die Summe insgesamt identisch zu den anderen Zeilen ist und ob die Summe der Summe an ausgewerteten Kategorien entspricht. Ist dies der Fall, so kann zum nächsten Schritt übergegangen werden.

In diesem Schritt wird jedem Platz ein Wert zugeordnet. Dabei trägt der „Platz 4“ den Wert 1, der „Platz 3“ den Wert 2, der „Platz 2“ den Wert 4 und „Platz 1“ den Wert 8. Danach wird für jedes Interview der entsprechende Wert pro Platz mit der Anzahl der belegten Plätze multipliziert. Diese werden danach summiert und einem Interview zugeordnet. Hiernach entsteht ein Ranking bzgl. traditioneller bzw. agiler Ausrichtung. Jedes Interview kann somit in Relation zu den anderen durchgeführten Interviews eher traditioneller bzw. agiler sein. Diese Auswertungsmethodik beruht darauf, dass keine Gewichtung der Kategorien vorgenommen wurde. Nachfolgend wurde diese Methodik zunächst für die traditionelle Ausrichtung und danach für die agile Ausrichtung verwendet.

7.4.2 Auswertung

Die Auswertung ist wie folgt aufgebaut. Zunächst wird die traditionelle Ausrichtungen betrachtet. Danach folgt eine nähere Beleuchtung der agilen Aspekte. Dabei wird das im Vorgang vorgestellte Verfahren beim Ranking genutzt.

Die Abbildung 7.5 beschreibt hierzu den Vergleich aller durchgeführten Neuuntersuchungen, basierend auf den Experteninterviews im Fokus der traditionellen Ausrichtung.

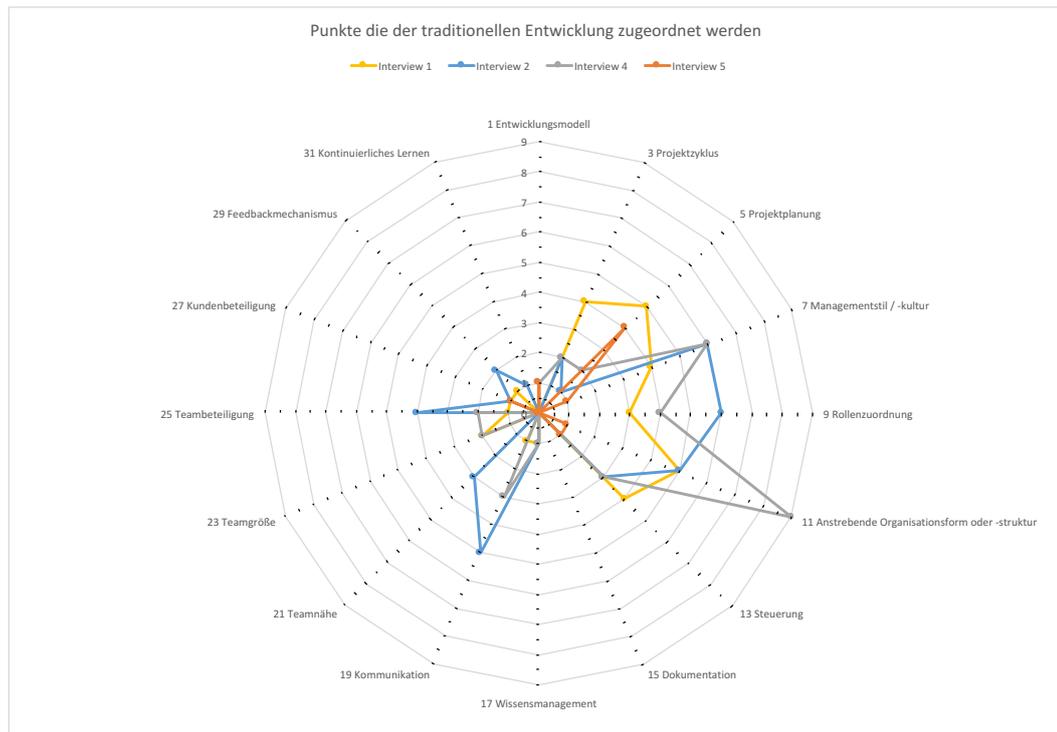


Abbildung 7.5: Vergleich der traditionellen Indizien

Im Fokus der Auswertung wurde der Grenzwert der verwendeten Methodik auf „Sieben“ gesetzt, sodass alles mit einer geringeren Summe nicht in die Bewertung eingeflossen ist. Hierdurch wurden die Aspekte „Entwicklungsmodell“, „Dokumentation“, „Wissensmanagement“, „Teamnähe“, „Teamgröße“, „Kundenbeteiligung“, „Feedbackmechanismus“ und „Kontinuierliches Lernen“ ausgeschlossen.

Im Ergebnis des Verfahrens sieht die Reihenfolge bzgl. der traditionellen Aspekte wie folgt aus:

- Interview 2
- Interview 4
- Interview 1
- Interview 5

Somit zeigen das Interview 2 sowie das Interview 4 stärkere traditionelle Ausrichtungen als das Interview 1 und 5. Zu erwähnen ist hierbei auch, dass die Unternehmen hinter Interview 2 und 4 als deutlich hierarchischer eingestuft werden können. Insbesondere haben bei der

Bewertung die Ausrichtungen „Management -Stil / - Kultur“, „Rollenzuordnung“, „Anstrebende Organisationsform oder -struktur“, „Kommunikation“ und die „Teambeteiligung“ eine Rolle gespielt.

Die Abbildung 7.6 beschreibt den Vergleich aller durchgeführten Neuuntersuchungen, basierend auf den Experteninterviews im Fokus der agilen Ausrichtung.

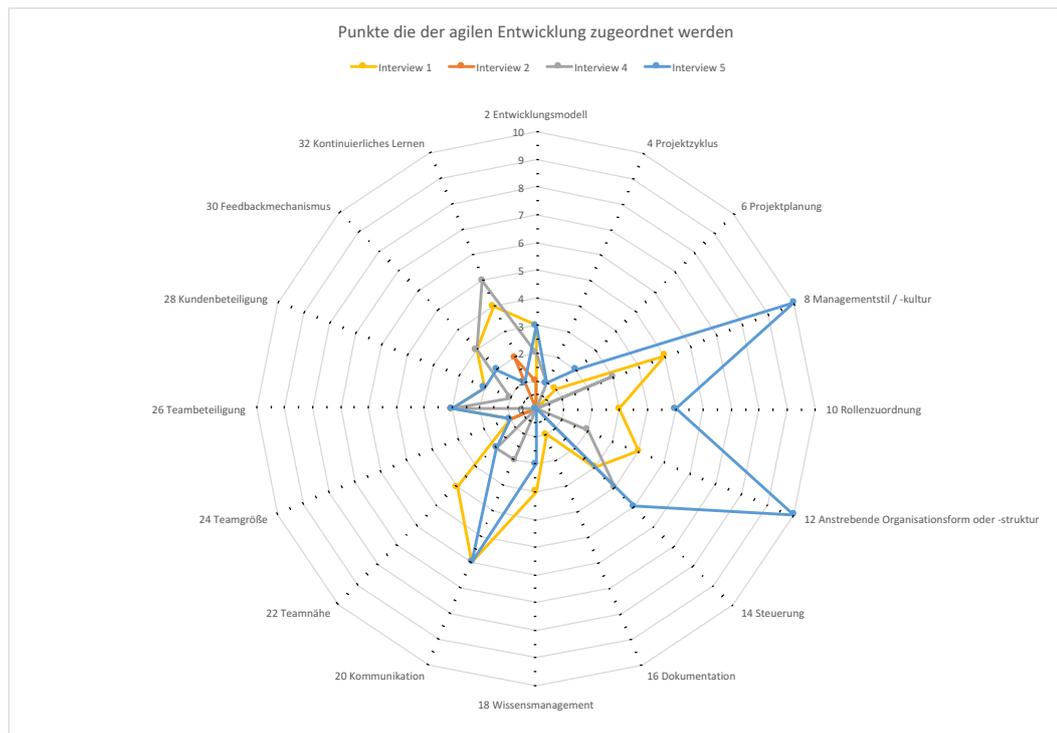


Abbildung 7.6: Vergleich der agilen Indizien

Im Fokus der Auswertung wurde der Grenzwert der verwendeten Methodik auf „Acht“ gesetzt, sodass alles mit einer geringeren Summe nicht in die Bewertung eingeflossen ist. Hierdurch wurden die Aspekte „Projektzyklus“, „Projektplanung“, „Dokumentation“, „Wissensmanagement“, „Teamgröße“, „Teambeteiligung“ und „Kundebeteiligung“ ausgeschlossen.

Im Ergebnis des Verfahrens sieht die Reihenfolge bzgl. der traditionellen Aspekte wie folgt aus:

- Interview 5
- Interview 1
- Interview 4
- Interview 2

Im Fokus der agilen Ausrichtung zeigt sich hierbei, dass das Interview 5 und 1 im Vergleich als eher agil betrachtet werden können, als das Interview 4 und 2. Zu dieser Reihenfolge haben dabei insbesondere die Kategorien „Management-Stil /-Kultur“, „anstrebende Organisationsform oder -struktur“, „Kommunikation“ und die „Rollenzuordnung“ gesorgt.

Betrachtet man die beiden Ranglisten, so entspricht die umgekehrte Reihenfolge der einen Liste, die Rangliste der anderen. Besonders die als eher hierarchisch in ihrer Unternehmensstruktur zu beschreibenden Firmen, sind im Vergleich als generell traditioneller einzustufen.

In der folgenden Abbildung 7.7 ist ein Vergleich der Summe aller traditionellen und agilen Punkte dargestellt. Zu betonen ist, dass durch besonders starke Ausrichtungen einzelner Komponenten über einzelne Interviews in dieser Grafik, entsprechend dominante Sichtweisen entstehen können.

Besonders auffällig sind in Bezug auf die traditionelle Ausrichtung die Punkte „Projektzyklus“, „Projektplanung“, „Rollenzuordnung“ und „Anstrebende Organisationsform oder -struktur“. Bei der Betrachtung dieser Ergebnisse wird ersichtlich, dass die meisten Punkte stark ausgerichtet sind, die auf das traditionelle Management zurückzuführen sind. Zwar sind auch Komponenten weniger stark ausgerichtet, die dieser Kategorisierung ebenfalls zuzuordnen sind, doch spiegelt das Bild insgesamt die klassischen Aufgaben des Managements außerhalb der Entwicklungsteams wieder.

In Bezug auf die agile Ausrichtung sind die folgenden Punkte „Kommunikation“, „Teamnähe“, „Teambeteiligung“, „Kundenbeteiligung“, „Feedbackmechanismus“, „Kontinuierliches Lernen“ und „Entwicklungsmodell“ auffällig. In der Interpretation dieser Ergebnisse lassen sich besonders Ausprägungen aufzeigen, die meist als Vorteile und „typische“ agile Eigenschaften angesehen werden. Auffällig ist in dieser Sicht die geringe Ausrichtung der Teamgröße in Bezug auf die Agilität.

Im Vergleich der traditionellen und agilen Ausrichtungen ist erkennbar, dass versucht wird, „Managementstil / -kultur“, „Rollenzuordnung“, „Anstrebende Organisationsform oder -struktur“ sowie „Steuerung“, agiler auszurichten. Zusätzlich fallen zur traditionellen Sichtweise „Pro-



Abbildung 7.7: Vergleich der Summe aller agil und traditionell ausgerichteten Punkte

jektzyklus“ und „Projektplanung“ auf. Hier kann interpretiert werden, dass bisher keine agilen Ansätze zur Verfügung stehen. Ebenso ist daraus ableitbar, dass das agile Ansätze nicht benötigt werden. Zu ergänzen ist dabei, dass die Untersuchung hier fokussiert werden sollte, um diese Interpretation zu stärken.

7.5 Fazit

Im Fazit dieser durchgeführten Untersuchung ist erneut zu berichten, dass der Hauptfokus der Interviews auf der Beleuchtung von Impediments lag. Hierdurch wurden generelle Probleme und Hindernisse angesprochen, die in Hinsicht der Neuuntersuchung die Sammlung von traditionellen Aspekten verstärkt. Trotzdem sind dabei viele agile Kategorien angesprochen worden. Die subjektiv zu bewertende Erfassung hat in der Gesamtauswertung der Interviews gezeigt, dass ein Vergleich möglich ist. Dazu beigetragen hat auch die Erhebung über die Rolle des Scrum Masters, welcher durch seine Definition, einen Blick auf den Prozess besitzt.

Besonders interessant sind in Bezug hierzu die Ausrichtungen der Aufgaben, die im klassischen Sinne dem Management zuzuordnen sind. Diese haben in der traditionellen Ausprägung den größten Anteil ausgemacht. Auch war dabei erkennbar, dass bereits Probanden agile Ansätze hierzu besitzen. Auffällig sind hierbei der Projektzyklus sowie die Projektplanung, die fast ausschließlich traditionell ausgerichtet sind. In Bezug auf diese Erkenntnis ist weiter zu prüfen, auf welche Charakteristika dies zurückzuführen sind und ob diese zu einem gegebenen Kontext positive Auswirkungen auf agile Teams besitzen.

Die verwendete Art und Weise der Neuuntersuchung hat ein Verfahren dargelegt, welches auch für andere Analysen hilfreich ist. Es ist dabei allerdings zu prüfen, ob alle Kategorien, die in dieser Untersuchung verwendet wurden, nützlich sind. Die Punkte, die auch hier nicht in der Endbewertung verwendet wurden, können dabei hilfreich sein. Auch ist diese Vorauswahl von Kategorien bei der Entwicklung einer weiteren Analyse unterstützend. Ein Beispiel für das Ziel einer Untersuchung wäre dabei die Feststellung eines Ist-Stands und einer nachträglichen oder wiederholenden Prüfungsanalyse. Hierdurch wäre der Stand und die Entwicklung der Agilität feststellbar.

8 Konsequenzen zur Herangehensweise

Die Implementierung von Scrum und agilen Verfahren stellt eine erhebliche Herausforderung dar. Die vorherige Untersuchung hat hierzu eine Auswertung aktueller Implementierungsstände in der Praxis agiler Softwareentwicklung aufgezeigt. Durch diese Untersuchung sowie den vorher dargestellten Herausforderungen der Implementierung und kontinuierlichen Adaption, stellen sich Konsequenzen in der Herangehensweise der Implementierung heraus. Hierzu erläutert dieses Kapitel zunächst eine kurze Zusammenfassung zur Vergangenheit der Softwareentwicklung. Darauf aufbauend wird auf die Identifikation des Ist-Zustandes eingegangen. Dieser stellt bei der Migration sowie der initialen Implementierung eine Voraussetzung dar, um im Kontext des Unternehmens realistische Ziele und nächste Schritte zu verfolgen. Im Anschluss daran wird auf das Setzen von Zielen eingegangen. Inhalt dieses Abschnitts ist die Darstellung von Fragen, die sich das Unternehmen und im Detail das einzelne Team stellt, um agile Verfahren zu implementieren. Das abschließende Fazit fasst die Ergebnisse dieses Kapitels nochmals zusammen.

8.1 Vergangenheit der Softwareentwicklung

Die Vergangenheit der Softwareentwicklung stellt für viele heutzutage noch die Verfolgung traditioneller Entwicklungsmethodiken dar. Dieser Aspekt wird sich in Zukunft weiterhin dahin entwickeln, dass sich immer mehr Unternehmen agilen Methoden annehmen [56, 57]. Besonders in der Betrachtung eines traditionellen Entwicklungsverfahrens als vorher verwendeter Entwicklungsprozess, stellt die Implementierung von Scrum oder ähnlichen Vorgehensmodellen eine enorme Herausforderung für die Unternehmen dar. Diese Thematik wurde hierzu in den Abschnitten 5.2 und 5.3 näher beschrieben.

