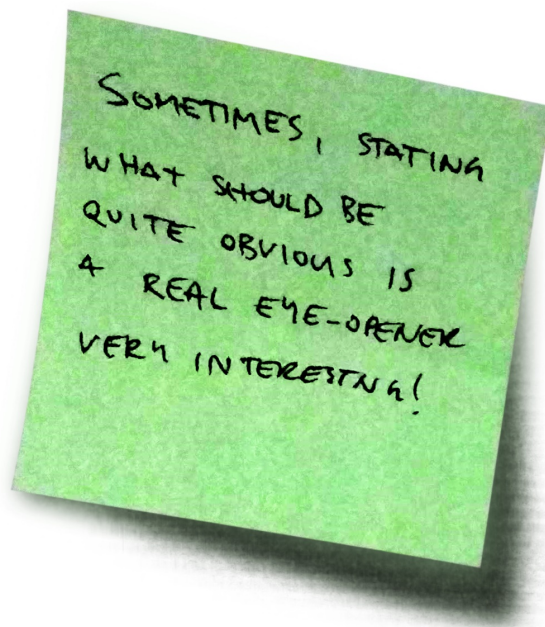


Ein Modell zum Konzept **KLARHEIT GEWINNEN** und  
dessen Ursachen und Auswirkungen auf die  
Zusammenarbeit in selbstorganisierten  
Softwareentwicklungsteams



vorgelegt von  
Helena Barke

**Dissertation**  
zur Erlangung des Grades eines Doktors der  
Naturwissenschaften  
am Fachbereich Mathematik und Informatik  
der Freien Universität Berlin

Berlin 2020

Erstgutachter: Prof. Dr. Lutz Prechelt  
Zweitgutachterin: Prof. Dr. Juliane Siegeris  
Disputation am : 03.07.2020

## Zusammenfassung

**Hintergrund:** Agile Softwareentwicklungsteams setzen Scrum unterschiedlich und individuell in der Praxis um. Anleitungen, Leitfäden und Handreichungen aus der Praxis und Forschung verlieren sich in den Details einzelner Werkzeuge, Faktoren oder Teile.

**Forschungsfrage:** Was hält selbstorganisierte Softwareentwicklungsteams zusammen und fördert eine gute Zusammenarbeit?

**Methode:** Es wurde mit Grounded-Theory-Methodologie nach Charmaz geforscht. Es wurden in 5 Scrum-Teams aus der Praxis intensiv Daten erhoben und weitere Interviews und Validierungen mit Expert\_innen durchgeführt.

**Ergebnisse:** Es wurde ein Modell entwickelt, das als zentrales Konzept **KLARHEIT GEWINNEN** enthält. Erstmals werden bekannte und neue Erkenntnisse zu einem gemeinsamen Modell verbunden und erklären Grundlagen funktionierender Zusammenarbeit in agilen Softwareentwicklungsteams.

**Fazit:** In der Validierung wird deutlich, dass agile Teams das Modell anwenden können, um ihre Zusammenarbeit zu analysieren und zu stärken.



## **Selbstständigkeitserklärung**

Name: Barke

Vorname: Helena

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

12. August 2020

Helena Barke



## Widmung

Für alle Informatikerinnen und Frauen.  
Insbesondere:

Juliane	Edna	Sünne
Dagmar	Kristina	Melanie
Elke	Natalia	Tine
Julia	Julia	Sophia
Wiebke	Marianna	Kerstin
Rashina	Shanti	Saskia
Evi	Gaby	Corinna
Nicola	Andrea	Andrea
Teresa	Tamara	Cati
Diana	u. v. m.	

Danke, mit Euch war es leicht!





## Danksagung

Mein herzlicher Dank gilt den vielen Menschen, die mich unterstützt haben. Um Einige zu nennen:

Lutz Prechelt, der stets ansprechbar war, mich motiviert und unterstützt hat.

Juliane Siegeris, die Rollenvorbild, Motivatorin und Ratgeberin in einer Person ist.

Allen Kolleg\_innen an der HTW Berlin und FU Berlin.

Allen Teams und Expert\_innen, die die Forschung unterstützt haben.

Edna, für die unglaublichen Ideen und Anregungen, von denen diese Arbeit profitiert hat.

Gaby, Stephan und Florian für ihre Zeit und die Freude am Lesen.

Andre, für deine Geduld, für deine Unterstützung, für alle unsere Projekte, die unser Leben schöner machen.

Marlies, für deine eigene Sprache der Liebe.

Den Frauenfördermaßnahmen wie dem Professorinnenprogramm und dem Berliner Chancengleichheitsprogramm, die mir die Finanzierung ermöglichten, mir Ziele aufzeigten, mir halfen und somit die defizitären Strukturen ein wenig verbessern.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>15</b>
<b>Tabellenverzeichnis</b>	<b>17</b>
<b>1 Einleitung</b>	<b>19</b>
1.1 Struktur der Arbeit . . . . .	22
1.2 Scrum . . . . .	22
1.2.1 Scrum-Team . . . . .	23
1.2.2 Iterativer Ablauf in Scrum . . . . .	23
1.2.3 Scrum-Artefakte . . . . .	25
1.3 Anmerkungen zum Scrum-Guide . . . . .	26
<b>2 Verwandte Arbeiten</b>	<b>29</b>
2.1 Relevanz menschlicher Aspekte in der Softwareentwicklung . . . . .	30
2.1.1 Menschliche Faktoren im agilen Umfeld . . . . .	30
2.2 Angrenzende Themen, die <i>nicht</i> betrachtet werden . . . . .	31
2.2.1 Kommunikation . . . . .	31
2.2.2 Agile Transformation in der Organisation . . . . .	32
2.2.3 Individuen reduziert auf Persönlichkeit . . . . .	33
2.3 Arbeiten, die sich <i>inhaltlich</i> mit Teams befassen . . . . .	34
2.3.1 Einflussfaktoren auf die Leistung selbstorganisierter Teams . . . . .	35
2.3.2 Hackman: selbstorganisierte Teams . . . . .	41
2.3.2.1 Relevanz und Einordnung . . . . .	41
2.3.2.2 Hackmans Forschungsergebnisse als Einführung in selbst- organisierte Teams . . . . .	41
Exkurs 1: Hackman in einer agilen Welt . . . . .	49
Exkurs 2: Warum Hackman nicht nur einfach validieren? Was ist der Mehrwert einer Grounded-Theory (GT)? . . . . .	50
2.3.3 Motivation . . . . .	51
2.3.4 Reflexion . . . . .	53
2.3.5 Hodas Grounded-Theory zu agilen Softwareentwicklungsteams . . . . .	55
2.3.6 Forschungsgruppe um Nils Brede Moe, Torgeir Dingsøyrr und Tore Dybå . . . . .	58
2.3.7 Verwandte Arbeiten aus entfernten Forschungsrichtungen . . . . .	60
2.3.7.1 Anthropologie: Teams in extremen Umgebungen . . . . .	60
2.3.7.2 Heintel: dynamische Prozesse in Gruppen . . . . .	62

<b>3</b>	<b>Anspruch der Arbeit: Grenzen, Lücken und Anknüpfungspunkte</b>	<b>65</b>
3.1	Forschungsbeitrag	67
<b>4</b>	<b>Methode</b>	<b>71</b>
4.1	Auswahl der Methode	71
4.2	Constructing Grounded-Theory nach Charmaz	73
4.2.1	Charmazs epistemologische Perspektive	74
4.2.2	Reflexion der Forschenden	75
4.2.3	Literaturreview	75
4.2.4	Forschungsfrage	77
4.2.5	Datenerhebung	77
4.2.6	Codieren	79
4.2.7	Memos schreiben	81
4.2.8	Paradigmatisches Modell	81
4.2.9	Evaluationskriterien	83
<b>5</b>	<b>Datenerhebung</b>	<b>85</b>
5.1	Zugang zum Feld	85
5.1.1	Theoretisches Sampling	86
5.1.2	Interviewtechnik	87
5.1.3	Reflexion der Forschenden	88
5.2	Daten	90
<b>6</b>	<b>Anwendung der Methode</b>	<b>93</b>
6.1	Begriffe: ihre Verwendung und Notation	93
6.2	Codieren	96
6.2.1	Initiales Codieren: Wort für Wort	97
6.2.2	Initiales Codieren: Zeile-für-Zeile und Ereignis-für-Ereignis	99
6.2.3	Fokussiertes und theoretisches Codieren	102
6.2.4	Memos	103
6.3	Anmerkung zur späteren Darstellung der methodischen Herleitung der Ergebnisse	104
<b>7</b>	<b>Ergebnisse</b>	<b>107</b>
7.1	Methodische Herleitung	107
7.1.1	Detaillierter Codierprozess des Konzeptes <b>SELBSTREFLEXION</b>	108
7.1.2	Detaillierter Codierprozess des Konzeptes <b>KLARHEIT GEWINNEN</b> und die Herleitung des Modells	111
7.2	Modell zur <b>GEWINNUNG VON KLARHEIT</b>	116
7.2.1	Ausgangspunkt: Selbstorganisiertes Team	121
7.2.2	1. Prozessschritt zur <b>GEWINNUNG VON KLARHEIT</b> : Lokales Rollenverständnis durch Selbstreflexion	122
7.2.3	2. Prozessschritt zur <b>GEWINNUNG VON KLARHEIT</b> : globales Rollenverständnis durch Teamreflexion	126
7.2.4	Rollen	131
7.2.5	Zusammenarbeit	133
7.2.6	Konsequenz: Teambestand	134

7.2.7	Monitoring und Feedback . . . . .	135
7.3	Ergebnis: Hypothesen . . . . .	136
7.4	Diskussion der Ergebnisse und des Forschungsbeitrages . . . . .	137
7.4.1	Ausgangspunkt: Selbstorganisierte Teams . . . . .	138
7.4.2	<b>KLARHEIT GEWINNEN</b> . . . . .	140
7.4.3	Rollen . . . . .	141
7.4.3.1	Selbstreflexion . . . . .	142
7.4.3.2	Teamreflexion . . . . .	143
7.4.4	Zusammenarbeit . . . . .	145
7.4.5	Konsequenz: Teambestand . . . . .	145
7.4.6	Diskussion der verwandten Arbeiten zu den Hypothesen aus Abschnitt 7.3 . . . . .	146
7.4.6.1	Hypothese 1: Selbstreflexion und proaktives Handeln ist notwendig, um <b>KLARHEIT</b> zu gewinnen und heraus- fordernd für die Teammitglieder in agilen Softwareent- wicklungsteams. . . . .	146
7.4.6.2	Hypothese 2: <b>KLARHEIT</b> auf eigener <i>und</i> auf Teamebene fördert eine positive Zusammenarbeit. . . . .	146
7.4.6.3	Hypothese 3: Gegenseitiger Konsens über die Fähig- keiten der Teammitglieder ist wichtiger als die Fähig- keiten selbst. . . . .	146
7.4.6.4	Hypothese 4: Scrum-Master und andere Teammitglie- der können unterstützen <b>KLARHEIT ZU GEWINNEN</b> . . . . .	147
7.4.6.5	Hypothese 5: Trotz aller Unterstützung kann der Pro- zess zu jedem Zeitpunkt scheitern insbesondere an per- sönlichen Sympathien und Emotionen. . . . .	147
<b>8</b>	<b>Validierung</b> . . . . .	<b>149</b>
8.1	Validierung der Ergebnisse während des Forschungsprozesses . . . . .	150
8.1.1	Validierung in $t_2$ . . . . .	150
8.1.2	Validierung in $t_3$ . . . . .	152
8.1.3	Validierung in $t_4$ . . . . .	152
8.1.4	Validierung in $t_5$ . . . . .	154
8.2	Validierung der finalen Ergebnisse . . . . .	154
8.3	Relevanz der Ergebnisse für die Praxis . . . . .	159
8.4	Hilfreiche Werkzeuge . . . . .	161
<b>9</b>	<b>Evaluierung und Grenzen des Modells</b> . . . . .	<b>165</b>
<b>10</b>	<b>Fazit und Ausblick</b> . . . . .	<b>169</b>
	<b>Literaturverzeichnis</b> . . . . .	<b>173</b>
<b>A</b>	<b>Interviewleitfaden</b> . . . . .	<b>187</b>



# Abbildungsverzeichnis

1.1	Der iterative Ablauf in Scrum . . . . .	24
2.1	Definition selbstorganisierter Teams (zweite Spalte von links: Self-managing Unit) entnommen aus Hackman [Hac86, S. 92] . . . . .	42
2.2	Prozess-Kriterien selbstorganisierter Teams entnommen aus Hackman [Hac86, S.110] . . . . .	45
2.3	Reflexionsmöglichkeiten im Scrum Prozess, entnommen aus Babb u.a. [BHN14] . . . . .	54
4.1	Vereinfachte Darstellung der methodischen Schritte der GTM nach Char-maz. . . . .	74
4.2	Abgrenzung objektivistischer GTM zu konstruktivistischer GTM, ent-nommen aus [Cha14, S.236] . . . . .	76
4.3	Paradigmatisches Modell in der Darstellung von Salinger, entnommen aus [Sal13, S.67] . . . . .	82
6.1	Eine vereinfachte Darstellung der Begriffe und ihre Entwicklung im Forschungsprozess . . . . .	96
6.2	Screenshot aus MAXQDA, Ausschnitt aus dem Dokumentenstruktur-baum . . . . .	100
6.3	Screenshot aus MAXQDA, Ausschnitt aus dem Codestrukturbaum . . . . .	101
6.4	Screenshot aus MAXQDA, Übersicht Memos, anonymisiert . . . . .	104
7.1	Grafische Darstellung der ersten methodischen Herleitungsschritte des Konzeptes SELBSTREFLEXION . . . . .	109
7.2	Grafische Darstellung der ersten methodischen Herleitungsschritte des Konzeptes TEAMFUNKTIONALITÄT . . . . .	112
7.3	Vereinfachte, retrospektive, grafische Darstellung des Forschungspro-zesses vom Konzept TEAMFUNKTIONALITÄT zum Modell. Erläuterungen siehe Abschnitt 7.1.2. . . . .	113
7.4	Ergebnis: Modell, um KLARHEIT ZU GEWINNEN und dessen Ausgangs-punkt und Konsequenzen in agilen Softwareentwicklungsteams. Eine Einführung zum Modell siehe Seite 118 und die nachfolgenden Ab-schnitte 7.2.1 bis 7.2.7. . . . .	117
7.5	Foto des Protokolls der Retrospektive von t3 am 06.02.2017: Maßnah-men zur Einbindung von t3dev3. (Namen anonymisiert und durch Syn-onyme ersetzt) . . . . .	127

8.1	Digitales Board der Retrospektive von t2 am 19.10.2017 zur Validierung der Forschungszwischenergebnisse . . . . .	153
8.2	Diskussionsergebnisse der Expertinnenrunde zum Modell am 24.08.2018	156
8.3	Reduziertes Modell aus der Präsentation zum Meet-up am 28.05.2019 .	157
8.4	Bilder des Meet-ups zur Validierung am 28.05.2019 . . . . .	158
8.5	Feedback nach dem Meet-up am 28.05.2018, leider in schlechter Bildqualität. Hier eine Auswahl von links nach rechts: „Great Workshop Competent!“ (ganz links); „Die Beispiele sind super!“ (u.l.); „großartiger Inhalt“ (grüner Klebezettel ganz unten); „Good discussion during exercise“ (gelber Zettel ganz unten). Einige Teilnehmenden haben ihre Email-Adresse hinterlassen, sie wurden anonymisiert. . . . .	160
8.6	Anonymisierte Beispiele von Teamzeichnungen aus Interviews mit jeweils unterschiedlichen strukturgebenden Elementen: o.r. funktionale Rolle, u.l. Introvertiert/extrovertiert und learning/designing, u.r. Sitzordnung . . . . .	163
A.1	Interviewleitfaden . . . . .	188
A.2	Zusätzliche Fragen für agile Expert_innen . . . . .	189
A.3	Datenschutzerklärung . . . . .	190
A.4	Datenblatt Interviewte . . . . .	191



# Tabellenverzeichnis

4.1	„Some of the key differences between the three main strands of grounded theory“, entnommen aus [SRF16], die Bezeichnungen der Strömungen wurden zur Konsistenz um die Namen Corbin und Charmaz ergänzt bzw. ersetzt. . . . .	72
5.1	Statistik zur Datenerhebung . . . . .	91
8.1	Validierungen der Forschungsergebnisse . . . . .	151



# Kapitel 1

## Einleitung

Es gibt eine Grundfeste in der Technik und im Denken von Informatiker\_innen<sup>1</sup>: Eins und Null. Zumindest ist das, das Ergebnis der Genderforschung von Wendy Faulkner [Fau00]. Sie nennt es Dichotomien. Ein Beispiel für dichotomisches Denken ist die Annahme, dass ein Mensch sich entweder mehr für Menschen *oder* für Technik interessieren kann. Frauen wird zugeschrieben, sich mehr für Menschen zu interessieren und sind daher Vorurteilen<sup>2</sup> ausgesetzt, wenn sie sich in einem technischen Umfeld bewegen. Umso bemerkenswerter ist die erste Aussage im agilen Manifest, das als Grundlage für die agile Softwareentwicklung gilt. Prioritäten werden folgendermaßen gesetzt:

Individuen und Interaktionen mehr als Prozesse und Werkzeuge [BBv<sup>+</sup>01]

Mit dieser Aussage wird das dichotomische Denken und die daraus resultierenden Vorurteile gleich mehrfach durchbrochen. Zum einen werden *beide* Seiten wertgeschätzt durch das „mehr“ und wie es weiter unten im Manifest heißt :

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein. [BBv<sup>+</sup>01]

Zum anderen werden Menschen und ihr Handeln als wichtiger bewertet als Technik und Prozesse. Aus dem Ingenieurwesen und der Genderforschung kommend, löste das agile Manifest ein außerordentliches Interesse an agilen Vorgehensweisen bei mir aus. Ich erforschte nach dieser ersten Erkenntnis Gender, IT und agilen Prozesse aus einer theoretischen Perspektive. Daraus entstand eine Tagung mit 79 Forscher\_innen und Teilnehmer\_innen und eine Sammelpublikation zu diesem Thema

---

<sup>1</sup>In der vorliegenden Arbeit wird als gendergerechte Schreibweise der Unterstrich gewählt. Zum einen verdeutlicht der Unterstrich, dass immer noch eine Lücke in den Teilhabe- und Machtverhältnissen der Geschlechter besteht z.B. im Gehalt oder der Anzahl von Frauen in Vorständen. Zum anderen beinhaltet der Unterstrich ein Kontinuum von Geschlechtern, das wiederum das dichotomische Denken in zwei Geschlechtern ablehnt und gleichzeitig die Konstruktion von Geschlecht unterstreicht. Der Unterstrich wird nur dann unnötig, wenn Geschlecht als Kontinuum in der Gesellschaft Normalität ist und eine Gleichbehandlung und Gerechtigkeit zwischen den Geschlechtern auf allen Ebenen besteht.

<sup>2</sup>Vorurteile beruhen darauf, dass Urteile gefällt werden, ohne die objektiven Tatsachen zu kennen. Sowohl die Annahme, dass ein Mensch sich nur für Technik *oder* für Menschen interessieren kann, als auch dass Frauen sich vorwiegend für Menschen interessieren, sind keine objektiven Tatsachen sondern Vorurteile.

[BSFK16]. Anschließend wendete ich mich der Praxis zu und erforschte agile Softwareentwicklungsteams mit der *Grounded-Theory-Methodologie* (GTM). GTM eignet sich insbesondere, um zwischenmenschliche Beziehungen und Zusammenhänge zu beleuchten. GTM bricht Daten auf, um sie tief zu durchdringen und die hinter den Handlungen stehenden Annahmen aufzudecken. In der vorliegenden Arbeit wird GTM nach Charmaz angewendet [Cha14]. Charmaz GTM zeichnet sich zum einen dadurch aus, dass sie auf einem konstruktivistischen Verständnis beruht, das von multiplen Realitäten ausgeht und die Forschende als Teil der Konstruktion ansieht<sup>3</sup>. Zum anderen wird in Charmaz GTM verlangt, strikt den Daten zu folgen und weniger methodischen Vorgaben, Vorannahmen oder Theorien. Durch das strikte Folgen der Daten änderte sich im vorliegenden Fall das Forschungsinteresse und ein anderes vordringlicheres Problem in der agilen Softwareentwicklung wurde sichtbar:

Die am häufigsten verwendete agile Methode in der Softwareentwicklung ist Scrum [Kom17]. Scrum ist ein Rahmenwerk, das auf einer schmalen Beschreibung von 19 Seiten beruht [SS17b]. Scrum erfordert von Softwareentwicklungsteams Selbstorganisation. Die Anforderungen an agile Softwareentwicklungsteams steigen, ohne dass der Scrum-Guide klare Antworten dazu gibt. Diese Voraussetzungen machen eine Umsetzung herausfordernd und führen dazu, dass jedes agile Team eine eigene individuelle Umsetzungsweise hat.

Die bisherige Forschung zeigt, dass das entscheidende Merkmal agiler Teams die Selbstorganisation ist. Hoda [Hod11] hat bereits 2011 erforscht, dass agile Teams die Verantwortung tragen, eine Balance zu finden zwischen *Freiheit und Verantwortung, ständigem Lernen und Iterations-Druck* und *Spezialisierung und Cross-Funktionalität*<sup>4</sup>. Moe u.a. [MDD08] haben die Hürden einer agilen Transition und spezielle Teamaspekte untersucht. Neben Hoda und Dybå gibt es zu verschiedenen Aspekten von agilen Teams bereits Forschung [BGGM14] [YP14] [Lin13] [AHS<sup>+</sup>05] wie dem Grad an Selbstorganisation [GGGM19], menschlichen Aspekten wie Motivation [BBH<sup>+</sup>08] oder Persönlichkeit [CdC15] [AGJ09] oder konkrete Werkzeuge und Baukästen speziell zu agilen Artefakten [KIK<sup>+</sup>10] [UMWB14] [Lin13] [DCP12] [O'C10] [MAD12] [LDvB16][AFM<sup>+</sup>11] oder Zusammenhänge dieser verschiedenen Aspekte [CVR15] [KMAB18] [GWA15] [AGJ09].

Hinzu kommt Forschung aus anderen Feldern wie dem Management [Hac03] [Hac90] [WBW91], Ethnologie [JBP03] oder Philosophie [Hei06] zu (selbstorganisier-ten) Teams und ihre Dynamiken.

Es besteht folgendes **Problem**: Hodas und Dybås Forschungen sind zu einer Zeit entstanden als agile Arbeitsweisen noch neu waren und Teams sich in einem Transformationsprozess befanden. Man erhoffte sich von agilen Methoden grundsätzlich eine Verbesserung der Arbeitswelt und stand ihnen positiv gegenüber. Heutige Teams haben eine Unmenge an Baukästen und Leitfäden zur Verfügung, die sie individuell einsetzen. Der Schwerpunkt der Forschung zu agilen Teams liegt auf agilen Praktiken und deren Umsetzung [HSGT17]. Jedes Team hat seine eigene individuelle Umsetzung. Es ist schwierig generische allgemeingültige und gleichzeitig detaillierte und

---

<sup>3</sup>Ein Grund weshalb in dieser Arbeit aktive Sprache und erste Person singular verwendet wird siehe auch Abschnitt 5.

<sup>4</sup>Übrigens interessanterweise auch eine Dichotomie die Faulkner erkannt hat: Spezialisierung und ganzheitliches Denken.

für die Praxis konkrete Anweisungen zur Umsetzung von Scrum zu entwickeln. Eine mögliche Lösung wäre statt einer immer detaillierteren Sicht auf einzelne Praktiken, eine abstraktere Perspektive einzunehmen und die **Frage** zu beantworten:

**Was hält selbstorganisierte Softwareentwicklungsteams zusammen und fördert eine gute Zusammenarbeit?**

Aus der Antwort dieser Frage lässt sich wohl möglich ableiten aus welchen individuellen Instrumenten sich ein agiles Team bedienen sollte.

Die vorliegende Arbeit nähert sich einer möglichen Beantwortung dieser Frage, indem sie anhand eigener Forschung und Datenerhebung neue Erkenntnisse generiert, mit bestehenden Erkenntnissen verknüpft und weiterentwickelt.

Die vorliegende Arbeit deckt ein Konzept auf, dass die (erfolgreiche) Zusammenarbeit agiler Teams bestimmt und unabhängig von der Umsetzung und Anwendung von Werkzeugen ist.

Meine Arbeit gibt keine allgemeingültigen Empfehlungen wie Scrum optimaler einzusetzen ist oder wie Werkzeuge und Praktiken besser eingesetzt werden sollten. Die Arbeit stützt vielmehr die Perspektive, dass jedes Team individuelle Lösungen braucht: Was in einem Team zwingend notwendig ist, führt in einem anderen Team zum Bruch. Eine Erkenntnis die Hackman [Hac86] bereits 1986 hatte und die bis heute ihre Gültigkeit hat. Eine Erkenntnis, die sich auch im Software Engineering inzwischen durchgesetzt hat. Ralph [Ral18] fordert mehr empirische Forschung an der Arbeitsrealität von Softwareentwickler\_innen, die Menschen sind und menschlich handeln. Lenberg [LFW15] fordert den Menschen und sein Handeln in den Mittelpunkt zu stellen, anstatt Werkzeuge weiter auszufeilen. Agile Arbeitsweisen scheitern nicht am Werkzeug, sondern an der Person, die es anwendet [LFW15]. Laut Floyd ist Softwareentwicklung Realitätskonstruktion, die durch Menschen konstruiert wird [Flo92].

Meine Arbeit zeichnet aus, dass nicht in Einsen und Nullen gedacht wird. Es wird kein dichotomisches Denken zugrunde gelegt. Es gibt kein einfaches Ursache-Wirkung-Prinzip und es gibt kein Problem-Lösungs-Prinzip:

1. Es werden keine einfachen Werkzeuge oder Methoden als Lösung des Problems vorgestellt.
2. Es wird kein Mindset, Persönlichkeitsmerkmale oder Ähnliches definiert, das zur „wahren“ Agilität nötig ist.
3. Es wird ein Modell erarbeitet, das es jedem Team ermöglicht selbst zu erkennen und zu entscheiden, wie es mit den unterschiedlichen Mindsets und Fähigkeiten im Team umgehen will. Es wird ein Grundstein gesetzt, der Teams befähigt einen eigenständigen und damit tatsächlich agilen, selbstorganisierten Umgang mit den in jedem Team unterschiedlichen Dynamiken zu finden, um erfolgreich zu sein.

Die Arbeit zeichnet insbesondere aus, dass alle praktischen und konkreten Schritte der GTM in einem außergewöhnlich hohem Detailgrad transparent dargelegt werden (siehe Abschnitte 5, 6 und 7.1).

## 1.1 Struktur der Arbeit

Die Arbeit ist folgendermaßen strukturiert: Es folgt eine kurze Darstellung des Scrum Guide und erste Lücken, die eine Umsetzung erschweren, werden angemerkt.

Zu Beginn der Arbeit wird ein aktueller Forschungsstand verwandter Arbeiten dargelegt in Kapitel 2. Die Relevanz der vorliegenden Forschung wird belegt in Abschnitt 2.1 und in inhaltlich relevante Arbeiten umfänglich eingeführt in Abschnitt 2.3. Aus den verwandten Arbeiten werden Forschungslücken aufgezeigt und der Forschungsbeitrag, den die vorliegende Arbeit leistet, in Kapitel 3 dargelegt.

Anschließend wird die angewandte Methode GTM nach Charmaz in Kapitel 4 erläutert und wie sie konkret umgesetzt und angewendet wurde in Kapitel 6. Die Datenerhebung wird in Kapitel 5 dargelegt.

Es folgen die Ergebnisse in Kapitel 7. Das beinhaltet die methodische Herleitung der konkreten Ergebnisse in Abschnitt 7.1, das Modell und die Untermauerung mit Datenquellen in Abschnitt 7.2, sich daraus entwickelnde induktive Hypothesen in Abschnitt 7.3 und die Diskussion der Ergebnisse im Zusammenhang mit den verwandten Arbeiten in Abschnitt 7.4.

Abschließend wird die Validierung in der Praxis dargelegt und der Nutzen des Modells erläutert in Kapitel 8, bevor die Evaluierung nach den Kriterien Charmazs in Kapitel 9 und ein Fazit in Kapitel 10 folgt.

## 1.2 Scrum

Scrum ist ein agiles Rahmenwerk, das von den Mitunterzeichnern des agilen Manifests Jeff Sutherland und Ken Schwaber entwickelt wurde. Die Anwendung und Umsetzung beruht auf dem 19-seitigen Scrum-Guide [SS17b]. Es existiert inzwischen eine Unmenge an weiterführender (nicht wissenschaftlicher) Literatur für die Praxis, die sich mit Scrum oder agilen Methoden und ihrer Anwendung auseinandersetzt [GH15] [McK16] [Max18] [Glo13] [GR14]. Sie reicht von der allgemeinen Anwendung hin zu spezifischen Artefakten wie User-Stories<sup>5</sup>[WM17] [PEH15] [Coh13] [AFM<sup>+</sup>11] oder Events wie Retrospektiven [DL12] [And14] [AE14] oder ganz konkreten Umsetzungen in einer Branche mit einer konkreten Entwicklungsumgebung [Nya19]. Dabei zeigen Titel wie „How to kill the Scrum Monster“ mit dem Hinweis

A practical guide that focuses on real-world applications of the Scrum methodology and not just its theory [Bib18]

oder auch „Agile - the good the hype and the ugly“ [Mey14], dass die praktische Umsetzung und seine Vielzahl an Varianten inzwischen von den Anwender\_innen in der Praxis kritisch betrachtet wird. Im folgenden werden die Scrum-Artefakte vor allem anhand des originalen Scrum-Guide erläutert [SS17b] und bereits erste Probleme, die dadurch sichtbar werden angemerkt.

Das Scrum-Rahmenwerk wird anhand einer Grafik erläutert, die während der agilen Transformation der studentischen Hochschulprojekte entwickelt wurden [SB19]

---

<sup>5</sup>User-Stories sind die übliche Form Anforderungen in agilen Umfeld zu formulieren. Im Scrum-Guide werden sie jedoch nicht erwähnt.

[SSB18]. Durch den Hochschulkontext weichen einzelne Bezeichnungen in der Abbildung 1.1 von der Praxis ab.

Scrum ist wie Ken Schwaber und Jeff Sutherland in ihrem Scrum-Guide zu Beginn festhalten und wie agile Teams in der Praxis und die Unmengen an weiterer Literatur verdeutlichen, leicht zu verstehen und schwer umzusetzen. Scrum zeichnet sich dadurch aus, dass iterativ und inkrementell gearbeitet wird. Das heißt es wird eine Reihenfolge von Events immer wieder durchlaufen, sogenannte Sprints (siehe Abbildung 1.1), die in der Praxis zwischen zwei und vier Wochen dauern. Bei jedem Sprint ist es Ziel das Produkt inkrementell zu erweitern und Kundennutzen zu realisieren. Dabei wird durch die wiederkehrende Iteration das Prinzip des *Inspect and Adapt* angewendet [SS17b].

Im Juli 2016 wurden dem Scrum-Guide Werte hinzugefügt: Mut, Fokus, Offenheit und Respekt. Das Ziel ist es Vertrauen zwischen den Teammitgliedern aufzubauen. Laut Scrum-Guide gelingt das bereits durch die Auseinandersetzung mit den Ereignissen, Rollen und Artefakten des Scrum-Guide [SS17b].

### 1.2.1 Scrum-Team

Ein Scrum-Team besteht aus dem Entwicklungsteam, Product-Owner (PO) und Scrum-Master (SM). Ein Scrum-Team soll möglichst interdisziplinär aufgestellt sein und möglichst alle Ressourcen enthalten, die es braucht, um ein Produkt zu entwickeln [SS17b].

**Entwicklungsteam:** Mitglieder des Entwicklungsteams sind für die Umsetzung und Qualität des Produktes selbstorganisiert verantwortlich. Laut Scrum-Guide gibt es keine Unterscheidung zwischen einzelnen Teammitgliedern weder in der Seniorität noch durch die Spezialisierung. Es trägt immer das Team gemeinsam die Verantwortung für die Erreichung der Ziele, niemals Einzelpersonen. Das Entwicklungsteam sollte zwischen 3-9 Personen groß sein [SS17b].

**Product-Owner (PO):** Product-Owner tragen die Verantwortung für die Wertmaximierung des Produktes. In der Regel werden Anforderungen an das Produkt in einem *Product-Backlog*: (siehe unten) gepflegt. Product-Owner sind dafür verantwortlich, dass das Entwicklungsteam die Anforderungen versteht, für die Priorisierung der Anforderungen und dafür den Wert der Arbeit zu optimieren.

**Scrum-Master (SM):** Scrum-Master sind dafür verantwortlich, dass Scrum verstanden und umgesetzt wird. Die Person unterstützt alle Teammitglieder durch Coachen, Hürden beseitigen und konkrete Hilfestellung. Sie vertritt die agilen Werte auch nach außen hin gegenüber Nicht-Teammitgliedern.

### 1.2.2 Iterativer Ablauf in Scrum

Bei der Anwendung von Scrum wird eine bestimmte Reihenfolge von Ereignissen immer wieder erneut durchlaufen sogenannte Sprints <sup>6</sup>. Ein Sprint ist ein geschützter

<sup>6</sup>Siehe Abbildung 1.1 dort ist die Dauer eines Sprints mit 3 Wochen angegeben. Neben dem Sprint inkludiert auch die vorangehenden und abschließenden Meetings Sprint Review, Sprint Retrospective und Sprint Planning

# Scrum Flow

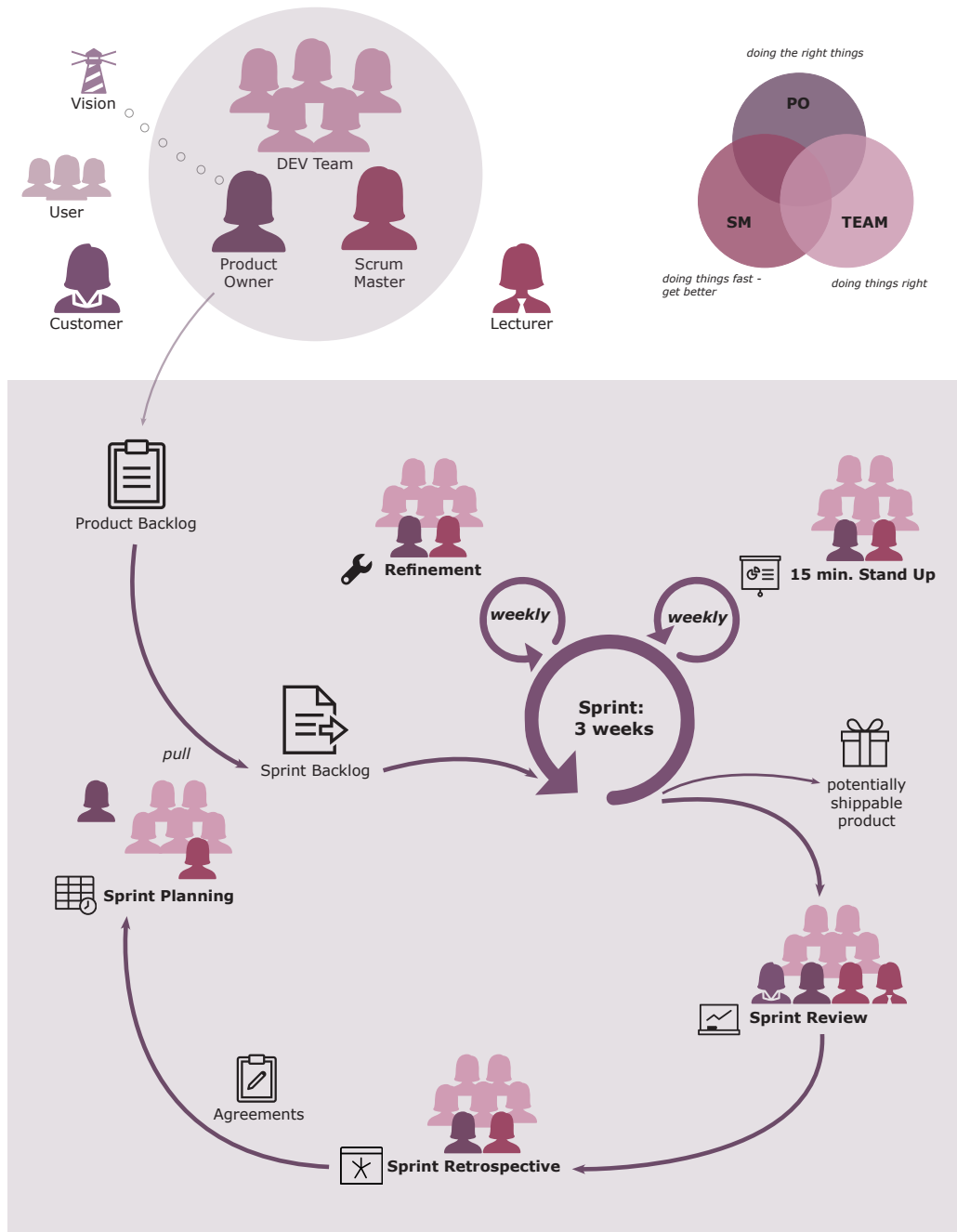


Abbildung 1.1: Der iterative Ablauf in Scrum



Raum, indem keine Änderungen an Anforderungen vorgenommen werden dürfen.  
Ein Sprint enthält folgende Ereignisse [SS17b] siehe auch Abbildung 1.1:

**Planning:** Im Planning entscheidet das Scrum-Team wie viele Anforderungen bzw. welches Produkt-Inkrement aus dem Product-Backlog es im nächsten Sprint umsetzen will. Es werden die beiden Fragen beantwortet, was im kommenden Sprint fertiggestellt werden kann und wie die Arbeit konkret und vor allem technisch erledigt werden kann. Es wird gemeinsam ein Sprintziel festgelegt.