Auch wenn sich das Unternehmen, die Abteilung oder ein spezielles Team für einen Wechsel zur agilen Softwareentwicklung entschieden hat, ist eine vollständige Migration des Prozesses hin zu agilen Verfahren meist nicht immer möglich. Je nachdem, ob man sich generell für einen evolutionären Wandel entschieden hat oder weitere Einflüsse dafür sorgen, dass sich

bestimmte Dinge nicht ändern sollen, verharren gewisse Rituale, Rollen oder Artefakte, die nicht der jeweils ausgewählten Methode zuzuordnen sind. Somit stellt die Art der verfolgten Entwicklung einen Faktor dar. Die Identifikation von Hindernissen, der Migration sowie die realistische Umsetzung einzelner Bestandteile ist in dieser Konsequenz eine wichtige Aufgabe. Hierzu wurde im Abschnitt 5.2.3 die agile Potenzialanalyse vorgestellt. [49]

Zu unterscheiden sind dabei insbesondere das Projektgeschäft mit dem Produktgeschäft und einer internen Entwicklung. Betrachtet man das Projektgeschäft, so unterliegen den Projekten zeitlichen Grenzen, nach denen das jeweilige Projekt abzuschließen ist. Das jeweilige Unternehmen setzt sich somit generell auf Methodiken fest, die zeitlich begrenzt sein können. Hierdurch wird im Vergleich eine höhere Flexibilität geboten, die allerdings nicht unbedingt immer erreicht wird.

Bei der Produktentwicklung, wie auch der internen Entwicklung, wo in der Regel ebenfalls interne Produkte entwickelt werden, sieht es anders aus. Zwar können wiederum je nach Produkt und Kontext, Weiterentwicklungen als interne Projekte angesehen werden, im Großen und Ganzen wird allerdings eine konkrete Anzahl von Produkten verfolgt. Das bedeutet, dass sich in der Regel länger mit einem Produkt auseinander gesetzt und dieses auch entsprechend komplex wird. Durch diesen zeitlichen Aspekt kann eine gewisse Starrheit entstehen, die einen Wandel verlangsamt und erschwert. Zu erkennen ist, dass Teile der alten Vorgehensweise mit übernommen werden, die nicht als konform mit agilen Methoden zu bezeichnen sind. [8]

Wie bereits im Kapitel 6 angedeutet, können in einem Unternehmen verschiedenste Rituale, Rollen und Artefakte existieren, die nicht Bestandteil der Softwareentwicklung nach Scrum sind. Dies muss nicht zwingend bedeuten, dass diese im Widerspruch zur Entwicklung stehen. Allerdings ist darauf zu achten, dass die agilen Werte nicht gestört werden. So können Meetings, die das gemeinsame Verständnis verstärken, um z. B. einen besseren Überblick über die verfolgte Softwarearchitektur zu geben, eine positive Auswirkung haben. Negative Aspekte bieten dabei Veranstaltungen, die wiederholend Informationen wiedergeben und eher effektives Arbeiten verhindern. [8, 35, 53]

Im Fokus der Rollen ist häufig festzustellen, dass Rollen aus vorherigen Methoden wiederzufinden sind. Generell ist dies nicht als Problem zu betrachten. Allerdings ist auch hier darauf zu achten, dass keine Verletzung der Agilität entsteht. Dabei ist zu unterscheiden, ob doppelte Rollen angenommen werden oder Rollen im neuen Vorgehen existieren, die übernommen wurden. Sind dem Prozess z. B. Teammitglieder zugeordnet, die die Qualitätssicherung über geeignete Testmechanismen sicherstellen sollen, so muss dies nicht bedeuten, dass diese das agile Vorgehen stört. Sie werden aber in diesem Kontext von Scrum als fester Bestandteil des

Teams verstanden. Auch Projektleiter oder Projektmanager stellen generell kein Problem dar. Häufig ist aber insbesondere bei Projektleitern und Projektmanagern festzustellen, dass diese starke Methodiken verwenden, die den Grad der Agilität negativ beeinflussen. Dies muss nicht zwingend von den Personen selbst ausgehen, sondern kann auch durch unternehmensweite Regularien hervorgerufen werden. Diese Personen stehen dabei einem enormen Druck und der Herausforderung gegenüber, die Rolle z. B. im Rahmen eines agilen Projektmanagements auszuüben, anstatt auf das klassische Projektmanagement zu setzen. Oft kann das Festhalten an alten Rollen dafür sorgen, dass sich generell richtige Rollen falsch verhalten oder sich falsch darstellen. Das bedeutet im konkreten Fall, dass z. B. der Product Owner zu viel Verantwortung erhält und traditionelle Rollen inkludiert, wie die des klassischen Projektleiters, aber auch Scrum Master, die parallel Entwickler sind. Selbst als einzelnes Teammitglied kann es sein, dass Verantwortungen entstehen, die dann auch häufig in Verbindung mit Anweisungsbefugnissen das Gleichgewicht des Teams in Bezug auf das jeweilige agile Vorgehensmodell stören. [15, 16, 35, 53, 60]

Bei der Betrachtung der Artefakte stellt sich dar, dass insbesondere Dokumentationspflichten, die sich das Unternehmen selbst gesetzt hat, weiter verfolgt werden. Die Gründe hierfür können sehr vielfältig sein. Die Begründung liegt unter anderem in äußerlichen Regularien, die sich nicht abschaffen lassen. In der Regel werden diese Dokumentationen allerdings verwendet, da immer noch eine andere Art des Anforderungsmanagements betrieben wird. Hinzu kommt, dass das Wissensmanagement zum Teil über diese Regularien geregelt oder die Berichterstattung entsprechend komplex im Unternehmen gefordert wird. Zusätzlich können neben der Dokumentation auch weitere Artefakte übernommen werden, die sich ebenfalls negativ auswirken. Es existieren allerdings auch Artefakte, die sich nicht im Widerspruch zur Agilität befinden und die sogar eine beschleunigte Wirkung besitzen können. [13, 35, 49, 53]

Es ist bei allen genannten sowie nicht erwähnten Punkten immer darauf zu achten, dass diese zur Agilität des Prozesses beitragen. Dabei muss diese Herausforderungen betrachtet und bewertet werden. Behilflich können dabei Personen sein, die einen hohen Grad an agilem Verständnis besitzen, wie gut ausgebildete, agile Coaches. Diese können dann entsprechend des Kontextes im Unternehmen sowie Team, bewerten und feststellen wie der Ist-Zustand ist, Ziele erarbeiten und Maßnahmen zur Verfolgung zusammen mit dem Team und Unternehmen entwickeln.

8.2 Identifikation des Ist-Zustands

Die Identifikation des Ist-Zustands stellt eine unabdingbare Voraussetzung zur Implementierung von agilen Verfahren dar. Ohne das Identifizieren und Analysieren der Gegebenheiten können Konzepte unter Umständen nicht umgesetzt werden.

Auch bei einer weiteren Adaption der Methodiken hilft es, dass Gegebenheiten analysiert werden, um Problemstellungen und Potenziale aufzudecken. Im Zuge dessen können etablierte Praktiken evaluiert werden. Allerdings unterscheidet sich dabei die Detailstufe der Analyse im Gegensatz zur Analyse einer initialen Implementierung. [13, 49]

Eine detaillierte Analyse kann Problemen der Implementierung und weiteren Adaption aufdecken. Im Hinblick auf die Agilität ist allerdings zu berücksichtigen, dass der Prozess des Ist-Zustands nicht bedeutet, dass eine komplette Analyse besonders bei der Betrachtung von Adaptionsverbesserungen zu fein und detailliert erfolgen sollte. Je nach Größe des Schritts, der getätigt werden soll, kann sich der Zeitraum der Vorabanalyse verlängern. Das bedeutet, dass ggf. kleine geschlossene Analysen deutlich hilfreicher sind, um kleine geschlossene Verbesserungen z. B. für ein Team durchzuführen. [13, 49]

Bei der Betrachtung von wissenschaftlichen Arbeiten können solche Analysen von Unternehmen allerdings auch in ausführlicher Form dafür sorgen, dass das Verständnis über Herausforderungen der Implementierung verbessert wird. Insbesondere können hierdurch aktuelle Herausforderungen ausgiebiger erforscht und in einen Bezug zur Wissenschaft gesetzt werden, um Theorien neu zu überdenken. [13, 49]

Eine Möglichkeit dieser generellen Analysetechniken stellt die agile Potentialanalyse dar, die im Abschnitt 5.2.3 erläutert wurde. In diesem Verfahren wird dabei der Kontext durch das Unternehmen in die Entscheidung zur Implementierung von agilen Praktiken verwendet. Insbesondere können somit angestrebte Ziele auf ihre Realisierungsmöglichkeit geprüft werden. Dieses Verfahren stellt dabei keine allgemeingültige Lösung bereit. Es wird vorausgesetzt, dass die Grundwerte der Agilität und der einzelnen Praktiken verstanden wurden. [13, 49]

In der Phase der Implementierung neuer Verfahren ist durch das eben genannte Verfahren nicht gewährleistet, dass diese erfolgreich verlaufen. Auch in der generellen Betrachtung des Ist-Zustandes setzt das Verfahren voraus, dass Hindernisse und Problemstellungen schon bekannt sind, die sich besonders in Bezug auf eine weitere Adaption bereits im Prozess befinden. Eine Unterstützung dieser Lokalisierungsproblematik stellt die Analyse der bestehenden Impediments dar. Hierzu wurde im Abschnitt 6.4 unter anderem eingegangen. Auch wurde

im Abschnitt 7.1 auf eine Analysemöglichkeit über Impediments in der Praxis der agilen Softwareentwicklung Bezug genommen. [3, 13, 45, 49]

Eine zusätzliche Hilfestellung, um ein besseres Verständnis der Agilität im Unternehmen oder Team zu erhalten, stellt die Analyse dar, die im Kapitel 7 erläutert wurde. Dabei können sowohl traditionelle sowie agile Ausrichtungen der Entwicklungsabteilungen aufgezeigt werden. Diese Untersuchung bietet dabei die Möglichkeit, dass der aktuelle Ist-Zustand aus Sicht der initialen Implementierung darstellbar ist. Hierzu können starke Ausprägungen der agilen Komponenten sichtbar werden. Besonders aus Sicht der weiteren Adaption wird aber durch diese Art der Analyse der Fortschritt durch iterative und vergleichbare Erhebungen erkennbar gemacht und der Erfolg oder Misserfolg einer Änderungen grafisch dargestellt. Die Grundlage dafür stellt hierzu die Methodik zur Erhebung der Daten dar. Die in dieser Arbeit verfasste Untersuchungsvariante auf Grundlage einer Impedimentsanalyse veranschaulicht hierzu nur den Versuch, um ein Auswertungsverfahren zu entwickeln. Auch die Identifikation über geeignete Probanden im Prozess, wie hier über den Scrum Master, sind dabei entscheidend. [13, 45, 49]

Die in dieser Arbeit erwähnten Techniken stellen Varianten dar, um die Ist-Analysen im Schritt der Migration sowie der weiteren Adaptierung zu unterstützen. Hierdurch lässt sich eine verbesserte Sichtbarkeit und Transparenz erzeugen, die letztendlich zusätzlich auch die Awareness der Softwareentwicklung bei der korrekten Verwendung verstärken kann. Die Analyse des Ist-Zustands ist erforderlich, um realistische Ziele setzen zu können. Die Thematik der Ziele soll hierzu im nächsten Abschnitt näher beschrieben werden.

8.3 Zielsetzung

Wie im vorherigen Abschnitt beschrieben stellt die Vorabanalyse eine wichtige Voraussetzung dar, um bestehende Problemstellungen zu lokalisieren und zu verstehen. Dabei stellt insbesondere die Aufgabe des Verständnisses eine entscheidende Rolle, um Lösungsvorschläge zu erarbeiten. In dieser Betrachtung stellt sich allerdings die Frage, welches Ziel verfolgt werden soll. So ist es ebenfalls wichtig, das Ziel zunächst zu definieren.

Die Ziele, die verfolgt werden können sind abhängig vom Unternehmen auf verschiedenen Wegen generierbar. Diese können unter anderem im Prozess der Analyse von agilen oder auch traditionellen Teams entstehen. Ein konkreteres Beispiel ist hierzu, dass aktuelle Herausforderungen und Misstände im Prozess aufgedeckt werden und das Ziel die Beseitigung dieser ist.

Die Erarbeitung von anzustrebenden Zielen kann auch über generelle Beratungen im Fokus des agilen Coachings dargestellt werden. Hierbei werden vor allem Ziele zusammen mit den Teams entworfen und näher festgelegt. Diese Art der Beratung stellt besonders im Sinne des agilen Gedanken eine Möglichkeit dar, in dem keine Zielvorgabe von einzelnen Personen manifestiert wird. Das Vorgehensmodell Scrum bietet hierzu bereits das Ritual der Retrospektive. Im Fokus dessen wird in einem kürzeren Zeitraum versucht, Herausforderungen aufgedeckt und Ziele zu beschreiben. Diese Methode eignet sich besonders, um nicht zu viele Veränderungen in einem Schritt umzusetzen. Des Weiteren steckt das Unternehmen hierbei in der Phase der weiteren Adaptierung des Prozesses, anstatt in der Initialisierung agiler Verfahren. [13, 49, 53]

Um weitere Ziele zu erarbeiten und besonders auch generelle Veränderungen anzustoßen, werden teilweise Workshop-Techniken miteinbezogen. In diesem Fall besteht die Möglichkeit, dass bei einer initialen Einführung agiler Methoden ein Verständnis aufgebaut und zusammen mit dem Team das Ziel näher beschrieben wird. Im Sinne des Verständnisses kann so ein agiles Grundverständnis transportiert und auf die neue Verantwortung eingegangen werden. [13, 49, 53]

Auch bei der Betrachtung von Abhängigkeiten zum Team sind Workshops notwendig, um auf die Thematiken wie Sprintschutz und Teamzusammenstellung einzugehen. In der Beleuchtung dieser genannten Thematiken steht der Fokus vor allem auf der Betrachtung des Prozesses auf einer Metaebene. Hierbei sind Fähigkeiten notwendig, die der Rolle des Scrum Masters zuzuordnen sind und dessen Ergebnisse nur in Kooperation mit dem Team umsetzbar sind. [12, 39]

Im Sinne der entworfenen Ziele wird es somit möglich, zu beschreiben, wo das Team und das Unternehmen sich gemeinsam hin entwickeln wollen. Eine Priorisierung ist hierzu zu empfehlen, um eine Verfolgungsstrategie aufbauen zu können. Dabei dienen die Ziele insbesondere, um anschließend auf die Realisierungsmöglichkeiten im Kontext aller erlangten Erkenntnisse einzugehen. Möglichkeiten bieten dabei z. B. agile Praktiken, die unterstützen können. Hierzu gilt, dass die Ziele weniger flexibel sind als die Vielfältigkeit der Varianten, um diese zu erreichen. [49]

„Der Weg ist flexibel, das Ziel ist es nicht.“ (Eckstein 16, S.27)¹

¹„Be flexible how you get there, but be inflexible in the focus.“ Martin Fowler, auf der International Conference on eXtreme Programming and Agile Processes in Software Engineering 2001, Sardinien, Italien.

8.4 Fazit

Dieses Kapitel hat auf Basis der vorherigen Kapitel dargestellt, dass bei der Implementierung sowie der weiteren Adaption agiler Methoden besondere Faktoren zu berücksichtigen sind. Hierzu gehört unter anderem, dass das tiefere Verständnis zuvor verfolgter Verfahren hilfreich ist. Es muss dabei verstanden werden, warum bestehende Prozesse für identifizierte Problemstellungen ungeeignet sind. Ebenso sind Fehler, die gemacht wurden, auch in den neu zu verfolgenden Entwicklungsmodellen und Praktiken zu beachten.

Zur Identifikation von bestehenden Herausforderungen und Problemstellungen ist somit eine Analyse im angemessenen Maße empfehlenswert. Hierbei können bereits Abhängigkeiten hervorgehoben werden, die dafür sorgen, dass bestimmte Praktiken oder Lösungswege nicht gegangen werden können. Es gilt somit zu verstehen, was Bestand hat, um zu verstehen, was folgen kann.

Bereits durch die Ist-Analyse werden direkte Ziele sichtbar. Zusätzlich stellt die Manifestierung dieser Ziele eine bedeutsame Aufgabe dar. Hierzu ist es auch notwendig, dass das beeinflusste Team, das Unternehmen und ggf. die Stakeholder involviert werden. Im Fokus dessen steht allerdings das Team selbst. Im diesem Sinne sind verschiedene Ebenen zur Erfassung der Ziele zu berücksichtigen, wie die Analyse der Ist-Zustände, das agile Coaching, die Workshop-Techniken und Metaergebnisse des Prozesses, die in eine priorisierte Reihenfolge zu stellen sind.

Dieses Kapitels hat zusammengefasst auf Hinweise aufmerksam gemacht, die in der Betrachtung einer Verfolgung von agilen Methoden wie Scrum hilfreich sind. Auch wird hierdurch deutlich, wie wichtig die Retrospektive in kurzen Abständen ist, um größere Projekte zur Analyse und Umsetzung zu vermeiden.

9 Schluss

Inhalt dieses Kapitels ist die Zusammenfassung sowie der Ausblick dieser Arbeit. Hierzu wird im nächsten Abschnitt jedes Kapitel in Abhängigkeit zueinander dargestellt und kurz rekapituliert. Im darauffolgenden und zugleich letzten Abschnitt dieser Masterarbeit wird ein Ausblick gegeben. Dabei steht der Fokus auf möglichen Anknüpfungspunkten zu dieser Arbeit sowie Fragestellungen, die sich aus dieser ergeben und Inhalt weiterer wissenschaftlicher Betrachtungen werden können.

9.1 Zusammenfassung

Diese Arbeit hat dargelegt, dass trotz der Gründe zum Festhalten an traditionellen Verfahren erhebliche Indizien existieren, dass eine Migration für Unternehmen mit Entwicklungsbezug für die Bewältigung von Herausforderungen der Zukunft gefordert wird. Hierzu wurden zunächst traditionelle Vorgehensmodelle der Softwareentwicklung im Kapitel 2 dargestellt. Dabei wurde auf die Thematik der Softwarekrise im Abschnitt 2.1 eingegangen und anschließend auf typische klassische Phasenmodelle im Abschnitt 2.2, die eine Relevanz zum Verständnis der Entwicklung unterschiedlicher Modelle aufweisen. Abschließend zu diesem Kapitel sind im Abschnitt 2.3 Gründe zum Festhalten am klassischen Vorgehen dargelegt worden.

Das Kapitel 3 diente in Beziehung zum vorherigen Abschnitt zur Darstellung der Motivationen, warum agile Verfahren in der Softwareentwicklung attraktiv sind. Um dies näher zu erläutern wurde der Abschnitt 3.1 herangezogen, indem Veränderungen der IT-Umwelt und des -Marktes erläutert wurden, die einen starken Einfluss nehmen. Zudem stellte der Abschnitt 3.2 die verschiedenen Vorteile dar, die sich durch agile Verfahren ergeben und die aktuellen und voraussichtlichen Anforderungen des Marktes besser abdecken können, als herkömmliche Modelle.

In Anlehnung zur Motivation der Veränderungen führte das Kapitel 4 agile Vorgehensmodelle auf. Dabei wurde zunächst auf das agile Manifest (Abschnitt 4.1) und auf weitere agile Vorgehensmodelle (Abschnitt 4.2) eingegangen. Hierbei wurde der Fokus nicht auf die detaillierte

Beschreibung der eigentlichen Vorgehensmodelle vorgenommen, sondern eher auf die relative Relevanz der Modelle zur Entwicklung und Anwendbarkeit.

Im Kapitel 5 wurde die Thematik der Umwandlung eines traditionellen Entwicklungsmodells in ein agiles Entwicklungsmodell näher betrachtet. Hierzu stellte der Abschnitt 5.1 einleitend die generellen Unterschiede der Vorgehensmodelle, traditionell zu agilen, gegenüber. Des Weiteren stellte der Abschnitt 5.2 die generellen Verfahren zur Migration sowie ein Modell zur kontextabhängigen Transition dar. Im Anschluss daran ist der Abschnitt 5.3 auf zwei Arbeiten eingegangen, die jeweils eine Transition aus der Praxis näher betrachtet haben.

Die Problemstellungen, die sich auch nach einer zunächst erfolgreichen Implementierung ergeben erläuterte das Kapitel 6. Hierzu wurde auf verschiedenste Hindernisse im Prozess eingegangen, die sich zum Teil auf ein mangelndes Verständnis von Scrum beziehen. Zusätzlich wurden generelle Problemstellungen im Kontext des Prozesses dokumentiert.

Um den Punkt der Herausforderung in agilen Prozessen, insbesondere bei der Verfolgung des Scrum Modells, in einen praktischen Bezug zu setzen, wurde eine empirische Untersuchung in dieser Arbeit präsentiert. Das Kapitel 7 stellte zu dieser eine ursprüngliche Untersuchung dar, die im Abschnitt 7.1 den Fokus der Impediment von Scrum Prozessen beleuchtete. Im Anschluss daran ist eine Neuauswertung vorgenommen worden, die im Abschnitt 7.2 erläutert sowie im Abschnitt 7.3 einzeln und im Abschnitt 7.4 im Zusammenhang ausgewertet wurde. Bei der Neuauswertung ist hierzu aufgezeigt worden, inwieweit durch eine Analyse von Herausforderungen auf traditionelle und agile Ausrichtungen von Unternehmen und insbesondere Teams geschlossen werden kann.

Im Anschluss an die Auswertung ist in Relation zur kompletten Arbeit und insbesondere zur durchgeführten, empirischen Erhebung auf Konsequenzen zur Implementierung sowie Migration agiler Verfahren im Kapitel 8 eingegangen worden. Hierzu wurde zu Beginn im Abschnitt 8.1 die Entwicklung der Softwareentwicklung beschrieben und insbesondere die Ergebnisse der unterschiedlichen Kapitel im Bezug hierzu wiedergegeben. In Anlehnung dazu stellte der Abschnitt 8.2 die Bedeutung der Analyse des Ist-Zustands zur Migration und Initialisierung agiler Prozesse dar. Im Fokus der enormen Herausforderungen diente der Abschnitt 8.3 zur Darlegung verschiedener Ziele, die sich aus dieser Arbeit ergeben haben. Dabei wurden keine global gültigen Muster beschrieben, sondern es ist vielmehr auf Problemstellungen eingegangen worden.

9.2 Ausblick

Inhalt dieser Arbeit war die Betrachtung des Stands der agilen Softwareentwicklung in der Praxis. Dabei wurden verschiedenste Punkte dargestellt, warum die traditionelle Verfolgung von Softwareprojekten nur im Kontext der aktuellen Herausforderungen hilfreich ist sowie auf Schwierigkeiten der Implementierung und Migration von agilen Verfahren aufmerksam gemacht. Auch in der Untersuchung von Unternehmen, die agile Methoden verwenden, zeigte sich, dass es weiterhin eine Vielzahl von Faktoren gibt, die eher traditionellen Vorgehensmodellen zuzuordnen sind. Trotz dieser Darstellung von Herausforderungen und Hindernissen lässt sich zeigen, dass es sich weiterhin lohnt, die agile Softwareentwicklung zu nutzen. Auch wurde dargelegt, dass besonders hierzu die Untersuchung des gegebenen Kontextes durch Schnittstellen zu Projekten und dem Unternehmen essentiell sind. Dabei ist die Unterstützung der Stakeholder des eigenen Unternehmens eine entscheidende Aufgabe. Eine interessante Betrachtung hierzu stellen die aktuellen Herausforderungen von Unternehmen im Kontext der sogenannten digitalen Disruption dar [47], die bzgl. ihrer Veränderungen auf den unterschiedlichen Unternehmensebenen eine Beeinflussung hierzu bedeuten könnten. Des Weiteren ist in Bezug auf diese Arbeit zu untersuchen, inwieweit sich die empirische Untersuchung zu anderen Gegebenheiten präsentieren lässt. So können bei einem anderen Fokus der Befragung und in Verbindung der Analyse mit den Erkenntnissen der Potenzialanalyse neue Verfahren entwickelt werden. Auf Grundlagen dessen bieten sich neue Erkenntnisse, die zur weiteren Beschleunigung der agilen Softwareentwicklung beitragen können.

Diese Arbeit hat sich sowohl auf die initiale sowie auf die weitere Adaption von agilen Prozessen fokussiert. Hierzu wurde ein aktueller Stand der agilen Softwareentwicklung in der Praxis dargelegt. Die Identifikation über die des Scrum Masters hat dabei gezeigt, dass eine Vielzahl von Hindernissen sowie traditionelle und agile Bestandteile des Prozesses sichtbar werden. Zusätzlich könnte der Wechsel der Identifikation vom Scrum Master zum Product Owner neue Erkenntnisse erbringen, die auf einer anderen Ebene ansetzt. Diese könnten sowohl das Verständnis der Rolle sowie eine bessere Eingliederung in den Prozess präsentieren. Eine weitere Betrachtung in Bezug auf diese Arbeit stellt die Untersuchung der agilen Verfahren als vergangene Entwicklungsmethode dar. Hierbei steht insbesondere die starke Veränderung des agilen Prozesses durch intensive Adaptierungsvorgänge sowie ein Wechsel auf andere agile oder nicht agile Verfahren im Fokus. Diese neue Betrachtung und die hier dargestellten Ergebnisse führen dazu, dass ein verbessertes Verständnis der Softwareentwicklung generiert werden kann, welches besonders durch die Bestandteile der Untersuchungen der angewandten Wissenschaften und praxisnahen Erhebungen unterstützt wird.

Literaturverzeichnis

- [1] AGILE ALLIANCE: Agiles Manifest. In: *OBJEKTSpektrum* (2016), Nr. Online Themenspecial Agilität, S. 1–6. – URL http://www.sigs-datacom.de/uploads/tx_dmjournals/Muth_Valentini_OTS_Agility_16.pdf
- [2] BARTON, B: All-Out Organizational Scrum as an Innovation Value Chain. In: *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, jan 2009, S. 1–6. – ISSN 1530-1605
- [3] BASS, J M.: Scrum Master Activities: Process Tailoring in Large Enterprise Projects. In: *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*, aug 2014, S. 6–15
- [4] BECK, Kent ; BEEDLE, Mike ; BENNEKUM, Arie van ; COCKBURN, Alistair ; CUNNINGHAM, Ward ; FOWLER, Martin ; GRENNING, James ; HIGHSMITH, Jim ; HUNT, Andrew ; JEFFRIES, Ron ; KERN, Jon ; MARICK, Brian ; MARTIN, Robert C. ; MELLOR, Steve ; SCHWABER, Ken ; SUTHERLAND, Jeff ; THOMAS, Dave: *Manifesto for Agile Software Development*. 2001. – URL <http://www.agilemanifesto.org/>
- [5] BECK, Kent ; BEEDLE, Mike ; BENNEKUM, Arie van ; COCKBURN, Alistair ; CUNNINGHAM, Ward ; FOWLER, Martin ; GRENNING, James ; HIGHSMITH, Jim ; HUNT, Andrew ; JEFFRIES, Ron ; KERN, Jon ; MARICK, Brian ; MARTIN, Robert C. ; MELLOR, Steve ; SCHWABER, Ken ; SUTHERLAND, Jeff ; THOMAS, Dave: *Principles behind the Agile Manifesto*. 2001. – URL <http://agilemanifesto.org/iso/en/principles.html>
- [6] BOEHM, B W.: Guidelines for Verifying and Validating Software Requirements and Design Specifications. In: SAMET, P A. (Hrsg.): *Euro IFIP 79: Proceedings of the European Conference on Applied Information Technology of the International Federation for Information Processing, London, 25-28 September, 1979*, North Holland, 1979, S. 711–719

- [7] BOEHM, Barry W.: A spiral model of software development and enhancement. In: *Computer* 21 (1988), may, Nr. 5, S. 61–72. – URL <http://ieeexplore.ieee.org/document/59>. – ISBN 0018-9162
- [8] BOEHM, Barry W. ; TURNER, Richard: *Balancing agility and discipline : a guide for the perplexed*. Addison-Wesley, 2004. – 266 S. – URL <http://proquest.tech.safaribooksonline.de/book/software-engineering-and-development/agile-development/0321186125>. – ISBN 0321186125
- [9] BOGNER, Alexander ; LITTIG, Beate ; MENZ, Wolfgang: *Interviews mit Experten : eine praxisorientierte Einführung*. Wiesbaden : Springer VS, 2014 (Qualitative Sozialforschung). – Online-Ressource (VI, 105 S. 1 Abb. in Farbe) S
- [10] CHOUDHARY, B ; RAKESH, S K.: An approach using agile method for software development. In: *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, feb 2016, S. 155–158
- [11] CONBOY, Kieran ; COYLE, Sharon ; WANG, Xiaofeng ; PIKKARAINEN, Minna: People over process: Key challenges in agile development. In: *IEEE Software* 28 (2011), Nr. 4, S. 48–57. – ISBN 0740-7459 VO - 28
- [12] DE O. MELO, Claudia ; S. CRUZES, Daniela ; KON, Fabio ; CONRADI, Reidar: Interpretative case studies on agile team productivity and management. In: *Information and Software Technology* 55 (2013), feb, Nr. 2, S. 412–427. – URL <http://www.sciencedirect.com/science/article/pii/S0950584912001875>. – ISSN 09505849
- [13] DIEBOLD, Philipp ; SCHERR, Simon A.: Software Process Models vs. Descriptions: What Do Practitioners Use and Need? In: *Proceedings of the International Workshop on Software and Systems Process*. New York, New York, USA : ACM Press, 2016, S. 66–75. – URL <http://dl.acm.org/citation.cfm?doid=2904354.2904356>. – ISBN 978-1-4503-4188-2
- [14] DINGSØYR, Torgeir ; NERUR, Sridhar ; BALIJEPALLY, Venugopal ; MOE, Nils B.: A decade of agile methodologies: Towards explaining agile software development. In: *Journal of Systems and Software* 85 (2012), Nr. 6, S. 1213–1221. – ISBN 0164-1212
- [15] DYBÅ, T ; DINGSØYR, T: Agile Project Management: From Self-Managing Teams to Large-Scale Development. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering Bd. 2*, 2015, S. 945–946. – ISSN 0270-5257

- [16] ECKSTEIN, Jutta: *Agile Softwareentwicklung in großen Projekten Teams, Prozesse und Technologien - Strategien für den Wandel im Unternehmen*. 2. Heidelberg : dpunkt.verlag GmbH, 2012. – ISBN 978-3-89864-790-8
- [17] EPPING, Thomas: *Kanban für die Softwareentwicklung*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. – 136 S. – URL <http://link.springer.com/10.1007/978-3-642-22595-6>. – ISBN 978-3-642-22594-9
- [18] FALLIS, A.G.: *Agile IT-Projekte erfolgreich gestalten*. 2013. – 1689–1699 S. – ISBN 9788578110796
- [19] FRIEDRICH, Jan ; HAMMERSCHALL, Ulrike ; KUHRMANN, Marco ; SIHLING, Marc: Das V-Modell XT. In: *Das V-Modell® XT*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009 (Informatik im Fokus), S. 1–32. – URL <http://link.springer.com/10.1007/978-3-642-01488-8>. – ISBN 978-3-642-01488-8
- [20] GOLL, Joachim ; HOMMEL, Daniel: Das agile Rahmenwerk Scrum. In: *Mit Scrum zum gewünschten System*. Springer Fachmedien Wiesbaden, 2015, S. 81–115. – URL http://dx.doi.org/10.1007/978-3-658-10721-5_4. – ISBN 978-3-658-10720-8
- [21] GOLL, Joachim ; HOMMEL, Daniel: *Mit Scrum zum gewünschten System*. 2015. – 185 S. – ISBN 3658107219
- [22] HANSER, Eckhart: *Agile Prozesse: Von XP über Scrum bis MAP*. 2010. – 1689–1699 S. – ISBN 9788578110796
- [23] HOFERT, Svenja: *Agiler führen*. Wiesbaden : Springer Fachmedien Wiesbaden, 2016. – URL <http://link.springer.com/10.1007/978-3-658-12757-2>. – ISBN 978-3-658-12756-5
- [24] HÖHN, R. ; HÖPPNER, S.: *Das V-Modell XT. Anwendungen, Werkzeuge, Standards*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008 (eXamen.press). – 629 S. – URL <http://link.springer.com/10.1007/978-3-540-30250-6>. – ISBN 9783540302490
- [25] HOOD, Colin: Das V-Modell ist tot. Lang lebe das V-Modell! In: *OBJEKTSpektrum* (2016), Nr. Online Themenspecial Agilität, S. 4. – URL http://www.sigs-datacom.de/uploads/tx_dmjournals/Hood_OTS_Agility_16.pdf