**Daily-Scrum:** Daily-Scrum oder auch Stand-up (in Abbildung 1.1 als „weekly“ bezeichnet) ist ein kurzes ca. 15 minütiges Treffen, das in der Praxis einmal täglich stattfindet. Es wird der Fortschritt des Teams überprüft im Hinblick auf das Sprint-Ziel in dem alle Teammitglieder folgende Fragen kurz beantworten:

- Was habe ich gestern erreicht, das dem Entwicklungsteam hilft, das Sprint-Ziel zu erreichen?
- Was werde ich heute erledigen, um dem Entwicklungsteam bei der Erreichung des Sprint-Ziels zu helfen?
- Sehe ich irgendwelche Hindernisse, die mich oder das Entwicklungsteam vom Erreichen des Ziels abhalten?

**Review:** Am Ende des Sprints erfolgt ein Meeting, in dem das neue Produkt-Inkrement und das Ergebnis des Sprints den Stakeholdern wie Kund\_innen, Anwender\_innen und der oder dem Product-Owner vorgestellt werden und Feedback eingeholt wird. Als Ergebnis des Meetings können sich Änderungen im Product-Backlog ergeben.

**Retrospektive:** In der Retrospektive reflektiert das Scrum-Team ohne weitere Stakeholder über seine eigenen Prozesse, Abläufe und die Zusammenarbeit im letzten Sprint. Ziel ist es potentielle Verbesserungen zu identifizieren und im nächsten Sprint anzuwenden.

### 1.2.3 Scrum-Artefakte

Im Scrum-Guide werden drei weitere Artefakte erwähnt [SS17b]:

**Product-Backlog:** Im Produkt-Backlog wird die Produkt-Vision in einzelne Anforderungen runtergebrochen, die es ermöglichen inkrementell Produktwert herzustellen. Es ist eine Liste priorisierter Anforderungen, Features und Verbesserungen. Das Product-Backlog wird in einem Meeting mit dem ganzen Team verfeinert und offene Fragen zur Umsetzung geklärt. Das Meeting heißt je nach Team Refinement, Grooming oder Storytelling.

**Sprint-Backlog:** Das Sprint-Backlog enthält die Anforderungen, die im laufenden Sprint umgesetzt werden sollen.

**Definition of Done:** Die Definition of Done beschreibt welche Arbeitsschritte notwendig sind, damit eine Anforderung voll und ganz umgesetzt ist und als abgeschlossen gilt. Sie ist für jedes Team individuell.

### 1.3 Anmerkungen zum Scrum-Guide

Der Scrum-Guide ist sehr schmal und viele in der Praxis vielfach angewendeten Artefakte und Prozesse sind nur oberflächlich oder gar nicht enthalten. Hier eine minimale Auswahl von Begriffen, die notwendig sind für das Verständnis der weiteren Arbeit und nicht im Scrum-Guide enthalten sind:

**Scrumboard:** Ein Werkzeug, das in der Praxis alle untersuchten Scrum-Teams anwenden ist ein Scrumboard. Es bildet den aktuellen Stand im Sprint ab und beinhaltet mindestens drei Spalten: Sprint-Backlog, Work-in-Progress, Done. Es werden die Anforderungen von links nach rechts durch das Board abgearbeitet.

**User-Stories:** In den meisten Scrum-Teams werden Anforderungen in Form von User-Stories erstellt. Sie stellen den Anwender\_innennutzen in den Vordergrund. Die Erstellung von User-Stories ist ausgesprochen komplex und es gibt vielfache Anwendungshilfen dazu [WM17] [PEH15]. User-Stories haben einen unterschiedlichen Detailgrad. Je höher sie priorisiert sind im Backlog, um so detaillierter werden sie.

**Schätzen:** Der Aufwand der Anforderungen soll laut Scrum-Guide geschätzt werden. Inzwischen hat sich durchgesetzt, die Komplexität von User-Stories anhand von Story-Points in verschiedenen Methoden zu schätzen [Coh12] [WM17].

**Ablauf Retrospektive:** Eine häufig angewendeter Ablauf einer Retrospektive, der im Scrum-Guide nicht erwähnt wird, ist folgender [DL12]:

**set the stage:** Teammitglieder werden willkommen geheißen und ein Icebreaker wird durchgeführt.

**gather data:** Daten zum letzten Sprint werden mit Hilfe einer ausgewählten Übung erhoben.

**generate insights:** Einzelne Ereignisse werden detaillierter betrachtet und reflektiert.

**decide what to do:** Es wird entschieden, was konkret als Verbesserung umgesetzt werden soll.

**closing it:** Die Teammitglieder werden verabschiedet.

Der Scrum-Guide vermittelt ein sehr klares und einfaches Verständnis eines Softwareentwicklungsprozesses und überlässt die konkrete praktische Umsetzung den Teams. Daher gibt es inzwischen die oben erwähnte Vielzahl an weiterführender (nicht wissenschaftlicher) Literatur. Bemerkenswert ist, dass laut Schwaber und Sutherland, die 2016 hinzugefügten Werte durch das Einhalten und Leben des Scrum-Rahmenwerkes erlernt werden:

Die Mitglieder des Scrum-Teams lernen und erforschen diese Werte, indem sie mit den Scrum-Ereignissen, Rollen und Artefakten arbeiten. [SS17b]

Trotzdem hielten es Ken Schwaber und Jeff Sutherland für notwendig, Werte explizit zu erwähnen und nachträglich hinzuzufügen, obwohl diese ja eigentlich bereits

durch den Prozess entstehen sollen. Daraus lässt sich möglicherweise ableiten, dass ein Prozess allein diese Werte nicht einfach hervor bringt sondern eine Orientierung an diesen Werten frühzeitig expliziert werden sollte. Gleichzeitig sind Werte an sich sehr abstrakt und herausfordernd im Praxisalltag zu leben.

Im Hinblick auf mögliche Rollen, lässt der Scrum-Guide nur drei offizielle Rollen zu. Er lehnt auch eine Differenzierung zwischen verschiedenen Teammitgliedern innerhalb des Entwicklungsteams ab:

- Scrum kennt für Mitglieder des Entwicklungsteams keine Titel. Dies ist unabhängig von der Arbeit, die diese Personen erledigen.
- Scrum kennt keine weiteren Unterteilungen innerhalb des Entwicklungsteams, ungeachtet der verschiedenen Themenfelder, mit denen das Team sich befasst, also z.B. "Test", "Architektur", "Betrieb" oder "Analyse".
- Einzelne Mitglieder des Entwicklungsteams können zwar spezialisierte Fähigkeiten oder Spezialgebiete haben, aber die Rechenschaftspflicht obliegt dem Team als Ganzem.

[SS17a, S.7]

In der ersten Ausgabe des Scrum-Guide wurde diese Aussage noch unterstreichen mit der deutlichen Aussage „*Es gibt keine Ausnahmen von dieser Regel*“ [SS17b]. Wie ein Umgang mit unterschiedlichen Kompetenzen, Spezialisierungen und Vorlieben innerhalb des Teams gefunden werden soll, bleibt ungesagt. Gleichzeitig werden äußerst hohe Ansprüche gestellt, denn die Teams sollen interdisziplinär zusammenarbeiten, aus Profis bestehen und ein Teamverständnis und ein gemeinsames Verantwortungsgefühl entwickeln. Als hilfreiches Instrument zu Reflexion zum Teamverständnis wird in den Teams häufig die Retrospektive verstanden. Die Retrospektive ermöglicht augenscheinlich eine offene Reflexion. Wie schwierig jedoch eine tatsächliche Reflexion ist, zeigt aktuelle Forschung [AHA17] und ganze Bücher, die sich nur mit der Umsetzung dieses Events beschäftigen [And14] [DL12].

Der Scrum-Guide ist also zu schmal für eine tatsächliche erfolgreiche Softwareentwicklung in der Praxis. Es gibt eine Unmenge an weiteren Handlungsanweisungen, insbesondere zu einzelnen Werkzeugen und Artefakten wie beispielsweise User-Stories. Die Rollen oder der Umgang mit verschiedenen Fähigkeiten im Team bleibt unterdifferenziert und wird wenig beleuchtet bei gleichzeitigem hohem Anspruch. Bisherige Werkzeuge zur Anregung von Reflexion in der Retrospektive bestehen zwar, führen aber nicht immer zum Ziel [And14] [AHA17]. Es bleibt die offene Frage, wie ein Team seine Zusammenarbeit gestalten muss, um die oben erwähnten Werte zu pflegen und produktiv zu sein. Eine reine Umsetzung des Rahmenwerks macht Scrum-Teams nicht automatisch erfolgreich. Worüber und wann muss reflektiert werden, damit die erfolgreiche Zusammenarbeit von Menschen mit unterschiedlichen Fähigkeiten in Scrum-Teams gestärkt wird? Eine Frage, deren Beantwortung ich mich in dieser Arbeit nähere.

Nach der Einleitung und die Einführung in den Scrum-Guide, folgt im nächsten Abschnitt die Einführung in verwandte Arbeiten aus der Forschung.

## *Kapitel 1. Einleitung*

## Kapitel 2

# Verwandte Arbeiten

Diese Arbeit berührt viele verschiedene Forschungsbereiche in unterschiedlichen Disziplinen. Die Schwierigkeit bei der Darstellung der verwandten Arbeiten besteht darin, die unterschiedlichen Forschungen der verschiedenen Disziplinen in einem angemessenen und für diese Arbeit förderlichen Umfang und Detailgrad darzustellen. Dies geschieht ausgehend von einem allgemeineren Blick auf die grundsätzliche Relevanz des Themas hin zu immer detaillierteren Darstellungen fachspezifischer Literatur.

In drei Abschnitten werden verwandte Arbeiten vorgestellt:

- Im ersten Abschnitt [2.1](#) wird die **Relevanz des Problems** hergeleitet. Systematische und tertiäre Literaturreviews, die die (historische) Entwicklung des Forschungsfeldes *menschliche Faktoren in der Softwareentwicklung* untersuchen werden dargestellt. Die Arbeiten **stützen** die Relevanz der vorliegenden Arbeit.
- Im zweiten Abschnitt [2.2](#) stelle ich kurz dar, welche Themengebiete **nicht** Teil des **Forschungsinteresses** sind. Arbeiten von denen sich meine Arbeit inhaltlich oder in der epistemologischen Perspektive abgrenzt und die später nicht erneut diskutiert werden.
- Im darauf folgenden umfangreichsten Teil [2.3](#) führe ich in Arbeiten ein, die inhaltlich relevant sind und auf die Diskussionsteil im Ergebniskapitel [7.4](#) erneut eingegangen wird. Inhaltlich relevante Arbeiten außerhalb und innerhalb des Forschungsfeldes werden teilweise ausführlich dargestellt.

Die Arbeiten geben wertvolle Hinweise zur Konzeption und Durchführung der Forschung (insbesondere Abschnitt [2.1](#) und [2.3](#)), decken Lücken im Forschungsstand auf und/oder erforschen angrenzende Themengebiete an die meine Arbeit anschließt. Sie führen zu meiner Arbeit hin, in dem sie die **Grenzen und Stützen zu dem Forschungsinteresse** darstellen. Die Arbeiten unterstützen die **Relevanz des Problems**, das gelöst werden soll und zeigen Forschungsstrukturen, -fragen, -felder und inhaltliche Grenzen auf. Anschließend wird in Abschnitt [3](#) der Forschungsbeitrag, den diese Arbeit leistet hergeleitet und dargestellt.

## 2.1 Relevanz menschlicher Aspekte in der Softwareentwicklung

Die Bedeutung menschlicher Aspekte in der Softwareentwicklung wird vielfach erforscht und ihre Relevanz immer wieder betont. In seiner Key Note der *International Conference on Software Engineering 2018* (ICSE) bezog sich Fred Brooks Jr. auf Weinberg [Wei71] und erklärte, dass sich dieses Buch von 1971 nach wie vor gut verkauft zum Erstaunen des Autors. Das liegt sicherlich nicht an den völlig veralteten technischen Details in diesem Buch, sondern an den *Human Aspects* (menschliche Faktoren oder Aspekte), die bis heute relevant sind und immer noch der Erforschung bedürfen, so Fred Brooks Jr. Lenberg u.a. beziehen sich 2015 ebenfalls auf Weinberg und argumentieren sehr ausführlich, warum es im Software Engineering mehr Forschung zu menschlichen Faktoren bedarf und fordern *Behavioral Software Engineering* (BSE) zu etablieren in Anlehnung an das Pendant aus der Psychologie und Wirtschaftsforschung [LFW15]. Die Forschungsrichtung soll

cognitive, behavioral and social aspects of software engineering performed by individuals, groups or organizations [LFW15, S.18]

erforschen. Forschungsarbeiten zu BSE betrachten drei verschiedene Ebenen (Individuen, Gruppen und Organisationen), ihre Beziehungen untereinander und lassen sich in diese drei Ebenen einordnen.

Nach Lenbergs u.a. Argumentation sind Forschungen in diesem Feld relevanter als Forschung zur exakten Umsetzung neuer Werkzeuge oder Prozessen. Als anschauliches Beispiel nennen sie Change-Prozesse: Der Erfolg ist maßgeblich abhängig vom Verhalten der Menschen und dem Widerstand, den sie einer Veränderung entgegen setzen. Der Widerstand der Menschen entwickelt häufig höhere Relevanz, als die tatsächliche Perfektionierung der angestrebten Prozesse [LFW15, S. 15f]. Sie erstellen ein systematisches Literaturreview (SLR) zu den verschiedenen Bereichen des BSE. Diese Arbeit liefert eine gute Analyse des Forschungsstandes von 2015 von menschlichen Aspekten im Software Engineering. Auch wenn inzwischen inhaltliche Arbeiten hinzugekommen sind, stellen Lenberg u.a. eine hilfreiche Struktur des Feldes und Analyseeinheiten vor. Ein Teil der Arbeiten aus dieser SLR werden im Folgenden beleuchtet.

### 2.1.1 Menschliche Faktoren im agilen Umfeld

Agile Methoden zu untersuchen ist aktuell und relevant [HSGT17] [DD08]. Lenberg u.a. [LFW15] untersuchen in ihrer SLR zu BSE Arbeiten aus den Zeitraum von 1997 bis 2013. Diesen Zeitraum haben sie bewusst gewählt, um insbesondere Arbeiten zu agilen Methoden mit einzubeziehen, die einen starken Einfluss auf Software Engineering und menschliche Aspekte haben. Cruz u.a. [CdC15, S. 100] sehen ebenfalls einen Zusammenhang der steigenden Publikationen zu menschlichen Faktoren mit der Einführung von agilen Methoden.

Hoda u.a. [HSGT17] geben in einem tertiären Review einen aktuellen Überblick zu Forschungsarbeiten im agilen Umfeld. Der Schwerpunkt bisheriger Arbeiten liegt

auf konkreten *agilen Praktiken*. *Menschliche Faktoren* bilden auf Platz fünf lediglich die Mitte der thematischen Schwerpunkte [HSGT17].

Zusammenfassend lässt sich festhalten, dass agile Methoden das Forschungsinteresse an menschlichen Faktoren gefördert haben und ein wichtiger Bestandteil im Forschungsfeld des BSE sind. Gleichzeitig konzentriert sich Forschung zu agilen Methoden vorwiegend auf die praktische Umsetzung und weniger auf die menschlichen Faktoren. Folgt man Lenbergs u.a. [LFW15, S. 16] Argumentation sind Forschungen zu den beteiligten Menschen wichtiger und relevanter als Forschung, die sich lediglich mit Werkzeugen und Prozessen beschäftigt. Insgesamt lässt sich daraus schlussfolgern, dass Arbeiten zu menschlichen Faktoren im agilen Umfeld eine hohe Bedeutung zukommt. Menschliche Faktoren sind bisher weniger umfänglich erforscht als Tools und Prozesse und es besteht gegebenenfalls eine Forschungslücke.

Ausführliche Darstellungen einzelner relevanter Arbeiten, die auch Teil der SLRs von Lenberg u.a. [LFW15], Hoda u.a. [HSGT17] und Dybå u.a. [DD08] sind, folgen in den weiteren Abschnitten 2.2 und 2.3.

## 2.2 Angrenzende Themen, die *nicht* betrachtet werden

In diesem Teil werden verwandte Arbeiten dargestellt, die angrenzende Themen erforschen, die für meine Arbeit durchaus relevant sind. Sie bilden eigene umfängliche Themengebiete. Die Themengebiete standen nicht im Fokus des **Forschungsinteresses** und daher wurden keine oder zu wenige Daten dazu erhoben. Das Ziel dieses Abschnittes ist es, die Grenzen des Forschungsbeitrages aufzuzeigen und insbesondere gegenüber drei wichtigen Themengebieten im agilen Umfeld abzugrenzen: Kommunikation, agile Transformation und Persönlichkeit.

### 2.2.1 Kommunikation

Kommunikation, insbesondere in agilen Teams, ist ein wichtiger Faktor in Teams und wird daher häufig erforscht. Auch in der hier vorliegenden Arbeit wird der Wert von Kommunikation von einigen Interviewten betont. Es gibt umfängliche Forschung zu Kommunikation in agilen Teams [PHO+08] [CMD16] [AHS+05] [HRH13b].

Hummel u.a. [HRH13b] erheben eine SLR zu Kommunikation im agilen Umfeld. Hummel bezieht sich dabei vor allem auf die informelle Kommunikation innerhalb des Entwicklungsteams und zu Geschäftskunden. Das agile Manifest fordert vor allem mündliche Kommunikation vor Plänen und Dokumentationen zu bevorzugen, so Hummel [HRH13a].

Die Forschungsarbeiten der SLR betonen, dass Kommunikation im agilen Umfeld an Relevanz gewinnt im Vergleich zu bisherigen Softwareentwicklungsprozessen. Aus den untersuchten Forschungsarbeiten entwickeln sie ein Modell zu Kommunikation in agilen Softwareentwicklungsteams. Kommunikation ist dabei abhängig von der Teamverteilung, -größe und der Projekt-Domäne. Kommunikation wird beeinflusst vom Prozess wie XP oder Scrum und wirkt sich auf den Erfolg des Projektes aus [HRH13b, S. 344]. Pikkarainen u.a. [PHO+08] stellen fest, dass durch agile Prozesse die interne Teamkommunikation verbessert wird. Sprint-Planung, Stand-ups und offene Bürogestaltung wirken sich positiv auf die Kommunikation zu Anforderungen,

Features und Projektaufgaben aus [PHO<sup>+</sup>08, S. 332]. In beiden Fällen wird Kommunikation also im Rahmen des Prozesses und der Tools betrachtet und weniger unter menschlichen Aspekten.

In meiner Arbeit spielt Kommunikation insofern eine Rolle, dass sie benötigt wird für die dargestellten Prozesse im Ergebnisteil. Dabei erforsche ich nicht Umfang, Qualität und Art der Kommunikation, sondern lediglich über welche *Inhalte* kommuniziert werden sollte zur Förderung der menschlichen Zusammenarbeit. Dass Kommunikation an sich eine große Herausforderung sein kann, muss berücksichtigt werden, ist aber nicht Bestandteil dieser Arbeit. Die vorliegenden Daten dieser Arbeit eignen sich dazu nur sekundär, da sie das Thema zwar berühren, es aber nicht angemessen zur Hauptsache erklären.

## 2.2.2 Agile Transformation in der Organisation

Die Organisation ist ein Definitionsaspekt des BSE (s.o). Der Organisationsrahmen und sein Einfluss auf Teams ist ein relevanter Aspekt, insbesondere in agilen Teams und Unternehmen. Der agile Transformationsprozess ist dabei Schwerpunkt vieler Forschungen u.a. mit Grounded-Theory-Ansatz [HN17] [JZA<sup>+</sup>14] [MB09] [MDD09]. Agile Transformation bezeichnet dabei den Prozess, den eine Organisation durchläuft wenn die Arbeitsprozesse und Organisationsstrukturen hin zu agilen Prozessen und Strukturen entwickelt und verändert werden.

Moe u.a [MDD09] stellen auf dem Weg zur erfolgreichen agilen Transformation verschiedene Hürden vor. Die Hürden auf Teamebene werden in Abschnitt 2.3.6 genauer beleuchtet. Auf Organisationsebene identifizieren Moe u.a. geteilte Ressourcen und die organisationale Kontrolle als Hürden. Sie erklären, dass in einem Fallbeispiel die Selbstorganisation des Teams gestört wurde, da Teammitglieder an verschiedenen Projekten gleichzeitig arbeiten mussten. Die Organisation hat durch Kontrolle der gebuchten Projektstunden Druck aufgebaut und die Selbstorganisation zusätzlich gestört.

Hoda [HN17] erklärt, dass agile Transformation aus fünf Dimensionen besteht:

- Software-Praktiken
- Kultur
- Reflexion
- Management
- Team-Praktiken

Bei Software-Praktiken geht es darum einen agilen Prozess wie Scrum zu etablieren. Die Kultur verändert sich von einer hierarchischen zu einer offenen Kultur. Reflexionen sind in den Praktiken nicht nur vorhanden, sondern inkludiert. Das Management muss das Team zu Eigenständigkeit ermächtigen, statt Aufgaben vorzuschreiben. Team-Praktiken sollen nicht vom Management angetrieben, sondern vom Team entwickelt werden.

Die Organisationsebene und agile Transition wird in meiner Arbeit nicht betrachtet. Die agile Transformation lag in den untersuchten Unternehmen nominell bereits



länger zurück oder die Unternehmen waren von Beginn an agil. Meine erhobenen Daten sind auf Teamprozesse begrenzt und treffen daher keine fundierte wissenschaftliche Aussage zu agilen Transformationsprozessen.

### 2.2.3 Individuen reduziert auf Persönlichkeit

Individuen wurden in den Arbeiten zu menschlichen Faktoren in der Einordnung des BSE zwischen 1997 und 2013 am häufigsten erforscht laut Lenberg u.a. [LFW15]. Individuen und Gruppen ist die häufigste Kombination zweier Konzepte<sup>1</sup>:

The top three combinations of BSE concepts were personality-group composition (nine studies), personality-communication (7) and personality-conflicts (3) [...]

The top studied BSE concepts, with the most included studies, are communication, personality and job satisfaction. [LFW15, S. 23]

Lenberg u.a. fordern für zukünftige Arbeiten über diese Aspekte hinaus zu gehen. Sie glauben, dass einzelne Unterthemen dieser Forschungsfelder miteinander verknüpft sind und Abhängigkeiten bestehen. Menschliches Verhalten sei zu komplex, um einzelne Aspekte zu betrachten.

Bei Lenberg u.a. [LFW15] bezieht sich die Perspektive auf Individuen vor allem auf *Personality*, die häufig mit Myers-Briggs-Typenindikatoren (MBTI) ermesselt wird. Sie wünschen sich die Perspektive durch andere Modelle auf *Personality* zu erweitern. Cruz u.a. [CdC15] stellen ebenfalls MBTI als weitverbreitetes Tool dar und definieren *Personality* in ihrem systematischen Literaturreview folgendermaßen:

The definitions clearly separate personality from other constructs like cognition, motivation, and behavior, [...]

Persönlichkeit, beruhend auf verschiedenen Persönlichkeitstests, ist ein eigenes Feld in des BSE, die sich zwar auf Verhaltensforschung und Motivation auswirkt, sich aber auch abgrenzt. Lenberg u.a. [LFW15] fordern daher, den Blick über eine solche formalisierte Betrachtungsweise hinaus zu werfen.

Festschreibungen von Persönlichkeiten via Tests stehen im Gegensatz zur hier angewendeten Methode von Charmaz siehe Kapitel 4. Charmaz' Methodik liegt zu Grunde, dass Personen in jeder Situation die Freiheit haben, individuell und verschiedenen zu handeln. Somit sollte das *Handeln* der Beteiligten im Blickfeld der Analyse stehen und nicht eine zugeschriebene Persönlichkeit. Das führt dazu, dass Individuen zwar ein wichtiger Teil dieser Arbeit sind, Persönlichkeit im Sinne von Persönlichkeitstest jedoch kein Bestandteil der vorliegenden Arbeit ist.

---

<sup>1</sup>Dabei verwendet Lenberg u.a. den Begriff Konzepte, um verschiedenen Themen innerhalb der SLR abzugrenzen. Er ist zu unterscheiden vom Konzepte Begriff innerhalb einer GT.

## 2.3 Arbeiten, die sich *inhaltlich* mit (selbstorganisierten Softwareentwicklungs-) Teams befassen und später erneut diskutiert werden

In diesem Teil wird in Literatur eingeführt, die im Ergebnisteil relevant ist und dort inhaltlich diskutiert wird. Das bedeutet, dass folgende Kapitel beschränkt sich auf die *inhaltliche* Darstellung anderer Arbeiten und die *methodische* Abgrenzung gegenüber der vorliegenden Arbeit.

Während der inhaltlichen Darstellungen werden einige Forderungen der verwandten Arbeiten an zukünftige Forschungsarbeiten **farblich hervorgehoben**. Wichtige Probleme und Herausforderungen, die verwandte Arbeiten im Forschungsprozess erkennen, werden ebenfalls farblich hervorgehoben. Die farblichen Hervorhebungen kennzeichnen Textstellen, die im folgenden Kapitel zur Forschungslücke und meinem Forschungsbeitrag **3.1** besonders wichtig sind und in meinem Forschungsbeitrag aufgegriffen werden.

Wichtige Inhalte die in den späteren Ergebnissen wiederzufinden und relevant sind werden durch Vorwärtsverweise hervorgehoben.

Die *Relevanz* der dargestellten Literatur ergibt sich dadurch, dass

- **(R1)** (Relevanz erster Grad) entweder ein angrenzendes, ähnliches oder identisches Themengebiet im gleichen Kontext erforscht wurde oder
- **(R2)** (Relevanz zweiter Grad) ein angrenzendes, ähnliches oder identisches Themengebiet in einer anderen Disziplin wie z.B. der Verhaltenspsychologie erforscht wurde .

Meine Arbeit grenzt **(G-Grenzen)** sich gegenüber den verwandten Arbeiten ab:

- **(G1-Methode)** in der Methode und/oder
- **(G2-Daten)** im Kontext der Datenerhebung (Art der Unternehmen, Standorte, Reife der Teams, Zeitpunkt vor oder zu Beginn des Aufkommens agiler Methoden) und/oder
- **(G3-Inhalt)** Inhalt und/oder Art der Ergebnisse (andere oder ähnliche Ergebnisse).

Die *inhaltliche Abgrenzung* und Diskussion der Ergebnisse **(G3-Inhalt)** verwandter Arbeiten gegenüber meiner Arbeit wird **ausführlich** im Ergebnisteil **7.4** vorgenommen. Dieses Vorgehen entspricht der Methodik der Grounded-Theory wie es in Kapitel **4** dargestellt wird. In dieser späteren Diskussion werden vor allem Berührungspunkte und Ähnlichkeiten aufgegriffen, dargestellt und diskutiert. Sie werden daher im folgenden Kapitel nur in Form von Vorwärtsverweisen ergänzt, wenn sie für das Verständnis der Relevanz erforderlich sind.

Es ergeben sich folgende Argumente zur Abgrenzung, die im Weiteren mit Beispielen belegt werden. Bei der inhaltlichen Abgrenzung **(G3-Inhalt)** gelten grundsätzlich folgende Argumente:

Im Falle von **(R2)** lassen:

1. *Unterschiede* in den Erkenntnissen Spezifika von Softwareentwicklungsteams erkennen und tragen zur Schärfung bei.
2. *Ähnlichkeiten* in den Erkenntnissen
  - (a) bestätigen, dass das methodische Vorgehen meiner Arbeit stringent und richtig war, da es zu gleichen, etablierten Ergebnissen geführt hat.
  - (b) lassen den Schluss zu, dass sich Teile meiner Erkenntnisse verallgemeinern lassen ohne umfängliche quantitative Untermauerung und gewinnen damit an Relevanz.

Im Falle von **(R1)** gelten letztgenannte Argumente zu Ähnlichkeiten in den Ergebnissen äquivalent. In diesen Fällen ergibt sich eine Abgrenzung zumeist über **(G1-Methode)** und/oder **(G2-Daten)** und/oder es wird eine ausführliche Gegenüberstellung der Ergebnisse zu meinen Ergebnissen im Ergebnisteil (siehe Abschnitt 7.4) vorgenommen **(G3-Inhalt)**.

Die Darstellung der verwandten Arbeiten beginnt mit einem Forschungsüberblick von Magpili und Pazos aus dem Jahr 2017 zu selbstorganisierten Teams innerhalb und außerhalb des Software Engineering in Abschnitt 2.3.1.

Anschließend wird die Arbeit Hackmans [Hac86] inhaltlich ausführlich dargestellt. Diese Arbeit aus dem Jahr 1986 liefert wichtige und bis heute relevante Grundlagen und dient als eine ausführliche Einführung in das Wesen und Strukturen selbstorganisierter Teams. In zwei kleinen anschließenden Exkursen wird Hackmans Arbeit im Kontext agiler Methoden reflektiert und der Nutzen einer GTM gegenüber einer bloßen Validierung erläutert.

Folgend werden zwei relevante Phänomene, Motivation und Reflexion in agilen Softwareentwicklungsteams, kurz für sich betrachtet (Abschnitt 2.3.3 und 2.3.4)

Im Umfeld der agilen Softwareentwicklung fallen drei besonders aktive Forschungsgruppen auf. Dabei stehen sie in dem Zusammenhang mit den Personen Rashina Hoda, Pekka Abrahamsson und rund um die drei Forscher Nils Brede Moe, Torgeir Dingsøyr und Tore Dybå. Neben einer Zusammenschau der Arbeiten von Rashina Hoda und Nils Brede Moe als Erstautor\_in in Abschnitt 2.3.5 und 2.3.6, finden sich vieler dieser Arbeiten der drei Gruppen unter den jeweiligen thematischen Überschriften.

Zu guter Letzt werden noch zwei Arbeiten aus der Anthropologie und Philosophie dargestellt in Abschnitt 2.3.7. Diese Arbeiten kommen aus anderen Forschungsrichtungen **(R2)**. Sie tragen aber hilfreiche Aspekte zur späteren Diskussion (Abschnitt 7.4) bei und werden daher hier dargestellt. Grundsätzlich sind sie jedoch weit außerhalb des eigentlichen Forschungsgebietes und nehmen daher nur eine nachgeordnete Rolle ein.

### 2.3.1 Einflussfaktoren auf die Leistung selbstorganisierter Teams

Magpili und Pazos untersuchen in einem systematischen Review Forschung zur Effektivität von selbstorganisierten Teams in unterschiedlichen Kontexten [MP18]. Dabei ziehen sie Arbeiten mit quantitativen, qualitativen und kombinierten Methoden ein **(G1-Methode)**. Sie zielen darauf ab, Faktoren auf individueller, Team- und Organisationsebene zu finden, die die erfolgreiche (auf Leistung bezogene) Implementierung

von selbstorganisierten Teams fördern. Dabei ist mit Leistung vor allem der Output und die Effektivität der Teams gemeint. Magpilis und Pazos Arbeit stellt einen relevanten und relativ aktuellen Überblick über den Stand der Forschung über die Disziplin des Software Engineering hinaus dar (R2). Hervorzuheben ist, dass ihr Review auch wichtige Arbeiten im Software Engineering berücksichtigt und damit auch den Stellenwert dieser Forschungsrichtung für andere Strömungen belegt (R1). Im Folgenden wird ihre Arbeit zusammengefasst dargestellt. Falls nicht anders angemerkt, beziehen sich alle weiteren Aussagen im folgenden Abschnitt auf den Artikel von Magpili und Pazos.

Magpili und Pazos definieren selbstorganisierte Teams folgendermaßen:

[A Self-managing team] is a group of individuals with diverse skills and knowledge with the collective autonomy and responsibility to plan, manage, and execute tasks interdependently to attain a common goal [MP18, S. 4]

Magpili und Pazos belegen, dass die Anzahl selbstorganisierter Arbeitsteams steigt und verweisen auf Douglas u.a. [DG04] (R2). Die Forschung geht davon aus, dass selbstorganisierte Teams flexibler sind und schneller auf den Markt reagieren können [dJHD<sup>+</sup>13]. Dabei verfügen selbstorganisierte Teams über Autonomie bezüglich Projektmanagement, Entscheidungen, Problemlösungen, Konfliktmanagement, Fortbildung und Leistungskontrolle [LdWRJ05] [TPPD02]. Die im Team geteilte Führung führt zur mehr Verantwortlichkeit. Der Zugewinn an Verantwortlichkeit führt zu höherer Motivation, besseren Arbeitsprozessen und damit zu mehr Produktivität [vBK10]. Trotz dieser Annahmen, belegter Ergebnisse verschiedener Forschungsarbeiten und den Trend immer mehr selbstorganisierte Teams einzusetzen, sehen Magpili und Pazos widersprüchliche Forschungsergebnisse wie effektiv diese Teams tatsächlich sind. *Es wird gefordert, Erfolgsfaktoren oder Hürden selbstorganisierter Teams zu ermitteln, anstatt Effektivität weiter zu erforschen* [Ste06]. Diese Lücke möchten Magpili und Pazos schließen, indem sie in einem Review verschiedener Forschungsarbeiten Hürden und positive Faktoren bei der Implementierung selbstorganisierter Teams identifizieren. Dabei wählen sie ein Teameffektivitätsrahmenwerk [MMRG08], um in bestehenden Arbeiten Faktoren zu identifizieren. Das Rahmenwerk geht davon aus, dass *Input* auf Organisations-, Team und individueller Ebene sich auf *Prozesse* auswirkt, die wiederum die *Leistung* in den Endergebnissen beeinflussen [MMRG08, S. 413]. Dieser Prozess ist zyklisch d.h. das Ergebnis ist auch wieder Input.

Magpili und Pazos haben 695 Arbeiten gefunden aus den Jahren 1980-2017. Davon erfüllen 165 Arbeiten (82 quantitativ, 83 qualitativ) die Inklusionskriterien wie die Definition von selbstorganisierten Teams und werden im Review berücksichtigt. Die Arbeiten stammen aus einer hohen Bandbreite von Forschungsrichtungen wie Wirtschaft, Bildung, Ingenieurwesen, Management, Psychologie und Soziologie. An dieser Stelle möchte ich anmerken, dass auch die wissenschaftlichen Ansprüche unterschiedlich sind und nicht alle Quellen die Magpili und Pazos zitieren, dieselbe Qualität aufweisen. Magpili und Pazos untersuchen und stellen die Ergebnisse nach den drei Input-Faktoren des Rahmenwerks dar. Die Ergebnisse auf Team und individueller Ebene werden hier dargestellt. Input-Faktoren auf Organisationsebene werden nur sehr verkürzt wiedergegeben, da sie nicht Teil dieser Arbeit sind (siehe Abschnitt

2.2.2).

Ein Ergebnis des Reviews ist, dass auf *individueller* Ebene die Leistung durch folgende Faktoren beeinflusst wird:

- Individuelle Autonomie,
- individuelle Rolle,
- Führung,
- Fähigkeiten (Selbstmanagement, allgemeine Fähigkeiten und Teamwork),
- Fähigkeit zur Änderung und
- Arbeitserfahrung.