- [26] HUMMEL, Markus ; ROSENKRANZ, Christoph ; HOLTEN, Roland: The role of communication in agile systems development: An analysis of the state of the art. In: *Business and Information Systems Engineering* 5 (2013), Nr. 5, S. 343–355. – ISBN 1157601303760
- [27] JEPPESEN, Klaus J.: How To Live With Software Problems. In: *Diffusing Software Product and Process Innovations*. Boston, MA : Springer US, 2001, S. 225–239. – URL http://link.springer.com/10.1007/978-0-387-35404-0_14
- [28] JIE, Jason Lee H.: Industrial Case Study of Transition from V-Model into Agile SCRUM in Embedded Software Testing Industries. In: *SIGSOFT Softw. Eng. Notes* 41 (2016), Nr. 2, S. 1–3. – URL <http://doi.acm.org/10.1145/2894784.2894793>. – ISSN 0163-5948
- [29] JUHOLA, T ; YIP, M H. ; HYRYNSALMI, S ; MAKILA, T ; LEPPANEN, V: The connection of the stakeholder cooperation intensity and team agility in software development. In: *Management of Innovation and Technology (ICMIT), 2014 IEEE International Conference on*, IEEE, sep 2014, S. 199–204. – URL <http://ieeexplore.ieee.org/document/6942425>
- [30] KAACK, Alexander: Methodische Entwicklung zur konzeptionellen Erfassung einer Softwarelösung für Scrum Master. In: *Berichte Masterprojekt Master Informatik 2015* (2015). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/kaack.pdf>
- [31] KAACK, Alexander: Empirische Erhebung von Impediments über Scrum Master in der Praxis agiler Softwareentwicklung. In: *Berichte Masterprojekt Master Informatik 2016* (2016). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2016-proj/kaack.pdf>
- [32] KNIBERG, Henrik.: *Kanban and Scrum - Making the most of both*. C4Media, Inc, 2009. – 1–49 S. – URL <https://www.infoq.com/minibooks/kanban-scrum-minibook>. – ISBN 9780557138326
- [33] KOMUS, Prof. Dr. A.: *Study Report Status Quo Agile 2014*. BPM-Laboratory of the Koblenz University of Applied Sciences, 2014. – URL <http://www.status-quo-agile.de/>
- [34] KRUCHTEN, Philippe: *The Rational Unified Process: An Introduction*. Addison-Wesley, 1998. – ISBN 0201604590

- [35] KUTSCHA, Sebastian: Immer Ärger mit dem Product Owner: Plädoyer für ein PO-Team. In: *OBJEKTSpektrum* (2016), Nr. Online Themenspecial Agilität, S. 3
- [36] LENT, Bogdan: *IT-Projektmanagement als kybernetisches System*. Wiesbaden : Springer Fachmedien Wiesbaden, 2013. – URL <http://link.springer.com/10.1007/978-3-8348-2512-4>. – ISBN 978-3-8348-2511-7
- [37] MÄRKTE, Ob: Antifragilität : Warum wir den Glauben an stabile Umgebungen aufgeben müssen. In: *OBJEKTSpektrum* (2016), Nr. Online Themenspecial Agilität, S. 1–2
- [38] MAYRING, Philipp: *Qualitative Inhaltsanalyse : Grundlagen und Techniken*. 12., übera. Weinheim [u.a.] : Beltz, 2015 (Beltz Pädagogik). – Online Ressource (924 KB, 152 S.) S
- [39] MOE, Nils B. ; DINGSØYR, Torgeir ; DYBÅ, Tore: A teamwork model for understanding an agile team: A case study of a Scrum project. In: *Information and Software Technology* 52 (2010), Nr. 5, S. 480–491. – URL <http://www.sciencedirect.com/science/article/pii/S0950584909002043>. – ISSN 09505849
- [40] NAUR, Peter ; RANDELL, Brian: Software Engineering: Report of a conference sponsored by the NATO Science Committee. In: *NATO Software Engineering Conference* (1968), Nr. October 1968, S. 231. – URL <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>. – ISBN 0073655783
- [41] NERUR, Sridhar ; MAHAPATRA, RadhaKanta ; MANGALARAJ, George: Challenges of Migrating to Agile Methodologies. In: *Communications of the ACM* 48 (2005), may, Nr. 2, S. 72–78. – URL <http://portal.acm.org/citation.cfm?doid=1060710.1060712>. – ISBN 0001-0782
- [42] NOORDELOOS, R ; MANTELI, C ; VLIET, H V.: From RUP to Scrum in Global Software Development: A Case Study. In: *2012 IEEE Seventh International Conference on Global Software Engineering*, aug 2012, S. 31–40. – ISSN 2329-6305
- [43] PETERSEN, Kai ; WOHLIN, Claes: The effect of moving from a plan-driven to an incremental software development approach with agile practices. In: *Empirical Software Engineering* 15 (2010), Nr. 6, S. 654–693. – URL <http://dx.doi.org/10.1007/s10664-010-9136-6>. – ISSN 1573-7616
- [44] POWER, K: Stakeholder Identification in Agile Software Product Development Organizations: A Model for Understanding Who and What Really Counts. In: *Agile Conference*

- (AGILE), 2010, IEEE, aug 2010, S. 87–94. – URL <http://ieeexplore.ieee.org/document/5562805>
- [45] POWER, K: Impediment Impact Diagrams: Understanding the Impact of Impediments in Agile Teams and Organizations. In: *Agile Conference (AGILE), 2014*, 2014, S. 41–51
- [46] QASAIMAH, M ; MEHRFARD, H ; HAMOU-LHADJ, A: Comparing Agile Software Processes Based on the Software Development Project Requirements. In: *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, 2008, S. 49–54
- [47] RISING, L ; JANOFF, N S.: The Scrum software development process for small teams. In: *IEEE Software* 17 (2000), jul, Nr. 4, S. 26–32. – ISSN 0740-7459
- [48] ROYCE, Dr. Winston W.: Managing the Development of large Software Systems. In: *Ieee Wescon* (1970), Nr. August, S. 1–9. ISBN 0-89791-216-0
- [49] SCHMITT, Anna ; ZEHLER, Thomas: Agile Potenzialanalyse : Der Weg zum passenden Grad an Agilität. In: *OBJEKTSpektrum* (2016), Nr. Online Themenspecial Agilität, S. 1–4. – URL http://www.sigs-datacom.de/uploads/tx_dmjournals/Diebold_Schmitt_Zehler_OTS_Agility_16.pdf
- [50] SCHUH, Peter: *Integrating Agile Development in the Real World*. 1st. Rockland, MA, USA : Charles River Media, Inc., 2005. – 346 S. – URL http://books.google.se/books/about/Integrating_Agile_Development_in_the_Rea.html?id=9J_0ly01QicC&pgis=1. – ISBN 1584503645
- [51] SCHWABER, K ; SUTHERLAND, J: *The Scrum Guide*. scrumguides.org, 2011. – URL <http://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>
- [52] SOMMERVILLE, Ian: *Software-Engineering*. München; Boston [u.\,a.] : Pearson Studium, 2007. – ISBN 9783827372574 3827372577
- [53] SVERRISDOTTIR, Hrafnhildur S. ; INGASON, Helgi T. ; JONASSON, Haukur I.: The Role of the Product Owner in Scrum-comparison between Theory and Practices. In: *Procedia - Social and Behavioral Sciences* 119 (2014), S. 257–267. – URL <http://www.sciencedirect.com/science/article/pii/S1877042814021211>. – ISSN 18770428

- [54] SWENSON, Keith D.: Designing for an innovative learning organization. In: *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, IEEE, sep 2013, S. 209–213. – URL <http://ieeexplore.ieee.org/document/6658281>. – ISBN 978-0-7695-5081-7
- [55] TANVEER, M: Agile for large scale projects - A hybrid approach. In: *2015 National Software Engineering Conference (NSEC)*, IEEE, 2015, S. 14–18. – URL <http://ieeexplore.ieee.org/document/7396338>
- [56] VERSIONONE: *10th Annual State of Agile Survey*. VersionOne, 2016. – URL <http://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>
- [57] VERSIONONE: *11th Annual State of Agile Report*. VersionOne, 2017. – URL <http://bit.ly/2qfL4Uk>
- [58] WAARDENBURG, Guus van ; VLIET, Hans van: When agile meets the enterprise. In: *Information and Software Technology* 55 (2013), Nr. 12, S. 2154–2171. – ISSN 09505849
- [59] WAGUESPACK, L J. ; SCHIANO, W T.: SCRUM Project Architecture and Thriving Systems Theory. In: *System Science (HICSS), 2012 45th Hawaii International Conference on*, jan 2012, S. 4943–4951. – ISSN 1530-1605
- [60] WIRDEMANN, Ralf: *Scrum mit User Stories*. München [u.a. : Hanser, 2011. – ISBN 9783446426603 3446426604
- [61] ZÖRNER, Stefan: *Requirements-Engineering und -Management : aus der Praxis von klassisch bis agil*. München : Hanser, 2014. – URL <http://d-nb.info/1056934166>. – ISBN 978-3-446-43893-4; 3-446-43893-9

Hallo Herr/Frau XYZ,

im Rahmen meiner Forschungsarbeiten innerhalb des Master-Studiengangs Informatik an der HAW-Hamburg beschäftige ich mich intensiv mit der Rolle des Scrum Masters. Hierzu habe ich vor, dass eine Vielzahl von Scrum Mastern über ein möglichst persönliches Interview befragt werden.

Das Ziel des Interviews ist es, dass Probleme und Herausforderungen der Scrum Master eingestuft werden können. Darunter ist zu verstehen, welche Kategorie die häufigsten Probleme bzw. Hindernisse hält. Zum anderen soll es möglich sein, dass eine Generalisierung der Probleme ermöglicht wird, die sich unabhängig von der Unternehmens- und Projektgröße darstellen lässt. Ggf. lassen sich bestimmte Probleme auch auf die Unternehmenskategorisierung referenzieren. Hierzu wurde der folgende Interviewleitfaden erfasst.

Die Interviews sollen als Einzelinterviews durchgeführt werden, um eine möglichst neutrale Bewertung der Scrum Master zu erhalten. Die Zeit der Befragung ist generell auf eine Stunde begrenzt, kann sich aber bei Interesse entsprechend verlängern. Wenn es möglich ist, so soll das Interview sprachlich aufgezeichnet sowie schriftlich festgehalten werden. Die Sprachaufzeichnung ermöglicht eine verbesserte Analyse des Interviews, nachdem es geführt wurde. Die Ergebnisse werden im Anschluss anonymisiert erfasst, damit firmeneigene Daten geschützt werden.

Der Ablauf des Interviews ist wie folgt gegliedert:

1. Vorstellung des Projekts
2. Befragung zur Kontextbildung des Scrum Masters
3. Befragung bzgl. Hindernissen in Projekten
4. Befragung zu nützlichen Tools bzw. Hilfsmitteln

Ich würde mich sehr freuen, wenn Sie mir bei meinen Forschungsarbeiten behilflich sein könnten.

Mit freundlichen Grüßen

Alexander Kaack

Rahmeninformationen						
Interview Nr.	Teilnehmer					
1	Person A					
2	Person B					
3	Person C					
4	Person D					
5	Person E					
Interview Nr.	Geschlecht					
1	Männlich					
2	Männlich					
3	Männlich					
4	Weiblich					
5	Männlich					
Interview Nr.	Alter / Altersgruppe (optional)					
1		37				
2		47				
3		35				
4		28				
5		45				
Interview Nr.	Firma (optional)					
1	Unternehmen A					
2	Unternehmen B					
3	Unternehmen C					
4	Unternehmen C					
5	Unternehmen D					
Interview Nr.	Ort					
1	Cafe					
2	Meeting Raum - Unternehmen B					
3	Meeting Raum - Unternehmen C					
4	Meeting Raum - Unternehmen C					
5	Meeting Raum - Unternehmen D					
Interview Nr.	Dauer in Minuten	Von	Bis			
1	151	16:34	19:05			
2	109	08:41	10:30			
3	90	13:00	14:30			
4	95	15:30	17:05			
5	76	11:47	13:03			
	521	=	8	Stunden	41	Minuten
Interview Nr.	Sonstiges					
1	/					
2	/					
3	/					
4	/					
5	/					

Anhang B: Auswertungstabellen der ersten Auswertung

Frage Nr. Befragung zum professionellen Kontext des Scrum Masters		
1	Interview Nr.	Wie viele Projekte haben Sie bisher als Scrum Master betreut? (Dauer?, Art?)
	1	2 x als Stellvertreter und 2 x als fester Scrum Master
	2	2 Projekte - Langzeitprojekte, Unterstützung / Organisation der Anwendungsentwicklung
	3	>10 ; kann nicht spezifiziert werden
	4	Mehrere kleine, 1 Team, 2 Jahre
	5	20 Projekte - im klassischen Sinne
2	Interview Nr.	Wie viele Jahre haben Sie als Scrum Master Erfahrungen sammeln können? (Durchgängig?, Pausiert?)
	1	< 1 Jahr, nicht durchgängig
	2	< 3 Jahre durchgängig
	3	4 Jahre durchgängig
	4	5 Jahre, nicht durchgängig
	5	Agiler Projektmanager 8 Jahre
3	Interview Nr.	Besitzen oder besitzten Sie während Ihrer Tätigkeit als Scrum Master parallel eine weitere Rolle im Unternehmen? Wenn ja welche?
	1	Entwickler
	2	Entwickler, Systembetreuer JIRA
	3	Projektleiter
	4	Agiler Coach, PL, Innovations-Manager, Discovery Coach
	5	Teamlead für andere Agile Projektmanager
4	Interview Nr.	Besitzen Sie Programmiererfahrung? Wenn ja, wie viele Jahre? (Wird noch aktiv ausgeübt?, Beruflich?)
	1	Ja, seit 30 Jahren
	2	16 Jahre, nur beruflich aktiv
	3	3 Jahre, nicht mehr aktiv
	4	Nein
	5	Nein
5	Interview Nr.	Über welchen Zeitraum erstreckten sich Ihre letzten drei Projekte, indem Sie als Scrum Master tätig waren? (Art?)
	1	2 Jahre, 2,5 Jahre und 1,5 Jahre
	2	3 Jahre, 2 Monate
	3	1,5 Jahre und 30 kleinere Projekte
	4	1 Projekt, 3 und 2 Jahre
	5	1-6 Monate, immer gleiche Projektarten
6	Interview Nr.	Betreuen Sie generell mehrere Projekte parallel, wenn ja wie viele im Durchschnitt?
	1	Nein
	2	2 bis 3
	3	Ja, max 4-5, mit dem Team
	4	Ja, max 4-5, mit dem Team
	5	Ja, 2-3, Stabile Teams

Anhang B: Auswertungstabellen der ersten Auswertung

7	Interview Nr.	Besitzen Sie Erfahrungen bei der Teilnahme und Führung von Scrum of Scrum Meetings? (Rolle?)
	1	Ja, 1 x moderiert
	2	Nein - machen PO's
	3	Ja, in unregelmäßigen Abständen, Vorstellung der Teams sowie Rückfragen zu Quell- und Zielterm am Board
	4	Nein
	5	Ja, aber selten
8	Interview Nr.	Wie ist die durchschnittliche Anzahl an Personen (gruppiert nach Rolle) im Projekt? (Welche Rollen sind anwesend?)
	1	Unterschiedlich 3-14 Entwickler, 1 Scrum Master, 1 Product Owner, 4 partielle Unterstützer aus dem DevOps Team
	2	7 Entwickler, 1 Product Owner, 1 Scrum Master
	3	15 inkl. Triade - Project Lead, Quality Assurance, Projektmanager
	4	15 inkl. Triade - Project Lead, Quality Assurance, Projektmanager
	5	7+-2 (inkl. Product Owner und Scrum Master)
9	Interview Nr.	Wie sind die Art der Projekte Ihrer Meinung nach zu beschreiben? (Projektbasierend?, Langzeitprojekte?, Wartung?, Erweiterungen?)
	1	Zum Großteil klassische Projekte, Start > Ende
	2	Langzeitprojekte, Change- und Organisationsprojekte, Wartung
	3	Alle Arten von Projekten
	4	Alle Arten von Projekten
	5	Projektbasierend
10	Interview Nr.	In welche Größenklasse lässt sich Ihr Unternehmen eingliedern? (klein, mittel, groß?)
	1	Mittel (120 Mitarbeiter)
	2	Großunternehmen (12.000 Mitarbeiter)
	3	Großunternehmen
	4	Großunternehmen
	5	Größeres, mittelständiges Unternehmen
11	Interview Nr.	Wie ist der Umgang mit Kommunikation in Ihrem Unternehmen zu beschreiben? (Direkter Kontakt? Über Mittelspersonen?)
	1	Eher direkter Kontakt, außer bei Ressourcenanfragen etc.
	2	E-Mail als Nachweis - kein Chat, Beschleunigt nur durch Bekannte, Standard Prozesse für Freigaben
	3	Direkter Kontakt nur bis zum Project Lead
	4	Nur über den SM in andere Teams
	5	Eher direkt, bei manchen über Hierarchie - Großunternehmensstruktur
12	Interview Nr.	Um was für eine Art von Unternehmen handelt es sich? (Branche, Bezeichnung)
	1	IT Dienstleister
	2	Versicherung
	3	Versand, Einzelhandel
	4	Versand, Einzelhandel
	5	IT-Produktentwicklung

Frage Nr. Befragung bzgl. Impediments in Projekten		
13	Interview Nr.	Was sind Ihre häufigsten Impediments mit einem großen Impact in Projekten, bezogen auf das Team? (Stört das Team bzw. die direkte Produktivität?)
	1	1. Continuous Integration (Konfig/Performance), 2. Ineffektive Meetings / Große Teams, 3. Technische Schulden
	2	1. Überlastung, 2. Organisationsstruktur -> Standard Prozesse, 3. Meetingräume - verteilte Entwickler
	3	1. Lehre / Realität - Durchfluss / Externe, 2. Adaption der Mitarbeiter bzgl. Scrum bei Unternehmen C, 3. Team Struktur (Teamfeeling)
	4	1. Durch Politik keine Eigenverantwortung, 2. Unklare Verantwortungen 3. Prozesse
	5	1. Auftragsklärung, 2. Entscheidungen durch Stakeholder von außen, 3. Austausch von Mitarbeitern
14	Interview Nr.	Was sind Ihre häufigsten Impediments mit einem großen Impact in Projekten, bezogen auf ihre Tätigkeit als Scrum Master? (Was Sie an Ihrer Tätigkeit hindert?)
	1	1. Gleichzeitig Entwickler / Rollenkonflikt, 2. Material / Beschaffung, 3. Milestones -> Sprintkonflikt
	2	1. Keine Videokonferenzräume buchbar, 2. Räumliche Distanz zum Team, 3. Feedbackmöglichkeit zum Scrum Master
	3	1. Adaptionproblematik Scrum im Unternehmen, 2. Teambuilding, 3. Kommunikation
	4	1. Plexo / Agiles Reporting auf klassisches Projektmanagement, 2. Mehr Projektleiter als Scrum Master, 3. Politik des Hauses, 4. Langsame Entscheidungswege
	5	1. Rollenkonflikte Scrum Master - Projektmanager, 2. Bereitschaft in Konflikte zu gehen
15	Interview Nr.	Was sind Ihre häufigsten Impediments, bezogen auf das Team, die einen großen Beseitigungsaufwand besitzen? (Langfristig, mittelfristig, kurzfristig?)
	1	1. Continuous Integration 2. Testdaten - abhängig von Projektkomplexität
	2	Nicht aussagefähig
	3	Langfristig - alle aus 13
	4	Langfristig - alle aus 13
	5	80-90% kurzfristig

Anhang B: Auswertungstabellen der ersten Auswertung

16	Interview Nr.	Was sind Ihre häufigsten Impediments, bezogen auf ihre Tätigkeit als Scrum Master, die einen großen Beseitigungsaufwand besitzen? (Langfristig, mittelfristig, kurzfristig?)
	1	Mittelfristig / kurzfristig - Aufgabenverteilung / Rollenkonflikt ; Kurzfristiges Material
	2	Nicht aussagefähig
	3	Siehe 15 langfristig
	4	Siehe 15 langfristig
	5	Mittel und kurzfristig siehe 14
17	Interview Nr.	Was sind Ihre häufigsten Impediments, bezogen auf das Team, die Sie besonders hoch priorisieren? (Welche sind warum besonders wichtig für Sie?)
	1	Steht in 13
	2	Steht in 13
	3	Steht in 13
	4	Steht in 13
	5	Steht in 13
18	Interview Nr.	Was sind Ihre häufigsten Impediments, bezogen auf ihre Tätigkeit als Scrum Master, die Sie besonders hoch priorisieren? (Welche sind warum besonders wichtig für Sie?)
	1	Steht in 14
	2	Steht in 14
	3	Steht in 14
	4	Steht in 14
	5	Steht in 14
19	Interview Nr.	Wann werden Impediments bei Ihnen erfasst? (Ritual, OnDemand?)
	1	Daily, jederzeit auf dem Whiteboard, Retro
	2	Retro erfasst aber nicht priorisiert, Gliffy Board, F2F/Tele/E-Mail
	3	OnDemand - Jederzeit
	4	OnDemand - Jederzeit
	5	OnDemand - Jederzeit
20	Interview Nr.	Wie werden Impediments bei Ihnen erfasst? (Tags an einer Metaplanwand?)
	1	Wasteboard - Dient auch der Retro - Prio über Teammitglieder
	2	Digital, Retro mit Kärtchen
	3	Metaplanwand - eher weniger, Direkt, E-Mail, Meetings
	4	JIRA, über Retrobeschluss, Haptische Boards nach JIRA
	5	Agile Health Check -> Online Survery, von Team zu Team unterschiedlich

Anhang B: Auswertungstabellen der ersten Auswertung

21	Interview Nr.	Wie werden Impediments bei Ihnen verfolgt und archiviert? (Vorgehen?)
	1	Wichtige Impediments werden über das Retroprotokoll archiviert, sonst nur temp. über Wasteboard
	2	Unzureichend, Teil über Retro -> Fotos
	3	Nein
	4	JIRA oder gar nicht
	5	Eher selber für SM, keine open list

Anhang B: Auswertungstabellen der ersten Auswertung

Befragung zu nützlichen Tools bzw. Hilfsmitteln		
Frage Nr.	Interview Nr.	Welche Tools unterstützen Sie in Ihrer Tätigkeit als Scrum Master? (Welches für was?)
22	1	0. Taskboard (Rückseite Technische Schuld Board), 1. Retroprotokoll, 2. Backlog Tasks keine Stories, 3. Karteikarten, 4. Wasteboard, 5. Persönliche Gespräche, 6. Jenkins, 7. Skype
	2	0. Confluence mit Gliffy, 1. JIRA
	3	1. Jira, 2. Wiki Confluence, 3/4 Excel und Plexo = Power Market Simulation
	4	1. Creative Methods + Design Thinking, 2. Visual Facilitation, 3. Jira, 4. Confluence, 5. Plexo -> Zeitbuchung Reporting
	5	Excel, Coaching Canvas, Personal Kanban, Trello Online Kanban, klassische Boards
23		Wie würden Sie die eingesetzten Tools im Verhältnis priorisieren? (Warum?)
	1	Frage 22
	2	Frage 22
	3	Frage 22
	4	Frage 22
	5	Keine Prio möglich, je nach Situation
24		Dient eines der verwendeten Tools zur Erhebung der Impediments? Wenn ja, welche Art von Impediments werden dort erfasst? (Impedimentsbacklog?)
	1	Wasteboard, Rückseite Taskboard als Technische Schuld Board, Retroprotokoll
	2	Confluence -> Retro (gesicherter Bereich)
	3	Nein
	4	JIRA, Visual Facs
	5	Excel Sheet
25		Was würde Ihrer Meinung nach eine zusätzliche Hilfestellung, auch bezogen auf die Kommunikationsunterstützung, für die Arbeiten des Scrum Masters bieten?
	1	Confluence (Dinge besser Erfassen, Dinge in Verbindung bringen, Aufgaben aus Retro, Verbesserte Dokumentation), anderes Tool als Skype
	2	Beamen (räumliche Trennung zu Dortmund), Bessere Präsenz durch digitale Unterstützungen
	3	Jemand der mit Menschen umgehen kann
	4	Flache Hierarchien, Parallele Überschneidung
	5	Personen zusammensetzen, Business und Tech besser verbinden
26		Sonstiges
	1	/
	2	Meeting Ablauf, Impactmessung, Retrospektive im ganzen Team (Dortmund), Scrum Master in anderen Projekten, Vollzeit Scrum Master
	3	Kein Standard Scrum
	4	Natürlicher Verlauf
	5	People over Interaction and Tools

Weitere Erkenntnisse

Großen Impact - Team

CI, Überlastung, Lehre/Realität, Durch Unternehmen keine Eigenverantwortung, Auftragsklärung
Ineffektive Meetings, Organisationsstruktur, Adaption der Mitarbeiter bzgl. Scrum, Unklare Verantwortungen, Entscheidungen durch Stakeholder von Außen
Technische Schulden, Meetingräume - verteilte Entwicklung, Team Struktur (Feeling), Prozesse, Austausch von Mitarbeitern

Großen Impact - Scrum Master

Rollenkonflikt, Räumlichkeiten für Meetings (Distanz Mitarbeiter), Adaptionproblematik im Unternehmen, Agile Reporting zu klassischen PM (Milestones), Beschaffung, Teambuilding, PL mehr als SM, Bereitschaft Konflikte, Feedbackmöglichkeit, Kommunikation, Politik des Hauses, Langsame Entscheidungswege

- Problem der Verifikation von Projekten - wie definieren
- direkt immer im Team - bei HUs eher über Lead Positionen falls eine Kommunikations außerhalb des Teams stattfinden soll.
- Zusammensetzung, Regeln, Vorgehen im Unternehmen unterschiedlich

- Probleme wiederholen sich

Tools

- Eher Visual Facts, bzw. persönlch
- Zum Teil Unterstützt durch Confluence / JIRA oder Excel oder gar nicht
- Tooling nicht ausschlaggebend für Scrum Master Tätigkeiten

- Frischer Wind von Außen

- DevOps

- TechiDenke - Technische Schuld - Rückseite Taskboard

- Thema -> Thema | Bauen Messen Lernen -> aber nur bauen bauen bauen - messen lernen und dann wieder ein neues Thema

- Frage nach dem minimal valuable Product

- Individuals and Interaction over Processes and Tasks

- MMM Fehler - Mehr Personen in ein Team - also schneller Fertig

- Wichtige Teams werden aufgestockt - "interne Splits"

HU=Hierarchische Unternehmen

- Impediments Denke fehlt - HU

- PO's im SOS - PO zu viel Verantwortung - HU

- Rollenkonflikt - HU

- Kein Knowledge Sharing - HU

- Langfristige Probleme - HU

- Technische (SM / Entwickler) - Beeinflussung der Rolle zu Befragung eher Richtung Technik

- People Fokus bei Teams mit längerer SM Erfahrung und "hohen" Scrum Level

- Aufgeschlossenheit zur Untersuchung

- Abweichung der Erfahrung von Scrum. Hiermit ist die Erfahrung nach Dauer zum

Weitere Erkenntnisse

tiefgründigen Verständnis gemeint.

-Adaption von Scrum - Unternehmensstrukturen/klassisches

Projektmanagement/Regularien

-Rechtfertigung der Rolle (Rollen-Fusion bzw. Rollen-Schizophrenie)

-Impediments

-Feedback der Rolle

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

Rahmeninformationen

Teilnehmer (optional): _____

Geschlecht (optional): _____

Alter / Altersgruppe (optional): _____

Firma (optional): _____

Ort: _____

Dauer (Von/Bis): _____

Sonstige Informationen: _____

Der Ablauf des Interviews ist wie folgt gegliedert:

1. Vorstellung des Projekts
2. Befragung zum professionellen Kontext des Scrum Masters
3. Befragung bzgl. Impediments in Projekten
4. Befragung zu nützlichen Tools bzw. Hilfsmitteln

1. Vorstellung des Projekts

Im Rahmen meiner Forschungsarbeiten innerhalb des Master-Studiengangs Informatik an der HAW-Hamburg beschäftige ich mich intensiv mit der Rolle des Scrum Masters. Hierzu möchte ich gern eine Vielzahl von Scrum Mastern über ein möglichst persönliches Interview befragen.

Das Ziel des Interviews ist es, dass Probleme und Herausforderungen der Scrum Master eingestuft werden können. Darunter ist zu verstehen, welche Kategorie die häufigsten Probleme bzw. Impediments (deut. Hindernisse/Probleme/Herausforderungen) beinhaltet. Außerdem soll es möglich sein, dass eine Generalisierung der Probleme ermöglicht wird, die sich unabhängig von der Unternehmens- und Projektgröße darstellen lässt. Ggf. lassen sich bestimmte Probleme auch auf die Unternehmenskategorisierung referenzieren. Hierzu wurde der folgende Interviewleitfaden erfasst.

Die Interviews sollen als Einzelinterviews durchgeführt werden, um eine möglichst neutrale Bewertung der Scrum Master zu erhalten. Die Zeit der Befragung ist generell auf eine Stunde begrenzt, kann sich aber bei Interesse entsprechend verlängern. Wenn möglich, soll das Interview sprachlich aufgezeichnet sowie schriftlich festgehalten werden. Die Sprachaufzeichnung ermöglicht im Anschluss eine verbesserte Analyse des Interviews. Die Ergebnisse werden im Anschluss anonymisiert erfasst, damit firmeneigene Daten geschützt werden.

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

2. Befragung zum professionellen Kontext des Scrum Masters

Die folgenden Fragen dienen der Kontextbildung des Scrum Masters. Hierdurch wird es möglich, die Aussagen gemäß Erfahrung, Unternehmen und Projekt zu kategorisieren.

1. Wie viele Projekte haben Sie bisher als Scrum Master betreut?
(Dauer?, Art?)

2. Wie viele Jahre haben Sie als Scrum Master Erfahrungen sammeln können?
(Durchgängig?, Pausiert?)

3. Besitzen oder besitzten Sie während Ihrer Tätigkeit als Scrum Master parallel eine weitere Rolle im Unternehmen? Wenn ja welche?