Zu den Punkten auf individueller Ebene im Einzelnen:

In der Diskussion zu individueller Autonomie beziehen sich Magpili und Pazos u.a. auf die Arbeiten von Hackman (siehe Abschnitt 2.3.2) und Moe (siehe Abschnitt 2.3.6), deren Arbeiten an anderer Stelle ausführlich diskutiert werden.

In ihrem Review beziehen sich Magpili und Pazos auf das Rollenverständnis von Hoda, das ebenfalls in dieser Arbeit in Abschnitt 2.3.5 an anderer Stelle diskutiert wird. In der Gesamtschau der Arbeiten erklären Magpili und Pazos, dass das Management von Rollen ein wichtiger Schlüsselfaktor in selbstorganisierten Teams ist. Dabei sind die Rollen flexibel und durchlässig und werden den sich ändernden Anforderungen des Unternehmens angepasst. Ein Verständnis das auch in der vorliegenden Arbeit diskutiert wird und teilweise in den Ergebnissen wiederzufinden ist, siehe Abschnitt 7.2.4 und 7.4.3. Besonders in hoch performanten Teams sind Fähigkeiten redundant besetzt und Rollenflexibilität unabhängig von Stellenbeschreibungen gegeben. Diese hohe Flexibilität kann auch zu Leistungseinbußen führen, wenn die Aufgabenorientierung und Prioritäten verloren gehen. Die erhöhte Verantwortlichkeit in den einzelnen Rollen führt zu einem höheren Stresslevel im Team [MP18].

In selbstorganisierten Teams ist es hilfreich, wenn die Teammitglieder über verschiedene Führungsstile verfügen, die sich an die flexiblen Anforderungen anpassen können. Ein transformationaler Führungsstil führt zur besseren Nutzung des sozialen Kapitals und besserer Leistung. Die Führungsfähigkeiten beruhen häufig auf praktischem Wissen und ggf. auch hohen technischen Kompetenzen. Dabei helfen erfolgreiche Führungspersonen dem Team Qualität im Blick zu behalten, Grenzen zu überwinden, zu coachen und soziale und hilfreiche Teamnormen zu entwickeln [MP18].

Wichtige selbstorganisatorische Fähigkeiten sind: Die Individuen akzeptieren die Verantwortung, können sich selbst regulieren und wollen Prozesse selbst mitgestalten. Die Bereitschaft Verantwortung zu übernehmen, hängt von der eigenen Erfahrung und Expertise ab. Verantwortung zu akzeptieren ist dabei eine wichtige Voraussetzung, die beinhaltet, Aufgaben zu übernehmen und das Team zu stützen und zu schützen. Diese Fähigkeiten sind in den Ergebnissen dieser Arbeit und dem Modell wiederzufinden und werden in einen strukturierten Zusammenhang gestellt (siehe Kapitel 7). Dabei wird die Bedeutung (des Fehlens) dieser Fähigkeiten in einer Reihe der Episoden sehr deutlich wie beispielsweise die Fähigkeit Verantwortung zu akzeptieren oder nicht und welche Folgen für das Team daraus entstehen (siehe Beispiel

Episode 3 in Abschnitt 7.2). Dabei spielt intrinsische Selbstreflexion bei der Entwicklung dieser Fähigkeiten eine wichtige Rolle, so ein Ergebnis dieser Arbeit.

Individuelles fachliches Wissen ist ebenfalls wichtig in selbstorganisierten Teams. Ein Verständnis das auch in der späteren Rollendefinition der vorliegenden Arbeit wiederzufinden ist (siehe Abschnitt 7.2.4). Laut Magpili und Pazos besteht sonst die Gefahr, dass Teammitglieder den Status quo verteidigen und keine Verantwortung tragen wollen. Dabei kann auch die Verfügbarkeit von Zeit und Ressourcen zur Entwicklung von Fähigkeiten helfen. Wertvoll sind dabei Personen, die domänenübergreifende Fähigkeiten haben und dabei dem Team helfen flexibel zu sein. Diese Fähigkeiten auszubauen, verlangsamt das Team zu Beginn, wirkt sich aber insgesamt positiv auf die Leistung aus<sup>2</sup>.

Teamfähigkeiten sind neben technischen Fähigkeiten eine weitere wichtige Voraussetzung auf individueller Ebene. Neben Fähigkeiten wie Kommunikation zählt dazu auch ein transformationaler Führungsstil, dem egoistisches Verhalten entgegensteht.

Veränderung akzeptieren und aushalten können, ist eine weitere Voraussetzung für Individuen in selbstorganisierten Teams. Dem gegenüber stehen Unsicherheiten wie der Wunsch nach Struktur, Angst vor Neuem, fehlender Erfahrung, fehlende Job-sicherheit und Ablehnung steigender Arbeitsanforderungen. Wie herausfordernd das für manche Teammitglieder sein kann ist in den Episoden der vorliegenden Arbeit wiederzufinden in Abschnitt 7.2.

Arbeitserfahrung ist in selbstorganisierten Teams wichtig, um Autonomie zu erreichen und den Gesamtkontext der Arbeit vollständig zu erfassen. Häufig entwickeln sich dadurch Personen mit längerer Arbeitserfahrung zu direkten oder indirekten Führungspersonen. Das birgt die Gefahr in alte Strukturen zu verfallen.

Auf Teamebene identifizieren Magpili und Pazos folgende Einflussfaktoren:

- Externe Führung,
- gegenseitige Kontrolle,
- Aufgaben-Charakteristik,
- Team-Autonomie und
- Team-Diversität.

Mit *externer Führung* sind Personen gemeint, die Einfluss auf das Team haben ohne im alltäglichen mit ihm direkt zusammenzuarbeiten. Grundsätzlich ist es fraglich, ob selbstorganisierte Teams eine solche Führung benötigen und es wird eine Abhängigkeit von der Reife der Teams und der Notwendigkeit hergestellt. Daher ist es Aufgabe externer Führungskräfte die Autonomie der Teams weiter zu entwickeln. Dazu gehört Selbstreflexion, Selbstkritik und Selbstverbesserung bei den Teammitgliedern zu fördern. Die Eigenständigkeit der Teams kann durch Unterstützung im Konfliktmanagement, der Kommunikation, in der Entscheidungsfindung und der Teamentwicklung

---

<sup>2</sup>Das hier von Magpili und Pazos vertretende Verständnis lässt sich auch als cross-funktionale Teams beschreiben, in denen es ein Ziel ist T-shape Fähigkeiten aufzubauen. T-shape ist ein Begriff aus dem agilen Umfeld und bedeutet, dass eine Person in einem Bereich ein vertieftes Verständnis und in vielen anderen Bereichen ein Grundverständnis hat.

ausgebaut werden. Externe Führung kann sich negativ auswirken, wenn sie durch enge Kontrolle und Anweisungen die Autonomie der Teams beschneidet. Das kann zu Rollenkonflikten führen. Sie ist dann sinnvoll, wenn das Team darum bittet oder eine Lösung für ein konkretes Problem sucht. Eine Aufgabe die in der vorliegenden Arbeit von Scrum-Mastern übernommen wird und sich im Modell, in den Episoden in Abschnitt 7.2 und der Hypothese 4 wiederfindet.

Gegenseitige Kontrolle kann dazu beitragen, soziale Kontrolle über individuelles Verhalten auszuüben. Im positiven Fall kann das dabei helfen, unproduktive Teammitglieder zu regulieren. Zu strenge Kontrollen wirken sich wiederum negativ auf die Autonomie aus. Im Ergebnis dieser Arbeit, im Modell findet die Ausübung dieser Kontrolle während der verschiedenen Formen der Teamreflexion statt siehe Abschnitte 7.2.3 und 7.2.2.

Bezüglich der Aufgaben-Charakteristik entfalten selbstorganisierte Teams bei Aufgaben mit hohen technischen Anforderungen, Innovationen, neuen Aufgaben, hohen Abhängigkeiten und wenig Routinen ihre Potentiale am Besten. Diese hohen Anforderungen fördern auch das Verantwortungsgefühl und den Gestaltungs-Spielraum sprich die Autonomie der Teams. Eine Herausforderung, die bei agilen Softwareentwicklungsteams grundsätzlich gegeben ist und daher im Modell zum Ausgangspunkt gehört (siehe Kapitel 7).

Team-Autonomie muss in einem angemessenen Maß gegeben sein. Dass heißt in der Organisation muss bezüglich Mindset, Management und Struktur Autonomie in angemessenen Umfang gegeben sein. Teammitglieder benötigen Einfluss und Kontrolle auf die Dinge für die sie verantwortlich gemacht werden. Dazu gehört auch, dass sie mit entsprechenden technischen Fähigkeiten ausgestattet sind. Ebenfalls eine Voraussetzung und damit ein Ausgangspunkt im Modell siehe Kapitel 7.

Im Hinblick auf Team-Diversität wird der Einfluss einer hohen Bandbreite von Fähigkeiten in Teams widersprüchlich diskutiert. Es gibt Studien, die eine Leistungsverbesserung bei der Lösung komplexer Aufgaben aufgrund vom Vorhandensein vielfältiger Fähigkeiten in Teams sehen. Gleichzeitig entstehen durch verschiedene berufliche Hintergründe der Teammitglieder auch Unterschiede bezüglich Wissen, Status, Erwartungen und Macht, die zu Hürden in der Entscheidungsfindung führen können. Eine Balance zu halten ist hier wichtig. Hilfreich ist ein klares Rollenverständnis, das durch Kommunikation und die Fähigkeit Perspektiven von Personen mit anderen Hintergründen zu verstehen erreicht wird. Eine Erkenntnis, die die vorliegende Arbeit teilt und in den Ergebnissen wiederzufinden ist (siehe Kapitel 7 und insbesondere 7.2.4). Eine ausführliche Analyse ist auch zu finden in einer weiteren Publikation [BP18].

Faktoren auf Organisationsebene sind: Unternehmenskultur, Unternehmenspolitik, nationale Kultur, Organisationsziele, Organisationsstruktur, Schulung, Ressourcen und Belohnungen.

Zusammenfassend lässt sich sagen, dass diese Faktoren die Autonomie der Teams, wozu auch das Eingehen von Risiken gehört, nicht untergraben dürfen. Weitere Faktoren wie Ziele setzen, Weiterbildungen oder Ressourcen sind auch im Abschnitt 2.3.2 zu finden und werden ähnlich diskutiert und daher hier nicht erneut dargelegt.

Magpili und Pazo schlußfolgern u.a., dass zukünftige Forschung die angemessene Balance zwischen Autonomie und Stabilität untersuchen sollte, damit Führungsmo-

delle entwickelt werden können unter Berücksichtigung der verschiedenen Kontexte selbstorganisierter Teams. Außerdem sollte zukünftige Forschung den Blick auf das Wohlbefinden der Teammitglieder legen, die durch Selbstorganisation höheren Anforderungen und Stress ausgesetzt sind.

Ergänzend zu Magpilis und Pazos Zusammenfassung über die aktuelle Forschung zu selbstorganisierten Teams und Leistung, werden hier noch drei weitere Arbeiten zu diesem Thema kurz betrachtet.

Lenberg und Feldt [LF18] stellen einen ersten Zusammenhang zwischen psychologischen Konzepten und Leistung in selbstorganisierten Softwareentwicklungsteams her. In der Psychologie gibt es Konzepte von *Team Norms*. Lenberg und Feldt [LF18] erklären, dass diese *Team Norms* und *Psychological Safety* in Teams umfänglich erforscht sind, aber Forschung zu agilen Softwareentwicklungsteams (G2-Daten) fehlt und erheben selbst eine erste quantitative Studie (G1-Methode). In der Studie stellen sie dar, dass *Team Norm Clarity* sich direkt auf *Team Performance* und *Job Satisfaction* auswirkt. *Team Norm Clarity* wirkt sich dabei mehr aus als *Psychological Safety*.

Damit belegen sie die Klarheit über Normen sich positiv auf das Team auswirkt. Die vorliegende Arbeit stellt einen ähnlichen Zusammenhang her, geht aber noch einen Schritt weiter und stellt das grundsätzliche Phänomen KLARHEIT GEWINNEN in den Mittelpunkt des Modells siehe Kapitel 7.

Hoegl und Gemuenden [HG01] untersuchen qualitativ (G1-Methode) 145 Software Teams in Deutschland (G2-Daten). Sie stellen dabei fest, dass die Leistung der Teams von ihrer Zusammenarbeit beeinflusst wird. Dabei wird Leistung subjektiv unterschiedlich wahrgenommen von Manager\_innen und Entwickler\_innen. Entwickler\_innen fühlen sich dann erfolgreich, wenn sie sich auch persönlich weiterentwickelt haben. Gute Teamzusammenarbeit definieren Hoegl und Gemuenden, als gute Kommunikation, Koordination, Ausgewogenheit der individuellen Beiträge, gegenseitige Unterstützung, Bemühungen und Zusammenhalt [HG01]. Dies ist eine interessante Definition von Leistung und Teamzusammenarbeit. Das erarbeitete Modell dieser Arbeit sieht den Zusammenhalt im Team als Folge guter Zusammenarbeit bzw. GEWONNENER KLARHEIT an (siehe Kapitel 7 und insbesondere Abschnitt 7.2.6).

Bunderson und Boumgarden [BB10] untersuchen selbstorganisierte Teams. Sie unterscheiden sich zu agilen Softwareentwicklungsteams, da es Produktionsteams mit immer wiederkehrenden, wenn auch komplexen Aufgaben sind (G2-Daten). Sie überprüfen ihre Hypothesen quantitativ mit Fragebögen (G1-Methode). Inhaltlich (G3-Inhalt) legen sie dar, dass strukturierte Teams eine höhere Lernfähigkeit besitzen. Unter Struktur verstehen sie u.a. klarere Rollenaufteilungen. Die Lernfähigkeit wirkt sich wiederum positiv auf die Leistung aus. Außerdem stellen sie einen positiven Zusammenhang von Struktur, psychologischer Sicherheit und einer geringeren Anzahl von Konflikten fest. Diese Ergebnisse sind für die vorliegende Arbeit insofern relevant, da sie einen quantitativen Zusammenhang zwischen klaren Rollen und Teamzusammenarbeit (psychologische Sicherheit und Anzahl der Konflikten) herstellt. Ein Verständnis, dass im Modell wiederzufinden ist (siehe Kapitel 7 und Abschnitt 7.2.4). Durch Bunderson und Boumgarden wird die quantitative Relevanz belegt.



### 2.3.2 Hackman: selbstorganisierte Teams

In diesem Abschnitt wird die Forschung Hackmans erläutert. Zuerst wird die Relevanz, Abgrenzung und Einordnung zur vorliegenden Arbeit vorgenommen. Anschließend wird seine wichtigste Arbeit [Hac86] zu selbstorganisierten Teams inhaltlich zusammengefasst. Bei der Zusammenfassung werden besonders Ergebnisse berücksichtigt, die aus heutiger Perspektive für agile Softwareentwicklungsteams relevant sind. Gleichzeitig dient die inhaltliche Zusammenfassung zur Einführung in die Besonderheiten und das komplexe Umfeld von selbstorganisierten Teams. Einzelne inhaltliche Aspekte werden später im Anschluss an den jeweiligen Ergebnisteil in Abschnitt 7.4 im Bezug zu den generierten Erkenntnissen dieser Arbeit erneut diskutiert.

#### 2.3.2.1 Relevanz und Einordnung

Hackman [WHL05] war ein Professor für Sozial- und Organisationspsychologie und ist damit Teil einer anderen Forschungsdisziplin (R2). Seine unzähligen Publikationen (eine minimale Auswahl zu Teams wäre beispielsweise: [Hac80, Hac86, Hac90, Hac03]) sind bis heute relevant und werden in aktuellen wissenschaftlichen Arbeiten zu agilen Softwareentwicklungsteams [MDD09] [MDD10] [LF18] herangezogen.<sup>3</sup>

Hackman erklärt nicht, nach welcher Methodik (G1-Methode) er vorgegangen ist. In [Hac90] wird deutlich, dass seine Erkenntnisse auf Beobachtungen von verschiedenen Teams beruhen. Man kann mit einer hohen Wahrscheinlichkeit davon ausgehen, dass er nicht mit GTM gearbeitet hat: In seinem Vorgehen wird nirgends auf ein Konzept einer GTM oder ähnliches Bezug genommen. Damit grenzt sich die vorliegende Arbeit methodisch ab.

Seine Daten (G2-Daten) wurden vor über 30 Jahren in selbstorganisierten Teams erhoben. In seinen Beispielen [Hac90, Hac03] taucht vor allem immer wieder eine selbstorganisierte Fluggesellschaft auf als auch ein Kammerorchester, Sportteam und eine Fließbandarbeitsgruppe. Dabei hat er die Fluggesellschaft über einen längeren Zeitraum verfolgt und dabei auch ihr Scheitern beobachtet.

Hackman stammt also aus einer anderen Disziplin (R2), verwendet eine andere Methode (G1-Methode) und unterscheidet sich deutlich im Kontext der Datenerhebung (G2-Daten) sowohl zeitlich als auch in der Art der Teams. Auf der Ebene der Ergebnisse finden sich Gemeinsamkeiten und Unterschiede. Womit die oben bereits genannten Argumente greifen bei (R2) und (G3-Inhalt): Inhaltlich lassen sich in den Unterschieden die Spezifika von Softwareentwicklungsteams erkennen. Methodisch werden die Erkenntnisse gestützt: Mein methodisches Vorgehen hat zu etablierten Erkenntnissen geführt, die sich daher ggf. auch verallgemeinern lassen.

#### 2.3.2.2 Hackmans Forschungsergebnisse als Einführung in selbstorganisierte Teams

Der folgende Abschnitt ist eine Zusammenfassung von [Hac86]. Falls nicht anderes angegeben, beruhen alle Aussagen auf [Hac86]. Es werden vor allem die Teile wiedergegeben, die entweder einen inhaltlichen Einstieg in selbstorganisierte Teams

---

<sup>3</sup> Merkwürdigerweise zitieren sie nicht sein Werk, dass sich direkt mit selbstorganisierten Teams beschäftigt [Hac86]. Vielleicht weil es bereits recht alt ist oder die Inhalte seiner Publikationen auf dieselben Grundlagen verweisen.

ermöglichen und/oder relevante Beiträge für die spätere Diskussion der Ergebnisse sind.

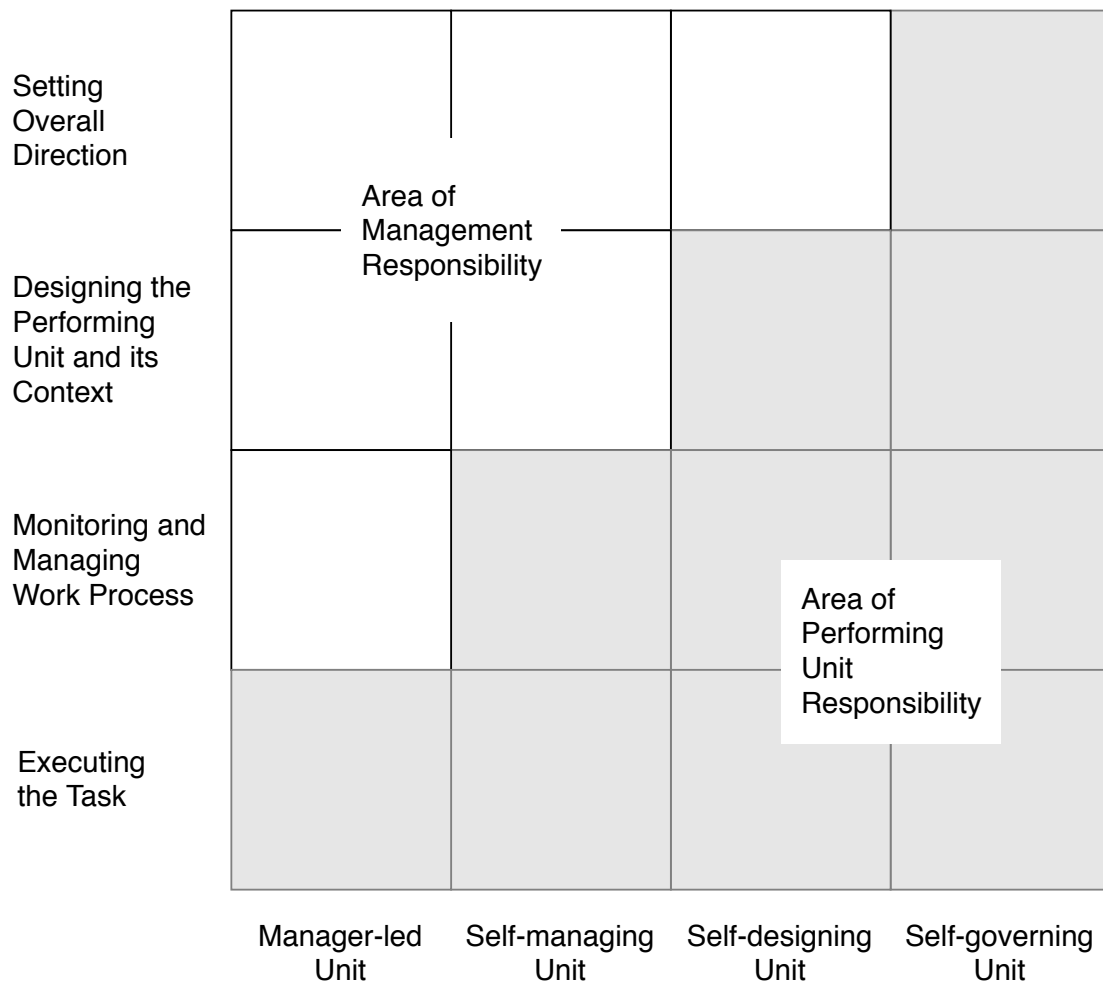


Abbildung 2.1: Definition selbstorganisierter Teams (zweite Spalte von links: Self-managing Unit) entnommen aus Hackman [Hac86, S. 92]

Zu Beginn nimmt Hackman eine Definition zu selbstorganisierten Teams vor und stellt sie in einer Grafik dar (siehe auch Abbildung 2.1) [Hac86, S. 92]<sup>4</sup>. Auf der Y-Achse sind vier Funktionen abgebildet, die in einer Organisation ausgeführt werden müssen, damit die Arbeit erledigt wird: Aufgaben ausführen, Arbeitsprozess managen, Arbeitseinheit zusammen setzen und die Richtung vorgeben. Die Aufgaben und Funktionen auf der X-Achse können entweder vom Management oder der Arbeitsgruppe ausgeführt werden. Eine selbstorganisierte Gruppe ist dabei so definiert, dass sie die beiden Funktionen *Aufgaben ausführen* und *Arbeitsprozesse managen* überneh-

<sup>4</sup> Die Grafik spiegelt mit „Self-designing Unit“ und „Self-governing Unit“ auch den aktuellen Trend von *New Work* [HWAB17] wider, in dem es neben der Globalisierung und Digitalisierung auch darum geht, das Personal- und Strategie- Management immer mehr in die Hände der Arbeitnehmer\_innen zu legen. Ein Trend mit dem ich mich nicht beschäftige, der aber zeigt wie aktuell Hackmans Erkenntnisse nach wie vor sind.

men. Hackman unterscheidet die Gruppen der *manager*, die vor allem dafür verantwortlich sind Aufgaben an andere zu delegieren und zu strukturieren und *member*, die Aufgaben ausführen. Eine *Performing Unit* definiert sich dadurch, dass eine einzelne Person oder eine Gruppe von Menschen Verantwortung teilt für die Bewältigung einer gemeinsamen Aufgabe. Eine wichtige Definition im Hinblick auf selbstorganisierte Teams, die auch später in der Diskussion in Abschnitt 7.4 erneut auftaucht ist dabei:

I will speak of performing units as if they consisted of several individuals working interdependently on a common task [Hac86, S.91]

Hackman verwendet die Begriffe „*self-organized performing unit*“ oder „*group*“ im Unterschied zum hier verwendeten Begriff *Team*. In anderen Forschungsarbeiten wird der Begriff Team und Gruppe gleichgesetzt und sogar Zweier-Beziehungen in die Forschung einbezogen [RPSG<sup>+</sup>17]. Eine genaue Differenzierung geht aber davon aus, dass es verschiedener Bedingungen bedarf, damit aus einer Gruppe ein Team wird [KDS<sup>+</sup>09]:

**Ziele** Es gibt individuelle und Team-Ziele.

**Zwischenmenschliche Beziehungen** werden durch Kommunikation und gegenseitiges Vertrauen gestärkt.

**Rollen-Klärung** Ein Verständnis der gegenseitigen Rollenerwartungen wird entwickelt.

**Problemlösung** Zur Bewältigung der Aufgaben wird gemeinsam ein Plan entwickelt.

Magpili und Pazo haben (s.o.) selbstorganisierte Teams als Gruppe definiert, deren Mitglieder in Abhängigkeit voneinander, autonom ein gemeinsames Ziel verfolgen. Hackmans Definition von selbstorganisierten Gruppen umfasst bereits ein Teil der Teamaspekte: Die Gruppenmitglieder arbeiten in Abhängigkeit zueinander an einer Aufgabe. Sie sind also in ihrer Zielerreichung aneinander gebunden. Wie im weiteren Verlauf deutlich werden wird, sieht Hackman auch weitere Aspekte wie die gemeinsame Problemlösung oder Kommunikation als Aufgabe selbstorganisierter Gruppen. Teams und damit auch Scrum-Teams sind nur eine Teilmenge von Gruppen. Wobei die Anforderungen und selbstorganisierte Gruppendifinition Hackmans bereits klare Bezüge zu Teams aufweisen. Im Mittelpunkt dieser Arbeit stehen (Scrum-)Teams. Daher wird im Folgenden auch der Begriff *Team* für Hackmans „*groups*“ verwendet und sich damit auf die für diese Arbeit relevante Teilmenge bezogen.

Eine *selbstorganisierte* Arbeitseinheit führt Aufgaben aus und gestaltet ihren Arbeitsprozess selbst.

Hackman erklärt fünf Anzeichen von selbstorganisierten Gruppen:

1. Personen übernehmen *persönliche Verantwortung* für ihre Arbeitsergebnisse und ihr Handeln.
2. Personen *kontrollieren* ihre eigene Leistung kontinuierlich und streben aktiv nach Feedback, um zu erfahren wie gut sie die Aufgaben erfüllen.

3. Personen *regeln* ihre eigene Leistung und korrigieren eigenständig ihr Verhalten, um ihre Leistung zu verbessern.
4. Wenn Personen etwas zur Leistungserbringung fehlt, verlangen sie aktiv *Unterstützung von der Organisation* auf eine konstruktive Weise.
5. Personen *helfen* aktiv anderen Mitgliedern, ihre Leistung zu verbessern. Dabei stellen sie zuvor sicher, dass sie ihren eigenen Verantwortlichkeiten gerecht wurden, bevor sie Hilfe anbieten.

Die Punkte 1-4 sind im Modell vor allem unter dem Aspekt der Selbst- und Team-reflexion wiederzufinden, siehe auch später im Kapitel 7 und insbesondere Abschnitte 7.4.3.1, 7.4.3.2 und 7.3.

Im darauffolgenden Abschnitt geht Hackman auf die Effektivität von selbstorganisierten Gruppen ein und ob sie grundsätzlich *besser* arbeiten als traditionelle Gruppen. Der Begriff *Effektivität* kann je nach Wertegrundlage unterschiedlich definiert werden. Hackmann definiert Effektivität einer Arbeitsgruppe auf drei Ebenen:

1. Die Arbeitsleistung und die Ergebnisse der Arbeitsgruppe (Output)
2. Das Ausmaß inwiefern die Gruppe ihre Arbeit so ausführt, dass sie auch zukünftig erfolgreich und gerne zusammenarbeitet.
3. Das Ausmaß inwieweit die Arbeitserfahrung zum Wachsen und Wohlbehalt der einzelnen Teammitglieder beiträgt.

Nach dieser Definition beruht Effektivität nicht allein auf dem Arbeitsergebnis, sondern auf den sozialen und persönlichen Fähigkeiten einer Gruppe. Ein Verständnis, dass durch das Modell in den Mittelpunkt gerückt wird siehe Kapitel 7. Insbesondere Punkt 2 Hackmans, der Teamzusammenhalt ist Ergebnis der durch sozialen Fähigkeiten geprägten Prozesses inwieweit **KLARHEIT GEWONNEN** wird.

Zum damaligen Zeitpunkt wird in der Forschung konträr diskutiert, ob selbstorganisierte Gruppen effektiver sind als traditionelle Gruppen. Hackman erklärt, dass diese Frage unbeantwortet bleibt. Seiner Erfahrung nach, sind selbstorganisierte Teams auf beiden Seiten der Effektivitätsskala zu finden. Daher sei nicht die Frage, ob selbstorganisierte Teams effektiver sind relevant, **sondern was selbstorganisierte Teams an einem Ende der Skala zu Teams am anderen Ende der Effektivitätsskala unterscheidet**. Was Magpilis und Pazos Frage nach Einflussfaktoren auf Leistung widerspiegelt (s.o. Abschnitt 2.3.1 und [MP18]). Eine selbstorganisierte Gruppe zu bilden, zu erhalten und darin zu arbeiten erfordert generell mehr Fähigkeiten und Engagement als in einer traditionellen Arbeitseinheit. Im weiteren Verlauf geht Hackman auf Bedingungen ein, die vor allem das erste Kriterium der Effektivität fördern in der Annahme, dass sich das positiv auf die beiden anderen Kriterien auswirkt. Hackman nennt fünf Bedingungen, die die Effektivität selbstorganisierter Teams fördern:

1. eine klare, motivierende Richtung,
2. eine leistungsfördernde Struktur der Arbeitseinheit,
3. eine unterstützende Organisationsstruktur,

4. Coaching durch Expert\_innen und
5. adäquate materielle Ressourcen.

Auf Punkt 2 bis 4 wird im Folgenden ausführlicher eingegangen, da die vorliegende Arbeit daran anschließt. Die weiteren Punkte werden ebenfalls kurz dargelegt.

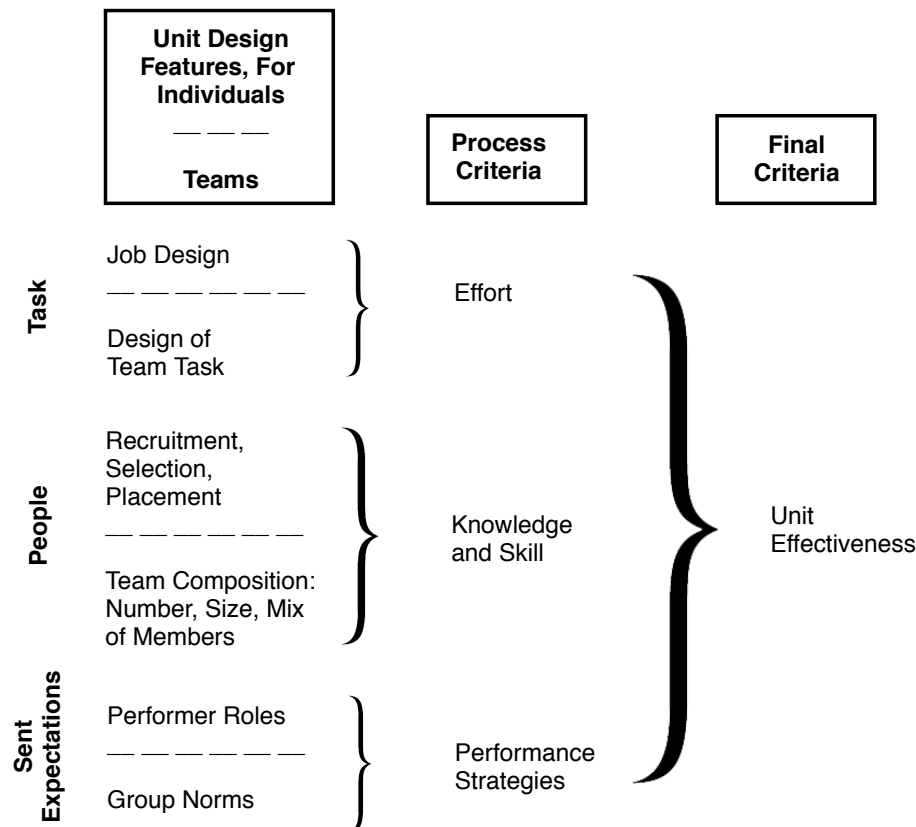


Abbildung 2.2: Prozess-Kriterien selbstorganisierter Teams entnommen aus Hackman [Hac86, S.110]

Das Management muss eine klare Richtung vorgeben und gleichzeitig den Teammitgliedern die Freiheit geben, eigenverantwortlich zu handeln. Die Waage zu halten ist dabei die große Herausforderung für Manager\_innen.

Nachdem eine Richtung und das Ziel vorgegeben sind, muss es der Arbeitseinheit ermöglicht werden zu arbeiten. Dabei sieht Hackman drei Hürden, siehe auch Abbildung 2.2, Mitte (Process Criteria):

1. Es braucht ausreichendes *Engagement* (effort), um die Aufgabe zu erledigen mit angemessener Leistung.
2. Es braucht entsprechendes *Wissen und Fähigkeiten*, die Aufgabe erfüllen zu können.
3. Es braucht entsprechende *Arbeitsstrategien*, die der Aufgabe und dem Umfeld angemessen sind.

Unter Arbeitsstrategien versteht Hackman wie die Arbeit praktisch erledigt wird: Wird in Untergruppen, gemeinsam oder nacheinander gearbeitet. Werden erst kreativ Lösungen erarbeitet und dann praktisch umgesetzt. Dabei bezieht Hackman natürlich auch Arbeitsstrategien ein, die nicht agil sind, da seine Forschung unabhängig agiler Strömungen entstanden ist. Die Prozess-Kriterien sind im späteren Rollenverständnis im Ergebnis dieser Arbeit wiederzufinden siehe Abschnitt 7.4.3.

Hackman empfiehlt bei Problemen im Team den drei Hürden entsprechend die Frage zu stellen, ob es ein Engagement-, Talent- oder Strategie- Problem ist, das das Team behindert. Eine Methode die als mögliches Reflexionstool in Abschnitt 8.4 aufgegriffen wird.

Diese drei Prozessindikatoren zur Einschätzung der Effektivität sind wichtig so Hackmann. Hackman sieht es jedoch kritisch, als Manager\_in in sie direkt einzugreifen, weil es die Selbstorganisation des Teams untergräbt. Er empfiehlt vielmehr sich auf folgende Eigenschaften zu fokussieren und sie im Team umzusetzen, da sie indirekt auf die drei Indikatoren wirken, siehe Abbildung 2.2, links:

- Aufgabengestaltung,
- Zusammenstellung des Teams und
- Erwartungen an das Verhalten von Teammitgliedern verdeutlichen.

Bei der *Aufgabengestaltung* sollten wiederum drei Kriterien erfüllt sein:

1. Die Aufgabe sollte für die Ausführenden *sinnvoll* sein. Das kann dadurch unterstützt werden, dass zur Erfüllung der Aufgabe (a) eine Bandbreite von ausgeprägten Fähigkeiten benötigt werden, (b) dass die Aufgabe ein abgeschlossenes Stück Arbeit mit Beginn und Ende ist, das ein konkretes wahrnehmbares Ergebnis hervorbringt und/oder (c), das Ergebnis wirkt sich auf andere Personen innerhalb oder außerhalb der Organisation aus.
2. Die Ausführenden sollten sich *persönlich verantwortlich* fühlen für das Ergebnis der Arbeit. Das ist gegeben, wenn die Aufgabe ein ausreichendem Maß an Entscheidungsautonomie erfordert und die Person sich nicht als ausschließlich ausführend und befehligt durch andere wahrnimmt.
3. Die Arbeit führt zu *Wissenszuwachs*. Das setzt voraus, dass während der Ausführung der Aufgabe vertrauensvolles und gehaltvolles Feedback zur Aufgabe und zum Arbeitsprozess gegeben wird.

Hackman widerspricht Annahmen, dass es Teams gibt, die eine grundsätzlich negative Arbeitshaltung haben, aus denen sich zumeist kontraproduktive Teamnormen ergeben. (Auch wenn er zustimmt, dass Menschen unterschiedliche Niveaus von Fleiß haben). Er geht davon aus, dass negative Teamnormen eher eine Folge von schlechten Arbeitsaufgaben sind. Er hält es daher für sinnvoller, die Arbeit neu zu gestalten, anstatt Teammotivation oder Teamnormen ändern zu wollen. **Diese Aussagen sind unter der Prämisse zu verstehen, dass soziale Systeme häufig vielfältigen, unvorhersehbaren, nicht unabhängigen Kräften ausgesetzt sind.** Er schlussfolgert daraus, dass eine gute Aufgabenstellung das Engagement (effort) positiv beeinflusst.