4. Besitzen Sie Programmiererfahrung? Wenn ja, wie viele Jahre?
(Wird noch aktiv ausgeübt?, Beruflich?)

5. Über welchen Zeitraum erstreckten sich Ihre letzten drei Projekte, indem Sie als Scrum Master tätig waren?
(Art?)

6. Betreuen Sie generell mehrere Projekte parallel, wenn ja wie viele im Durchschnitt?

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

- 7.** Besitzen Sie Erfahrungen bei der Teilnahme und Führung von Scrum of Scrum Meetings?
(Rolle?)

- 8.** Wie ist die durchschnittliche Anzahl an Personen (gruppiert nach Rolle) im Projekt?
(Welche Rollen sind anwesend?)

- 9.** Wie sind die Art der Projekte Ihrer Meinung nach zu beschreiben?
(Projektbasierend?, Langzeitprojekte?, Wartung?, Erweiterungen?)

- 10.** In welche Größenklasse lässt sich Ihr Unternehmen eingliedern?
(klein, mittel, groß?)

- 11.** Wie ist der Umgang mit Kommunikation in Ihrem Unternehmen zu beschreiben?
(Direkter Kontakt?, Über Mittelspersonen?)

- 12.** Um was für eine Art von Unternehmen handelt es sich?
(Branche, Bezeichnung)

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

3. Befragung bzgl. Impediments in Projekten

Um die häufigsten Impediments (deut. Hindernisse/Probleme/Herausforderungen) in einem Unternehmen zu erfassen, müssen die folgenden Fragen gestellt werden. Hierdurch wird es auch möglich sein, dass die Impediments bzgl. Auswirkungen (engl. *Impact*), Beseitigungsaufwand und Priorisierung gruppiert werden können.

13. Was sind Ihre häufigsten Impediments mit einem großen Impact in Projekten, bezogen auf das Team?
(Stört das Team bzw. die direkte Produktivität?)

14. Was sind Ihre häufigsten Impediments mit einem großen Impact in Projekten, bezogen auf ihre Tätigkeit als Scrum Master?
(Was Sie an Ihrer Tätigkeit hindert?)

15. Was sind Ihre häufigsten Impediments, bezogen auf das Team, die einen großen Beseitigungsaufwand besitzen?
(Langfristig, mittelfristig, kurzfristig?)

16. Was sind Ihre häufigsten Impediments, bezogen auf ihre Tätigkeit als Scrum Master, die einen großen Beseitigungsaufwand besitzen?
(Langfristig, mittelfristig, kurzfristig?)

17. Was sind Ihre häufigsten Impediments, bezogen auf das Team, die Sie besonders hoch priorisieren?
(Welche sind warum besonders wichtig für Sie?)

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

- 18.** Was sind Ihre häufigsten Impediments, bezogen auf ihre Tätigkeit als Scrum Master, die Sie besonders hoch priorisieren?
(Welche sind warum besonders wichtig für Sie?)

- 19.** Wann werden Impediments bei Ihnen erfasst?
(Ritual, OnDemand?)

- 20.** Wie werden Impediments bei Ihnen erfasst?
(Tags an einer Metaplanwand?)

- 21.** Wie werden Impediments bei Ihnen verfolgt und archiviert?
(Vorgehen?)

Einzelinterview eines Scrum Masters

Datum: 26.01.2016

4. Befragung zu nützlichen Tools bzw. Hilfsmitteln

Um die aktuell am häufigsten genutzten Unterstützungen für Scrum Master zu erfassen, wurden die nachfolgenden Fragen entworfen. Das Ergebnis wird einen aktuellen Stand widerspiegeln, welche Tools erfolgreich oder weniger erfolgreich eingesetzt und darauf aufbauend für Impediments verwendet werden.

22. Welche Tools unterstützen Sie in Ihrer Tätigkeit als Scrum Master?
(Welches für was?)

23. Wie würden Sie die eingesetzten Tools im Verhältnis priorisieren?
(Warum?)

24. Dient eines der verwendeten Tools zur Erhebung der Impediments? Wenn ja, welche Art von Impediments werden dort erfasst?
(Impedimentsbacklog?)

25. Was würde Ihrer Meinung nach eine zusätzliche Hilfestellung, auch bezogen auf die Kommunikationsunterstützung, für die Arbeiten des Scrum Masters bieten?

Sonstiges

Kodierregeln

Kodierregeltabelle

Nr.	Kategorie	Definition	Kodierregel
1	Traditionelles Entwicklungsmodell	Punkte, die auf ein Lebenszyklus Modell (Wasserfall, Spiral, oder Variationen) schließen.	Interviewabschnitt, welcher die Definition erfüllt.
2	Agiles Entwicklungsmodell	Punkte, die auf ein evolutionäres Bereitstellungsmodell schließen.	Interviewabschnitt, welcher die Definition erfüllt.
3	Traditioneller Projektzyklus	Punkte, die auf eine Bestimmung des Projektzyklus über Aufgaben und Aktivitäten schließen	Interviewabschnitt, welcher die Definition erfüllt.
4	Agiler Projektzyklus	Punkte, die auf eine Bestimmung des Projektzyklus über Produktfeatures schließen.	Interviewabschnitt, welcher die Definition erfüllt.
5	Traditionelle Projektplanung	Punkte, die darauf schließen, dass die Projektplanung im Voraus durchgeführt wird.	Interviewabschnitt, welcher die Definition erfüllt.
6	Agile Projektplanung	Punkte, die darauf schließen, dass die Projektplanung kontinuierlich durchgeführt wird.	Interviewabschnitt, welcher die Definition erfüllt.
7	Traditionelle(r) Management-Stil / -Kultur	Punkte, die auf Kommando- und Kontrollstrukturen im Management schließen.	Interviewabschnitt, welcher die Definition erfüllt.
8	Agile(r) Management-Stil / -Kultur	Punkte, die auf Leader und Zusammenarbeiten bzw. responsive Managementstrukturen schließen.	Interviewabschnitt, welcher die Definition erfüllt.
9	Traditionelle Rollenzuordnung	Punkte, die auf Individuen bzw. favorisierte Spezialisierungen bei der Rollenzuordnung schließen.	Interviewabschnitt, welcher die Definition erfüllt.

Kodierregeltabelle

Nr.	Kategorie	Definition	Kodierregel
10	Agile Rollenzuordnung	Punkte, die auf ein selbstorganisiertes Team schließen bzw. Rollen zur Austauschbarkeit untereinander begünstigen	Interviewabschnitt, welcher die Definition erfüllt.
11	Traditionell anstrebende Organisationsform oder -struktur	Punkte, die auf eine mechanistische (Bürokratie mit hohem Formalismus) Organisations-Form oder -Struktur schließen.	Interviewabschnitt, welcher die Definition erfüllt.
12	Agil anstrebende Organisationsform oder -struktur	Punkte, die auf eine organische (flexibel und teilnehmend, begünstigt, kooperative, soziale Interaktionen) Organisations-Form oder -Struktur schließen.	Interviewabschnitt, welcher die Definition erfüllt.
13	Traditionelle Steuerung	Punkte, die auf eine Prozess-zentrierte Steuerung schließen.	Interviewabschnitt, welcher die Definition erfüllt.
14	Agile Steuerung	Punkte, die auf eine Menschen-zentrierte Steuerung schließen.	Interviewabschnitt, welcher die Definition erfüllt.
15	Traditionelle Dokumentation	Punkte, die auf eine substantielle Dokumentation schließen.	Interviewabschnitt, welcher die Definition erfüllt.
16	Agile Dokumentation	Punkte, die auf eine minimale Dokumentation schließen.	Interviewabschnitt, welcher die Definition erfüllt.
17	Traditionelles Wissensmanagement	Punkte, die auf ein explizites (zwingende Dokumentation) Wissensmanagement schließen.	Interviewabschnitt, welcher die Definition erfüllt.
18	Agiles Wissensmanagement	Punkte, die auf ein implizites Wissensmanagement schließen.	Interviewabschnitt, welcher die Definition erfüllt.
19	Traditionelle Kommunikation	Punkte, die auf formale und nur notwendige Kommunikation schließen.	Interviewabschnitt, welcher die Definition erfüllt.
20	Agile Kommunikation	Punkte, die auf informelle und kontinuierliche Kommunikation schließen.	Interviewabschnitt, welcher die Definition erfüllt.
21	Traditionelle Teamnähe	Punkte, die auf verteilte Teams bzw. Teammitgliedern schließen lassen.	Interviewabschnitt, welcher die Definition erfüllt.

Kodierregeltabelle

Nr.	Kategorie	Definition	Kodierregel
22	Agile Teamnähe	Punkte, die darauf schließen, dass das Team sich in unmittelbarer Nähe verteilt, sodass z.B. die Meetings von Angesicht zu Angesicht ohne digitale Unterstützung möglich sind.	Interviewabschnitt, welcher die Definition erfüllt.
23	Traditionelle Teamgröße	Punkte, die auf Teams schließen, die größer als 10 Personen sind.	Interviewabschnitt, welcher die Definition erfüllt.
24	Agile Teamgröße	Punkte, die auf Teams schließen, die kleiner als 10 Personen sind.	Interviewabschnitt, welcher die Definition erfüllt.
25	Traditionelle Teambeteiligung	Punkte, die darauf schließen, dass die Teambeteiligung im Prozess nicht vorgeschrieben ist.	Interviewabschnitt, welcher die Definition erfüllt.
26	Agile Teambeteiligung	Punkte, die darauf schließen, dass die Teambeteiligung im Prozess als notwendig betrachtet wird und diese auch stattfindet.	Interviewabschnitt, welcher die Definition erfüllt.
27	Traditionelle Kundenbeteiligung	Punkte, die auf eine Kundenbeteiligung schließen, die normalerweise nur während der Projektanalyse als wichtig erachtet werden.	Interviewabschnitt, welcher die Definition erfüllt.
28	Agile Kundenbeteiligung	Punkte, die auf eine Kundenbeteiligung schließen, die als kritisch und kontinuierlich betrachtet werden.	Interviewabschnitt, welcher die Definition erfüllt.
29	Traditioneller Feedbackmechanismus	Punkte, die Feedback als nicht einfach beschaffbar deklarieren.	Interviewabschnitt, welcher die Definition erfüllt.
30	Agiler Feedbackmechanismus	Punkte, die Feedback als gewöhnlichen Bestandteil mehrerer Mechanismen vorsieht.	Interviewabschnitt, welcher die Definition erfüllt.
31	Traditionelles, kontinuierliches Lernen	Punkte, die das Lernen im Prozess als nicht regelmäßig einstufen.	Interviewabschnitt, welcher die Definition erfüllt.
32	Agiles, kontinuierliches Lernen	Punkte, die das Lernen im Prozess als festen Bestandteil sehen.	Interviewabschnitt, welcher die Definition erfüllt.

Bericht zum Einzelinterview 1.

Befragung - Person A - Unternehmen A

Alexander Kaack
alexander.kaack@haw-hamburg.de

Interviewdatum: 13.01.2016

1 Beschreibung des Berichts

Inhalt dieses Berichts ist die Neuauswertung eines Experteninterviews. Das durchgeführte Interview hatte dabei einen anderen Fokus, als der, der in dieser Untersuchung verfolgt wird. Hierzu werden die Audioaufzeichnungen erneut gehört und sukzessiv die getätigten Aussagen mittels der Kodierregeln untersucht. Die verwendeten Kodierregeln sind ebenfalls in diesem Bericht enthalten.

2 Rahmeninformationen

Folgend werden die Rahmeninformationen des Interviews in der Tabelle 1 dargestellt.

Tabelle 1: Rahmeninformationen

	Wert
Interview Nummer	1
Unternehmen	Unternehmen A
Entwicklungsart	Projektgeschäft
Teilnehmer	Person A
Geschlecht	Männlich
Alter	37
Ort	Cafe
Von	16:34
Bis	19:05
Dauer	151 Minuten

3 Anwenden der Kodierregeln

Folgend wurden die definierten Kodierregeln angewendet um entsprechende Stellen im Interview zu lokalisieren. Diese sind in der Tabelle 2 in zeitlicher Reihenfolge notiert worden. Die erste Spalte stellt dabei die Verknüpfung zu den Kodierregeln dar.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
2,30,32	03:36	[...] die Retro geleitet und solche Sachen.
2	04:24	Scrum selber mache ich seit [...]
9	06:00	Die Rolle des Entwicklers immer parallel.
18,20,22,30,32	06:44	Wir machen einmal im Monat immer [...] hinten dran gibt es dann immer bisschen Essen und gibt nen Bier. Sitzt man noch ein bisschen zusammen [...] kann über die Vorträge reden und sich technisch bisschen austauschen. [...] Ist so ein bisschen informell.
27,29	12:39	Ein klassisches Review hatten die gar nicht.
3,5	15:33	Das ist so eine Rahmenbedingung, auf die ich als Scrum Master überhaupt keinen Einfluss habe, weil das vor Projektbeginn schon feststeht.
25,28	16:09	Und da gibt es aber auch harte Meilensteine [...] fahren 2 Leute von uns zum Kunden und dann wird gemeinsam da drauf geguckt, auch ohne Entwickler.
9,17	19:12	Kunde XY, du bist der einzige der das Projekt kennt [...]

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
8	19:40	Ich versuch da durchaus auch mal rein zu -grätschen und zu sagen, Moment.
23	24:19	Es gab Teams bei [...] 6 Entwickler [...] es gab aber auch welcher die hatten über 10, 12, 14 Entwickler.
9,23	24:32	Das aktuelle Team sind im maximum 9 Entwickler gewesen plus 2 Studenten [...] plus nen PO [...] ich bin einer von den neun Entwicklern, Scrum Master und dann noch 3 Leute ausm DevOps Bereich die uns partiell unterstützt haben.
12,13,19	29:05	Es gibt Fällen in denen wir angehalten sind mit der Abteilungsleitung sozusagen die Erlaubnis einzuholen, dass wir die Ressource benutzen dürfen.
12,20	29:34	Wenn ich eine Frage habe [...] kannst du mir mal 2 Minuten was zeigen, das ist kein Thema.
7,11,13	29:57	Mit der Abteilungsleitung würde ich sprechen dann.
5,6	40:40	Die wollen einem dann nicht mehr Ressourcen geben, freiwillig.
18,20,22,26,30,32	43:03	Dabei hält man die Kommunikationswege recht kurz und diese gemeinsame Verantwortung führt halt auch dazu [...] Da sitzt dann ein Entwickler und ein Ops Mann (DevOps) vorm Rechner und die lösen gemeinsam das Problem. [...] Das du das Knowhow von dem Opsler und dem Entwickler gleichzeitig vor dem Rechner sitzen hast.
2	46:32	Mein erstes Scrumprojekt war, wo ich keine Erfahrung hatte.
3,5,6,13	48:00	Unser letzter Meilenstein lag hat so, dass er mitten in einem Sprint lag.
7,11	56:46	[...] angefangen eine Abteilungsleiterebene einzuführen [...]
8	57:23	Eine Kultur von [...] weiß ich nicht, weißt du besser, kannst du besser entscheiden, mach du, gibt.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
8,12,14,18,26,32	58:49	Wie können wir als Firma besser werden [...] Input von unten [...] was können wir machen, damit wir besser arbeiten können [...] das sich das Personal auch wohlfühlt.
11,13	01:02:00	Wir haben nicht genug Magnete, dann dauert das zwei bis drei Wochen bis die da sind.
8,10,12,14	01:03:42	[...] damit zusammen, dass bei uns und das ist auch in vielen anderen Teams so die ich kenne, jedes Team bestimmt, wie es selber arbeitet.
10,12,20	01:20:57	[...] losgeht zur Infra und sagt so hier wir brauchen nochmal [...]
3,5	01:24:38	Weil Scrum eigentlich [...] festen Meilensteinen nur bedingt passt. [...] wie können wir diesen Meilenstein halten.
16	01:27:58	Im Zweifelsfall, wenn es gar nicht anders geht irgendwo zu dokumentieren, auch wenn man das eigentlich gar nicht machen möchte, weil der Code soll ja die Dokumentation sein.
3,5,7,11	01:37:23	Wenn die Meilensteine vor Projekt stehen [...] sollen ja gehalten werden. Als Scrum Master kann ich da gar nicht viel machen. [...] Einen Meilenstein zu verschieben muss mit dem Kunden geklärt werden
8,10,22	01:56:42	Dann sind sie schnell auch schon selber am Start. [...] Frage: Selbst organisiert? ja [...]
14	01:58:21	[...] Für die Teamstimmung.
20	02:05:08	[...] Einfach mal [...] dazu gesetzt oder gestellt, sag mal was ist eigentlich los. [...]
22,24	02:05:20	Wir sitzen mit 10 Mann in einem Büro.
26,28	02:22:18	Mit dem Kunden gemeinsam im [...] arbeiten. [...] da werden User-Stories geschrieben und der Kunde kommentiert das.
20	02:28:31	Was Scrum tatsächlich tut ist, das Teammitglieder offener miteinander umgehen müssen. [...] Der Einzelne von sich mehr preisgeben muss.