### 2.3. Arbeiten, die sich inhaltlich mit Teams befassen

Die Teamzusammensetzung wirkt sich auf den zweiten Prozessindikator (*Wissen und Fähigkeiten*) aus (siehe Abbildung 2.2). Dabei geht es nicht ausschließlich darum, dass es hoch qualifizierte Teammitglieder mit der Aufgabe entsprechenden Wissen im Team geben muss. In selbstorganisierten Team sind vor allem auch Fähigkeiten gefordert wie Verantwortung zu übernehmen und Entscheidungen zu treffen. Das Wissen und die Fähigkeiten der Teammitglieder wirken sich dabei direkt auf das Endergebnis aus.

Wäre ein Teammitglied beispielsweise dauerhaft überfordert, wirkt sich das direkt auf seine oder ihre Leistung aus. Daher ist es wichtig, Personen auszuwählen, die durch einen selbstorganisierten Kontext positiv motiviert werden. Außerdem sollten Komponenten wie Teamgröße, Homogenität und Heterogenität und die Kompetenz gemeinschaftlich zu arbeiten, berücksichtigt werden.

Die dritte Maßnahme, Erwartungen kommunizieren, dient der Verbesserung des dritten Prozessindikators (*angemessenes Arbeitsstrategien*). Dazu müssen sich die Teammitglieder verantwortlich fühlen und daher kontinuierlich ihren Arbeitsprozess bewerten und verbessern. Dazu bedarf es regelmäßiger Kommunikation der Erwartungen an das Team. Erwartungen können zum Beispiel durch formale Rollenbeschreibungen abgesteckt und kommuniziert werden. In selbstorganisierten Teams wird die Situation dadurch komplexer, dass Erwartungen zusätzlich durch sich entwickelnde Teamnormen verfestigt werden.

Hackman fasst an dieser Stelle zusammen, dass die drei Prozessindikatoren unterstützt werden können durch motivierende Aufgaben, ein gut zusammengestelltes Team und Strukturen, die regelmäßige Kommunikation zu Erwartungen und Selbstregulation ermöglichen. In den Ergebnissen der vorliegende Arbeit werden gerade die Strukturen zur Kommunikation der Erwartungen und der Selbstregulation innerhalb der Selbst- und Teamreflexion aufgegriffen (siehe Abschnitte 7.4.3.1 und 7.4.3.2). Interessant ist auch Hackmans Aspekt, das es im Team nicht nur hochqualifizierte Mitglieder bedarf. Ein Aspekt der in der späteren Hypothese 3 genauer beleuchtet wird.

Damit hat das Team eine unterstützende Struktur und die Richtung ist vorgegeben. Nun geht es darum eine sinnvolle Organisationsstruktur zu entwickeln.

Hackman bezieht sich erneut auf die drei Faktoren einer Organisation, die sich auf das *Engagement*, *Wissenszuwachs* und *Arbeitsstrategien* beziehen und auswirken.

Der erste Faktor ist das Belohnungs- oder Anreizsystem des Unternehmens. Dabei ist es grundsätzlich wichtig, dass Unternehmensangehörige ihren Beitrag zum Unternehmen wahrnehmen können und dafür belohnt werden. Daher sollten bei einem großen Unternehmen die Bewertungseinheiten des Anreizsystems kleiner als das ganze Unternehmen sein. Bei selbstorganisierten Teams ist es zusätzlich wichtig auf Teamebene zu bewerten und nicht Individuen, da das den Teamzusammenhalt gefährden könnte.

Der zweite Faktor ist das Bildungssystem im Unternehmen. Es müssen relevante Fortbildungsmöglichkeiten geschaffen werden, die auch für die entsprechenden Personen zugänglich sind. Wobei Hackman darauf hinweist, dass gerade Zweites nicht zu unterschätzen sei. Umstrukturierung hin zu Selbstorganisation bedeutet auch, dass Unternehmensmitglieder, die zuvor evtl. wenig bis keinen Zugang zu Fortbildungsressourcen hatten, diesen erhalten sollten, was gegebenenfalls zu politischen

Spannungen führen kann.

Der dritte Faktor ist das Informationssystem. Das Team braucht ausreichende Informationen, um die Aufgabe planen und angemessene *Arbeitsstrategien* zu entwickeln. Wichtige Informationen sind dabei

- Aufgabenanforderungen, Einschränkungen und Möglichkeiten, die positiven oder negativen Einfluss auf die Arbeitsstrategien haben,
- vorhandene Ressourcen und
- Personen, die das Produkt am Ende erhalten, beurteilen oder nutzen und gegebenenfalls die dazugehörigen Standards die angewendet werden.

Im folgenden Teil beschäftigt Hackman sich mit Coaching durch Expert\_innen. Hackman geht davon aus, dass unabhängig davon wie gut das Team ausgewählt wurde, die Passung zwischen Person und Aufgabe niemals perfekt ist. Daher ist es hilfreich jemanden zu haben, die den Personen hilft, ihr Verhalten zu beobachten und den ständig ändernden Aufgabenanforderungen oder Situationen anzupassen und angemessene Arbeitsstrategien zu entwickeln. Auch hier kann ein Coach auf drei Ebenen helfen:

- Engagement: Unterstützung bei der Motivation und Koordination untereinander und dem Verantwortungsgefühl der Arbeit und der Gruppe gegenüber.
- Wissenszuwachs: Unangemessene Überbewertung von Ideen und Beiträgen verhindern, die den Prozess behindern und stattdessen Expertise zu teilen und gegenseitiges Lernen zu unterstützen.
- Unterstützung dabei kreative Wege und Prozesse zu finden und nicht durch ungenaue Umsetzung eines Planes Prozessverluste zu erleiden.

Dabei ist es nicht Ziel, konfliktfreie reibungslose Zusammenarbeit zu schaffen. Die Arbeit kann durch die Überwindung von Konflikten an Kreativität und Gruppenzufriedenheit zugewinnen. Aufgaben die dem Scrum-Master zugeordnet sind und die im Modell (Kapitel 7) und in den späteren Hypothese 4 wiederzufinden sind.

Der letzte Punkt der fünf Bedingungen, die sich auf die Effektivität auswirken (siehe Seite 45) ist *adäquate Ressourcen*. Wie Hackman anmerkt, eigentlich ein leichter Punkt, der aber häufig von Unternehmen übersehen wird.

Im weiteren Teil seiner Arbeit erklärt Hackman, dass seine Arbeit einen eher allgemeinen Blick auf selbstorganisierte Einheiten wirft und nicht auf die Auswirkungen des Verhaltens Einzelner eingeht [Hac86, S.118]. Hinzu kommt ein weiterer Teil zu Führung von selbstorganisierten Teams, der hier weniger detailliert als die vorherigen Teile dargestellt wird. Zum einen, weil es das Thema der Arbeit nur am Rand betrifft, zum anderen, weil es in agilen Teams mit einem Scrum-Master und Product-Owner anders gehandhabt wird.

Er führt bezüglich Führen von selbstorganisierten Teams zuerst aus, dass es wesentlich herausfordernder ist, als in traditionellen Organisationen. Dabei ist es Aufgabe der Führungskraft zu überwachen und kontrollieren (monitor). Dazu gehört die



aktuelle und zukünftige Situation zu diagnostizieren. Eine weitere Aufgabe ist entsprechend zu handeln, um dem Team zu helfen, die gewünschten Ziele und Strategien umzusetzen. Diese beiden Aufgaben, Diagnostizieren und Handeln, müssen auf allen fünf Ebenen, die zu Effektivität beitragen, erfüllt werden (siehe auch Seite 45): Richtungsvorgaben, Struktur der Arbeitseinheit, Organisationskultur, Coaching und Ressourcen.

Abschließend stellt Hackman drei Qualitäten fest, die Führungskräfte in selbstorganisierten Unternehmen brauchen:

- **Mut:** Gegen den Strom schwimmen und unbequeme Wahrheiten ansprechen, Konflikte riskieren, sich persönlich unbeliebt machen, das Team aus negativen Verhaltensweisen herausbringen und zu mehr Effektivität hinführen.
- **Emotionale Reife:** Die Fähigkeit sich auf unsicherem Terrain zu bewegen und auf Ängste zuzugehen, anstatt sich von ihnen abzuwenden.
- **Werte:** Klare und angemessen persönliche Werte, die die Ziele der Organisation und das Wohlbefinden der Individuen gleichermaßen wertschätzen. Nur mit angemessenen Werten können ausgewogene Entscheidungen getroffen werden.

In seiner Zusammenfassung weist Hackman darauf hin, dass Effektivität nicht auf einzelne, direkte und leichte Weise beeinflusst werden kann. Es ist wichtig die Gesamtaspekte zu betrachten und zu fördern und keinen Aspekt zu vernachlässigen.

**Außerdem sind Personen in Teams unterschiedlich und keine Gruppe gleich. Selbst für zwei Gruppen mit denselben Aufgaben können zwei völlig unterschiedliche Strategien notwendig sein.**

#### **Exkurs 1: Hackman in einer agilen Welt**

Hackmans Arbeit ist vor dem Aufkommen agiler Arbeitsmethoden entstanden. Daher ist es an dieser Stelle notwendig, seine Forschung vor dem aktuellen Stand agiler Vorgehensweisen kurz zu diskutieren. Die Definition eines selbstorganisierten Softwareentwicklungsteams deckt sich insofern mit Hackmans, dass im Scrum-Guide selbstorganisierte Scrum-Teams folgende Eigenschaften zugeschrieben werden:

Selbstorganisierende Teams entscheiden selbst, wie sie ihre Arbeit am besten erledigen, anstatt dieses durch andere Personen außerhalb des Teams vorgegeben zu bekommen. [SS17b]

Hinzu kommt, das Scrum-Teams interdisziplinär sein sollen:

Interdisziplinäre Teams verfügen über alle Kompetenzen, die erforderlich sind, um die Arbeit zu erledigen, ohne dabei von Personen außerhalb des Teams abhängig zu sein. [SS17b]

Agile Arbeitsmethoden nehmen einige Anforderungen, die Hackman geäußert hatte auf bzw. vorweg: Die Teamgröße von 7-9 Personen, motivierte Teammitglieder (agiles Manifest) und eine Arbeitsstrategie die einen zyklischen Rahmen setzt (Sprints). Hinzu kommen Scrum-Master die coachen statt Manager.

Durch Meetings wie das Planning werden Informationen gegeben. Durch Retrospektiven wird regelmäßiges Feedback gegeben.

Im Scrum-Guide sind Werte enthalten, die auch Mut abdecken.

Hackman selbst erklärt abschließend, dass die Granularität seiner Theorien eher allgemein ist. (Team-) Ergebnisse lassen sich niemals endgültig erklären und ein reines Ursache-Wirkung Prinzip ist unterkomplex für Teams. Soziales Verhalten ist vielmehr komplex und nicht vorhersehbar. Seine Forschung deckt die individuelle Ebene und teamspezifische Faktoren nicht ab. Er stellt fest, dass es methodisch schwer zu beleuchten ist. Er schlägt gegebenenfalls eine Case Study vor oder die Entwicklung neuer Methoden, die sorgsam die Beleuchtung sozialer Systeme berücksichtigt.

## Exkurs 2: Warum Hackman nicht nur einfach validieren? Was ist der Mehrwert einer Grounded-Theory (GT)?

Erstaunlich ist, dass bisher niemand den Versuch gewagt hat, Hackmans Erkenntnisse in agilen Softwareentwicklungsteams zu validieren, obwohl er so oft (siehe oben) zitiert wurde. Das kann auf verschiedene Gründe zurückgeführt werden: Im ersten Schritt hätte grundsätzlich die Relevanz eines solchen Forschungsvorhabens erkannt werden müssen. Die Voraussetzungen der Teams von Hackmans und agiler Softwareentwicklungsteams sind äußerst unterschiedlich. Die Teamkontexte Hackmans und die agile Softwareentwicklungsteams liegen inhaltlich und vor allem zeitlich weit auseinander, was gegen eine direkte Validierung spricht. Erst die Ergebnisse der Forschung zu agilen Softwareentwicklungsteams auf abstrakter und theoretischer Ebenen führen zu Gemeinsamkeiten und ermöglichen die Erkenntnisse überhaupt zu vergleichen. Das ist vermutlich der Grund, warum Hackmans Arbeiten bisher vor allem zur Stütze der eigenen Erkenntnisse verwendet wurden. Eine Vergleichbarkeit lässt sich im Vorhinein nur schwer antizipieren. Daher findet ein Vergleich bisher meist retrospektiv oder deduktiv statt (Vergleich [MDD09] [MDD10] [LF18]).

Falls man doch die grundsätzliche Forschungsidee gehabt hätte - z.B. durch die Annahme, dass es unabhängig vom Kontext Ähnlichkeiten zwischen selbstorganisierten Teams gibt - bestünde immer noch die Herausforderung der Methodik, Hackmans abstrakte Aussagen in der Praxis agiler Softwareentwicklungsteam zu validieren.

Demgegenüber hat die in dieser Arbeit angewendete Methode (GTM) folgende Vorteile: Bei der reinen Validierung wäre ein klarer und enger Rahmen abgesteckt und die Datenerhebung wäre darauf reduziert und forciert. Unterschiede und Spezifika zu Softwareentwicklungsteams wären ggf. gar nicht erst erhoben oder nicht erkannt worden. Gleichzeitig wären eventuell Aspekte Hackmans abgefragt worden, die damit einen Stellenwert eingenommen hätten, der dem tatsächlichen Stellenwert nicht entspricht. In Charmaz' GTM, die hier angewendet wird, wird daher davon ausgegangen, dass der größere Teil der Literaturrecherche *nach* der Erhebung der Erkenntnisse gemacht wird. In GTM wird es als *forcing* verstanden, bestehende Erkenntnisse abschließlich validieren zu wollen.

GTM schließt methodisch eine Lücke: GTM geht nicht von einem Ursache-Wirkungs-Prinzip aus und ist Hackmans Forderung entsprechend darauf ausgelegt komplexe menschliche Zusammenarbeit zu beleuchten. Die Ergebnisse der vorliegenden Arbeit spiegeln daher einen Teil der komplexen Realität wider.

Der Vorteile der GTM sind:

1. GTM ermöglicht der komplexen Realität des zwischenmenschlichen Zusammenspiels näher zu kommen.
2. Bestehende Arbeiten aus anderen Disziplinen wie Hackmans dienen anschließend als Reflexionsfeld der eigenen Erkenntnisse. Die vorherige offene Herangehensweise, möglichst frei von bisherigen Ergebnissen, verhindert *forcing* und die damit einhergehende Reduzierung der Datenerhebung auf bestimmte Elemente. So können Spezifika von Softwareentwicklungsteams aufgedeckt werden.
3. Quellen wie Hackman, denen gegebenenfalls nicht die notwendige Relevanz zugeschrieben worden wäre, da veraltet und anderer Kontext, gewinnen Bedeutung und können gewinnbringend herangezogen werden.

Durch die Ergebnisse der vorliegenden Arbeit lässt sich zeigen, dass ein Teil von Hackmans Ergebnissen in agile Softwareentwicklungsteams in heutiger Zeit wiederzufinden ist. Dass diese Zusammenhänge und Kriterien erkannt wurden, ist erstaunlich und zeichnet gleichzeitig die hohe Qualität der vorliegenden Arbeit aus. Die Ergebnisse sind unabhängig von Hackmans Forschungen entstanden. Die Strukturen in selbstorganisierten Teams sind sehr spezifisch und das Erkennen von Ursache und Wirkung nicht unbedingt naheliegend. Die konsequente Anwendung der GTM hat zum einen zu bereits etablierten Ergebnissen geführt und diese gleichzeitig ausdifferenziert und zum anderen sie für den spezifischen Kontext weiterentwickelt.

Nachdem ein Überblick über die Leistung selbstorganisierter Teams in verschiedenen Disziplinen und eine ausführliche Darstellung selbstorganisierter Teams nach Hackman gegeben wurde, folgen nun einzelne Phänomene, die im Zusammenhang mit selbstorganisierten Softwareentwicklungsteams stehen.

### 2.3.3 Motivation

Im Bezug auf Magpili und Pazos [MP18] siehe Abschnitt 2.3.1, ist Motivation im Zusammenhang mit Selbstmanagement zu sehen und könnte sich auf die Leistung von selbstorganisierten Teams auswirken. Dasselbe gilt für Hackmans Strukturen wie *persönliches Verantwortungsgefühl* und *motivierende Richtung*, die sich auf die Effektivität der selbstorganisierten Teams auswirken, siehe Abschnitt 2.3.2.

Motivation ist ein wichtiges Element auf individueller Ebene im Rahmen der BSE so Lenberg u.a. [LFW15].

The question of what is significant for an individual's well-being and job satisfaction has been one of the most important research areas in organizational psychology since the 1920s [LFW15, S. 16]

Fehlende Motivation ist einer der wichtigsten Faktoren für Projektmisserfolg [BBH<sup>+</sup>08, S. 861]. Gleichzeitig ist sie in Bezug auf Softwareingenieur\_innen unzureichend beleuchtet und widersprüchlich, da sie sehr individuell ausgeprägt ist (ebd.).

Motivation und agile Arbeitsweisen stehen dabei in einem wichtigen Zusammenhang. Agile Methoden werden eingesetzt, um die Motivation der Teammitglieder

durch einen Zuwachs an Autonomie zu erhöhen. Ob die Motivation steigt, hängt auch davon ab, wie erfolgreich eine Implementierung agiler Methoden gelingt [dF12, S. 224].

Es gibt bereits ausführliche SLRs im Software Engineering zu Motivation [BBH<sup>+</sup>08] [SBB<sup>+</sup>09] [dF12]. Im Hinblick auf die vorliegende Arbeit lassen sich aus diesen Forschungsarbeiten folgende relevante Aussagen herleiten:

Motivation von Softwareingenieur\_innen ist

1. individuell sehr verschieden,
2. steht im Spannungsverhältnis zur Teambotivation,
3. und ist beliebig komplex, da sie von einer Vielzahl von Faktoren abhängig ist.

Diese Aussagen lassen sich aus den Forschungsarbeiten folgendermaßen herleiten: Zu Punkt 1: Um Motivation von Softwareingenieur\_innen zu erklären, stellt sich zuerst die Frage, ob sie eine eigene Gruppe bilden, die es zu erforschen gilt. Studien dazu sind widersprüchlich, ob Softwareingenieur\_innen eine eigene Gruppe bilden, die sich von Nicht- Softwareingenieur\_innen unterscheidet und damit spezielle Motivationsfaktoren bedarf. Beecham u.a. schließen daraus, dass es vom individuellen Kontext abhängt, ob sich Softwareentwickler\_innen als homogenen Gruppe empfinden:

These mixed findings from the 1980s to today lead us to conclude that whether or not software engineers form a homogeneous group with similar motivational needs depends on their individual context. [BBH<sup>+</sup>08, S. 871]

Beecham u.a. [BBH<sup>+</sup>08] haben 2008 einen umfangreichen Literatur-Review zu Motivation erhoben, aus dem Sharp u.a. [SBB<sup>+</sup>09] ein Modell zur Motivation von Software Ingenieur\_innen entwickelt haben. Dieses wiederum wurde von Da Silva u.a. [dF12] verfeinert. Beecham u.a. stellen bereits 24 Faktoren vor, die Personen inhärent sind und ihre Motivation modulieren und schließen daraus:

This structure implies a different profile of characteristics for every individual Software Engineer, [BBH<sup>+</sup>08, S. 866]

Motivation muss zusätzlich im Kontext von Rollen, Zeit, Kultur, Erfahrung und Alter betrachtet werden: Als Beispiel führen sie an, dass es in den 90er aufregend und motivierend war, mit objektorientierten Sprachen zu programmieren. Das gilt heute nicht mehr.

Motivation ist somit ausgesprochen individuell und kontextabhängig.

Zu Punkt 2: In Beecham u.a. wird darauf hingewiesen, dass Unterschiede in den Persönlichkeiten größer sind als die Gemeinsamkeiten mit anderen und daher nicht ignoriert werden können [BBH<sup>+</sup>08, S. 872]. Da Silva u.a. [dF12] beziehen sich auf eine Studie aus ihrem Review und erklären, dass individuelle Faktoren und Teamfaktoren sich unterscheiden und Motivationsstrukturen dadurch komplexer werden.

[...] the motivation process of teams is even more complex to deal with than individual motivation, since there may be team-level motivators which differ from individual-level motivators. [dF12, S. 217]

Da Silva u.a. [dF12] und Beechem u.a. [BBH<sup>+</sup>08] unterstreichen damit, dass individuelle und Teammotivation in einem komplexen Spannungsverhältnis stehen.

Zu Punkt 3: Die SLRs und Arbeiten von Beechem u.a. [BBH<sup>+</sup>08] und Sharp u.a. [SBB<sup>+</sup>09] und da Silva u.a. [dF12] bauen jeweils aufeinander auf. Sie verschaffen sich über die Vielzahl von existierenden Modellen zu Motivation von Softwareentwickler\_innen einen Überblick, versuchen sie zu vereinheitlichen und fügen dem vorherigen Modell weitere Komplexität hinzu.

Die Modelle betrachten dabei vor allem Motivationsfaktoren und nicht den Umgang im Team mit (fehlender) Motivation. Das Modell dieser Arbeit greift den Konflikt zwischen Selbst- und Teammotivation auf und analysiert seine Folgen für das Team und den Teamzusammenhalt (siehe Kapitel 7). Der Konflikt zwischen Selbst- und Teammotivation wird dabei in vielen Episoden deutlich und konnte vielfach analysiert werden (siehe Abschnitt 7.2).

### 2.3.4 Reflexion

Reflexion ist ein wichtiger Bestandteil in der agilen Softwareentwicklung und ebenso ein wichtiger Teil des späteren Modells. In der Einleitung zur Ausgabe *The reflective Software Engineer* betonen Dybå u.a. wie wichtig und schwierig Reflexion in der Softwareentwicklung ist [DMG14]. Prior u.a. [PFL16] versuchen daher schon ihren Studierenden Reflexion während der Softwareentwicklung näher zu bringen ((R1), (G2-Daten)). Sie haben ein Modell entwickelt, in dem die Studierenden auf vier Ebenen Fragen in einem Reflexionstagebuch beantworten sollen:

**Bericht** Was ist passiert?

**Reaktion** Wie hast Du dich dabei gefühlt?

**Analyse** Warum ist es passiert oder warum hast du so reagiert?

**Verbesserung** Was hast du davon gelernt? Was würdest du das nächste Mal ändern?

Babb u.a. [BHN14] haben Reflexionstechniken in agilen Softwareentwicklungsteams theoretisch nach Schöns's Kategorien „*reflection in action*“ und „*reflection on action*“ [Sch83] untersucht ((R1), (G1-Methode)). Sie haben daraus ein „*Reflective Agile Learning Modell (REALM)*“ entwickelt (siehe Abbildung 2.3). Sie ordnen den agilen Praktiken Möglichkeiten zu, worüber in der jeweiligen Situation reflektiert werden kann. Gegenstand der Reflexion sind dabei projektbezogene Inhalte wie eine gemeinsame Produktvision zu schaffen oder Inhalte auf Wissen und Fähigkeiten bezogen.

Andriyani [AHA17] hat Reflexion in Retrospektiven in agilen Softwareentwicklungsteams untersucht. Methodisch grenzt (G1-Methode) sich die Arbeit folgendermaßen ab: Andriyani erhebt eine Fallstudie und beobachtet und interviewt vier agile Softwareentwicklungsteams. Die Interviewfragen sind dabei sehr zielgerichtet auf Reflexion ausgerichtet, beispielsweise: „*Based on the three main points discussed in a retrospective (i.e. what went well, what went wrong, what can you improve), which one(s) do you think are most helpful for your team's reflection?*“. Die Beobachtungen beziehen sich lediglich auf eine Retrospektive pro Team. Alle vier beobachteten Teams kommen aus derselben Firma (G2-Daten).

Andriyani unterscheidet drei Ebenen der Reflexion:

Learning and reflection opportunities  
in the Reflective Agile Learning Model (REALM).

Agile practices	Reflection-in-action
Release planning	Learning about the project, domain, and product
Collective estimation and iteration planning	Learning about each other's skills and areas of expertise, and the team's collective capacity to delivery (velocity)
Group programming	Learning new skills and expertise from each other
Pair-in-need	Novices learning about the team and project as they pair with experts; pairs learning a new technology or skill as they work on a challenging task
Project wiki	Capturing project-based learning about the domain and individual reflections (blogs)
Demo and acceptance testing	Learning from acceptance test failures that fuel refinements and guide product vision
Agile practices	Reflection-on-action
Daily standup	Reflection on action performed the previous day, planning for the day's actions, and highlighting impediments
Learning spike	Adapted practice that embodies reflection on action as some members devote exclusive time to new learning
Retrospective	Standard agile practice devoted exclusively to reflection on actions performed in previous iteration

Abbildung 2.3: Reflexionsmöglichkeiten im Scrum Prozess, entnommen aus Babb u.a. [BHN14]

**Reporting and responding** bedeutet, dass Teammitglieder ihre Erfahrungen schildern und teilen.

**Relating and reasoning** bezieht sich darauf Geschehnisse bewerten zu können und einzuordnen.

**Reconstructing** bedeutet konkret Schlüsse aus der Reflexion zu ziehen und einen Plan zu entwickeln.

Andriyani stellt in der Praxis der agilen Softwareentwicklungsteams fest, dass die Teams auf der Ebene des *reporting and responding* sich über Abhängigkeiten, Schwierigkeiten bei der Aufgabenabarbeitung und Gefühle während des Sprints auf Grund von Ereignissen austauschen.

Auf Ebene des *relating and reasoning* reflektieren die Teams über vorherige *Action Points* aus dem letzten Sprint, identifizieren Gründe für aktuelle Probleme und leiten daraus zukünftige *Action Points* ab.

Abschließend wird während der Retrospektive auf der Ebene des *Reconstructing* ein Plan für den nächsten Sprint entwickelt.

Andriyani schlägt vor, die Ebenen der Reflexion den fünf Phasen der Retrospektive - set the stage, gather data, generate insight and decide what to do, close the retrospective - zuzuordnen. Sie stellt fest, dass nicht alle beobachteten Teams jede Ebene der Reflexion erreichen, insbesondere die dritte Ebene wird nicht immer erreicht und die Ebene der Gefühle wird vernachlässigt. Sie empfiehlt sicher zu stellen, dass während der Retrospektive alle Ebenen der Reflexion erreicht werden.

Das spätere Modell dieser Arbeit greift Reflexionen auf Selbst- und Teamebene auf [7.4.3.2](#).

### 2.3.5 Hodas Grounded-Theory zu agilen Softwareentwicklungsteams

Hoda [Hod11] hat agile Methoden in ihrer Dissertation und darüber hinaus ausführlich erforscht. Aus ihrer Doktorarbeit sind verschiedene Publikationen hervorgegangen [HNM10a] [HNM10b] [HNM11] [HNM12] [HNM13].

Sie hat mit Grounded-Theory-Methodologie gearbeitet und sich mit der Anwendung im Software Engineering auseinandergesetzt [HNM10b] [HN17]. Sie hat einen wichtigen Beitrag geleistet zur Anwendung der Methode im Software Engineering. Sie erklärt detailliert die praktische Umsetzung und mögliche Probleme. Die Unterschiede und Gemeinsamkeiten im Vorgehen der Methodik (**G1-Methode**) werden bei der konkreten praktischen Umsetzung der Methode im jeweiligen Abschnitt in Kapitel 6 deutlich.

Hoda untersucht agile Teams mit Grounded-Theory-Methodologie (GTM). Damit gleichen sich Methodik und Forschungsfeld zur vorliegenden Arbeit (**R1**). Grenzen bestehen bei genauerer Betrachtung bezüglich Kontext (**G2-Daten**) und Inhalt und Art der Ergebnisse (**G3-Inhalt**).

Hoda untersucht agile Teams in Neuseeland und Indien nach dem verstärkten Aufkommen agiler Methoden ca. zwischen 2007 und 2010 (**G2-Daten**)<sup>5</sup>. Die Arbeit setzt immer wieder den Vergleich zu Wasserfallmethoden und betont die Unterschiede. Das Anliegen ist grundsätzliche Mechanismen von Scrum zu erklären. Es werden Prozesse wie die Erstellung von User-Stories, Management und die Einbindung der Kunden immer wieder erklärt und angeführt. Das zeigt, dass zu diesem Zeitpunkt agiles Vorgehen noch keine Selbstverständlichkeit war. Agile Methoden werden positiv dargestellt im Vergleich zum Wasserfall. Die vorliegende Arbeit knüpft an Hodas Arbeit an. Sie unterscheidet sich im Kontext, sowohl zeitlich als auch örtlich. Agile Methoden sind inzwischen selbstverständlich. Gleichzeitig hat sich die ausschließliche positive Sicht relativiert. Agile Methoden gehen über allgemeine Tools und Mechanismen hinaus. Nach der Etablierung der Prozesse ergeben sich vor allem Hindernisse in der Zusammenarbeit und damit menschliche Aspekte. Das ist der Schwerpunkt der vorliegenden Arbeit (**G3-Inhalt**).

Inhaltlich werden hier zwei für die vorliegende Arbeit wichtige Veröffentlichungen vorgestellt. Die erste Veröffentlichungen beschäftigt sich mit grundsätzlichen Mechanismen in agilen Teams [HNM10a]. Die zweite Veröffentlichung beschäftigt sich mit Rollen in agilen Teams [HNM13]. Dabei war Hodas erste wichtige und zentrale Erkenntnis, dass agile Teams *selbstorganisiert* sind und dass dieses prägende Merkmal ausschlaggebend für die Betrachtung agiler Teams ist <sup>6</sup>.

Hoda beschreibt als zentrales Element selbstorganisierter Teams „*Balancing acts*“. Sie erklärt, dass Teammitglieder zwischen folgenden Faktoren eine Balance finden müssen:

- Freiheit und Verantwortung
- Crossfunktionalität und Spezialisierung

<sup>5</sup>Hoda erwähnt in ihrer Arbeit 2010 [HNM10a], dass sie 2,5 Jahre Teilnehmer\_innen interviewt hat. Daraus abgeleitet ergibt sich ein vermutlicher Zeitraum.

<sup>6</sup>Dies spiegelt sich auch in den in dieser Arbeit vorgestellten und diskutieren verwandten Arbeiten wider. Ein Großteil beschäftigt sich aus verschiedenen Disziplinen mit selbstorganisierten Teams siehe Abschnitt 2.3.1 und 2.3.2.

- kontinuierliches Lernen und Iterations-Druck

Mit Selbstorganisation, so die Teilnehmer\_innen aus Hodas Studie, geht mehr Freiheit einher, die Arbeit und dabei konkret die Aufgabenabarbeitung eigenständig zu handhaben. Das erfordert mehr Selbstkontrolle und -management und ein erhöhtes Verantwortlichkeitsgefühl [HNM10a, S. 6f]. Dabei stellt Hoda Teamziele und „Self-Assignment“ (Selbstzuweisung der Arbeitsaufträge) als Gegensatzpaar dar.

Ein Teamziel für einen Sprint festzulegen ist dabei eine *Gruppenaufgabe*. Das Team hat dabei die *Freiheit* selbst den Umfang zu bestimmen. Welche Aufgaben von wem abgearbeitet werden, sind individuelle Entscheidungen, da die Teammitglieder selbst wählen können. Mit dieser Freiheit geht Verantwortung einher bei gleicher Priorität und Business Value einer Aufgabe, die richtige zu wählen und abzuarbeiten und auch andere Teammitglieder die Möglichkeit zu geben, sich zu entwickeln und gleichzeitig das Teamziel zu erreichen.

Hoda erklärt anhand eines Beispiels, in dem Senior-Entwickler\_innen Entscheidungen alleinig übernommen hatten, dass die Selbstorganisation untergraben wird. Andere Teammitglieder wurden unzureichend in die Entscheidungen eingebunden. In diesem Fall hat das Management eingegriffen und die Senior-Entwickler\_innen versetzt.

Beim zweiten *Balancing Act* von Hoda geht es darum, dass agile Teams möglichst crossfunktional also interdisziplinär zusammen arbeiten sollten. Sie gibt Beispiele in denen *Agile Coaches* testen oder *Business Analysten* leichte Programmieraufgaben übernehmen [HNM10a, S. 8] und nennt dies Crossfunktionalität über funktionale Rollen hinweg. Gleichzeitig gibt es Crossfunktionalität auch innerhalb einer funktionalen Rolle. Insbesondere bei Entwickler\_innen, die sich in möglichst allen technischen Bereichen der Software soweit auskennen sollten, dass sie Aufgaben abarbeiten können. Das bedeutet beispielsweise, dass Entwickler\_innen sowohl Datenbanken als auch GUI-Entwicklungsaufgaben übernehmen können. Dabei fühlen sich die Entwickler\_innen meist in einem Bereich (ihrer Spezialisierung) sicherer, als in anderen Bereichen, in denen sie weniger arbeiten.

Hoda führt ein Beispiel an, in dem die Balance in ein Missverhältnis gerät, als ein *Business Analyst* unabgesprochen Programmieraufgaben übernimmt und dem Code Schaden zufügt. In diesem Fall musste ein *Agile Coach* eingreifen.

Im dritten *Balancing Act* geht es darum, dass unter hohem Druck am Ende des Sprints bzw. Iteration ein Ergebnis gefordert ist. Gleichzeitig erfordert die Komplexität der Aufgaben und die Anforderung der Crossfunktionalität von den Teammitgliedern ständiges Lernen und Selbstoptimierung. Unter Zeitdruck, Ressourcen zu finden für Lernmöglichkeiten ist schwierig. Laut Hoda kann durch Paarprogrammierung und Retrospektiven eine Balance gefunden werden. So können schwierige Aufgaben gemeinsam gelöst werden und bieten Möglichkeiten voneinander zu lernen und gleichzeitig Hürden bei der Erreichung des Sprintziels zu meistern. In den Retrospektiven kann über Lernerfolge reflektiert und die Planung für eine Balance optimiert werden. Hoda gibt hier ein Beispiel, in dem ein\_e Tester\_in das Team verlässt und das Team daraufhin in der Retrospektive Maßnahmen festlegt, die ermöglichen dass andere Teammitglieder sich die nun fehlenden Fähigkeiten aneignen.

Alle drei *Balancing Acts* wirken sich jeweils auf einen wesentlichen Bestandteil



selbstorganisierter Teams aus [HNM10a].

- Autonomie wird bestimmt durch die Balance zwischen Freiheit und Verantwortlichkeit.
- Gegenseitige Befruchtung (Cross-Fertilization) wird bestimmt durch die Balance zwischen Spezialisierung und Crossfunktionalität.
- Selbst-Transzendenz wird bestimmt durch kontinuierliches Lernen und Iterationsdruck.

Damit beschreibt Hoda die Balance der jeweiligen Gegensatzpaare als prägend für die erfolgreiche selbstorganisierte Zusammenarbeit in agilen Softwareentwicklungsteams. Die Gegensatzpaare sind indirekt teilweise auch im Modell wiederzufinden wie später in der Diskussion deutlich wird siehe Abschnitt 7.4.

In einer weiteren Publikation, die auf denselben Daten und Methodik beruht, erklärt Hoda Rollen in agilen Softwareentwicklungsteams. In Hodas Rollenverständnis sind Rollen informell, implizit, durchlässig und spontan [HNM13, S. 422]. Sie benennt sechs konkrete Rollen:

**Mentor\_in** führt und unterstützt das Team bei der Einführung agiler Methoden, ermutigt und bestärkt das Team bei der stetigen Verbesserung.

**Koordinator\_in** verhandelt als Repräsentant\_in des Teams Kundenerwartungen und koordiniert die Zusammenarbeit zwischen Kund\_innen und Team.

**Übersetzer\_in** versteht sowohl die Kund\_innensprache, die auf Geschäftsprozessen beruht als auch die technischen Terminologien des Softwareentwicklungsteams. Sie verbessert die Kommunikation zwischen beiden Gruppen.

**Meister\_in** sichert die Unterstützung des Senior Managements für das agile Team.

**Förder\_in** sichert die Unterstützung der Kund\_innen für die agile Zusammenarbeit.

**Terminator\_in** sorgt dafür, dass Teammitglieder, die die agile Teamzusammenarbeit gefährden, vom Senior-Management aus dem Team entfernt werden. [HNM13, S. 423]

Hoda erklärt, dass je nach Situation und Notwendigkeit ein und dieselbe Person in unterschiedlichen Situationen verschiedene Rollen annehmen kann [HNM13, S. 439]. Sie stellt fest, dass ähnliche Rollen bereits gefunden wurden. Hodas Rollen berücksichtigten allerdings die Besonderheiten agiler Softwareentwicklungsteams. Dabei definiert sie die Rollen als durchlässig (siehe oben) und erklärt, dass durch Rol-  
lendefinition ein Team selbstorganisiert wird [HNM13, S. 440].

In der Praxis sollen die definierten Rollen Mitglieder eines Teams dabei unterstützen, ihre eigene Rolle in der agilen Zusammenarbeit besser zu verstehen. Hoda erarbeitet Vorschläge welche Personen dabei, welche Rollen annehmen können. Beispielsweise brauchen Teammitglieder, die mit der Adaption von agilen Methoden hadern, eine Mentorin. Oder sie sollten von einer Terminatorin ausgeschlossen werden [HNM13, S. 440], wenn es ihnen misslingt agile Methoden zu adaptieren. *Agile*

*Coaches* sollen dabei unterstützen, dass das Team die Verantwortung der jeweiligen Rollen erkennt und sie einnimmt. Rollen dürfen nicht vorgeschrieben werden, sondern Teammitglieder sollen ermutigt werden, sie anzunehmen.

Das im Ergebnis dieser Arbeit vorgestellte Rollenmodell teilt Hodas grundsätzliche Idee und entwickelt diese weiter, in dem keine Rollen mehr benannt werden, sondern eine Rollendefinition erarbeitet wird (siehe Abschnitt 7.2.4 und Abschnitt 7.4.3).

### 2.3.6 Forschungsgruppe um Nils Brede Moe, Torgeir Dingsøy und Tore Dybå

Die Forschungsgruppe um Nils Brede Moe arbeitete über einen längeren Zeitraum zusammen bei SINTEF Information and Communication Technology und haben jeweils Abschlüsse von der Norwegian University of Science and Technology erhalten. Ihre Erkenntnisse beruhen auf einem Datensatz, der im Laufe der Zeit erweitert wurde.

Moe u.a. Arbeiten [MDD08] [MD08] [MDD09][MDR09] [DD12] [MDD10] [MAD12] [MFCF16] [GMD11] sind eine Quelle relevanter Erkenntnisse für meine Arbeit (R1). Lässt man dabei außer acht, dass Moe u.a. Teams im agilen Transformationsprozess erforscht hat und Hinweise für eine gelungene Transformation gesucht hat, gibt es für die vorliegende Arbeit inhaltlich interessante Teamaspekte, die auch unabhängig von einer Transformation interessant und relevant sind. Diese Differenzierung beschreibt auch die Abgrenzungen dieser Arbeiten: Sie wurden in unterschiedlichen zeitlichen und inhaltlichen Kontexten erhoben (G2-Daten), weisen aber im Ergebnis inhaltlich Ähnlichkeiten auf (G3-Inhalt).

In der Datenerhebung und Auswertung unterschieden sich die Arbeiten folgendermaßen (G1-Methode): Moe u.a. arbeiten an einer Case-Study und analysieren die Daten entsprechend. Bezüglich des Kontext der Datenerhebung (G2-Daten) erklärt Moe u.a. in [MAD12], dass er Interviews und Beobachtungen in 4 Teams über 11-12 Monate erhoben hat. Er hat umfangreiche Interviews geführt, Beobachtungen notiert und Meetings aufgenommen und transkribiert. Ohne genaue Angaben über den Umfang der Observationsnotizen (die eine Zusammenfassung beinhalten) zu haben, scheint es so, als wäre die Quantität der Daten hoch. Moes Daten stammen ausschließlich aus Teams die mit Scrum beginnen und sind vor ca. 10 Jahren entstanden. Moe u.a. Ergebnisse stellen häufig Anfangsschwierigkeiten beim Transformationsprozess dar, die sich auf das grundsätzliche Verständnis von Scrum beziehen (z.B. Welche Meetings werden abgehalten oder eine fehlende *Definition of Done*). Die Ergebnisse sind sehr am Scrum-Prozess orientiert: Wo entstehen Probleme beim Abstimmen von Management-Zielen und Team-Ressourcen? Wie sollte ein *Product-Backlog* besser gestaltet werden?

Im Vergleich zu Moes u.a. Arbeiten stammen die Daten in der vorliegenden Arbeit aus einer höheren Bandbreite unterschiedlicher Unternehmen, alle Teams arbeiteten bereits seit einer Weile agil und haben einen agilen Arbeitsalltag (Change- Prozess weitestgehend abgeschlossen)(G2-Daten). Die in der vorliegenden Arbeit verwendet GTM greift tiefer in die Daten (G1-Methode) ein und bringt daher weniger Ergebnisse hervor, die sich auf Pläne, Business-Prozesse oder den Umgang mit Werkzeugen

oder Artefakten beziehen (**G3-Inhalt**). Die Zusammenhänge in den Ergebnissen (**G3-Inhalt**) werden ausführlich im Ergebnisteil 7 diskutiert.

Moe u.a. beziehen sich auf Langfred [Lan00], dessen Forschungsergebnisse Autonomie in Teams beschreiben. Langfreds Forschung beruht auf Untersuchungen in einer öffentlichen Einrichtung und dem Militär und kommt unter anderem zu folgendem Ergebnis:

As this study has shown, autonomy at the individual level may conflict with autonomy at the group level [Lan00, S. 581]

Moe u.a. untersuchen in ihrer Publikation [MDD08] Autonomie in einem selbstorganisierten Softwareentwicklungsteam anhand einer Fallstudie. Sie kommen zu dem Ergebnis, dass es eine Herausforderung ist eine Balance zu finden zwischen individueller und interner Autonomie. Individuelle Autonomie bedeutet, dass Teammitglieder eigenständig über ihre Aufgaben und deren Abarbeitung entscheiden. Interne Autonomie bezieht das Team mit ein und geht davon aus, dass Entscheidungen durch das Team getroffen werden (wobei das Team auch gemeinsam entscheiden kann, die Entscheidung einer einzelnen Person zu überlassen). Beide Arten der Autonomie sind in agilen Softwareentwicklungsteams vorhanden, wobei in der Theorie die individuelle Autonomie sinkt zugunsten gemeinsamer Teamziele und Aufgaben. Moe. u.a. beschreibt, dass es in dem untersuchten Team Schwierigkeiten gab, individuelle Autonomie zu reduzieren. Softwareentwickler\_innen waren es gewohnt, für ein Modul verantwortlich zu sein und hatten ein entsprechend spezialisiertes Wissen. Unter Zeitdruck gelang es nicht, diese Spezialisierung zu durchbrechen zugunsten von interner Autonomie. Ein solches Verhalten beschreiben Moe u.a. als gegenläufig zur Selbstorganisation:

Thus the highly specialized skills and corresponding division of work resulted in a lack of redundancy. This lack of redundancy reduced the flexibility and, thus, the internal autonomy of the team, and was therefore the most important barrier for self-organizing. [MDD08, S. 83]

Durch redundantes Wissen (es gibt mehr als eine Person, die Wissen besitzt), entsteht Zusammenarbeit, Teammitglieder können sich gegenseitig in ihrer Arbeit unterstützen und flexibel Aufgaben unter sich aufteilen. Fehlt Redundanz entsteht keine tatsächliche Selbstorganisation. Aufgaben werden lediglich nach Wissen aufgeteilt und individuell abgearbeitet (individuelle Autonomie).

In einer weiteren Publikation [MDD09] wurde der Datensatz erweitert und das Ergebnis fließt als ein Teil zu Überwindung von Hindernissen auf dem Weg zu Selbstorganisation ein. In dieser Publikation werden die unterschiedlichen Level von Autonomie als eine von drei wichtigen Hürden auf dem Weg zu selbstorganisierten Teams dargestellt. Sie beschreiben erneut, dass Entwickler\_innen - die zuvor sehr eigenständig gearbeitet haben - Schwierigkeiten haben, Verantwortung, Arbeitsprozess und Entscheidungen mit dem Team zu teilen, da sie vorzugsweise bis ausschließlich an *„ihrem“* Modul der Software arbeiten.

Das Modell dieser Arbeit beschreibt wie sich die Autonomiekonflikte im Team auswirken und welche Folgen sie haben (können) siehe Abschnitt 7.4 und werden in einer Vielzahl der Episoden, die zum Ergebnis geführt haben sichtbar siehe Abschnitt 7.2.

## 2.3.7 Verwandte Arbeiten aus entfernten Forschungsrichtungen

### 2.3.7.1 Anthropologie: Teams in extremen Umgebungen

Jeffrey C. Johnson ist Professor der Anthropologie an der *University of Florida* und Professor am *Institute for Software Research* an der *Carnegie Melon University*. Er hat eine große Themenbandbreite in seinen Veröffentlichungen. Neben Forschung zur Methodik wie Interviewtechniken und Ethnografie, Teams in isolierten Umgebungen publiziert er auch technische und medizinische Forschungsergebnisse.

Im Folgenden wird eine Publikation [JBP03] von ihm inhaltlich dargestellt, da sie später zur Diskussion der Inhalte beiträgt (R2). Sie dient damit auch als relevantes Beispiel aus dem Forschungsfeld der Anthropologie. Zuvor zeige ich die Grenzen zur vorliegenden Arbeit im methodischen Vorgehen auf (G1-Methode) und ergänze es um ein weiteres Beispiel aus der Psychologie und Soziologie, das stellvertretend für einen Teil dieser Forschungsrichtung steht.

Das methodische Vorgehen unterscheidet sich sehr stark zu dem der vorliegenden Arbeit. In der Arbeit von Johnson u.a. und auch in den Arbeiten auf die er sich bezieht, werden die Teammitglieder üblicherweise hierarchisch sortiert oder Personen zu Rollen zugeordnet, in dem Sätze mit Namen vervollständigt werden wie „... ist eine gute Führungskraft“. Ein Vorgehen das „assigning types to People“ von Charmaz entspricht. Charmaz lehnt ein solches Vorgehen ab, da es nicht die komplexe Realität abbildet siehe auch u.a. Seite 80.

Hinzu kommen Erhebungen zu Interaktionen zwischen den Teammitgliedern. Dafür werden sie aufgefordert, zu unterschiedlichen Zeitpunkten auf einer Skala zwischen 1 und 10 anzugeben wie häufig sie mit jedem Teammitglied interagiert haben. Methodisch entspricht das einem quantitativen Vorgehen.

Was beide zuvor erklärten Vorgehensweisen gemein haben, ist dass sie das Verständnis der untersuchten Personen nicht hinterfragen. Beispielsweise wird nicht ausdifferenziert wie eine gute Führungsperson von verschiedenen Personen definiert wird, was als Interaktion gewertet wird und was *viel* und was *wenig* Interaktion ist. Diese Annahmen können von Person zu Person sehr unterschiedlich sein. Auch wenn die Methodiken auf vorherigen Interviews mit anderen Personen beruhen, zielt diese Art der Methodik eher darauf ab, Veränderungen in einem System aus Personen zu erheben, als die qualitative Differenzierung einzelner Phänomene. Darin unterscheidet sich die vorliegende Arbeit in der Methodik (G1-Methode).

Auch in einem Review aus der Forschungsrichtung der Psychologie und Soziologie von Ramos-Villagrasa u.a. [RVMQNR18] werden die Unterschiede in der Methodik deutlich, die häufig auf Statistiken beruhen. In dem Review wird gezeigt, dass Teams komplexe adaptive Systeme sind, was von der Annahme her der vorliegenden Arbeit nahe kommt. Daraus wird geschlossen, dass Daten quantitativ zu unterschiedlichen Zeitpunkten und Ausmaß erhoben werden müssen (G2-Daten). Das grenzt sich klar von dem empirischen Vorgehen der GTM ab.

Johnson u.a. [JBP03] untersuchen Teams in extremen Situationen im konkreten Polarexpeditionen, um das Wissen auf Marsflüge anwenden zu können. Sie unterscheidet sich damit maßgeblich im Kontext (G2-Daten).

Nach dieser methodischen Abgrenzung wird nun die Forschung Johnsons inhaltlich dargestellt, da sie inhaltlich später zur Ergebnisdiskussion beiträgt. Alle folgen-

den Aussagen beziehen sich auf [JBP03], falls nicht anderes angegeben.

Johnson u.a. merkt an, dass bisher wenig über die strukturellen Faktoren wie Position, Status, Rolle und die Entwicklung von Gruppennormen in (isolierten) Teams erforscht wurde, da sich bisher auf Gesundheit und psychologische Probleme konzentriert wurde. Strukturelle Probleme werden ab einer Gruppengröße über 7 Personen relevant und sind daher auch in der vorliegenden Forschung relevant.

Führung ist ein entscheidender Faktor für kleine Gruppen so Johnson u.a. [JBP03] und wurde intensiv und ausführlich erforscht. Dazu gibt es verschiedene Modelle und Theorien. Ein interessanter Aspekt ist, dass in den Führungstheorien unterschieden wird zwischen informeller (emergent) und formeller Gruppenstruktur. Johnson bezieht sich auf Smith und erklärt, dass informelle Rollen im Laufe der Zeit formelle Rollen ersetzen [Smi66].

Klassische Rollen sind in der Führungsforschung „Leaders“ und „Followers“. Dabei decken diese Bezeichnungen nicht die ganze Bandbreite möglicher Rollen ab. Insgesamt ist es schwierig, alle formalen Stellungen und Rollen in Gruppen abzudecken. Johnson bezieht sich auf verschiedene Forschungen [JJ71] [JQJ72] [JTG80] [KC69] (Basketball-Teams, Militär) und erklärt, dass erfolgreiche Teams einen hohen Konsens über ihre Rollen haben. Außerdem werden unterschiedliche Erwartungen besser erfüllt, wenn es eine erhöhte Bandbreite an Vorlieben für Aufgaben gibt. Es entstehen Rollenkonflikte, wenn Individuen überschneidende Aufgabenfelder für sich beanspruchen. Ein Maß an Heterogenität ist daher vorteilhaft für Teams. Rollen werden dabei unter anderem durch Heterogenität und Persönlichkeit gebildet. Weitere wichtige Aspekte sind Kommunikation, Interaktion und gegenseitiges Bestärken, da sie zu einer guten Moral und hohen Effektivität führen. Gute Kommunikation resultiert wiederum aus heterogenen Rollen (untersucht am Beispiel von Basketballteams).

Im Bezug auf Teams in extremen Umgebungen gibt es zwei Arten von Abweichler\_innen. Abweichler\_innen haben einen niedrigen Gruppenstatus. Die ersten weichen durch negatives Verhalten ab wie Alkoholmissbrauch, nicht Erreichen von Arbeitsroutinen und Ähnlichem. Sie untergraben die Arbeitsmoral und schädigen das Team. Gleichzeitig schweißen sie das Team zusammen, indem durch sie Grenzen und akzeptables und inakzeptables Verhalten definiert und damit Gruppennormen forciert werden.

Eine weitere Gruppe von Abweichler\_innen sind die Gruppenclowns. Die häufig nicht aus der Forschergruppe kommen, sondern aus der Zivilbevölkerung wie die/der Köch\_in. Sie lösen Stress und sind in der Lage, Unzufriedenheit über Fehlverhalten anderer zu kommunizieren durch ihren Humor. Wenn das richtige Maß eingehalten wird, unterstützt Humor die Moral. Clown\_innen schlagen Brücken zwischen verschiedenen Gruppen.

In der Forschung gibt es unterschiedliche Ansichten zu Gruppenstabilität, ihren Auswirkungen und wie sie erreicht wird. Johnson geht von der Hypothese aus, dass je stabiler die Rollen und die Übereinstimmung von informeller und formeller Führung, umso stabiler das Team. Die untersuchten Teams sind von äußeren Einflüssen abgeschnitten und in einer Einflussdimension limitiert. In Teams entwickeln sich üblicherweise Subgruppen: Frauen vs. Männer, Raucher\_innen vs. Nichtraucher\_innen, wissenschaftliches Personal vs. Vertragspersonal usw.. Johnson gibt soziale Rollen vor und bittet die Teammitglieder bei der Datenerhebung sie einander zuzuordnen. Da-

bei interessiert er sich vor allem für die Führungen und die Abweichler\_innen. Es wird zwischen lokal und global kohärenten Gruppen unterschieden. In lokal kohärenten Gruppen bestehen Untergruppen mit jeweils gemeinsamen Normen. Global kohärente Gruppen haben *ein* gemeinsames Verständnis und haben mehr Harmonie. Wenn formale und informelle Rollen insbesondere im Hinblick auf Führungsrollen zusammenfallen, wirkt es sich positiv auf das globale Verständnis aus. Gruppenstabilität schlussfolgert Johnson, ist nicht grundsätzlich positiv oder negativ. Es kommt darauf an, wohin Bewegung führt. Führt zeitweilige Instabilität zu einem globaleren Verständnis, ist das Ergebnis positiv.

Soziale Rollen vermischen sich in verschiedenen Personen und es macht das Zusammenspiel verschiedener Komponenten aus, ob ein Team in extremen Umgebungen gut zusammen arbeiten kann. Eine gute Führung allein beispielsweise reicht nicht aus, da sie auch abhängig ist von anderen sozialen Rollen und ihrem Zusammenspiel mit beispielsweise positiven und negativen Abweichler\_innen. **Johnson stellt selbst fest, dass klare Rollendefinitionen schwierig sind, ihr Zusammenspiel komplex und genauer betrachtet werden sollte.** Eine Aufgabe, die die vorliegende Arbeit aufgreift und eine generische Rollendefinition entwickelt (siehe Abschnitt 7.2.4).

### 2.3.7.2 Heintel: dynamische Prozesse in Gruppen

Es gibt eine Forschungsrichtung zu Gruppendynamiken. Zu den Forschenden gehören zum Beispiel Peter Heintel (Philosoph und Hochschullehrer), Rudolf Wimmer (Professor für Führung und Organisation) und Roswita Königswieser (Organisationsberaterin). Ihre Publikationen sind zu finden im Verlag für Sozialwissenschaften, Springer und Zeitschriften zu Organisationsentwicklung [HK15] [Uko06] [KWF13]. Sie entwickeln (theoretische) Modelle zu Teams weiter und erproben sie in Trainingsgruppen und der Praxis. Dabei ist die Methodik nicht eindeutig (**G1-Methode**) und bei der Datenerhebung ist nicht klar, inwiefern die Ergebnisse ausschließlich auf Erfahrungswissen beruhen oder systematisch validiert wurden (**G2-Daten**). Eine klare wissenschaftliche Einordnung ist daher schwierig. Grundsätzlich sind ihre Aussagen und Ideen, die teilweise auf jahrzentelangen Beobachtungen und Überlegungen herrühren, interessant für die vorliegende Forschung. Da sie nicht unter streng wissenschaftlichen Kriterien erhoben wurden, dienen sie im weiteren Verlauf als Diskussionsanregung. Dabei changieren sie irgendwo zwischen Glaser's "all is data" und ernstzunehmender wissenschaftlicher Quelle.

Das Buch, das ich hier etwas genauer vorstellen möchte, ist ein Sammelwerk. Interessant ist bereits die Themenauswahl. Neben den Trainingsgruppen geht es um (Selbst-) Reflexion, Prozessgeschehen und Diversity. Alles Bereiche, die in dieser Arbeit berührt werden.

Heintel kommt aus der Philosophie und betrachtet Trainingsgruppen (T-Gruppen). Dabei beschäftigt sich eine Gruppe eine Woche lang mit sich als Gruppe und ihren Dynamiken:

Die im Gruppenprozess auftauchenden Themen betreffen z.B. die Herausbildung und Veränderbarkeit individueller Rollen und Funktionen in Gruppen, die Entstehung von Normen und Standards, die Bedeutung von

Einfluss und Vertrauen als strukturbildende Elemente des sozialen Geschehens, Konflikte in Gruppen, die Bedeutung von Feedback für individuelles und kollektives Lernen, die Entstehung und Bedeutung von Autorität und Führung in Gruppen, Phasen der Gruppenentwicklung u.a.m. [Hei06, S. 27]

Die T-Gruppe dient dabei als Lernraum, um über Gruppendynamiken etwas zu lernen. Das Werk richtet sich an Manager\_innen und andere Personen, die Gruppen leiten und lenken sollen. Die Idee ist, dass Trainingsgruppenerfahrung Teams arbeitsfähig und erfolgreicher machen kann.

In der Geschichte der Gruppendynamik haben sich zwei Lager gebildet, ein größeres, welche die Gruppe als Instrument für die therapeutische Bearbeitung individueller Befindlichkeiten nützt und ein kleineres, das sich mehr mit dem Verhältnis von Gruppe und Organisation sowie im ausgeführten Sinn mit der Entstehung und Veränderung sozialer Strukturen beschäftigt. Obwohl mit Lewin (der selbst mehr war als nur akademischer Psychologe) die Gruppendynamik auch ihre psychologischen Wurzeln hat, wird sie von verschiedenen Quellen gespeist, Soziologie, Ethnologie, Psychoanalyse und den verschiedensten cross-over-Kombinationen wie Sozialpsychologie, Ethnopschoanalyse u.a. Als angewandte Sozialwissenschaft beschäftigt sie sich mit sozialen Strukturen und Prozessen und wie diese beeinflusst und gesteuert werden können [Hei06, S. 20]

Im Vorwort weist Krainz darauf hin [Kra06], dass eine Gruppe immer erst in eine „bestimmte Form“ kommen muss, um zusammenarbeiten zu können. Wie diese Form Gestalt annimmt ist nicht vorhersehbar und lässt sich nicht von Qualifikationsbeschreibungen definieren.

Vielmehr sind in Gruppen hohe Interdependenzen beobachtbar, weshalb sich auch das Wort Dynamik rechtfertigt. Sobald sich nämlich eine Beziehung verändert, wirkt sich dies auf andere Beziehungen in der Gruppe aus und beeinflusst diese. [Hei06, S. 17]

Das erzeugt eine feedbackintensive Situation, die durch Kommunikation gestaltet wird.

Interessant ist auch die Betonung des Selbst. Im Vorwort wird erklärt, dass Selbstreflexion sich durch alle Teile des Buches zieht und ein Abschnitt diesem Thema gewidmet ist:

Man hat es also in gruppendynamischen Trainingsgruppen mit zwei verschiedenen Formen von Selbst zu tun, einem (mehrfach gegebenen) individuellen Selbst und einem kollektiven Selbst, wobei diese noch miteinander interagieren. Die anderen beeinflussen mich, ich beeinflusse die anderen, emergent entsteht daraus die Ganzheit der Gruppe, deren jeweilige Form wieder auf mich zurückwirkt usw. Dass dabei Emotionen viel stärker ins Spiel kommen als in Alltagssituationen, liegt an der unvermeidlichen Daueraufmerksamkeit und Selbstbezogenheit, die durch die Trainingsgruppe erzwungen wird. [Hei06, S. 17]

Auch zu Reflexion sind in diesem Buch interessante Hinweise zu finden unter anderem von Königswieser. Königswieser arbeitet zu Gruppen-Dynamiken und ist Coach. Sie erklärt, dass unter Reflexion die „gezielte Auseinandersetzung mit der eigenen Person“ verstanden wird [Kön06]. Dabei geht es darum einen Prozess im Rückblick zu bewerten. Gruppen können sich dabei verschiedene Fragen stellen und eine Trainerin kann dabei unterstützen, „dass nicht mehr als nötig geflüchtet wird“. Dabei gilt es, die eigenen Gefühle zu beachten, da sie der Resonanzboden für Probleme sind.

Damit Meta-Kommunikation stattfinden kann, bedarf es in ausgewogenem Maß sowohl einer gewissen Autonomie der Teilnehmer – Zivilcourage und Mut, auch Heikles zu sagen – als auch verbindlicher Spielregeln, die dieser Art von Kommunikationsprozessen förderlich sind. Hier ist vor allem die Vorbildwirkung der Normsetzer nicht zu unterschätzen. Will man aus alten Denkschienen herauskommen, müssen Querdenker Platz haben, Tabus ausgesprochen werden dürfen, muss Verrücktes erlaubt sein. [Kön06, S. 73]

Königswieser sieht folgende Vorteile in der Reflexion. Reflexion...

- bringt Orientierung und Entlastung: Da verschiedene Perspektiven, über Vergangenheit, Gegenwart und Zukunft hinweg, neue Optionen, Pläne und Strukturen sichtbar machen können, Unsicherheiten nehmen und Sicherheit geben.
- fördert Kreativität und Mut: Gerade in Krisensituation kann es Ängste lösen, gemeinsam die Situation in größeren Zusammenhängen zu betrachten, die Gemeinschaft wahrzunehmen und sich nicht alleine zu fühlen.
- erzeugt Selbstbewusstsein und hebt implizites Wissen: Indem eigene Denkmuster und Begrenzungen reflektiert und aufgehoben werden.
- ist Weiterbildung, denn es fördert Sozialkompetenz.
- hat Hürden zu überwinden, denn es geht darum Gefühle einzugestehen und mitzuteilen.[Kön06]



## Kapitel 3

# Anspruch der Arbeit: Grenzen, Lücken und Anknüpfungspunkte zu verwandten Arbeiten

Im vorherigen Kapitel 2 wurden verwandte Arbeiten vorgestellt. Im nun folgenden Abschnitt werden die verwandten Arbeiten in ihrer Zusammenschau und die sich daraus ergebenden Forschungsfragen und -lücken diskutiert, bevor der Forschungsbeitrag der vorliegenden Arbeit vorgestellt wird.

In Abschnitt 2.1.1 wird die Relevanz menschlicher Aspekte in der agilen Softwareentwicklung und die geringe Forschung dazu hergeleitet. Lenberg u.a. stellen fest, dass Veränderungen der Prozesse am Menschen scheitern und nicht an den Werkzeugen [LFW15]. Gleichzeitig fordern sie Menschen nicht auf Persönlichkeitsmerkmale zu reduzieren (siehe Abschnitt 2.2.3). Ralph [Ra18] fordert empirische Forschung an der Lebensrealität der Entwickler\_innen und Hoda [HSGT17] stellt in ihrem tertiären Review fest, dass das Hauptaugenmerk bisher auf den agilen Praktiken und deren Umsetzung liegt. Es bedarf also weiterer fundierter Forschung zu menschlichen Aspekten in der agilen Softwareentwicklung.

Hoda hat bereits durch eine umfängliche GT festgestellt, dass das prägende Merkmal in agilen Softwareentwicklungsteams die Selbstorganisation ist. Hackmans [Hac86] Forschung bietet interessante Einblicke in selbstorganisierte Teams und wird in aktueller Forschung zu agilen Teams [MDD09] [MDD10] [LF18] vielfach zitiert. Leider fehlen genaue Angaben zur Forschungsmethode, seine Anregungen sind durch agile Praktiken bereits teilweise umgesetzt (siehe Exkurs 1) und lassen sich nur schwer validieren (siehe Exkurs 2).

Im aktuellen Review von Magpili und Pazos [MP18] wird deutlich, dass es auf drei Ebenen Faktoren gibt, die sich auf die Leistung selbstorganisierter Teams auswirken: individuelle, Team- und Organisationsebene. Die vorliegende Arbeit befasst sich vorwiegend mit den ersten beiden Ebenen. Alle von Magpili und Pazos genannten Faktoren ausgenommen „Aufgaben-Charakteristik“ beziehen sich auf menschliche Faktoren wie „individuelle Autonomie“ und „gegenseitige Kontrolle“. Abschließend stellen Magpili und Pazos fest, dass es Forschung zum Zusammenspiel zwischen Autonomie und Stabilität, das durch die aufgedeckten Faktoren beeinflusst wird, bedarf. Ein Phänomen das sowohl Hoda [HNM10a] in ihren „balancing acts“ als auch Moe u.a.

[MDD08] [MDD09] in seinen Hürden hin zur Agilität in ihren Forschungen zu agilen Softwareentwicklungsteams beschreiben. Moes Hürden, der Konflikt zwischen individueller und interner Autonomie, lassen sich als „*balancing acts*“ bzw. *vis versa* beschreiben. Jeder „*balancing act*“ ist eine Balance zwischen individueller und interner Autonomie. Die Balance zwischen Autonomie und Stabilität ist ein zentrales Phänomen agiler Softwareentwicklungsteams.

Ein weiteres Phänomen, das Hoda beschreibt, sind Rollen in agilen Teams [HNM13]. Sie versteht Rollen als durchlässig, benennt sechs Rollen und überlässt die weitere Rollendefinition den Teams. Die Frage wie Rollen an sich definiert sind, wird bei Hoda nicht beantwortet und auch Johnson [JBP03] erachtet Rollen als prägend in Teamgefügen, findet aber keine abschließende Definition und nur unzureichende soziale Rollen bezogen auf Führung und Abweichter\_innen. Rollen sollen laut Hoda [HNM13] den Teams helfen, ihre Zusammenarbeit zu reflektieren. Reflexion ist herausfordernd und gelingt laut Andriyani [AHA17] nicht allen Teams im gleichen Umfang (siehe Abschnitt 2.3.4). Dabei wird der einzige von Andriyani angeführte direkte menschliche und nicht prozesshafte Aspekt „*Gefühle*“ häufig vernachlässigt. Welche Bedeutung menschliche Aspekte beispielsweise bei der Reflexion von Abhängigkeiten und Schwierigkeiten bei der Aufgabebearbeitung zukommt, erklärt Andriyani nicht. Im Reflexionsmodell REALM von Babb u.a. [BHN14] werden menschliche Aspekte durch „*Learning about each other's skills and areas of expertise*“ berücksichtigt. Die weiteren Reflexionen sind vorwiegend Projekt- oder Technologie orientiert. Inwiefern menschliche Aspekte indirekt einfließen und reflektiert werden, wird nicht geklärt. Königswieser betont dabei die Relevanz von Reflexion insbesondere über Gefühle und menschliche Aspekte [Kön06].

Das Ziel dieser Arbeit ist es, die bisherige Forschung zu verbinden und zu erweitern. Am Beispiel Motivation (siehe Abschnitt 2.3.3) wurde der Detailgrad der Forschung immer ausgeprägter: Es besteht ein komplexer Zusammenhang zwischen individuellen Entscheidungen, Aufgabenverteilung (die auf Entscheidungen beruht) und Motivation. Gleichzeitig lassen sich Motivationsfaktoren nicht abschließend erklären. Unter Berücksichtigung der bisherigen Erkenntnisse ist Motivation beliebig komplex und unterliegt sich ständig ändernden Faktoren, von der jeweiligen Persönlichkeit, vom Team und von der sich ständig ändernden Projekt- und Entwicklungsumgebung abhängig sind. Hackman erklärte solche Phänomene bereits damit, dass Personen in Teams unterschiedlich sind und keine Gruppe gleich ist. Selbst für zwei Gruppen mit denselben Aufgaben können zwei völlig unterschiedliche Strategien notwendig sein, so Hackman [Hac86].

Der Anspruch meiner Arbeit besteht nicht darin, einen höheren Detailgrad zu erreichen und bisherige Erkenntnisse weiter auszudifferenzieren. Das Ziel ist es, mit Hilfe der GTM einen sinnvollen Abstraktionsgrad zu erreichen, der es erlaubt, menschliche Aspekte und ihre Auswirkung auf die Zusammenarbeit in agilen Softwareentwicklungsteams zu betrachten und zu analysieren.

Beim Gesamtblick auf die bisherige Forschung und die farblich markierten Forderungen in Abschnitt 2.3 bleibt festzuhalten:

- Agile Softwareentwicklungsteams haben spezifische Hürden zu überwinden.
- Während Werkzeuge und Methoden häufig erforscht werden, fehlt differenzier-

te Forschung zu menschlichen Aspekten.

- Laut einigen Forschenden sind menschliche Aspekte relevanter als Werkzeuge und Methoden, da agile Softwareentwicklungsteams an Menschen scheitern.
- Menschliche Aspekte zu untersuchen ist komplex. Sie sind in jedem agilen Softwareentwicklungsteam individuell verschieden.
- Bisherige Forschung betrachtet einzelne Bestandteile und Hürden selbstorganisierter Softwareentwicklungsteams.
- Es besteht die Forderung, dass Forschungsergebnisse sich an der Realität von Softwareentwicklungsteams orientieren sollen, damit sie einen Nutzen entfalten.

Mögliche **Forschungsfragen** wären daher:

Wie wäre es möglich, die bisherigen Forschungserkenntnisse zu verbinden unter der besonderen Berücksichtigung menschlicher Aspekte?

Welche weiteren relevanten Phänomene oder Konzepte gibt es unter besonderer Berücksichtigung menschlicher Aspekte?

Können die Forschungserkenntnisse zu menschlichen Aspekten in agilen Softwareentwicklungsteams in einem Kontext dargelegt werden, der möglicherweise Softwareentwicklungsteams in der Praxis helfen kann die Hürden der selbstorganisierten Zusammenarbeit zu erkennen und zu überwinden?

Antworten auf diese Fragen könnten auch weiterführende Fragen klären wie:

Wie lassen sich Rollen in agilen Softwareentwicklungsteams definieren?

Über welche menschlichen Aspekte bedarf es der Reflexion, damit eine Stabilität zwischen individueller Autonomie und Teamautonomie geschaffen werden kann?

### 3.1 Forschungsbeitrag

Das Ziel meiner Forschungsarbeit war es, eine Grounded-Theory zu entwickeln. In der Praxis habe ich reichhaltige Daten aus verschiedenen Teams erhoben (siehe Kapitel 5). Die Daten habe ich umfangreich nach Charmaz' Methodik der *Constructing Grounded-Theory* analysiert (siehe Kapitel 4.2 und 6). Daraus entstand im Ergebnis ein Modell (siehe Kapitel 7).

Der Prozess der Analyse ist im Sinne von Charmaz Methode ein wichtiger Bestandteil des Ergebnisses und wird in dieser Arbeit detailliert und ausführlich dargelegt (siehe Kapitel 6 und Abschnitt 7.1). GTM eignet sich insbesondere dafür menschliche Aspekte in Softwareentwicklungsteams zu erforschen, wie die verwandten Arbeiten, u.a. von Hoda [Hod11], zeigen. Wenn in den bisherigen verwandten Forschungsarbeiten GTM angewendet wurde, fehlt entweder eine Angabe zu der verwendeten

Methode gänzlich [SRF16] oder es wurde vorwiegend nach Glaser oder Strauss und Corbin gearbeitet. Diese Arbeit legt eine umfängliche Grounded-Theory in der Softwareentwicklung vor, die nach den Prinzipien Charmaz' erarbeitet wurde und den Forschungsprozess umfangreich, transparent und nachvollziehbar darlegt. Ich erhoffe mir dadurch, den Zugang zu dieser GTM für Forschende zu erleichtern.

Durch die akribische Umsetzung der GTM nach Charmaz schließen meine Ergebnisse an vorherige Arbeiten sinnvoll an, ohne dass es explizit intendiert war. Im vorherigen Abschnitt 2.3 wurden verwandte Arbeiten vorgestellt. Dabei wird bereits an den mehrfachen Vorwärtsverweisen deutlich, wie vielfältig und umfangreich meine Ergebnisse an bisherige Arbeiten anknüpfen.

Die Ausgangslage der Forschung war, dass zum einen agile Softwareentwicklungsteams in der Praxis inzwischen Normalität geworden sind [Kom17]. Zum anderen jedes Team seinen eigenen individuellen Weg hat, agile Arbeitsweisen umzusetzen und mit den sozialen Herausforderungen umzugehen. Agile Softwareentwicklungsteams müssen sich aus einem wachsenden Baukasten von Theorien, konzeptionierten Phänomenen und Handlungsempfehlungen bedienen. Und obwohl in der Forschung der Stellenwert menschlicher Faktoren deutlich wird [MP18] [LFW15], bezieht sich der größere Teil der Forschung auf agile Praktiken [HSCT17].

Menschliche Faktoren gewinnen an Bedeutung im Software Engineering. Gleichzeitig wird deutlich, dass menschliche Faktoren im Software Engineering und in anderen Disziplinen komplex sind, nicht immer dem Ursache-Wirkung-Prinzip folgen und sich nur schwer in Forschungsergebnissen vollumfänglich nachzeichnen lassen.

Die bisherigen Arbeiten beleuchten daher Elemente (z.B. Hodas Balancing Acts [HNM10a]), Modelle zu einzelnen Faktoren (z.B. zu Motivation [BBH<sup>+</sup>08]) oder Hilfestellung zu analysierten Hürden (z.B. Moe u.a. [MDD08]).