4 Grafische Auswertung

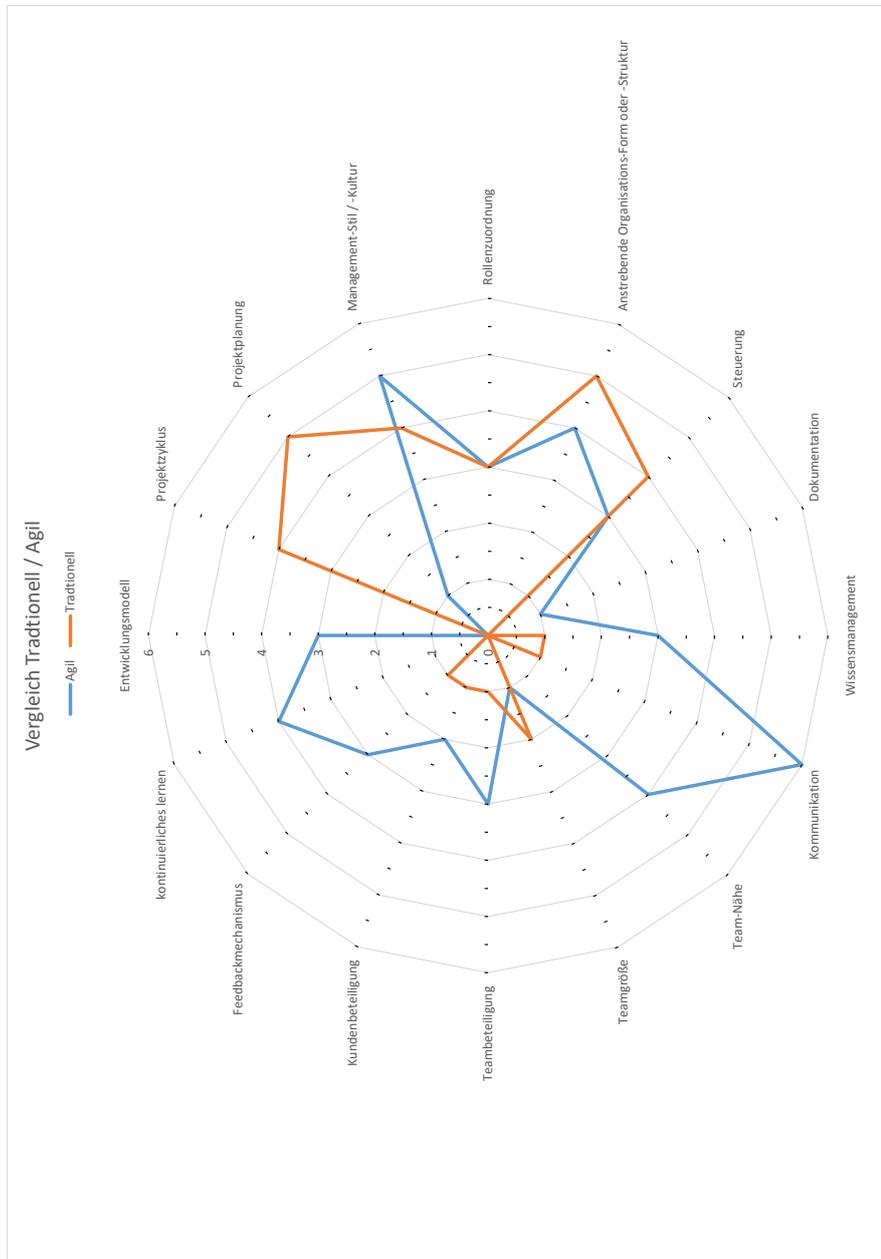


Abbildung 1: Vergleich der Indizien

Bericht zum Einzelinterview 2.

Befragung - Person B - Unternehmen B

Alexander Kaack
alexander.kaack@haw-hamburg.de

Interviewdatum: 13.01.2016

1 Beschreibung des Berichts

Inhalt dieses Berichts ist die Neuauswertung eines Experteninterviews. Das durchgeführte Interview hatte dabei einen anderen Fokus, als der, der in dieser Untersuchung verfolgt wird. Hierzu werden die Audioaufzeichnungen erneut gehört und sukzessiv die getätigten Aussagen mittels der Kodierregeln untersucht. Die verwendeten Kodierregeln sind ebenfalls in diesem Bericht enthalten.

2 Rahmeninformationen

Folgend werden die Rahmeninformationen des Interviews in der Tabelle 1 dargestellt.

Tabelle 1: Rahmeninformationen

	Wert
Interview Nummer	2
Unternehmen	Unternehmen B
Entwicklungsart	Interne Entwicklung
Teilnehmer	Person B
Geschlecht	Männlich
Alter	47
Ort	Meeting-Raum Unternehmen B
Von	08:41
Bis	10:30
Dauer	109 Minuten

3 Anwenden der Kodierregeln

Folgend wurden die definierten Kodierregeln angewendet um entsprechende Stellen im Interview zu lokalisieren. Diese sind in der Tabelle 2 in zeitlicher Reihenfolge notiert worden. Die erste Spalte stellt dabei die Verknüpfung zu den Kodierregeln dar.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
2	02:40	Wir machen halt Scrum, weil wir es propergiegen.
3	02:32	Wir haben aber natürlich keine shipping build Inkrements
3	02:37	Wir erledigen Aufgaben
7,9	03:48	Wo ich auch als Scrum Master nominiert, eingeteilt bin
7,9	05:54	Wir wechseln ja ständig Scrum Master, mach du das jetzt mal
9	07:03	Fulltime Scrum Master ist nicht vorgesehen.
7,25	14:00, 14:35	Scrum of Scrum: Das sind dann eher die PO's die das machen. - Es heißt Scrum of Scrum, aber das ist es nicht
7,9,25	14:52	Bei uns sind die PO's auch gleichzeitig die Projektleiter
9,17	15:37	Viele sind nicht ersetzbar und der ist definitiv nicht ersetzbar
24	17:02	Wir haben 1 PO ... 1 Scrum Master ... 7 Entwickler

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
21,25	50:40	... Weil die Dortmunder tendenziell vernachlässigt werden, weil sie weniger sind und das meiste halt in Hamburg passiert.
29,31	21:41	Die Scrum Master hier im Unternehmen mal zusammenzuholen ... Eigentlich ist es auch Aufgabe des Scrum Master sich mit anderen Scrum Masters ... auszutauschen
32	22:20	Scrum nach dem Scrum Guide ... ich glaub es macht wirklich keiner. Es ist immer die Frage von welcher Seite man kommt. ... Ich glaube man kann Sachen modifizieren , wenn man es einmal richtig gemacht hat. Wenn man es kann.
11,13	28:20	Hat das Verhalten einer Behörde
13,19	29:42	E-Mail braucht man immer als Nachweis
11,13,19	30:45	Standardmäßig gibt es für alles irgendein Prozess ... Es ist durchformalisiert
5	33:53	Gib mir was im Projektmanagement-Tool, damit ich die Zeiten buchen kann. Die sind ja auch für das Jahr verplant. Da muss dann auch wirklich vorher sagen, wir brauchen nächstes Jahr einen mit dem und dem Knowhow.
7,11	44:17	Wenn in der Retro aufkommt, das und das stört, dann sortieren wir immer die Sachen aus, die außerhalb des Teams liegen, die wir sowieso nicht ändern können.
7,11	47:50	Impediments, die in der Unternehmenskultur liegen ... die schon gar nicht mehr erwähnt werden.
9	50:07	Letztendlich fühle ich mich als Scrum Master meistens als einer der die Räume bucht und die Einladungen verschickt für die Meetings.
29	56:24	Wie sammelt man als Scrum Master Feedback ein
11	1:02:53	Das die Firma nicht ein junges dynamisches Startup ist ... ist ein mega Impediment und stört auch mehr als fehlende Videokonferenzen
19,21	1:24:40	Was würde dir als Scrum Master helfen ... beamen .. Personen ... in einen Raum bringen

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
21	1:25:58	Mehr Video (Präsenz) wäre besser ... Meetings ... optisch untermalen kann
19	1:27:45	Präsenz und Kommunikation is everything
19,25,27	1:35:58	DevOps ... um Sachen klarer zu kriegen ... Wir haben einen großen Gap zwischen den Entwicklern / Anwendungsentwicklung und der EDV ... lauter Baustellen
32	1:33:57	Vorstand ... haben halt erkannt .. das man die Firma drehen muss .. um auf die ganze Digitalisierungsgeschichte auch vorbereitet zu sein

4 Ergebnisse

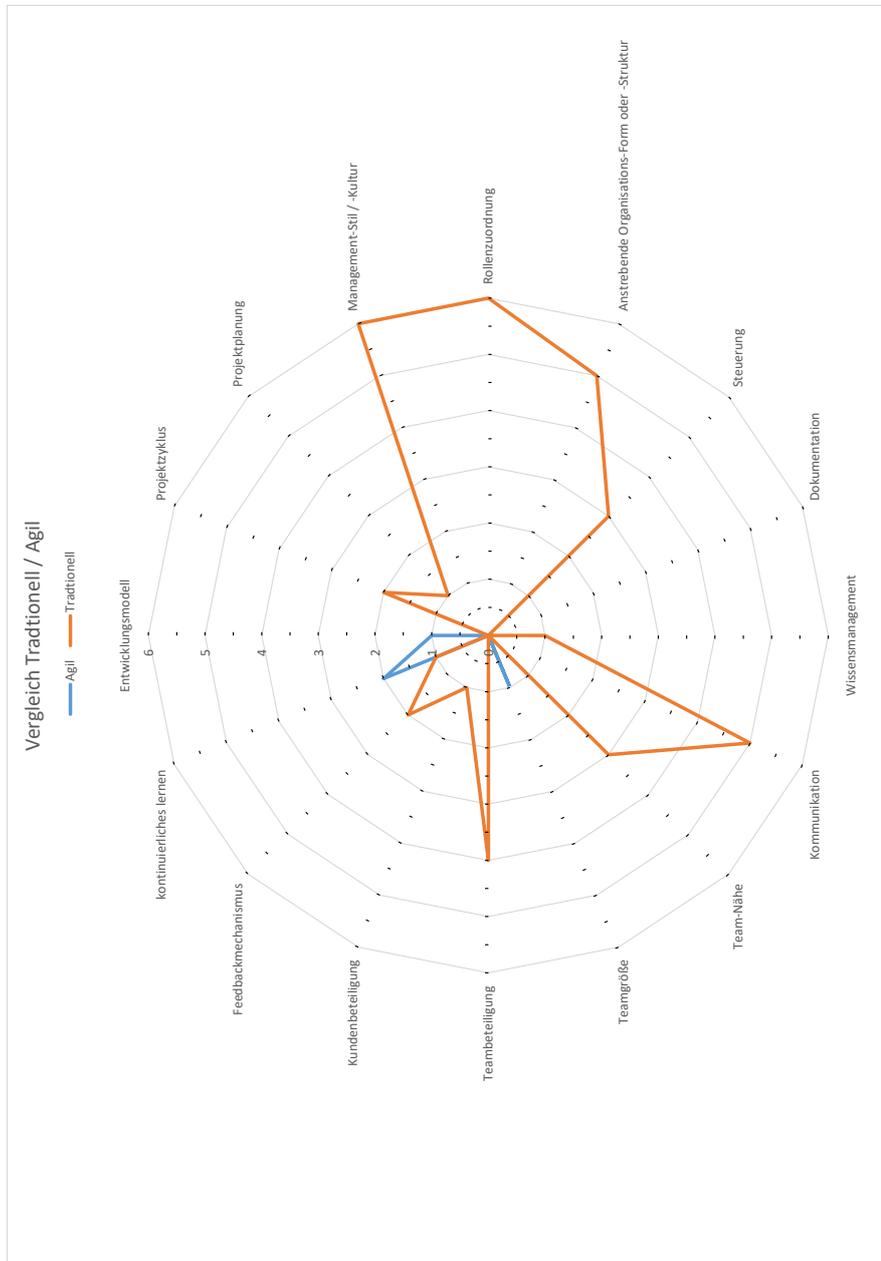


Abbildung 1: Vergleich der Indizien

Bericht zum Einzelinterview 4.

Befragung - Person D - Unternehmen C

Alexander Kaack
alexander.kaack@haw-hamburg.de

Interviewdatum: 26.01.2016

1 Beschreibung des Berichts

Inhalt dieses Berichts ist die Neuauswertung eines Experteninterviews. Das durchgeführte Interview hatte dabei einen anderen Fokus, als der, der in dieser Untersuchung verfolgt wird. Hierzu werden die Audioaufzeichnungen erneut gehört und sukzessiv die getätigten Aussagen mittels der Kodierregeln untersucht. Die verwendeten Kodierregeln sind ebenfalls in diesem Bericht enthalten.

2 Rahmeninformationen

Folgend werden die Rahmeninformationen des Interviews in der Tabelle 1 dargestellt.

Tabelle 1: Rahmeninformationen

	Wert
Interview Nummer	4
Unternehmen	Unternehmen C
Entwicklungsart	Interne Entwicklung
Teilnehmer	Person D
Geschlecht	Weiblich
Alter	28
Ort	Meeting-Raum Unternehmen C
Von	15:30
Bis	17:05
Dauer	95 Minuten

3 Anwenden der Kodierregeln

Folgend wurden die definierten Kodierregeln angewendet um entsprechende Stellen im Interview zu lokalisieren. Diese sind in der Tabelle 2 in zeitlicher Reihenfolge notiert worden. Die erste Spalte stellt dabei die Verknüpfung zu den Kodierregeln dar.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
9,11	06:33	Ich bin seit [...] Jahren bei [...] in der Rolle des Projektleiters, wo man ja diese hybride Rolle hat.
2,30,32	07:40	Design Thinking als Discovery Methode
7,9,11	08:00	Als Scrum Master oder Agile Coach [...] das ist eine Rolle die es selten schafft 100 % gelebt zu werden, auf Grund der Umgebung [...] Du hast nicht nur die Rolle des Projektmanagers
4	14:42	Man sollte in Themen denken [...] du gehst über Funktionalität
3,5	15:14	Okay mein Geld ist alle, aber ich habe die Funktionalität noch gar nicht. Okay, kriegst du mehr Geld.
2	30:40	Ein Service Team das mit Kanban arbeitet und ein Produktteam was mit Scrum arbeitet.
14	33:40	Hauptsache du kriegst die Menschen zum arbeiten und sie sind glücklich dabei.
23	34:28	Pro Team sind das 15 Personen
11,13	36:12	Ich brauche Hardware, also Bestellprozesse [...] Beschaffungsprozesse

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
32	37:04	Alles was mit Lernen ist freies Gut [...] da sparen sie nirgends [...] Weiterbildungen / Konferenzen / Zertifikate
20	38:02	Im Team habe ich die Kommunikation auch immer als sehr ehrlich und direkt empfunden.
8,14	39:22	Sprint-Schutz nehmen wir sehr ernst.
11,19	43:00	Das klassische Hierarchieproblem, du hast keine Ahnung was da ankommt.
7,11,19,25	43:45	Protokolle bekommt nur die bestimmte Hierarchie.
7,11,17	45:05	Jeder hat schon sein Königreich
11,19	46:40	Wir haben schon eine Kultur von Unterlagen nicht weitergeben
3,5,7	56:44	[...] schafft die Schnittstelle zum klassischen Projektmanagement
26,28,30	59:50	Wir gehen auch mit den Stakeholdern durch den Prozess
8,22	01:09:30	Müssen im Minutentakt zusammen sein [...] zusammenarbeiten.
14	01:12:37	Es sind schließlich Menschen.
23	01:42:30	Mein Team war 3 Leute kleiner. Also 12
7,9,11,13,25	01:15:22	Ein Denkfehler was Sie hier haben ist, je mehr Menschen du rein wirfst um so schneller wirst du.
7,9,11,13	01:16:50	Durch Politik keine Eigenverantwortung
22	01:20:15	Wir hatten auch einen der [...] gearbeitet hat
26,32	01:22:41	Design Thinking ganz groß jetzt.
30,32	01:24:30	Haben wir gelernt und hatten eine witzige Zeit und dann geht man wieder ins Daily Business
12,32	01:26:28	Offen für Änderungen und für Lernen zu sein
8,12,14,20,26	01:31:44	Alle möglichst regelmäßig [...] Was sind deine Erwartungen [...] wo Fehlt dir das gerade noch [...] der Ansprechpartner zu sein.