In der bestehenden Forschung zu selbstorganisierten Teams aus vielen verschiedenen Forschungsrichtungen wird deutlich, dass selbstorganisierte Teams eigenen Gesetzen und Faktoren unterworfen sind. Ein wichtiger Aspekt, der auch auf agile Softwareentwicklungsteams zutrifft, ist der Zugewinn an Autonomie und die Übernahme an Verantwortung. Zu den Anfangszeiten der agilen Methoden wurde dieses Phänomen in Arbeiten wie in denen von Hoda [Hod11] oder Moe [MDD09] bereits erkannt. Die Phänomene wurden konzeptionalisiert und Handlungen abgeleitet. So benennt Hoda Rollen mit Namen und Moe gibt konkrete Hinweise, wie der Transformationsprozess oder der Scrum-Prozess angewendet oder verbessert werden können.

Um die oben stehenden Fragen zu beantworten, habe ich in der vorliegenden Arbeit ein Modell mit Hilfe von GTM entwickelt, das bisher bekannte Erkenntnisse zu selbstorganisierten Teams aufgreift, ergänzt und zu einem schrittweisen Prozess weiterentwickelt. Menschliche Aspekte stehen dabei im Mittelpunkt des Modells.

Eine GT und auch das Modell sind grundsätzlich vorrangig analytisch. Menschliche Prozesse in agilen Softwareentwicklungsteams werden betrachtet und analysiert, um den Kern der Zusammenarbeit zu erkennen. Dabei findet keine Wertung statt und es wird nicht in das Geschehen aktiv eingegriffen, um z.B. Änderungen oder Verbesserungen zu erzeugen. Nichtsdestotrotz ist das nachfolgende Ziel, dass durch eine solche grundlegende Analyse ein wesentlicher Nutzen für Softwareentwicklungsteams entsteht. Das Modell beschreibt, wie einzelne Teammitglieder und das Team **KLARHEIT** erreichen können. Das Modell legt einen Zusammenhang nahe, dass durch

diese Klarheit über verschiedenen Faktoren, die Team-inherent verschieden sind, die Zusammenarbeit verbessert werden kann. Außerdem ist ein Ergebnis der Analyse, dass bei fehlender **KLARHEIT** das Teamgefüge und der Bestand des Teams gefährdet sind. Umso wichtiger erscheint es, **KLARHEIT** in Softwareentwicklungsteams zu gewinnen. Was das im Konkreten beinhaltet, wird ausführlich im Ergebniskapitel 7 dargelegt. Ein solches Modell für agile Softwareentwicklungsteams ist bisher einzigartig. Es ermöglicht erstmalig, den Gesamtprozess der Zusammenarbeit selbstorganisierter Softwareentwicklungsteams zu verstehen, in ihn einzugreifen und zu steuern, soweit er steuerbar ist. Teams wird dadurch ermöglicht, ihre jeweils einzigartige Konstellation im Hinblick auf Individuen, Ziele und die Zusammenarbeit abzubilden, zu beleuchten und daraus eigene Handlungen abzuleiten. Die Autonomie der Teams wird dadurch bestärkt, indem sie individuelle Lösungen finden, anstatt auf generische Lösungen zugreifen zu müssen.

Meine Forschungsbeiträge sind folgende:

Es wurde eine GT nach der Methodik Charmaz, entwickelt und der gesamte Forschungsprozess wird umfangreich und detailliert dargestellt.

Ein neues Konzept **KLARHEIT GEWINNEN** wurde erkannt und in das Zentrum der Teamzusammenarbeit gestellt.

Agile Softwareentwicklungsteams wurden in ihrer Gesamtheit mit Hilfe der GTM nach Charmaz analysiert. Das Modell beschreibt in differenzierten Schritten, wie **KLARHEIT** in agilen Softwareentwicklungsteams gewonnen werden kann.

Das analytische Modell hilft voraussichtlich agilen Softwareentwicklungsteams, auf einfache und sinnvolle Weise die Zusammenarbeit zu analysieren, zu reflektieren und zu verbessern.

Das Modell berücksichtigt, dass jedes Team anders handelt und individuelle Lösungen suchen muss.

Die Forschung ergab, dass fehlende **KLARHEIT** zu Konflikten und im schlimmsten Fall zum Bruch in agilen Softwareentwicklungsteams führen kann.

Das Modell enthält außerdem eine generische Definition von Rollen und erweitert damit Hodas Verständnis von Rollen in agilen Softwareentwicklungsteams.

Trotz seiner Komplexität ist das Modell für Praktiker\_innen einfach zu verstehen und anzuwenden, wie die Validierung zeigt (siehe Kapitel 8).

Das Modell ist von seinem Aufbau und Inhalt so gestaltet, dass es die Kernelemente und Hürden agiler Softwareentwicklungsteams aufgreift, in einen vereinfachten stringenten Prozess abbildet und gleichzeitig komplexe menschliche Zusammenarbeit versucht abzubilden. Das Modell wird damit den Forderungen verwandter Arbeiten gerecht, menschliche Aspekte zu beleuchten und für die Praxis relevante Ergebnisse zu liefern. Das Ziel ist es, agile Softwareentwicklungsteams mit der analytischen Hilfe des Modells zu unterstützen ihre Zusammenarbeit erfolgreicher zu gestalten.



# Kapitel 4

## Methode

In diesem Abschnitt wird die Wahl der Methodik begründet und die Methodik dargestellt. Anschließend wird die Datenerhebung in Kapitel 5 und die konkrete Umsetzung der Methode in Kapitel 6 dargestellt. Während bei vielen Forschungsmethoden erst Hypothesen gebildet werden, um diese in der Praxis zu prüfen, wird in der GTM die Praxis beobachtet, um daraus Hypothesen bzw. eine Theorie zu entwickeln. Das Ziel der GTM ist es, Daten aufzubrechen, die impliziten Annahmen und Denkweisen offenzulegen und zu einer Theorie zu verbinden.

### 4.1 Auswahl der Methode

Agile Softwareentwicklungsteams, die nach Scrum arbeiten, sind das Forschungsfeld dieser Arbeit. Die Forschung wurde mit Grounded-Theory-Methodologie (GTM) durchgeführt aus folgenden Gründen: GTM eignet sich, um zwischenmenschliche Prozesse (in Softwareentwicklungsteams) zu erforschen. GTM ist im Software Engineering inzwischen etabliert [HNM12] [SRF16] [Bad13]. Zur Erforschung agiler Methoden zeichnet sie sich besonders aus [HNM12] [JZA<sup>+</sup>13] [Bad13]. GTM und agile Methoden haben Gemeinsamkeiten: sie sind inkrementell und iterativ in ihrem Vorgehen, propagieren minimale Vorabplanung und stellen Menschen und ihre Handlungen in den Mittelpunkt [HNM12]. Badreddin [Bad13] stellt fest, dass agile Softwareentwicklung eines von drei Themengebieten ist, das häufig mit GTM erforscht wird. Eine der ersten Arbeiten mit GTM und agilen Methoden ist dabei die Arbeit von Kahkonen [KA03] zu extreme Programming.

Seit der Entwicklung der GTM durch Glaser [GS67] in den 70ern wurde die Methodik vielfach weiter entwickelt. Stol u.a. [SRF16] betrachten die Anwendung der GTM im Software Engineering und stellen drei zentrale GTM Varianten heraus (in chronologischer Reihenfolge): Glaser [GS67], Strauss und Corbin [SC10] und Charmaz [Cha14]<sup>1</sup>. Sie vergleichen die drei Strömungen und stellen sie in einer übersichtlichen Tabelle dar, die hier übernommen wurde (siehe Tabelle 4.1)

---

<sup>1</sup>Häufig wird im englischen von *glaserian* oder *straussian* GT gesprochen und Charmaz Variante als *constructing* GT benannt. Diese Bezeichnungen stellen sich in eine historische Tradition und nennen die männlichen Autoren und unterschlagen die Autorinnen Corbin und Charmaz. Obwohl die Bezeichnung *constructing* GT aussagekräftiger ist, verwende ich daher *Charmazs* GT, solange sich für Glasers und Strauß und Corbins Varianten keine ähnlichen Begriffe durchgesetzt haben.

## Kapitel 4. Methode

Element	Classic/Glaserian GT	Corbin/Straussian GT	Charmazs GT
Research Question	Should not be defined a priori, but emerge from the research—this makes the RQ relevant to the field. The researcher starts with an ‘area of interest.’ Literature in other areas may be consulted to increase the researcher’s “theoretical sensitivity.” Defining a RQ a priori is considered ‘forcing’ [Gla92].	Research question may be defined upfront, derived from the literature or suggested by a colleague; RQ is often broad and open-ended	Research begins with “initial research questions,” which evolve throughout the study [Cha14].
Role of the literature	An extensive literature review should be delayed until after the theory is emerging to prevent the influence of existing concepts on the emerging theory. Until the researcher has defined the RQ, it is not clear which literature should be consulted. Existing concepts such as gender and age should not be included a priori, but must ‘earn’ their way into the emerging theory.	The literature may be consulted throughout the process, as concepts from the literature may be used if applicable; to enhance theoretical sensitivity, as a secondary data source; to formulate questions for data collection or stimulate questions during analysis; to suggest areas for theoretical sampling [SC98] (p. 49).	Acknowledges not only Glaser’s reasons for delaying the literature review but also the impracticality of this strategy. Charmaz highlights the need to tailor a literature review to fit the purpose of the GT study [Cha14] (p. 306).
Coding procedures	<b>Open coding:</b> ‘fracturing’ of the data; line by line coding is recommended to achieve full theoretical coverage, but does not reject coding sentences or paragraphs, or whole documents [Gla92]. <b>Selective coding:</b> delimiting coding to only those variables that relate to one (or in some cases, several) core variables to establish a parsimonious theory. The core variable guides further data collection. <b>Theoretical coding:</b> establishing conceptual relations between substantive codes, resulting in the development of hypotheses. Glaser proposes several ‘coding families,’ which are theoretical codes that can be used by researchers, though these must ‘earn’ their way into the emerging theory	<b>Open coding:</b> generation of ‘categories’ and how they vary dimensionally. Coding can be done line by line or by sentence or paragraph, or even the whole document [SC98]. <b>Axial coding:</b> putting back data in new ways after open coding by identifying relationships between categories; this is effectively Glaser’s theoretical coding. Use of the ‘paradigm model’ or ‘conditional matrix’ (an analytical tool in Corbin and Straussian GT [SC98], Ch. 12) to identify context, conditions, action/interaction strategies and consequences. <b>Selective coding:</b> deciding on the central category that all major categories can link to [SC98].	<b>Initial coding:</b> examining data word-by-word, line-by-line or incident-by-incident to make sense of the text without injecting the researcher’s assumptions, biases, motivations. Similar to Glaser’s open coding. Charmaz recommends “coding with gerunds.” <b>Focused coding:</b> selecting categories from the most frequent or important codes, and using them to categorize the data; does not require a single core category or variable. <b>Theoretical coding:</b> specifying the relationship between categories to integrate them into a cohesive theory.
Questions asked during analysis	<ul style="list-style-type: none"> <li>• What is this data a study of? [Gla78]</li> <li>• What category or what property of what category does this incident indicate?</li> <li>• What is actually happening in the data?</li> </ul>	Asking questions about whom, when, where, how, with what consequences, and under what conditions phenomena occur, helps to ‘discover’ important ideas for the theory [SC94]. ‘Freewheeling flights of imagination’ [Cha14]	<ul style="list-style-type: none"> <li>• What is this data a study of? [Cha14]</li> <li>• What do the data suggest? Pronounce? Leave unsaid?</li> <li>• From whose point of view?</li> <li>• What theoretical category does this specific datum indicate? [Cha14]</li> </ul>
Philosophical influences	<b>Objectivism:</b> There exists a single, correct description of reality; the researcher therefore discovers grounded theory from data [BC07].	<b>Pragmatism</b> and <b>symbolic interactionism:</b> actors engage in a world that requires reflexive interaction; reality is constructed through interaction and relies on language and communication [CSMU13].	<b>Social constructionism:</b> social reality is constructed by our individual and collective action. GT emerges from “shared experiences and relationships with participants”; Observers are not neutral [Cha14].
Evaluation criteria	The generated categories must <b>fit</b> the data, the theory should <b>work</b> (it must be able to explain or predict what will happen); the theory must have <b>relevance</b> to the action of the area, and the theory must be <b>modifiable</b> as new data appear [Gla78] (p. 4-5).	Seven criteria for the research process e.g. information on sample selection, major categories, derived hypotheses and discrepancies. Eight criteria regarding the empirical grounding, e.g. “are concepts generated?” “is variation built into the theory?” [SC98].	<b>Credibility</b> (e.g. is there sufficient data to merit claims?), <b>originality</b> (do your categories offer new insights?), <b>resonance</b> (does the GT make sense to participants), <b>usefulness</b> (does the GT offer useful interpretations?) [Cha14] (p. 337).

Table 4.1: „Some of the key differences between the three main strands of grounded theory“, entnommen aus [SRF16], die Bezeichnungen der Strömungen wurden zur Konsistenz um die Namen Corbin und Charmaz ergänzt bzw. ersetzt.



Die verschiedenen Strömungen stehen sich nicht immer einvernehmlich gegenüber und grenzen sich in unterschiedlichem Ausmaß in ihrer epistemologischen Perspektive oder auch philosophischen Hintergründen wie Stol sie nennt (siehe Abbildung 4.1 unten) und ihrem methodischen Vorgehen ab<sup>2</sup>. Die Entwicklung geht von einer positivistischen Perspektive Glasers zu einer immer mehr konstruktivistischen Perspektive Charmazs. Glaser hat das Ziel eine externe objektive Realität freizulegen. Glaser geht davon aus, dass die subjektive Perspektive des Forschenden durch eine Methodik, die sich an die Daten hält und vielfache Überprüfung und ständiges Vergleichen beinhaltet, relativiert wird. Strauß und Corbin haben in ihren Auflagen von 1998 zu 2008 eine Entwicklung zu einer mehr konstruktivistischen Perspektive durchgemacht [Cha14, S. 243]. Charmaz [Cha14] stellt die konstruktivistische Perspektive in den Mittelpunkt ihres Vorgehens (siehe nächsten Abschnitt 4.2).

Während die Methodiken im Spannungsfeld einer positivistischen und einer konstruktivistischen Perspektive diskutiert werden, argumentiert Ralph [Ral18] hingegen, dass das eigentlich zu diskutierende Paradigma zwischen *Rational* und *Empirical Design Research* liegt. Er führt dabei an, dass das rationale Paradigma unterstellt, dass Softwareentwicklung rational ist und umgesetzt wird und beispielsweise Teams tatsächlich Methoden eins zu eins umsetzen. Er unterstützt dabei das empirische Paradigma, das davon ausgeht, dass Softwareentwicklung komplexer ist und nicht durchweg durch das hinzufügen und etablieren "besserer" Methoden zu lösen ist. Er fordert dabei, dass aktuelle Forschung auf empirischen Design beruht, um in der Lebensrealität der Softwareentwickler\_innen einen Nutzen zu entfalten.

Stol u.a. [SRF16] haben Forschungsarbeiten im Software Engineering, die auf GTM beruhen, überprüft und daraus Guidelines für weitere Forschungsarbeiten abgeleitet. Sie beobachten eine inflationäre Verwendung des GTM Begriffs, ohne dass in den Forschungsarbeiten dargelegt wird, wie die praktische Umsetzung aussah. Da es verschiedene GTM existieren und die saubere Umsetzung von der Theorie zur Praxis schwierig und anspruchsvoll ist, ist es notwendig, den tatsächlich umgesetzten Prozess transparent und detailliert darzulegen (ebd.). Diesen Forderungen leistet die vorliegende Arbeit umfassend Folge. In den kommenden Abschnitten wird zuerst die ausgewählte GTM nach Charmaz (Abschnitt 4.2) vorgestellt und dann die konkrete praktische Anwendung (Kapitel 6) in dieser Arbeit. Im Ergebnisteil wird die methodische Herleitung der Ergebnisse erneut ausführlich dargelegt (siehe Abschnitt 7.1)

## 4.2 Constructing Grounded-Theory nach Charmaz

Firstly, it is important to study grounded theory before starting. As several authors have noted, grounded theory may suffer from its 'apparent simplicity' [SRF16]

In diesem Zitat fordern Stol u.a. sich intensiv mit der ausgewählten Methodik zu befassen. Die augenscheinliche Einfachheit der Methodik verleitet dazu, sich nur oberflächlich mit ihr zu beschäftigen. Das führt zu einer vielfach fehler- und lückenhaften Umsetzung der Methode. Die grundsätzliche Auswahl für GTM wurde bereits

---

<sup>2</sup>Charmaz betrachtet beispielsweise die Forschenden als Teil der Konstruktion des Ergebnisses und wird dafür von Glaser heftig kritisiert [Gla12].

in Abschnitt 4.1 begründet. Im folgenden Teil wird die GTM nach Charmaz detaillierter erläutert und auf spätere Abschnitte, in denen die methodische Umsetzung dargelegt wird, verwiesen.

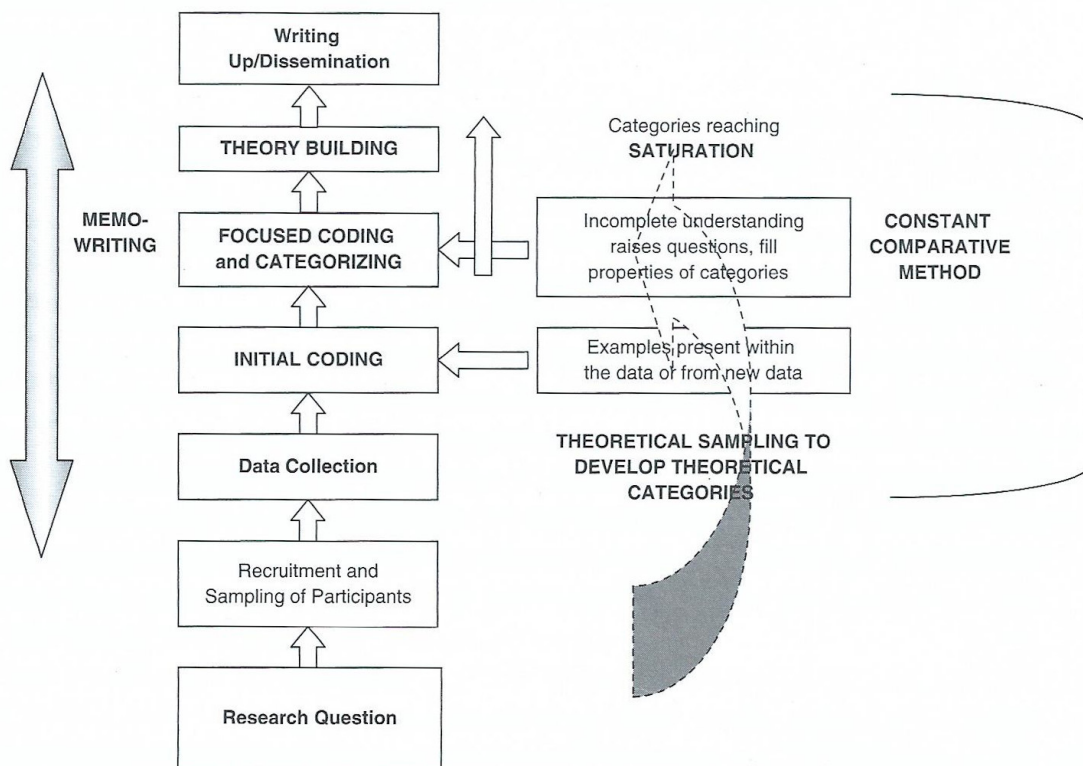


Abbildung 4.1: Vereinfachte Darstellung der methodischen Schritte der GTM nach Charmaz. Entnommen aus [Cha14, S.18]. Diese sehr vereinfachte Darstellung dient zur ersten Orientierung bezüglich der erforderlichen methodischen Schritte und umfasst nicht die vollständige tatsächliche Ausführung der einzelnen Schritte. Zum tatsächlichen Verständnis der einzelnen Schritte ist es notwendig die Abschnitte 4.2 und 6 zu berücksichtigen. Beispielsweise ist die Abfolge der Schritte nicht linear und Charmaz geht von einer initialen Forschungsfrage aus, die sich im Prozess verändert.

#### 4.2.1 Charmazs epistemologische Perspektive

In Tabelle 4.1 und Abschnitt 4.1 wird bereits erläutert, dass die konstruktivistische Perspektive Charmazs GTM auszeichnet. Sie sagt dazu:

Wir suchen nach den Annahmen, auf deren Grundlage Forschungsteilnehmende Handeln und Sinn konstruieren. [Cha11, S. 185]

Charmaz geht davon aus, dass sowohl die Datenerhebung als auch die Analyse ein konstruktivistischer Akt ist.:

Demgegenüber wurzelt die konstruktivistische Grounded-Theory-Methodologie im Pragmatismus und in einer relativistischen Epistemologie. Sie geht von multiplen Wirklichkeiten – und von multiplen Perspektiven auf diese

Wirklichkeiten – aus. Daten sind weder unabhängig von denen, die beobachten, noch von denen, die beobachtet werden, sondern sie werden durch Interaktion gemeinsam konstruiert. [Cha11, S. 192]

Damit unterscheidet sich ihre Perspektive eindeutig zu Glasers objektiver GTM, wie Charmaz in einer Tabelle verdeutlicht, siehe Abbildung 4.2. Aus der Gender- und Diversity- Forschung kommend [BSN14] [BSFK16] teile ich die Annahmen der *Konstruktion* von Wahrheit oder Wirklichkeit. Der Begriff *Gender* geht von einer sozialen Konstruktion der Geschlechter aus [Sch10]. Charmaz stellt in ihrer GTM die Handlungen und die dahinterstehenden Annahmen in den Mittelpunkt des Forschungsinteresses (siehe Zitat oben). Das kommt einem Verständnis des Doing-Gender gleich. Doing-Gender geht davon aus, dass durch das alltägliche Handeln und Interagieren von Subjekten miteinander Gender konstruiert wird [FW04]. Charmaz [Cha14, S. 14] erklärt, dass ihre Position nachträglich als ein Statement zur sozialen Konstruktion interpretiert wurde und stimmt dieser Interpretation zu. Daher war es naheliegend sich für Charmaz GTM zu entscheiden. Im Folgenden wird Charmaz GTM, ihre methodischen Schritte und Forderungen anhand des Prozesses beschrieben.

### 4.2.2 Reflexion der Forschenden

Eine wichtige Forderung Charmazs, die in Stols u.a. Tabelle 4.1 leider vernachlässigt wird, ist die Reflexion der Forschenden während des Forschungsprozesses. Charmaz betont, dass eine reflexive Haltung im gesamten Forschungsprozess eine wichtige Grundlage der konstruktivistischen GTM ist. Dabei bleibt es der Entscheidung und Erwägungen der Forschenden überlassen, ihre Haltung zu erörtern und offen darzulegen oder aus Rücksicht auf den Prozess, die Beforschten oder der eigenen Privatsphäre zu schweigen [MM11, S. 187]. Grundsätzlich geht sie davon aus, dass die Forschenden Teil der Konstruktion sind und damit wichtiger Bestandteil bei der Generierung von Erkenntnissen [MM11, S. 188].

In der vorliegenden Arbeit wird der Anteil der Forschenden durch aktive Sprache und einen transparenten Forschungsprozess deutlich gemacht (siehe dazu auch Kapitel Datenerhebung 5 und die verschiedenen Teile zur Methodik in Kapitel 6, Abschnitt 7.1) Hinzu kommt eine beispielhafte, ausführliche Reflexion der Forschenden anhand der Datenerhebung in Abschnitt 5.1.3.

### 4.2.3 Literaturreview

Charmaz hält es für wenig hilfreich, ohne jegliches Vorwissen ein Feld erforschen zu wollen. Wissen ermöglicht dabei Zugänge und schärft Perspektiven. Gleichzeitig soll eine intensive vorherige Literaturliteraturarbeit nicht die Kreativität und individuelle Perspektive der Forschenden einschränken (siehe [Cha14, S. 305ff]). Sie fordert im Literaturreview zu demonstrieren, dass man den aktuellen Forschungsstand erfasst hat und relevante frühere Arbeiten kennt. Die eigenen Ergebnisse sollen in diesen Kontext eingeordnet und abgegrenzt werden. Es geht darum, explizite Bezüge und überzeugende Verbindungen zur eigenen GT herzustellen. Der Anspruch einer eigenen GT soll aufgezeigt und untermauert werden. Konkret sollen vorherige Ergebnisse anderer Arbeiten dargestellt und bewertet werden. Die Lücken in diesen Arbeiten und



<i>Foundational Assumptions</i>	<i>Foundational Assumptions</i>
<ul style="list-style-type: none"> <li>• Assumes an external reality</li> <li>• Assumes discovery of data</li> <li>• Assumes conceptualizations emerge from data analysis</li> <li>• Views representation of data as unproblematic</li> <li>• Assumes the neutrality, passivity, and authority of the observer.</li> </ul>	<ul style="list-style-type: none"> <li>• Assumes multiple realities</li> <li>• Assumes mutual construction of data through <i>interaction</i></li> <li>• Assumes researcher constructs categories</li> <li>• Views representation of data as problematic, relativistic, situational, and partial</li> <li>• Assumes the observer's values, priorities, positions, and actions affect views.</li> </ul>
<i>Objectives</i>	<i>Objectives</i>
<ul style="list-style-type: none"> <li>• Aims to achieve context-free generalizations</li> <li>• Aims for parsimonious, abstract conceptualizations that transcend historical and situational locations</li> <li>• Aims to create theory that fits, works, has relevance, and is modifiable. (Glaser)</li> </ul>	<ul style="list-style-type: none"> <li>• Views generalizations as partial, conditional, and situated in time, space, positions, action, and interactions</li> <li>• Aims for interpretive understanding of historically situated data</li> <li>• Specifies range of variation</li> <li>• Aims to create theory that has credibility, originality, resonance, and usefulness.</li> </ul>
<i>Implications for Data Analysis</i>	<i>Implications for Data Analysis</i>
<ul style="list-style-type: none"> <li>• Views data analysis as an objective process</li> <li>• Sees emergent categories as forming the analysis</li> <li>• Sees reflexivity as one possible data source</li> <li>• Gives priority to researcher's analytic categories and voice.</li> </ul>	<ul style="list-style-type: none"> <li>• Acknowledges subjectivities throughout data analysis</li> <li>• Views co-constructed data as beginning the analytic direction</li> <li>• Engages in reflexivity throughout the research process</li> <li>• Seeks and (re)represents participants' views and voices as integral to the analysis.</li> </ul>

Abbildung 4.2: Abgrenzung objektivistischer GTM zu konstruktivistischer GTM, entnommen aus [Cha14, S.236]

wie sie die eigene Arbeit füllt, sollen dargelegt werden. Damit wird die eigene Arbeit positioniert und die Berechtigung verdeutlicht [Cha14, S. 309f].

In der vorliegenden Arbeit werden relevante Arbeiten in Kapitel 2 dargestellt. Anschließend werden die Lücken in der aktuellen Forschung aufgezeigt und der eigene Forschungsbeitrag untermauert in Abschnitt 3.1. Nach der Darstellung der Ergebnisse werden die Ergebnisse im Kontext der verwandten Arbeiten diskutiert, abgegrenzt und der Mehrwert der eigenen GT belegt (siehe Abschnitt 7.4)

#### 4.2.4 Forschungsfrage

Charmaz betrachtet die Forschungsfrage als ein Werkzeug zur Generierung neuer Erkenntnisse. Es kann zu Beginn mehrere initiale Forschungsfragen geben [Cha14, S.215]. Durch dieses Forschungsinteresse wird sich an das Feld angenähert und bestimmt welche Daten sinnvollerweise erhoben werden sollten [Cha14, S.79]. Grundsätzlich vergleicht sie das Forschungsinteresse mit einem Blick durch verschiedene Linsen. Die Perspektive soll sich dabei durch die Datenerhebung und die daraus entstandenen Erkenntnisse fokussieren [Cha14, S.26]. Für Charmaz kann, beziehungsweise *muss* sich das Interesse und die Forschungsfrage während der Forschung wandeln [Cha14, S.26]. Dabei entscheidet die Qualität der Forschungsfrage auch darüber, wann *theoretische Sättigung* der Konzepte erreicht wird [Cha14, S. 215]. Einfache Fragen erlangen schnell theoretische Sättigung naheliegender Kategorien. Theoretische Sättigung bedeutet, dass Kategorien ausdefiniert sind und durch die Erhebung weiterer Daten keine neuen Erkenntnisse dazu zu erwarten sind (siehe auch Abschnitt 4.2.5). Nur eine für das Forschungsfeld neue und angemessene Frage bringt eine qualifizierte GT hervor, die relevante Erkenntnisse birgt. In der Einleitung wurde die Weiterentwicklung des Forschungsinteresses dargelegt. In Kapitel 3 werden die Lücken im Forschungsfeld, mögliche Forschungsfragen und die Fragen, die die vorliegende Arbeit beantwortet erläutert.

#### 4.2.5 Datenerhebung

Charmaz widmet der Erhebung von reichhaltigen Daten und der Entwicklung einer angemessenen Interviewtechnik drei volle Kapitel [Cha14, Kapitel 2-4]. Bei der Erhebung reichhaltiger Daten erklärt sie, wie unterschiedliche Quellen herangezogen werden können. Neben den klassischen Interviews sind ethnografische Beobachtungen im Feld und Dokumente hilfreich. Tätigkeiten und Prozesse in der Praxis sollen beobachtet, aufgenommen und dokumentiert werden. Dabei sollen auch die Rahmenbedingungen in den jeweiligen Situationen festgehalten werden [Cha14, S. 35]. Als weitere Dokumente können Fragebögen, Meetingprotokolle o.ä. herangezogen werden.

Es ist wichtig Zugang zum Feld zu erlangen und involviert zu werden. Dazu gehört vor allem den Akteuren gegenüber offen zu sein und den Daten zu folgen. Hilfreich dabei ist 1.) direkten Zugang zu Ereignissen, Personen und Geschehnissen zu haben; 2.) über einen längeren Zeitraum teilnehmen zu können; 3.) Beobachtungen in Echtzeit und 4.) die Möglichkeit anschließend erneut Kontakt aufnehmen zu können bei sich im nachhinein ergebenden Fragen und Problemen [Cha14, S.39].

Die Reichhaltigkeit der Daten wird von einer guten Interviewtechnik geprägt. Dazu gehört, den Daten offen gegenüber zu bleiben und zu folgen, anstatt sich ausschließlich an vorgefertigte Fragebögen zu halten. In Interviews gehört vor allem dazu den Interviewten Zeit zu geben, ihre eigene Perspektive darzulegen, nachzufragen und ein Verständnis zu entwickeln. Sie gibt konkrete Hinweise zum Verhalten in Interviews. Dazu gehören Vertrauen aufzubauen, die Interviewten zu ermutigen eigene Wörter zu verwenden und sich auf Unerwartetes einzulassen [Cha14, S.70].

Die Daten sollen dabei nach dem Prinzip des *theoretischen Sampling* erhoben werden. In anderen Forschungsmethoden werden Daten häufig nach folgenden Prinzipien erhoben [Cha14, S. 197]:

Sampling, um die initiale Forschungsfrage zu beantworten.

Sampling, um eine Durchschnittsbevölkerung abzubilden.

Sampling, um negative Fälle zu finden.

Sampling bis keine neuen Daten auftauchen.

Beim theoretischen Sampling geht es um all diese Dinge nicht. Ein initiales Forschungsinteresse führt zu einer ersten Datenerhebung. Danach wird den Daten gefolgt. Das bedeutet, es werden Daten erhoben, die helfen, die entstehenden Kategorien und Konzepte weiterzuentwickeln. Dabei erkennt Charmaz die Schwierigkeit durchaus an, einen Zugang zur Praxis zu erhalten, der ein solches Sampling ermöglicht. Zu Beginn der Forschung kann gegenüber den Forschungsteilnehmer\_innen keine endgültige Aussage gemacht werden, wie viel, wie lange und welche Daten erhoben werden sollen. Die Teilnehmer\_innen müssen dem Prozess gegenüber aufgeschlossen sein. Trotz der Aufgeschlossenheit können im Prozess Zugänge zu weiterführenden Daten verwehrt bleiben [Cha14, Kapitel 8].

Das Ziel ist es *theoretische Sättigung* zu erreichen. Dabei bestimmt nicht die Menge an Daten, wann dieser Punkt erreicht ist, sondern die Qualität der Daten und der Datenauswertung. Die Qualität der Daten wird von der konsequenten Anwendung des theoretischen Sampling bestimmt. Vereinfacht ausgedrückt, ist theoretische Sättigung erreicht, wenn neue Datenerhebung keine neuen theoretischen Erkenntnisse den Kategorien hinzufügt [Cha14, S. 213]. Allerdings hängt dies auch von der Qualität des Forschungsprozesses ab. Wenn nur einfache Forschungsfragen gestellt werden, dem Codierungsprozess wie ein Rezept gefolgt wird, statt den Daten gegenüber offen zu bleiben oder die analytischen Ebenen zu kurz greifen, ist theoretische Sättigung schnell erreicht, ohne einen Mehrwert zu schaffen [Cha14, S. 213ff]. Charmaz bezieht sich unter anderem auf Mark Mason:

[...] suggests that ten interviews, conducted by an experienced interviewer will elicit richer data than 50 interviews by an inexperienced or novice interviewer[Mas10]

Mason resümiert, dass Doktorand\_innen häufig eine übermäßig hohe Zahl an Daten erheben, um sich abzusichern oder eine zuvor zugesicherte Anzahl zu erreichen, anstatt aufzuhören wenn Sättigung erreicht ist.

Analytische Tiefe wird unter anderem durch *ständigen Vergleich* der Daten hergestellt und eine theoretische Abstraktion. Charmaz empfiehlt sich die folgenden Fragen zu stellen, um zu prüfen, ob theoretische Sättigung erreicht wurde [Cha14, S.214]:

Which comparisons do you make between the data, within and between categories?

Which sense do you make of these comparisons?

Where do they lead you?

How do your comparisons illuminate your theoretical categories?

In what other directions, if any, do they take you?

What new conceptual relationships, if any, might you see?

Nur wenn diese Fragen sinnvoll beantwortet werden können, ist theoretische Sättigung erreicht.

Die vielfältige Datenerhebung dieser Arbeit wird in Kapitel 5 erläutert. In Kapitel 6 und Abschnitt 7.1 wird der Codierprozess dargelegt. In Kapitel 9 wird die Erreichung theoretischer Sättigung diskutiert.

#### 4.2.6 Codieren

In der GTM werden die Daten *codiert*, um sie zu analysieren. Die Daten werden erst detailliert analysiert und aufgebrochen, um anschließend (neue) Zusammenhänge zu erkennen, die zu einer Theorie verknüpft werden. Charmaz unterscheidet zwischen zwei Codierschritten (Siehe auch Abbildung 4.1 und [Cha14, S.113]): *Initial coding* und *focused coding*.

Zu Beginn während des initialen Codierens werden Teile der Daten Wort-für-Wort oder Zeile-für-Zeile betrachtet. Charmaz bevorzugt es, die Daten Zeile-für-Zeile zu betrachten [Cha14, S.124], nah an den Daten zu bleiben und spontane direkte Codes zu vergeben [Cha14, S.113ff]. Zur Analyse werden Fragen beantwortet wie ([Cha14, S.116] und Tabelle 4.1):

What is the data a study of?

What do the data suggest? Pronounce? Leave unsaid?

From whose point of view?

What theoretical category does this specific datum indicate?

Charmaz gibt verschiedene Fragen und Hinweise, die bei der Analyse der Daten hilfreich sind. Zwei wichtige Hinweise, die Charmaz methodisches Vorgehen charakterisieren sind aus meiner Perspektive: *not assigning types to people* und stattdessen *coding with gerunds*.

Es sollen nicht Personen analysiert werden, sondern ihre Handlungen. Ein Mensch kann in jeder Situation neu handeln. Personen mit Codes zu belegen, legt sie fest und macht sie eindimensional. Charmaz sagt dazu:

Assinging types to people casts them with static labels. In effect you make them one-dimensional although the behavior on which you based the label may represent a small part of who they are and what they do. Both may change as events change. Thus, such coding freezes people in time and space and also erases or minimizes defining variation in the studied phenomenon. [Cha14, S.117]

Charmaz empfiehlt zur Codierung von Handlungen Gerundien. Gerundien sind Verben in Form eines Substantives und werden im Englischen durch die Endung -ing gebildet. In der deutschen Sprache gibt es leider keine Gerundien. Trotzdem kann auf die Codierung von Handlungen geachtet werden (siehe auch Kapitel 6 für ein Beispiel).

Charmaz empfiehlt im weiteren Verlauf Ereignisse zu codieren (*incident by incident*) und Ereignisse zu vergleichen (*incident with incident*). Beim Vergleichen von gleichen oder unterschiedlichen Ereignissen, werden Prozesse, Gemeinsamkeiten und Unterschiede sichtbar [Cha14, 128ff]. Das Prinzip des *ständigen Vergleichens* ist Teil des gesamten Codierprozesses. Es hilft Codes zu sortieren, zu analysieren und weiterzuentwickeln. Es entstehen erste *fokussierte Codes*. Codes werden zu fokussierten Codes oder später Kategorien zusammengefasst, die durch ständiges Vergleichen in einem Zusammenhang stehen, z.B. Ähnliches beschreiben oder Teile eines Prozesses wiedergeben. Während des *fokussierten Codierens* entwickeln sich mehr *konzeptuelle Codes* [Cha14, S.138]. Es wird abstrakter über die Daten und die bisher entstandenen offenen Codes nachgedacht, ohne das Detail zu verlieren. Dabei ist der Abstraktionsgrad, der ein neues Licht auf die Daten wirft wichtiger, als die Häufigkeit in der ein Code auftaucht. Charmaz sagt dazu:

Sometimes students believe that the same code must reappear time and again to be used as a focused code and subsequently a possible category. Not at all. If the code is telling, use it. Occasionally, you will craft a code that explains much more than the data from which you constructed it. The code can give you a flash of insight, a way if looking at your data. It's exhilarating! Allow these moments of exhilaration to occur. [Cha14, S. 145]

Sich entwickelnde fokussierte Codes werden erneut in den Daten geprüft oder führen dazu, dass neue Daten erhoben werden. Die entstehenden Kategorien kristallisieren sich nach und nach heraus.

Durch das Arbeiten an den Daten, dem ständigen Vergleichen und Konzeptionalisieren, steigt die *theoretische Sensibilität* der Forschenden gegenüber den Daten und dem Feld [Cha14, S.244]. Theoretische Sensibilität hilft dabei den Blick auf die Daten zu schärfen und neue Perspektiven und Erkenntnisse zu generieren.

Im letzten Schritt werden die Kategorien zu einer Theorie verbunden. Dabei kann *theoretisches Codieren* helfen<sup>3</sup>. Charmaz bezieht sich bei dem Begriff des theoretischen Codierens auf Glaser [Gla78][Gla05]. Externe Quellen wie Vorwissen, verwandte Arbeiten oder theoretische Modelle (wie das paradigmatische Modell siehe Abschnitt 4.2.8) können bei der Theoriebildung helfen. Theoretische Codes helfen die Daten

---

<sup>3</sup>Stol u.a. siehe Tabelle 4.1 nennen dies als eigene Schritt im Codierprozess. Tatsächlich ist es nur ein möglicher Schritt im Codierprozess.



zu konzeptionalisieren, zu abstrahieren und Beziehungen sichtbar werden zu lassen [Cha14, S.150f]. Dabei legt Charmaz großen Wert darauf, dass externe Theorien sich ihren Weg in die Daten verdienen [Cha14, S.153]. Charmaz betont, dass es außerordentlich wichtig ist zu unterscheiden, ob theoretische Codes auf die Daten angewendet werden oder aus ihnen hervorgehen [Cha14, S.150]. Gleichzeitig können sie die sich entwickelnde Theorie bereichern, damit sie überzeugender und stimmiger wird [Cha14, S. 151]. Richtig angewendet verhilft theoretisches Codieren zu mehr Klarheit und Präzision (ebd.).

In den Kapitel 6 und Abschnitt 7.1 wird der Codierprozess der vorliegenden Arbeit dargelegt.

#### 4.2.7 Memos schreiben

Charmaz schlägt vor ein methodisches Journal zu führen, in dem Entscheidungen und Dilemmata festgehalten werden [Cha14, S. 165]. Es ist eine Art Forschungstagebuch und erleichtert die retrospektive Rekonstruktion des Forschungsprozesses.

Grundsätzlich werden während des Forschungsprozesses weiterführende und ergänzende Memos geschrieben. In Memos können verschiedene Informationen festgehalten werden. Sie begleiten die entstehenden Daten und werden über die Zeit hinweg immer konkreter und genauer. Zu Beginn beinhalten sie noch allgemeine Notizen wie Beobachtungen oder Anmerkungen zum Interview, neue Ideen für Datenerhebung u.ä.[Cha14, S. 169]. Im späteren Verlauf des Forschungsprozesses beschreiben sie sich entwickelnde Konzepte und Kategorien. Charmaz empfiehlt Rohdaten in die Memos direkt aufzunehmen. Das hilft bei der späteren Darlegung der Kategorie und verdeutlicht die Herkunft [Cha14, S.170]. In Memos werden verschiedene Datenquellen verglichen. Sie helfen beim *ständigen Vergleichen*. In Memos soll sich erlaubt werden, ausschweifend und weitreichend Gedanken zu sich entwickelnden Kategorien zu machen. Schließlich werden Memos während des fokussierten Codings zur Konzeptualisierung der Kategorien verwendet [Cha14, S. 180]. Charmaz weist darauf hin, dass man sich beim Schreiben von Memos frei machen soll von Ansprüchen an die Formulierungen oder richtige Schreibweise. Es geht darum Gedanken schnell und urteilsfrei festzuhalten. Später oder vor einer Veröffentlichung können Informationen präziser ausformuliert werden. Außerdem sollte kenntlich sein, welche Teile eines Memos auf Daten beruhen und was vorerst nur freie Assoziationen oder Ideen sind. Der Umgang der vorliegenden Arbeit mit einem methodischen Journal und Memos wird in Kapitel 6, Abschnitt 7.1 und insbesondere Abschnitt 6.2.4 dargestellt.

#### 4.2.8 Paradigmatisches Modell

Die GTM von Strauss und Corbin enthält einen Codierschritt mehr als Charmaz GTM: *axiales codieren* (siehe auch Tabelle 4.1, theoretisches codieren bei Charmaz ist nur optional). Während dieses Codierens werden verschiedene Schemata angewendet, um Konzepte zu vergleichen und Beziehungen herzustellen [SC96, S. 78ff]. Charmaz erkennt an, dass es leichter erscheint kleineren Codierschritten zu folgen, die konkrete Schemata anwenden. Gleichzeitig bemängelt sie, dass so Kreativität und Offenheit gegenüber den Daten verloren geht. Charmaz unterscheidet nur zwischen initialem

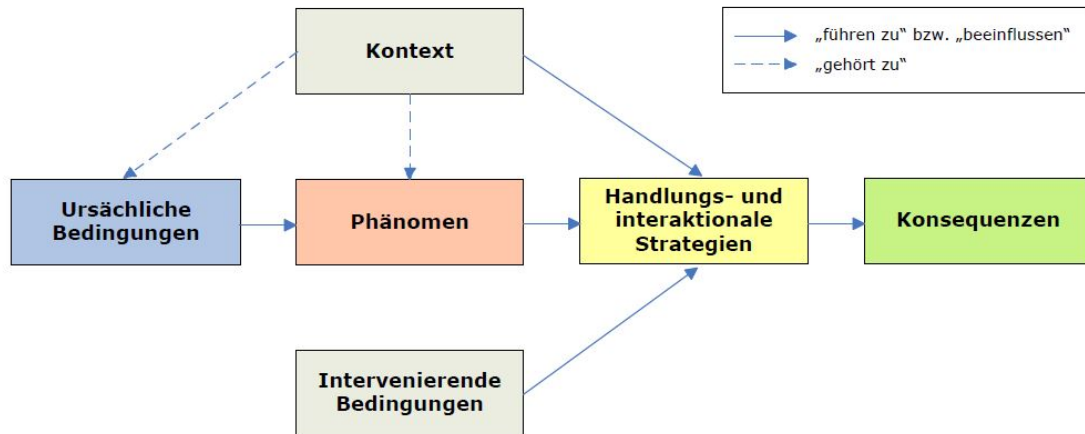


Abbildung 4.3: Paradigmatisches Modell in der Darstellung von Salinger, entnommen aus [Sal13, S.67]

codieren und fokussierten codieren (ein weiterer Schritt des theoretischen Codierens ist lediglich optional und nur unter gegebenen Umständen sinnvoll, siehe oben). Sie fordert den Daten zu folgen, statt vorgefertigte Modelle auf die Daten anzuwenden [Cha14, S. 147ff]. Ungeachtet dessen kann es Modelle geben, die passend erscheinen. Trotzdem sollen die Daten immer den Vorrang erhalten und Modelle ihnen angepasst werden.

Das in dieser Arbeit entwickelte Modell ist inspiriert vom paradigmatischen Modell Strauss' und Corbins. Daher wird das Modell hier kurz vorgestellt. Bei der Erarbeitung des Modells wurde den Grundsätzen Charmazs gefolgt: Es wurde den Daten gefolgt, das Modell wurde erst spät im Codierprozess angewendet und den Daten angepasst (siehe Abschnitt 7.1).

Eine hilfreiche Grafik zum paradigmatischen Modell hat Salinger in seiner Dissertation erstellt (siehe Abbildung 4.3). Kategorien werden im paradigmatischen Modell miteinander verknüpft und beschrieben. Dabei soll das Modell helfen, systematisch über die Daten nachzudenken [SC96, S. 78]. Die Bestandteile des Modells werden bei Strauss und Corbin [SC96, S.75] folgendermaßen definiert und sind in Abbildung 4.3 wiederzufinden:

**Ursächliche Bedingung:** Ereignisse, Vorfälle, Geschehnisse, die zum Auftreten oder der Entwicklung des Phänomens führen.

**Phänomen:** Die zentrale Idee, das Ereignis, Geschehnisse, der Vorfall, auf den eine Reihe von Handlungen oder oder Interaktionen gerichtet ist, um ihn zu kontrollieren oder zu bewältigen oder zu dem die Handlungen in Beziehung stehen.

**Handlungs- und interaktionale Strategien** helfen ein Phänomen unter einem spezifischen Satz wahrgenommener Bedingungen zu bewältigen, damit umzugehen, es auszuführen oder darauf zu reagieren.

**Konsequenzen:** Ergebnisse oder Resultate von Handlungen und Interaktionen.

**Intervenierende Bedingungen:** Die strukturellen Bedingungen, die auf Handlungs- und interaktionale Strategien einwirken, die sich auf ein bestimmtes Phänomen

beziehen. Sie erleichtern oder hemmen die verwendeten Strategien innerhalb des spezifischen Kontexts.

**Kontext:** Die spezifische Reihe von Eigenschaften, die zu einem Phänomen gehören; d.h. die Verortung der Ereignisse oder Vorfälle in einer Dimension/ einem Spektrum, die sich auf ein Phänomen beziehen. Der Kontext stellt den besonderen Satz von Bedingungen dar, in dem die Handlungs- und interaktionalen Strategien stattfinden

#### 4.2.9 Evaluationskriterien

In der Tabelle 4.1 von Stol u.a. werden vier Kriterien zur Evaluierung der entstandenen Theorie benannt. Tatsächlich weist Charmaz darauf hin, dass es je nach Forschungsrichtung unterschiedliche, relevante Kriterien geben kann. Die vier Kriterien sollen lediglich eine Vorstellung wie eine GT evaluiert werden kann geben. Sie werden wie folgt ausgeführt [Cha14, S.337f]:

##### **Credibility:**

- Has your research achieved intimate familiarity with the setting or topic?
- Are the data sufficient to merit your claims? Consider the range, number, and depth of observations contained in the data.
- Have you made systematic comparisons between observations and between categories?
- Do the categories cover a wide range of empirical observations?
- Are there strong logical links between the gathered data and your argument and analysis?
- Has your research provided enough evidence for your claims to allow the reader to form an independent assessment-and agree with your claims?

##### **Originality:**

- Are your categories fresh? Do they offer new insights?
- Does your analysis provide a new conceptual rendering of the data?
- What is the social and theoretical significance of this work?
- How does your grounded theory challenge, extend, or refine current ideas, concepts, and practices?

##### **Resonance**

- Do the categories portray the fullness of the studied experience?
- Have you revealed both liminal and unstable taken-for-granted meanings?

- Have you drawn links between larger collectivities or institutions and individual lives, when the data so indicate?
- Does your grounded theory make sense to your participants or people who share their circumstances? Does your analysis offer them deeper insights about their lives and worlds?

#### **Usefulness**

- Does your analysis offer interpretations that people can use in their every-day worlds?
- Do your analytic categories suggest any generic processes?
- If so, have you examined these generic processes for tacit implications?
- Can the analysis spark further research in other substantive areas?
- How does your work contribute to knowledge? How does it contribute to making a better world?

In der vorliegenden Arbeit wird viel Wert darauf gelegt, den Forschungsprozess ausgesprochen detailliert und transparent darzulegen, um ein hohes Maß an Glaubwürdigkeit zu erreichen (siehe Kapitel 6 und Abschnitt 7.1). Die Ergebnisse werden dadurch nachvollziehbar für die Leserschaft. Die Ergebnisse ermöglichen neue Einsichten, beinhalten neue Konzepte und Codes und beziehen bisherige Arbeiten und Erkenntnisse ein.

Zusätzlich wurde Aufwand betrieben, die Resonanz und Nützlichkeit mit Hilfe von Expert\_innen aus der Praxis mehrfach zu prüfen (siehe Kapitel 8).

In Kapitel 9 wird zusammengefasst, inwiefern Charmaz Kriterien damit erfüllt wurden.

## Kapitel 5

# Datenerhebung

GT ist ein komplexer und anspruchsvoller Prozess. Die Datenerhebung und die Darstellung bilden die Grundlage. Die Forscherin ist nach Charmaz ein wichtiger Teil der Konstruktion der Ergebnisse<sup>1</sup>. Daher ist die Reflexion der Forschenden ein wichtiger Bestandteil einer GT nach Charmaz. Aus diesem Grund und zur Sicherung und Steigerung der Qualität wie sie Stol u.a. fordert [SRF16] (s.o.), versuche ich den Prozess und meinen Anteil der Konstruktion in diesem Prozess von Beginn an so transparent wie möglich darzustellen. Dies tue ich auf zweierlei Art:

1. Ich verwende aktive Sprache. Durch die klare aktive Formulierung in erster Person Singular wird meine Rolle als Forscherin transparent. Die vorliegende Arbeit weicht in ihrer Ausdrucksweise damit von der üblichen wissenschaftlichen passiven Schreibweise ab. Auch wenn die aktive Schreibweise zuerst weniger wissenschaftlich erscheint, wird das Gegenteil erreicht: Sie benennt woher bzw. durch wen die Erkenntnisse generiert wurden.
2. Ich erkläre außerordentlich detailliert und an Beispielen das konkrete praktische Vorgehen in der Auswertung. Praktische Beispiele und Belege erhöhen die Glaubwürdigkeit und Nachvollziehbarkeit der Ergebnisse.

### 5.1 Zugang zum Feld

Finding participants can be difficult at best and extremely challenging at worst.[HNM12]

In dem obenstehenden Zitat verdeutlicht Hoda wie aufwändig und schwierig es sein kann, Zugang zu Softwareentwicklungsteams zu erhalten, die bereit sind, sich beobachten und beforschen zu lassen. Durch ein Mentoringprogramm für Studentinnen, das ich unabhängig von der Forschungsarbeit leitete, hatte ich ein Netzwerk zu Informatikerinnen aus der Praxis (aus dem sich die Mentorinnen rekrutieren). Dies wurde erfolgreich genutzt, um 4 der 5 Teams für die Forschung zu gewinnen. In zwei der Teams waren die jeweiligen Mentorinnen auch Teammitglieder. In diesen Fällen

---

<sup>1</sup>Zu Beginn bin ich es als Forscherin allein, die vor den Daten sitzt und Entscheidungen, Einordnungen und Abstraktionen fällt (wenn auch diese in vielfachen Prozessen überprüft, validiert und immer wieder zurück zu den Daten gegangen wird).

wurden nicht die Mentorinnen, sondern ausschließlich andere Teammitglieder interviewt. Dadurch sollte eine Beeinflussung der Ergebnisse durch Kontakte außerhalb der Forschung im Mentoringnetzwerk verhindert werden. Das Mentoringprogramm half auch als Türöffner: Auf der XP2017, einer Konferenz für agile Methoden, habe ich im ersten Schritt Unternehmen auf mögliche Mentorinnen angesprochen und im zweiten Schritt auch ein weiteres Scrum Team für die Forschung rekrutiert.

Der Kontakt zu den Expert\_innen entstand unter anderem dadurch, dass regelmäßig Praxisexpert\_innen und Consultants an die Hochschule eingeladen werden, um Einblicke für die Studierenden zu gewähren. Drei dieser Expert\_innen waren bereit auch an der Forschung teilzunehmen. Weitere Interviews sind durch die Teilnahme an Konferenzen und Meet-ups entstanden.

### 5.1.1 Theoretisches Sampling

Die Angaben beziehen sich immer auf den Zeitpunkt, zu dem die Beobachtungen begonnen wurden. Der Großteil der Daten stammt von fünf Softwareentwicklungsteams. Weitere Daten wurden durch Expert\_inneninterviews erhoben. Im Folgenden werden die Rahmenbedingungen der fünf Teams (t1 bis t5) beschrieben. Alle Teams gaben an, mit Scrum zu arbeiten. Neben der Arbeit mit Scrum stellen sie ein breites Spektrum im Hinblick auf Arbeitskontext, Firmenkultur, Produkt, Teamszusammensetzung und Erfahrung mit Scrum dar:

- t1 kam aus einem Unternehmen mit 90 Mitarbeiter\_innen, entwickelten eine Business-Management-Software und arbeiteten seit 6 Jahren mit Scrum.
- t2 kam aus einem Unternehmen mit 1500 Mitarbeiter\_innen. Das Team war über 3 Länder verteilt, kam von einem Wasserfall-Hintergrund und startete vor etwa einem Jahr agil. Sie entwickelten ein Hardware-Planungstool für den vorwiegend internen Gebrauch.
- t3 war in einer großen öffentlichen Institution (15.000 Mitarbeiter, 15 Scrum Teams entwickeln ein großes Informationssystem mit komplexer Geschäftslogik) mit externem Teammitgliedern einer Servicegesellschaft und einem langen Wasserfall Hintergrund. Es war seit 2 Jahren agil.
- t4 war in einem kleinen Startup von weniger als 25 Mitarbeiter\_innen, sie entwickelten ein Webportal und arbeiteten seit 4 Jahren agil. Sie hatten mit einem einzigen Scrum-Team begonnen und sich vor 1,5 Jahren in zwei Scrum-Teams aufgeteilt.
- t5 arbeitete an einem E-Commerce-Webportal in einem Unternehmen mit 150 Mitarbeiter\_innen für eine viel größere Muttergesellschaft. Die Aufgabe bestand insbesondere aus dem Import von Daten aus externen Datenbanken ihrer Lieferanten. Sie hatten mit Scrum angefangen, waren zu Kanban gewechselt und dann zurückgewechselt zu Scrum.

Die Datenerhebung wurde parallel und iterativ zur Datenauswertung durchgeführt. Die Datenhebung wurde durch die parallele Datenauswertung und die generierten Ergebnisse getrieben. Beispielsweise wurden in t2 vor allem Retrospektiven beobachtet. Die Auswertung ergab, dass während Aufgabendefinition, Verteilung und

Entscheidungsfindungen wichtige Teamprozesse stattfinden. In t3 und allen weiteren Teams wurden daher auch Plannings und Refinements/Groomings beobachtet. In t2 und t5 konnte bei der Auswahl der Teammitglieder ebenfalls nach *theoretischen Sampling* vorgegangen werden. Das bedeutet: Zu Beginn habe ich ein Interview mit einem Teammitglied geführt, das sich bereit erklärt hat. Nach weiteren Arbeitsbeobachtungen und Auswertung des Interviews habe ich gezielt ein bzw. zwei bestimmte Teammitglieder um ein Interview gebeten. In beiden Fällen wurden die weiteren Teammitglieder ausgewählt, da aus den vorherigen Ergebnissen zu schließen war, dass sie diese bestmöglich erweitern würden und teilweise konträre Perspektiven dazu haben. In t1 und t3 haben die Teams selbst bzw. die Scrum-Master entschieden, wer für ein Interview zu Verfügung steht. In t4 waren keine Einzelinterviews mit aktuellen Teammitgliedern möglich. Stattdessen wurde ein ausführliches Interview mit einem Scrum-Master, die sich zu diesem Zeitpunkt in Elternzeit befand, geführt. Nach Auswertung der Daten konnte als Feedbackmeeting eine Gruppendiskussion in t4 geführt werden, bei der alle Teammitglieder zu Wort gekommen sind. Bezüglich der Art der Meetings, die beobachtet wurden, den interviewten Teammitgliedern und auch den Fragebogen, der sich im Laufe der Zeit verändert hat, konnte *theoretisches Sampling* berücksichtigt werden. Die Auswahl der Teams beruht *teilweise* auf *theoretischem Sampling*. Teams vorab nach Kriterien des theoretischen Samplings auszuwählen, ist schwierig bis unmöglich. Aus folgenden Gründen:

1. Es kann von außen kaum beurteilt werden, ob das Team bestimmte Kriterien tatsächlich erfüllt, die für ein *theoretisches Sampling* interessant sind.
2. Diese Annahmen oder Selbstauskünfte zur Erfüllung der Kriterien entsprechen nicht zwingend der realen Situation oder es gibt andere vordringlichere Probleme im Team.
3. Der Zugang zu Teams ist nicht leicht und wird noch erschwert, wenn die Auswahl durch zu strenge Kriterien reguliert wird.

Daher wurde zuerst einmal jeder Zugang zu Teams genutzt, wenn das Team von sich behauptete, mit Scrum zu arbeiten.

t4 wurde gewählt und einbezogen, obwohl Einzelinterviews nicht möglich waren, da es sehr cross-funktional war und damit ein wichtiges Kriterium im *theoretischen Sampling* erfüllte.

Während der Datenerhebung in den Teams wurden jeweils die Daten parallel ausgewertet und anschließend entschieden, welche und wie viele Meetings noch beobachtet werden sollen. Außerdem standen immer Teammitglieder zur Verfügung, um Fragen im Nachgang zu Meetings oder Interviews zu klären. Die Datenerhebung selbst erstreckte sich über einen Zeitraum von 3 Jahren (Start September 2015).

### 5.1.2 Interviewtechnik

Charmaz führt sehr ausführlich aus, wie reichhaltige Interviewtechnik erreicht werden kann und wie schwierig das ist [Cha14]. Auch Hoda hat diese Erfahrung gemacht [HNM12]. Im Folgenden stelle ich die Entwicklung meiner Interviewtechnik dar.

Der erste von mir verwendete Interviewleitfaden wurde in einem Workshop für qualitative Datenerhebung 2015 in einem Team aus Teilnehmer\_innen erarbeitet. Die Fragen wurden später genau wie die Antworten codiert (s.u) oder mindestens reflektiert. Der Fragebogen wurde im Prozess angepasst wie Charmaz es als neunten Punkt der „Do’s“ ihrer Interviewtechniken fordert [Cha14, S.70]. Beispielsweise wurde eine Frage nach Kompetenzen gestrichen, da sie das Konzept **KOMPETENZ** erzwingt. Im gleichen Schritt wurde die Aufforderung: *„Erzähl mir etwas über Dein Team.“* gestrichen, da die Antworten auf sehr allgemeiner wohlwollender Ebene blieben. Beide Fragen wurden durch die sehr effektive Aufforderung: *„Erzähl mir etwas über jedes Teammitglied.“* ersetzt. Diese Aufforderung ermöglicht eine differenzierte Reflektion des Teams und gleichzeitig können (soweit die interviewete Person sie für relevant hält) auch Kompetenzen einfließen. Dieser kleine Unterschied in der Formulierung war ausschlaggebend für die späteren Ergebnisse und zeigt, dass es wichtig ist, iterativ zu arbeiten und im Auswertungsprozess des Codierens auch die eigenen Fragen und Aussagen gegebenenfalls zu berücksichtigen. Der am häufigsten verwendete Interviewleitfaden ist im Anhang A zu finden.

Die Interviewtechnik selbst hat sich im Prozess ebenfalls weiterentwickelt. Der erste „Do“ auf Charmaz Empfehlungen für Interviews:

Listen, Listen, Listen and listen some more

ist gerade zu Beginn schwierig auszuhalten. Insbesondere der Stille zu lauschen und den Interviewten die Möglichkeit zum Nachdenken zu geben, ohne der Versuchung nachzugeben, die zuvor lang durchdachte Frage kurzerhand umzuformulieren. Mit der Erfahrung, dass alle Interviewten ihre Antwort finden, wenn man ihn oder ihr nur genügend Zeit gibt, wird es leichter sich zurückzuhalten. Zumeist reichte ein wörtliches Wiederholen der Frage und evtl. der erneute Hinweis, dass es um die persönliche Perspektive geht und die Antwort so beantwortet werden soll, wie sie oder er es für richtig hält. Ein weitere Punkt aus Charmazs Do’s Liste erwies sich als äußerst hilfreich:

Ask the participant to elaborate, clarify, or give examples for his or her view

Nach konkreten Beispielen zu fragen verhindert, dass die Interviewteilnehmenden die Geschehnisse bereits direkt auf ihrer persönlichen Ebene werten und interpretieren. (Natürlich ist eine Wertung bereits dadurch gegeben, dass sie sich entscheiden bestimmte Dinge zu erzählen und andere auszulassen.) Die Interpretation der Geschehnisse ist damit zumindest zu einem Hohen Grad dem Codierungsprozess überlassen. U.a. aus diesem Grund weist Charmaz auch darauf hin, dass die Erhebung und Auswertung der Daten von ein und derselben Person durchgeführt werden soll, um ein realistisches Verständnis der Aussagen zu ermöglichen.

### 5.1.3 Reflexion der Forschenden

Hier stelle ich die Forschung mit t2 dar, um beispielhaft meine Rolle als Forscherin bei der Konstruktion der Daten zu reflektieren. Ich teile Charmaz Perspektive, dass der Akt der Datenerhebung bereits Teil der Konstruktion ist. Im folgenden Beispiel wird



das deutlich: In t2 habe ich über einen langen Zeitraum intensiv geforscht. Es wurden Interviews und Beobachtungen von Meetings durchgeführt. Außerdem gab es weitere Meetings mit dem PO des Teams und einer externen Consultant und nach der Präsentation der Ergebnisse weitere Schritte wie ein Online-Fragebogen, die Präsentation der teambezogenen Zwischenergebnisse und eine von mir gestalteter Retrospektive. Zu Beginn wurden konsequent im Sinne der GTM Daten erhoben ohne aktiven Eingriff oder Einflussnahme. Erst nach der Ausarbeitung erster Zusammenhänge wurde begonnen, aktiv mit dem Team zu interagieren und zu arbeiten, um tiefere Erkenntnisse und die Validierung der bisherigen Erkenntnisse zu erreichen. Bereits während des Prozesses der Datenerhebung gibt es unvermeidliche Eingriffe der Forschenden, die sich allein durch die Anwesenheit der Forscherin, die Gesprächsführung und Interaktion im Interview ergeben.

Bei t2 beruht die intensive Forschung u.a. darauf, dass teamführende Personen der Forschung und der Forscherin positiv gegenüber standen. Ein Teammitglied das wortführend in den Meetings war und gleichzeitig durch sein dominantes Verhalten von anderen Teammitgliedern als äußerst schwierig wahrgenommen wurde, hat im Interview zur Datenerhebung sehr offen reagiert. Während sein Verhalten dem Team gegenüber häufig nicht konstruktiv war<sup>2</sup>, war er der Forschung und der im Rahmen einhergehenden Veränderung eher aufgeschlossen. Dies beruht wohl vor allem auf dem starken Wunsch nach Veränderung dieses Teammitgliedes, aber auch einer gewissen Sympathie der Forschenden gegenüber. Aus Perspektive der Forscherin war die Person hoch relevant für das Team gerade durch sein provokantes und federführendes Verhalten. Die entgegengebrachte Sympathie, die mit einem Vertrauen einherging, war insofern äußerst hilfreich. Die Person war im Interview äußerst offen und (soweit sich das beurteilen lässt) ehrlich und hat Situationen und Eindrücke und vor allem die eigene Motivation dargelegt, wie sie den anderen Teammitgliedern bisher nicht zugänglich waren. Im weiteren Schritt war die Person auch explizit aufgeschlossen mitzuarbeiten. Dies änderte sich nach einiger Zeit und statt Unterstützung, boykottierte die Person bewusst die weiteren Teammeetings, an denen ich teilgenommen habe.

Es wird deutlich, dass damit die Ergebnisse in diesem Fall auch von dem Faktor der persönlichen Sympathie abhängen, die weder künstlich noch neutral hergestellt werden kann und damit auch nie Grundlage vermeintlich objektiver Forschung sein kann. Es ist daher besonders wichtig, es an dieser Stelle transparent zu machen.

Ohne das persönliche Interesse der Forschungsteilnehmer\_innen, welches auch auf der Wahrnehmung meiner Person als Forscherin beruhte, wäre eine solch langwierige und für die Forschung fruchtbare Arbeit nicht gelungen. Zudem ist davon auszugehen, dass durch meine Anwesenheit und durch meine Forschung die Teamdynamik beeinflusst wurde. Im Umkehrschluss bedeutet das, dass dieselben Vorgehensweisen zu einem anderen Ergebnis geführt hätten, wären sie von einer anderen Person durchgeführt worden. Ein Argument für - und ein Beispiel warum - die Perspektive der konstruktivistischen GTM relevant ist. Gleichzeitig zeigt dieses Beispiel wie wichtig es ist, verwoben und verbunden mit seinem Forschungsfeld zu sein, denn nur so können tiefgreifende Einsichten ins Forschungsfeld gelingen (vgl. [Cha14, S.

---

<sup>2</sup>Er wurde verbal laut, verwendete zahlreiche Schimpfwörter und griff Teammitglieder persönlich an.

240]). Die Forschende ist hier eindeutig auch Teil der Konstruktion. Das erfordert eine offene Reflexion der Forschenden, der an dieser Stelle genüge getan sein soll.

## 5.2 Daten

Ich habe während der Forschungsarbeit insgesamt in 5 verschiedenen Scrum- Teams Interviews und Arbeitsbeobachtungen durchgeführt. Das Ziel war es jeweils mit mindestens zwei Teammitgliedern ein Interview zu führen, um verschiedene Perspektiven zu erhalten.

In  $t_4$  war nur ein Interview mit einer Scrum-Masterin möglich (siehe Tabelle 5.1), die zu diesem Zeitpunkt seit wenigen Tagen in Elternzeit war.  $t_5$  arbeitet eng mit einem weiteren Team  $t_6$  zusammen und hielt beispielsweise die Reviews und Planning I gemeinsam ab. Daher wurden auch die Reviews und Plannings von  $t_6$  beobachtet und ausgewertet, dort aber keine Interviews geführt.  $t_6$  befand sich zu diesem Zeitpunkt in einer Umbruchphase mit wechselnden Teammitgliedern und war teilweise nur mit zwei Teammitgliedern ausgestattet. Es wurden insgesamt 44 Personen in Scrum-Teams beobachtet.

Insgesamt gibt es über 20 Stunden Tonaufnahmen. Davon entfallen etwa 16 Stunden auf Einzelinterviews, die weitestgehend transkribiert wurden. Die restlichen Aufnahmen sind während Feedbackmeetings und Diskussionsrunden entstanden und wurden zur weiteren Differenzierung einiger Konzepte verwendet. Insgesamt gibt es neben Transkripten und Beobachtungsprotokollen, 94 verschiedene Dokumente wie Entscheidungstemplates, Coach-Übungen, Fotos von Boards, Meetingprotokollen der Teams oder Ergänzungen zu den Teams (siehe auch Abbildung 6.2 Dokumentenstruktur als Beispiel).

In Tabelle 5.1 ist zu ersehen, dass zu Beginn der Forschungsarbeit vor allem Retrospektiven beobachtet wurden, während später alle Meetings berücksichtigt wurden. Die Meetings waren von sehr unterschiedlicher Dauer und im Durchschnitt etwa eine Stunde lang. Stand-ups waren etwa 15 Minuten lang. Die beobachteten Reviews in  $t_5$  und  $t_6$  etwa 30 Minuten, die Retrospektiven etwa 1-1,5 Stunden, die Plannings meist sehr lang und reichten bis zu 2,5 Stunden. Während der Beobachtungen wurde soviel wörtlich mitgeschrieben wie möglich. Dabei half ein Sitzplan, der jeweils zu Beginn notiert wurde und der es ermöglichte, später schnell Sprecher\_in namentlich zu zuordnen. Die Teams verwendeten team- und projektspezifische Fachbegriffe, Bezeichnungen und Abkürzungen. Insbesondere im Planning wurde in sehr technischen, für Außenstehende schwer verständlichen Details, diskutiert. Für die Forschung ist jedoch vor allem die Metaebene des Gesprächs relevant. Die inhaltlichen technischen Details zum Verständnis der Diskussion sind nicht zwingend erforderlich. Daher wurden sie im Protokoll meist durch Platzhalter wie *“erklärt technische Details”* wiedergegeben.

Bei Sprintwechseln wurden an einem Tag über mehrere Stunden mehrere Meetings hintereinander beobachtet wie Retrospektive und Planning. Natürlich lässt dann die Konzentration beim Protokollieren nach. Trotzdem habe ich versucht möglichst viel wörtlich mitzuschreiben. Insbesondere, da häufig erst am Schluss jeder Diskussion klar war, wohin sie geführt hat und ob sie dadurch für die Forschung von Interesse

Tabelle 5.1: Statistik zur Datenerhebung

Team <sup>a</sup>	Interviews	Zeitspanne	Anzahl der beobachtenden Meetings				Stand-up	Außerdem
			Review	Retro	Planning	Refinement		
t1	dt1ev1, t1dev2	ein Tag 2015					1	
t2 <sup>b</sup>	t2dev1, t2test1, t2test3	Nov16-Okt17		5		1	2	Teamanalyse, Anonyme Onlineumfrage, Retro durchgeführt
t3	t3test1, t3sm	Feb-Mai17		2	3	1	1	Feedback Juni17
t4	t4Sm2	Juni-Sep17		1	1	2		Gruppendiskussion
t5(t6)	t5sm1, t5dev1	Sep17-Jun18	4	3	4	2		Feedback mit SMs Januar und Juni18, Feedback gesamtes Team Jan18
Gesamtanzahl Meetings							32	

<sup>a</sup>Weitere Informationen zum Teamkontext sind u.a. in Abschnitt 5.1.1 zu finden

<sup>b</sup>Eine detaillierte Diskussion der Beziehung zwischen t2 und der Forscherin ist in Abschnitt 5.1.3 zu finden

ist, war es wichtig, parallel mitzuschreiben. In Einzelfällen, wenn mir die ersten Aussagen fehlten und die Diskussion an Relevanz gewann, habe ich versucht, die Aussagen im selben Moment zu rekonstruieren. Insgesamt sind so in den 32 Meetings mehr als 100 handbeschriebene DIN A5 Seiten Beobachtungsprotokolle entstanden. Ein Teil davon wurde ins Reine geschrieben und anonymisiert. Ein weiterer Teil liegt als abfotografierte pdfs vor.

Hier ein Auszug aus einem Protokoll zu einer Entscheidungsfindung während eines Plannings in t3. Da es keine wörtlichen Zitate sind, sind die Aussagen nicht in Anführungszeichen gesetzt. Falls wörtliche Rede mitgeschrieben wurde, ist das durch Anführungszeichen in späteren Ausschnitten sichtbar gemacht.:

CNr. 1: t3pm: Was hat das für Auswirkungen?

t3ao: erklärt welche Möglichkeiten es gäbe das Problem zu diskutieren im Planning II oder nächste Woche und wie sich das auswirken würde auf den Code

t3aoext: erklärt nochmal, was technisch wichtig ist

t3po: das legen wir hier (US)<sup>3</sup> nicht fest.

t3aoext: Doch Folgendes wird hier [...] festgelegt.

Für zukünftige Arbeiten würde ich empfehlen ein Schreiberkennungsprogramm auf einem Tablet o.ä. zur Erstellung von Beobachtungsprotokollen zu verwenden.