4 Ergebnisse

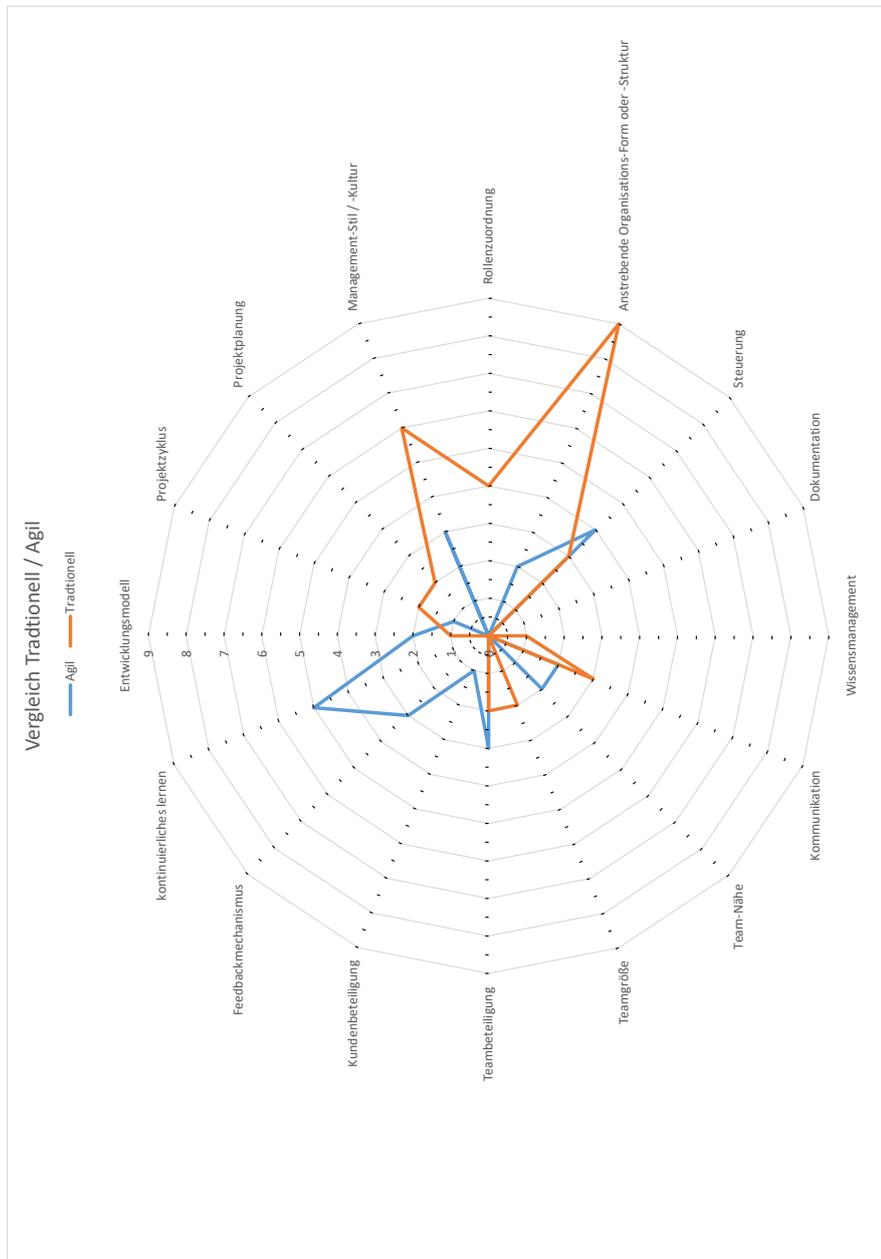


Abbildung 1: Vergleich der Indizien

Bericht zum Einzelinterview 5.

Befragung - Person E - Unternehmen D

Alexander Kaack
alexander.kaack@haw-hamburg.de

Interviewdatum: 01.03.2016

1 Beschreibung des Berichts

Inhalt dieses Berichts ist die Neuauswertung eines Experteninterviews. Das durchgeführte Interview hatte dabei einen anderen Fokus, als der, der in dieser Untersuchung verfolgt wird. Hierzu werden die Audioaufzeichnungen erneut gehört und sukzessiv die getätigten Aussagen mittels der Kodierregeln untersucht. Die verwendeten Kodierregeln sind ebenfalls in diesem Bericht enthalten.

2 Rahmeninformationen

Folgend werden die Rahmeninformationen des Interviews in der Tabelle 1 dargestellt.

Tabelle 1: Rahmeninformationen

	Wert
Interview Nummer	5
Unternehmen	Unternehmen D
Entwicklungsart	Produktgeschäft
Teilnehmer	Person E
Geschlecht	Männlich
Alter	45
Ort	Meeting-Raum Unternehmen D
Von	11:47
Bis	13:03
Dauer	76 Minuten

3 Anwenden der Kodierregeln

Folgend wurden die definierten Kodierregeln angewendet um entsprechende Stellen im Interview zu lokalisieren. Diese sind in der Tabelle 2 in zeitlicher Reihenfolge notiert worden. Die erste Spalte stellt dabei die Verknüpfung zu den Kodierregeln dar.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
2,6	03:45	... agile Kontext unterwegs sind ... es geht darum iterativ vorzugehen.
12,14	05:00	Wert Nr. 1 ist ja: People und Interactions over Processes and Tools
5	04:25	... mit einem klassischen Projektmanagementansatz ... das Was .. wie wollen wir das Ziel erreichen haben wir alle Mittel, alle Voraussetzungen...
8,12	07:45	Teamlead ... sind aber nicht Teil des Developmentteams ... die unterstützen das Team, wenn es z.B. um Impedements geht.
10	08:34	Du willst ein selbstorganisiertes möglichst autonomes, autarkes, selbst steuerndes Team haben.
4,10,14	19:42	... die Teams auch inhaltlich so zu schneiden das die autonom und autark auf ihrem Produkt arbeiten können
8,12	20:25	... ist halt einer von diesen agilen Projektmanagern dabei, um zu sehen, dass es halt funktioniert.
24	20:50	.. sieben plus minus zwei Leute

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
10,14,20,22,26	20:55	Klein genug um, eine enge Zusammenarbeit zu haben. Das die Leute zum eingeschworenen Team werden, um auch auf kurzen Dienstwegen und auf Zuruf reagieren zu können.
12	23:04	Die Strukturen so zu halten, das du kurze Wege hast.
8,12,20	23:30	Mit mehr Leuten wird es schwieriger die Kommunikation aufrecht zu erhalten. Da ist dann der Mehrwert eines agilen Projektmanagers wieder da, der sich dann darum kümmert, dass die Kommunikation auch läuft. Oder auch erkennen kann, wann ist erforderlich mit wem zu sprechen.
5,7,13	23:45	Ab einer bestimmten Größe macht das Thema Projektmanagement dann immer mehr Sinn.
7	29:57	Weil das Management plötzlich sagt, okay, so habe ich mir das aber nicht vorgestellt.
5,18,27	30:07	Wir haben Aufwand, mehr und mehr in die Initialisierung gesteckt um zu gucken, das wir im Vorhinein alle das gleiche Zielverständnis haben, inkl. Vorstand, wenn es um größere Projekte geht.
8,10,12	31:30	Das das Team, wenn es einmal das Ziel genannt bekommen hat, dann selbst organisiert das Ziel erreichen kann. Möglichst autonom und autark.
2,12,26	32:03	Lass uns auf diesen Prozess iterieren und bindet uns ein.
28	32:40	Stakeholdermanagement. Wenn es zu viele Stakeholder gibt die da mitkochen. ... Dann fängt das ganze an, nicht mehr so effizient abzulaufen.
8,12	33:30	Du hilfst als Kommunikator, als Riskmanager, wie auch immer, das die Impediments nicht zu großen Schaden im Team anrichten.
20	34:03	... deine interne Vernetzung in der Organisation
8	37:20	Die Bereitschaft in Konflikte zu gehen.
8,10	37:58	Das ist ein Problem, dass zu viel Verantwortung in diese Rolle des Product Owners gelegt wird.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
32	39:08	Es gibt immer irgendwelche Experimente die man machen kann und die man auch machen soll. Man soll sich nicht sklavisch an irgendwelche agilen Mechaniken halten. ... Wie kann ich das für mich am sinnvollsten adaptieren um es so zu machen, dass es Nutzbringend ist.
2	39:48	Scrum ist eher ein revolutionärer Prozess. Das heißt, wenn du in eine Firma kommst, die damit bisher noch nichts am Hut hatte und du kommst dann mit Scrum rein, dann kann das sein, dass du dadurch sehr viel Staub aufwirbelst, dass du sehr viel Widerstände erhebst, weil es schon eine grundlegende Umstellung ist.
2	40:11	Wogegen Kanban eher ein einfacheres Mittel ist, weil es respektiert die Situation vor Ort, es respektiert die gegebenen Rollen, die da vorherrschen, tut nicht gleich so weh und legt halt Wert darauf, wo sind wir hier gerade und was ist der größte "Pain-Point" und wie können wir den zusammen lösen. Deswegen ist das die smartere Variante um Agilität in ein Unternehmen zu bringen, was mit Agilität bisher nicht so viel hatte.
20	47:05	Das kann in der Kaffeeküche passieren, das dir da jemand aus dem Team sagt, passe mal auf, hier und hier.
6,8,14,30	47:36	Das ist ein agiler Health-Check. ... einmal im Quartal ... mit 5 ... Fragen, wie denn der ... "Gesundheitszustand" des Teams und dessen Projekt ist. ... Sind wir inhaltlich an den richtigen Dingen dran ... das "Was"... Machen wir das, was wir machen richtig ... sind wir schnell genug ... kriegen wir genügend Support, den wir brauchen ... wie ist die Motivation im Team
12	50:36	Nicht alle Unternehmensteile sind agil ... das kann ein nerviger Prozess sein.

Tabelle 2: Erfasste Interviewstellen

Kodierregel-Nr.	Zeit	Zitat
8,12,14,22	01:04:03	Es ist okay Tools zu haben ... Es ist immer wichtiger mit den Menschen irgendwie zu arbeiten und zu gucken was haben die für Interaktionen untereinander. ... Das die Teams zusammenarbeiten.
20	01:04:55	Das du Face to Face Kommunikation ermöglichst.
8,12,18,20,26,28,30	01:07:07	Du hast Leute aus dem Business Bereich, du hast Leute aus dem Tech Bereich, die sprechen nicht immer die selbe Sprache. Du als Scrum Master bist auch dabei diese Brücke zu schließen ... Die Leute lieber zusammen zu haben, um über Dinge reden zu können... Face to Face, physisch vor Ort.

4 Ergebnisse

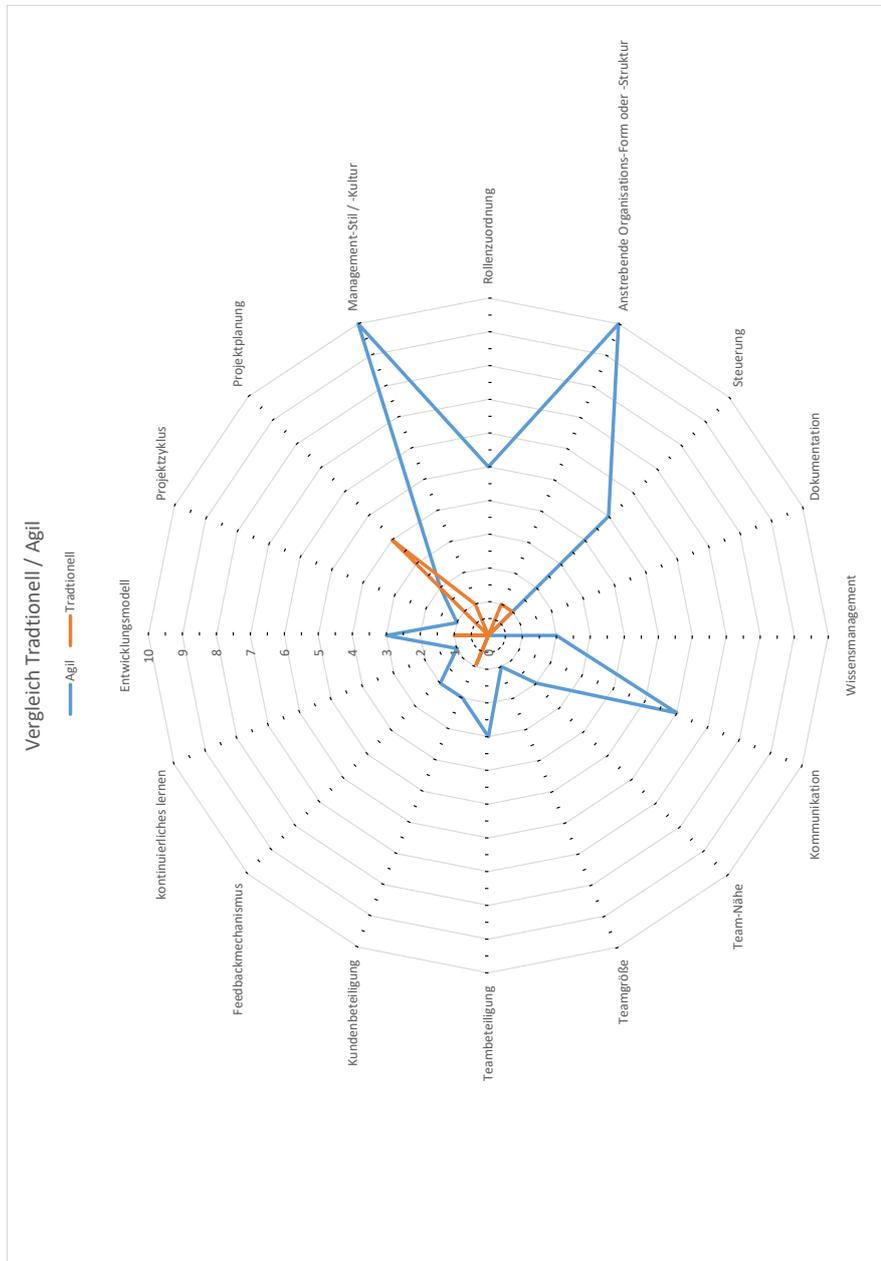


Abbildung 1: Vergleich der Indizien

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 06.06.2017

Alexander Kaack