Die erhobenen Daten aus den 5 Teams dienen als eine wichtige Quelle für die Forschungsarbeit. Darüber hinaus wurde mit 7 Expert\_innen Einzelinterviews geführt, transkribiert und ausgewertet (und sind ebenfalls Teil der 20 Stunden Tonaufnahmen). Zu den Expert\_innen gehörten Consultants aus deutschlandweit bekannten Beratungsagenturen, Agile Coaches mit mehrjähriger Berufserfahrung und zwei international bekannte agile Expert\_innen, die auf der Konferenz XP 2017 als Redner\_innen waren.

Zur Erhebung der Daten habe ich 17 Einzelinterviews geführt und aufgenommen, davon 7 Expert\_innen und 10 Teammitglieder. Weitere Interviews wurden zur Validierung der Ergebnisse geführt (siehe dazu auch später Abschnitt 8). Ein Teil dieser Interviews wurde ebenfalls aufgenommen und transkribiert. Dazu wurden beispielsweise 3 Teammitglieder von t5, die das Unternehmen inzwischen verlassen hatten, interviewt, die Interviews aufgenommen und transkribiert.

Darüber hinaus habe ich u.a. eine Abschlussarbeit einer Bachelorstudentin betreut, die in einem Team 7 Teammitglieder und zwei agile Expert\_innen zu ihrer agilen Zusammenarbeit interviewt hat. Diese Interviews lagen ebenfalls transkribiert vor. Da die Interviews sich im Forschungsinteresse unterscheiden, der Interviewleitfaden abweicht und ich bei den Interviews nicht anwesend war, wurden diese Daten ausschließlich im späteren Forschungsprozess zur Anreicherung der Konzepte ausgewertet.

---

<sup>3</sup>US= steht in den Protokollen für User-Story

## Kapitel 6

# Anwendung der Methode

Secondly, researchers should describe their implementation of GT, not GT in principle. [SRF16]

Die obenstehende Forderung von Stol u.a. betont wie wichtig es ist, die eigene konkrete Umsetzung der GTM darzulegen und nicht bei der allgemeinen, lehrbuchartigen Darstellung stehenzubleiben. Das tue ich im folgenden Abschnitt umfänglich und detailliert.

Zur Einführung in die Methodik habe ich an einem zweitägigen *GESIS Workshop Grounded-Theory-Methodologie* 2015 teilgenommen. Geleitet wurde der Workshop von Günter Mey und Sebastian Ruppel. Günter Mey ist Professor für Entwicklungspsychologie an der Hochschule Magdeburg Stendal. Er ist Experte für GTM in der Psychologie [MM10] und zusammen mit Katrin Mruck Herausgeber des GT Readers, in dem sowohl Glaser, Straus, Corbin und Charmaz zu Wort kommen [MM11].

Im Workshop wurde neben der theoretischen Einführung vor allem das direkte Codieren an realen Daten von Workshopteilnehmer\_innen geübt. Meine ersten Codierungen beruhen auf den Erfahrungen aus diesem Workshop. Zu Beginn war dieses Vorgehen ausreichend und hilfreich, um Übung zu erlangen (siehe auch Abschnitt 6.2.1). Während beim ersten offenen Codieren, die verschiedenen GTM-Strömungen noch relativ ähnlich vorgehen, unterscheiden sie sich in den weiteren Schritten. Spätestens zu diesem Zeitpunkt, nach dem allerersten offenen Codieren, musste eine eindeutige Entscheidung für eine der GT Methodiken getroffen werden. Aus oben genannten Gründen wurde Charmaz als Methodik ausgewählt. Die Annäherung an das Forschungsfeld und die Forschungsfrage entsprach bereits der Methodik Charmazs. Durch die Entscheidung für Charmaz wurde methodische Sicherheit und Klarheit gewonnen. Die bis zu diesem Zeitpunkt entstandenen Codes wurden daraufhin erneut nach den Kriterien Charmazs geprüft und ggf. bearbeitet. Das heißt konkret: unbenannt, gelöscht und anders gewichtet.

### 6.1 Begriffe: ihre Verwendung und Notation

Bevor die einzelnen Schritte des Codierens erläutert werden, hier eine kurze Begriffsklärung. In der Literatur wird teilweise strikt zwischen *Code*, *fokussierten Code*, *Kategorie* und *Subkategorie* unterschieden. Charmaz verlangt eine grundsätzliche Offenheit

den Daten und der sich entwickelten Theorie gegenüber. Daher verwende ich die Begriffe folgendermaßen (siehe auch Abbildung 6.1 von unten nach oben):

**Coding** bezeichnet ein *Zitat* aus den Daten und wird häufig synonym verwendet. Ein (offener) Code beruht beispielsweise auf einem Coding/Zitat. Im späteren Entwicklungsstadium der Theorie beruht ein Code oder Konzept üblicherweise auf einer Vielzahl von Codings. Der Begriff Coding stammt aus der verwendeten Software MAXQDA und wird dort für die Zitatensammlung genutzt. Es lässt sich z.B. die Anzahl der *Codings* für einen *Code* anzeigen.

Gleichzeitig ist es der englische Begriff für das Codieren (wie *initial coding*) und ist von der deutschen *großgeschriebenen* Verwendung *Coding* zu unterscheiden.

Codings aus Interviews, Beobachtungsprotokollen und Gruppendiskussionen werden so dargestellt: CNr. 2: *“Dies ist ein direktes Zitat”* (Person) Jedes Zitat erhält eine laufende Nummer, Codingnummer (CNr.), um es referenzieren zu können. Die Zitate werden immer in der Originalsprache (Englisch oder Deutsch) wiedergegeben. Kommt in längeren Interviewausschnitten auch die Forscherin zu Wort, ist das mit einem großgeschrieben *“I”* wie *“Interviewerin”* gekennzeichnet. Die Zitate sind für die Lesbarkeit und das Verständnis gegebenenfalls gekürzt. Auf Paraphrasierungen wurde verzichtet, um die Authentizität der Zitate zu erhalten. Eine Paraphrasierung, die das Lesen eindeutig erleichtern würde, ist gleichzeitig eine Interpretation der Daten. Für kurze wissenschaftliche Publikationen ist das wichtig und angemessen. In der hier vorliegenden Dissertation werden ausführliche Zitate und Mitschriften immer im Original wiedergegeben. Da gesprochene Sprache selten zielgerichtet und vollständig ist, ist das sowohl für die Forschenden während der Analyse als auch für die Lesenden eine Herausforderung. Es schafft jedoch Transparenz und ist unbedingte Voraussetzung zur wissenschaftlichen Nachvollziehbarkeit, der Einordnung und Interpretation der Daten.

Alle Einzelinterviews, sowohl mit Teammitgliedern als auch mit Expert\_innen, sind nach dem Alphabet deklariert. Expert\_innen haben als Angabe nur den Großbuchstaben wie *Expertin A*. Bei Teammitgliedern wird zur besseren Transparenz und Zuordnung nur ein Personenkürzel angegeben. Das Personenkürzel setzt sich aus der Teambezeichnung t1-t6, der Rolle und einer fortlaufenden Nummerierung je Rolle, je Team zusammen. Die Rollen sind wie folgt abgekürzt:

**sm** steht für Scrum-Master

**po** steht für Product-Owner

**dev** steht für Entwickler\_in

**test** steht für Tester\_in

**ao** steht für Architekt\_in<sup>1</sup>

**m** steht für Marketingperson

---

<sup>1</sup>Diese abweichende Notation ao entstand direkt während der Datenerhebung und wurde von der Rollenbezeichnung im ersten beobachteten Team übernommen. Um später nicht den Zusammenhang zu den Daten zu verlieren, wurde sie beibehalten.

**ext** steht ergänzend für ein externe\_r Teilnehmer\_in, die nicht zum Team gehört

Die Bezeichnung t4dev2, bezieht sich also auf ein\_e Entwickler\_in aus Team 4, das aus mindestens zwei Entwickler\_innen besteht. Tauchen einzelne Personen nur in Interviews von Expert\_innen einmalig auf, sind sie durch einen Anfangsbuchstaben und XXX gekennzeichnet wie beispielsweise "Axxx".

Ereignisepisoden (oder *incidents*) sind beispielhafte Ausschnitte, die Konzepte leichter verständlich machen. Episoden setzen sich zumeist aus verschiedenen Datenquellen zusammen wie Zitate einer oder mehrerer Personen und Beobachtungen innerhalb eines Teams oder längere paraphrasierte Zitate einer Person zumeist aus Experteninterviews. Sie haben eine laufende Nummer zur Referenzierung und werden so dargestellt:

**Episode 1.** *Dies ist eine beispielhafte Episode, in der ein längerer Prozess dargestellt wird.*

**Code** oder **Konzept** wird synonym verwendet als Überbegriff zu Kategorie, Subkategorie und verwandten Begriffen der GTM. Es werden Begriffe wie Kategorie oder Subkategorie vermieden und stattdessen der Begriff *Code* verwendet. Wann ein einfacher offener Code zu einem fokussierten Code oder einer gefestigten Kategorie/Konzept wird, ist eher schwierig zu benennen und ein Kontinuum. Ob ein Code nur kurzweilig oder bis zum Ende der Ausarbeitung Bestand hatte, wird vernachlässigt in der Bezeichnung und Darstellung. In der Zusammenschau werden die Beziehungen der Codes untereinander ausreichend deutlich.

Ein Code wird so dargestellt: **CODE**. Um die Lesbarkeit zu erhalten, werden Codes gelegentlich konjugiert und dem Textfluss angepasst.

**Phänomen** dieser Begriff wird von Charmaz selbst nicht verwendet. Ein Phänomen steht im Mittelpunkt des paradigmatischen Modells (siehe Abbildung 4.3). Es besteht aus einem oder mehreren Konzepten. Es vereint und erklärt als Bindeglied und Mittelpunkt andere Konzepte.

**Initial coding** ist der Fachbegriff Charmazs für das erste initiale codieren. Im Fließtext wird zur Lesbarkeit synonym die deutsche Variante wie "initiales codieren" verwendet.<sup>2</sup> Offenes Codieren beinhaltet initiales codieren und wird daher gelegentlich synonym verwendet. Offene Codes, die zu einem direkten Coding gehören, werden so dargestellt: **DIES IST EIN OFFENER CODE, DER SICH AUF EIN CODING BEZIEHT**

In Abbildung 6.1 ist der Zusammenhang der Begriffe sehr vereinfacht dargestellt. Ein oder mehrere Codings führen zu Codes. Ein oder mehrere Codes können in der weiteren Entwicklung Konzepte werden. Wie oben bereits angemerkt ist zu beachten, dass dieser Prozess nicht linear ist und daher die Begriffe Code und Konzept ggf. synonym verwendet werden. Viele Codes entwickeln sich nicht weiter (siehe Abbildung rechts). Wenn das paradigmatische Modell im Prozess sinnvoll angewendet werden kann, können mehrere Konzepte ein Phänomen beschreiben.

---

<sup>2</sup>wobei die vom Duden empfohlene Schreibweise 'codieren' verwendet wird und nicht die alternativ mögliche Variante 'kodieren'.

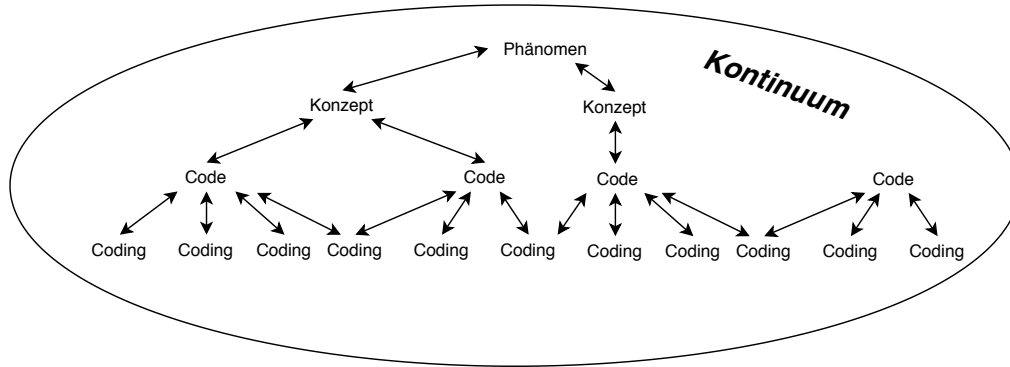


Abbildung 6.1: Eine vereinfachte Darstellung der Begriffe und ihre Entwicklung im Forschungsprozess

## 6.2 Codieren

Charmaz unterscheidet nur zwischen *initial* und *focused coding* im Gegensatz zu den bisherigen GT Methodiken, die sehr detailliert zwischen den einzelnen Codierschritten unterscheiden und sie teilweise auch konträr und ausführlich diskutieren. Die ersten Schritte des *initial coding* nach Charmaz sind sehr am offenen codieren nach Glaser orientiert, auf dessen Fragen sie sich auch bezieht.

Als Hilfestellung für das gesamte Codieren wurde ein Memo angelegt, das wichtige Hinweise Charmazs zusammenfasst mit einer Seitenangabe aus [Cha14]. Es war immer dann hilfreich, wenn das Codieren ins Stocken geriet oder Unsicherheiten auftraten. Hier der Inhalt des Memos:

Was untersuchen die Daten?

Was schlagen die Daten vor? Sprechen sie aus? Bleibt ungesagt?

Aus wessen Perspektive?

Welche theoretische Kategorie gibt dieser spezifische Bezugswert vor?

[Cha14, S. 116]

-Offen bleiben

-Bleib in der Nähe der Daten

-Halte die Codes einfach und präzise

-Konstruiere kurze Codes

-Handlungen bewahren

-Daten mit Daten vergleichen

-Schnell durch die Daten [Cha14, S. 120]

-Breche die Daten in ihre Komponenten, Bestandteile oder Eigenschaften

-Definiere die Handlungen, auf denen sie ruhen

-Suche stillschweigende Annahmen, die implizite Handlungen und Bedeutungen ausdrücken

-Kristallisiere die Bedeutung der Punkte

-Vergleiche Daten mit Daten

-Identifiziere Lücken in den Daten [Cha14, S. 125]

-Welcher Prozess ist hier?



- Wie kann er definiert werden?
- Wie entwickelt sich das?
- Wie verhält sich der oder die Forschungsteilnehmer\_in, während sie oder er an diesem Prozess beteiligt ist?
- Was zeigt der oder die Forschungsteilnehmer\_in in diesem Prozess, wie er/sie denkt und fühlt?
- Was könnte man aus dem beobachtetem Verhalten schließen?
- Wann, warum und wie verändert sich der Prozess?
- Was sind die Konsequenzen des Prozesses? [Cha14, S. 127]

Charmaz fordert, dass nicht Personen mit Codes belegt werden, sondern Handlungen analysiert werden (mit Gerundien siehe 4.2.6). Handlungen zu codieren, fördert die Mehrdimensionalität, macht Codings besser vergleichbar, reichhaltiger und erfordert mehr Abstraktion. Hier dazu ein Beispiel, folgendes Zitat wurde codiert: CNr. 3: *„Und das ist halt was, wo ich halt das Gefühl habe, ich brauche einfach so ein bisschen meine Zeit. Aber (.)Wenn ich manchmal sehe, wie die Dinge geschätzt sind und wie ich die abarbeite, habe ich auch das Gefühl, es ist einfach (.) völlig im Rahmen. Also es ist jetzt nicht so, dass ich länger brauche oder ständig die Zeiten überziehe ((lacht)) oder solche Sachen. Es ist, glaube ich, echt in Ordnung“* (t1dev1\_A)

Bei dem oben eingefügten Zitat wurde der Code **SELBSTREFLEXION** vergeben. Es wäre auch der Code **PERSON IST UNSICHER** oder **UNSICHERHEIT** möglich. Ein Vergleich mit anderen Codes, wäre dann auf andere **UNSICHERE** Personen beschränkt. Eine Person kann in unterschiedlichen Situationen unterschiedlich handeln (also einmal selbstsicher und einmal unsicher). Durch den handlungsorientierten Code **SELBSTREFLEXION** werden auch Codings mit einem anderen Reflexionsergebnis einbezogen. (Also auch die Person, die selbstsicher handelt.) Der Prozess wird genauer beleuchtet, wie die Person zu dem Ergebnis kommt, in welchen Situationen der Prozess stattfindet, wie er angeregt werden kann, wo er fehlt. Beim Codieren von Handlungen werden diese Prozesse unabhängig der Person berücksichtigt. Sie sind dadurch reichhaltiger und komplexer wie das Beispiel zeigt.

Während des offenen Codierens ist es wichtig, Charmazs Hinweisen zu folgen „Bleib in der Nähe der Daten“ und gleichzeitig die Daten weit genug aufzubrechen und Handlungen zu codieren, statt lediglich Typen oder Begriffe zu vergeben, zu sammeln und zu sortieren. Es muss ein (individuelles) Maß<sup>3</sup> bei der Umsetzung für diese beiden Hinweise gefunden werden. Um das individuelle Maß transparent darzulegen, wird den Ergebnisteilen eine methodische Herleitung vorangestellt (siehe Abschnitt 7.1).

### 6.2.1 Initiales Codieren: Wort für Wort

Zu Beginn habe ich einige Abschnitte ausführlich Wort für Wort codiert. Konkret heißt das, ich habe für jedes Wort ein Memo geschrieben und eine längere Assoziationskette zu diesem Wort formuliert. Dieses Vorgehen war äußerst kleinteilig und

<sup>3</sup> Während des GESIS Workshops und auch in Charmaz [Cha14] wird dabei von persönlichen Codierstil oder Vorlieben gesprochen.

mühselig und in MAXQDA schwierig zu händeln. Hoda [HNM12] und Allan [All03] halten ein solches Vorgehen für potentiell irreführend und auch Charmaz hat diesem Teil des Codierens nur einen sehr kleinen Abschnitt gewidmet [Cha14, S. 124]. Tatsächlich hat auch mich diese generierte Menge an Codes überfordert und gleichzeitig die Sorge ausgelöst im weiteren Schritt, durch die Reduzierung der Codes wichtige Informationen zu verlieren. Das Vorgehen war jedoch überaus wichtig, denn zwei entscheidende Lernschritte, die grundlegend waren für alles weitere Vorgehen, haben sich daraus ergeben:

1. *Abstraktionstraining*: Durch das ausführliche Wort für Wort codieren, habe ich Abstraktionen zu einzelnen Begriffen trainiert. Hoda u.a. [HNM12] beschreiben dies als schwierigen Schritt, der insbesondere für untrainierte Softwareingenieur\_innen eine Herausforderung ist. Neben meinen zusätzlichen sozialwissenschaftlichen Qualifikationen hat mir dieses Training geholfen, im weiteren Verlauf schnell wichtige Schlüsselwörter und Begriffe ausfindig zu machen, ihre Bedeutung im vorliegenden Kontext zu hinterfragen und die dahinterstehende Aussage zu abstrahieren. Das erleichtert auch den vorherigen Schritt, welche Abschnitte in den Daten relevant sind und detaillierter untersucht werden müssen.
2. *Verlust und Verdichtung*: Im zweiten Schritt musste ich einen Großteil der offenen Codes verwerfen. Dies löste die Angst aus, wichtige Erkenntnisse zu übersehen und stellte den Sinn des zuvor sehr zeitaufwändigen Vorgehens in Frage. Erst durch die offene Assoziation und die anschließende Verdichtung durch Abstraktion, entstand tatsächlicher Erkenntnisgewinn im Gegensatz zu einer reinen Klassifizierung oder Nacherzählung. Ein Teil der Codes verliert dabei seine Relevanz, ohne dass dabei der Erkenntnisgewinn leidet. Der Zweck dieser Codes war die Anregung neuer Erkenntnisse und somit kann irrelevanter Code auch archiviert werden. Diese Erkenntnis war sehr wichtig und half später sich auf die wichtig erscheinenden Codes und Konzepte zu konzentrieren ohne Angst um Verluste. Reduzierung ist ein Gewinn für die Erkenntnis.

Ein Beispiel zum Wort-für-Wort codieren:

CNr. 4: *“Meine erste Frage: Ich würde dich jetzt bitten, an ein abgeschlossenes Projekt zu denken, aus dem letzten halben Jahr, von dem du denkst: Das war ein ganz typisches Projekt!”*  
(Interviewerin)

Offene Codes zu diesem Zitat/Coding:

**erste Frage** AUFZÄHLUNG; ES FOLGEN NOCH WEITERE FRAGEN

**abgeschlossen** GEHEIM; BEENDET; ABGETRENNT; GEGENTEIL VON AUFGESCHLOSSEN; VERBARRIKADIERT

**letzten** SCHLECHTESTE?; BIS ZULETZT; GEGENTEIL VOM ERSTEN

**Du denkst** DENKEN=SELBSTVERANTWORTLICH; EIGENE MEINUNG; SCHWAMMIG

**ganz typisch** GÄNGIG WIEDERHOLEND; TYPISCH= ABWERTEND?; LANGWEILIG; KLASSISCH

**Projekt** ABGESCHLOSSENE GEMEINSCHAFTSARBEIT; KOMPLEX?; NICHT KLAR OB ES FERTIG WIRD; KONTRÄR ZU ABGESCHLOSSEN

An dieser Stelle habe ich meine eigene Frage meiner ersten beiden Interviews ausführlich codiert. Dies war zu Beginn des Forschungsprozesse und half die eigenen Fragen erneut zu hinterfragen, zu prüfen und anzupassen: Im ersten Moment wird der Sinn dieser detaillierten Codes überhaupt nicht deutlich. Im weiteren Prozess wurde klar, dass die Interviewten sich mit der Beantwortung der Frage schwer taten. Bei der weiteren detaillierten Codierung ihrer Antworten wird ersichtlich, dass die ausgelösten Assoziationen bzw. Codierungen von **ABGESCHLOSSEN**, **GANZ TYPISCH** und **PROJEKT** nicht zu ihrer Wahrnehmung agiler Sprints, die immer anders, komplex und selten abgeschlossen sind, passt. Die Frage wurde daraufhin durch eine Frage nach dem letzten Sprint ersetzt. Hinzu kommt die Erkenntnis, dass einzelne Worte wie z.B. **TYPISCH** wichtige Schlüsselworte sein können, die beim Codieren direkt hinterfragt werden müssen.

## 6.2.2 Initiales Codieren: Zeile-für-Zeile und Ereignis-für-Ereignis

Die Daten wurden an interessanten Stellen *Zeile-für-Zeile* oder *Ereignis-für-Ereignis* codiert. Dabei waren die Codes noch relativ ausführlich. Durch ständiges Vergleichen der Daten wurden erste zusammenfassende Begriffe gefunden wie z.B. **EIGENE POSITION - ICH** oder **ARBEITSVERTEILUNG ALS HERAUSFORDERUNG**. Codes unter einem zusammenfassenden Begriff wurden verglichen, sortiert, ergänzt und verworfen (*ständiges Vergleichen*). Dabei wurde ein erstes Memo zu den Zusammenhängen dieser Codes verfasst und einzelne Zitate im Memo eingefügt. In den Abbildungen 6.2 und 6.3 ist beispielhaft eine Dokumentenstruktur und Codestruktur zu sehen wie sie in der Software während des Codierens entstanden ist. Als Beispiel: Es wurden in den Daten mehrere Codings gefunden, in denen es um die Position der Interviewten im Team geht. Diese Codings wurden zuerst unter der Überschrift **EIGENE POSITION-ICH** gesammelt. Einer dieser Codes war: **SCHÄTZEN ALS SELBSTREFLEXION, EINSCHÄTZEN DER EIGENEN FÄHIGKEITEN** der zum oben bereits erwähnten Coding Nummer 3 gehört. Als nächstes habe ich alle gesammelten Codings zu **EIGENE-POSITION - ICH** angeschaut, ein Memo angelegt und zu jedem Coding einen zweiten ausführlicheren offenen Code geschrieben. (In MAXQDA ist die Länge der Codes beschränkt. Im Memo konnten die bisherigen Codes unter Reflektion der ersten zusammenfassenden Überschrift ausführlicher ausformuliert werden.) Wenn mehrere Codings dasselbe ansprachen, wurden sie in einem offenen Code zusammengefasst. Durch erneutes Vergleichen und Zusammenfassen dieser offenen Codes entstanden die Konzepte **TEAM-UND SELBSTREFLEXION** und das dazugehörige Konzept **INDIREKTE FEEDBACKPROZESSE**, die im Memo darunter abstrahiert ausformuliert wurden. Die offenen Codes und die erste theoretische Ausarbeitung, die bereits ein Schritt im *fokussierten Codieren* ist, stehen im selben Memo und stellen so sicher, dicht an den Daten zu bleiben. Gegebenenfalls wurden ins Memo außerdem gesondert freie offene Gedanken und Assoziationen formuliert, Mutmaßungen und weitere Schritte die teilweise zum *fokussierten Codieren* führten (siehe nächsten Abschnitt zu *fokussierten Coding*).

The screenshot shows a window titled "Liste der Dokumente" with a toolbar and a list of folders and documents. The list is as follows:

Item	#
Coaches, Consultants und Einzelpersonen	275
Team5	115
Team4	130
Team3	227
Planning I und II am 09.05.17	33
refinement Analysestory und Stand-up 09.3.17	31
planning I 7.3.17	9
retro 6.3.16	29
erste Übung	0
Bewertung Gespräche über 3 Iterationen	0
Bewertung US Teil 1	0
bewertung US teil 2	0
Retro am 06.02.17	23
Retro 06.02.17	0
InterviewK_T3	63
InterviewI T3 t3sm	39

Red callout boxes provide the following information:

- A box pointing to the number 275: "Anzahl der markierten Codings in einem Dokument oder Ordner"
- A box pointing to the number 33: "Offene Memos zum Team und zu einem Transkript"
- A box pointing to the list of documents under Team3: "Verschiedene Dokumente zu einem Team"

Abbildung 6.2: Screenshot aus MAXQDA, Ausschnitt aus dem Dokumentenstrukturbaum

**Liste der Codes**

- pflichtbewusst
  - selbstreflexion\_ Team klarheit**
    - verantwortung
    - coaching, Sm
    - positive effekte
    - episoden
    - einbringungsvoraussetzungen
    - epsiode gesamter Prozess selbstreflexion
    - Nicht übereinstimmende Aufgabendefinition
    - schnelle auffassungsgabe als kompetenz
    - eigene wahrnehmung der person als voraussetzung zur Klärung
    - eigenständigkeit als erfolg ungleich team
    - selbstständigkeit= Aufgabenverständnis
    - selbstwahrnehmung als Voraussetzung für das Team
    - Hilfe bei der Selbsterkenntnis als Lösungsweg
    - selbstwahrnehmung als Prozess zum Ergebnis

7  
60  
0  
0  
13  
12  
5  
2  
1  
1  
1  
0  
0  
1  
3  
1

Theoriememos zu den Codes

Anzahl der markierten Codings zu einem Code

Sich entwickelnder Code "Selbstreflexion" und weitere offene Codes, die darunter gesammelt sind

Abbildung 6.3: Screenshot aus MAXQDA, Ausschnitt aus dem Codestrukturbaum

Bei dem oben erwähnten Konzept der **TEAM- UND SELBSTREFLEXION** gab es beispielsweise in den Daten zwei Ereignisse einer gelungenen bzw. gescheiterten Integration eines neuen Teammitglieds. Die beiden Ereignisse wurden miteinander verglichen (*Ereignis für Ereignis*). Dies führte zu reichhaltigen Erkenntnissen zum Prozess der **SELBST- UND TEAMREFLEXION** und wurde im Memo festgehalten. Der in diesem Beispiel dargestellte methodische Vorgehensweise war nicht linear, sondern zyklisch. Das bedeutet z.B. durch den Vergleich der Ereignisse ergaben sich neue Erkenntnisse zur **SELBSTREFLEXION**, woraufhin die Daten erneut geprüft wurden, neue Codes entstanden und das Memo bzw. das Konzept angereichert wurde. Der Prozess erstreckte sich über mehrere Jahre und beinhaltete auch die erneute Datenerhebung.

### 6.2.3 Fokussiertes und theoretisches Codieren

Das Zusammenfassen offener Codes in Memos war der erste Schritt im *focused coding*. Daraus ergaben sich Abstraktionen, die zu weiteren freien Assoziationen und Ideen führten, die jeweils separat im Memo festgehalten wurden. Anschließend habe ich diese Erkenntnisse genutzt, um erneut in die Daten zu schauen oder um theoretische Codes zu entwickeln, die ich dann in den Daten geprüft habe. Sobald ein Code eine höhere Abstraktionsebene erreicht hatte und der Kern sich herauskristallisierte, wurden häufig sehr viel mehr Codings relevant, die erneut eine Betrachtung des Codes erforderten und diese schärften und erweiterten. Als Beispiel: Beim Codieren zu cross-funktionalen Teams habe ich aus einer Gruppendiskussion von t3 den Code **LEVEL OF DETAIL** abstrahiert. Die Gruppendiskussion bezog sich konkret auf die cross-funktionale Zusammenarbeit. Durch die Abstraktion **LEVEL OF DETAIL** fielen mir auf einen Schlag eine Unzahl von ähnlichen Situationen und Diskussionen in anderen Teams dazu ein. Ich habe die Daten daraufhin erneut zu diesem Code geprüft und untersucht, woraufhin noch weitere relevante Daten gefunden wurden, woraufhin der Code an Substanz gewann und sich ausgestaltete.

Das bis dahin rein induktive Vorgehen wurde durch deduktive theoretische Ideen angereichert, die vor allem aus meiner theoretischen Sensitivität hervorgingen. Wenn dieser Prozess jeweils so weit war, dass erste Strukturen und Zusammenhänge erkennbar wurden, wurde das im Memo festgehalten. Wenn das Memo eine Qualität erreicht hatte, die es ermöglichte, vorläufige Erkenntnisse zu teilen, begann der nächste und langwierige Schritt im *focused coding*. Es wurden vorläufige Erkenntnisse in unterschiedlichen Kontexten reflektiert, sortiert, neu durchdacht und überprüft. Dazu gehörten Forschungsmeetings, zeitliche Distanz, erneute Datenerhebung und Diskussionen in Meet-ups. Aus diesem iterativen Prozess ergaben sich vor allem theoretische Codes, die Überprüfung in den Daten und die Festigung und Abgrenzung bisheriger Codes.

Im persönlichen Gespräch mit Kolleg\_innen, Expertinnen wie Rashina Hoda und auch im GESIS Workshop zu GTM, an dem ich teilnahm, wurde die hohe Schwierigkeit dieses Schrittes betont vom initialen zum fokussierten Codieren. Die Empfehlung war, geduldig weiterzuarbeiten im Vertrauen darauf, dass der Erkenntnismoment kommt. Die Situation war vor allem von Unsicherheit geprägt: Wohin führt das? Dies ist eine Schlüsselsituation, die jeder/m GTM Forscher\_in begegnet. Ein Beispiel dafür wird daher später in der methodischen Herleitung 7.1 transparent und

detailliert dargelegt.

Erneut geschärft und geprüft wurden die Ergebnisse bei der Erarbeitung der ersten Publikationen. In der Zusammenarbeit mit Kolleg\_innen und Expert\_innen wurden die Strukturen erneut überarbeitet, reflektiert und sortiert.

In diesen Prozessen wurde vor allem die subjektive Perspektive der Forscherin durchbrochen und durch andere vielfältige Perspektiven angereichert, gegenüber gestellt und geprüft. Dieser langwierige Prozess hat vor allem die Qualität der Ergebnisse gefördert: Die Darstellung wurde geprüft und vereinfacht, die Benennung der Kategorien und ihre Zusammenhänge und Rückschlüsse reflektiert, verändert und erweitert. Dieser Prozess wird in Abschnitt 7.1 dargelegt.

Im einem weiteren Schritt wurde bisherige Literatur herangezogen, wodurch die theoretische Sensitivität gefördert wurde und die Unterschiede und Gemeinsamkeiten zu bisherigen Ergebnissen beleuchtet wurden. Dadurch wurden die Ergebnisse geschärft und abgegrenzt. Dieser Prozess wird ebenfalls in Abschnitt 7.1 dargelegt. Die Diskussion und Abgrenzung findet in Abschnitt 7.4.

Nachdem für verschiedene Teilbereiche zusammenhängende Strukturen existierten, habe ich sie im Zusammenhang betrachtet und Schnittstellen tiefer beleuchtet. Dazu habe ich erneut in die Daten und die bisherigen Memos geschaut und das Prinzip des ständigen Vergleichs weitergeführt. Das Ziel ist am Ende alle relevanten Teilerkenntnisse in eine zusammenhängende Theorie zu bringen.

#### 6.2.4 Memos

Ich habe verschiedene Arten von Memos verwendet (siehe auch Abbildung 6.4):

**Journal:** In dem Journal habe ich regelmäßig tagebuchartig festgehalten, welche Daten ich unter welchem Aspekt codiert habe und welche nächsten Schritte erfolgen sollen z.B. welche offenen Fragen sich durch die Daten ergeben, welche Daten als nächstes codiert werden sollten oder Ideen ob und welche Daten als nächstes erhoben werden sollten. Dies diente vor allem dazu, nach Arbeitspausen den letzten Arbeitsstand rekapitulieren zu können, ähnlich einem Arbeitsprotokoll.

**Planungsmemos:** In Planungsmemos wurden während des Codierens Ideen und Gedanken zu möglichen nächsten Schritten festgehalten (In Abbildung 6.4 am *L* zu erkennen. MAXQDA stellt eine begrenzte Auswahl an Buchstaben zur Verfügung)

**Offene Memos:** Offene Memos wurden zum einen für Hinweise und Hintergrundinformationen zu einzelnen Datenquellen und zum anderen für offene Codes zu einzelnen Textstellen angelegt.

**Theoriememos:** In dieser Art Memos wurden erste Zusammenhänge von einzelnen Codings zu einem gemeinsamen Code festgehalten.(In Abbildung 6.4 am *T* zu erkennen.) Sie dienten dem ständigen Vergleich der einzelnen Daten. Später kamen Erkenntnisse und Ideen, die weiterhalfen zur Entwicklung eines Konzeptes hinzu (siehe auch Beispiel Abbildung 7.1). Dabei wurden Rohdaten im Memo aufgenommen. Freie Assoziationen wurden im Memo getrennt festgehalten.

**Konzeptuelle Ideen:** Neue Kategorien, Strukturen und Gedanken wurden hier festgehalten.

Dokumentg...	Dokument	Code	Dokumentset	Titel	Autor	Geändert am	Vorschau	Her
		Aufgaben als ...		Aufgaben als ...	barke	...4.2017 10:36	Hier geht es ...	Cod
theoriebildun...	Core Kategori...			Core Kategori...	barke	...5.2017 11:17	das zentrale ...	Dok
Team 2_				Team 2_	barke	...7.2017 15:25	Vergleich Lite...	Dok
Team 2_	ananoyme on...			ananoyme on...	barke	...7.2017 09:09	Analyse1. An...	Dok
		Experten		Experten	barke	...7.2017 09:30	dimension "V...	Cod
				Retro	barke	...8.2017 10:21	Retro: frust a...	Frei
				intital Coding ...	barke	...8.2017 10:40	Was untersuc...	Frei
		kompetenz d...		kompetenz d...	barke	...8.2017 15:16	hier ringen di...	Cod
		Schätzen		Schätzen	barke	...8.2017 13:48	Schätzen erfü...	Cod

Abbildung 6.4: Screenshot aus MAXQDA, Übersicht Memos, anonymisiert

### 6.3 Anmerkung zur späteren Darstellung der methodischen Herleitung der Ergebnisse

Das konkrete, praktische, methodische Vorgehen zur Erlangung der dargestellten Ergebnisse wird in Abschnitt 7.1 detailliert beschrieben unter Berücksichtigung der folgenden Rahmenbedingungen:

Das methodische Vorgehen in der GTM ist im Allgemeinen und auch im vorliegenden Fall nicht linear. Die einzelnen Codierschritte gehen in einander über, wiederholen sich und sind selten endgültig abgeschlossen.

Es ist nicht vorhersehbar, welche initialen Codes im Forschungsprozess Relevanz entwickeln und in welchen Zusammenhängen sie zueinander stehen. Neue Erkenntnisse, die die Relevanz und Zusammenhänge in großem oder kleinem Ausmaß verändern, können zu jeder Zeit und auf jeder Ebene des Codierens entstehen. Während des *initialen* Codierens kann ein neues Coding neue Zusammenhänge eröffnen. Neue Aspekte oder bisher nicht betrachtete Themen und Codes können unvorhersehbar an Relevanz gewinnen. Durch neue Erkenntnisse während des *fokussierten* Codierens können sich der Blickwinkel und die Zusammenhänge, d.h. die Ordnung der Ergebnisse ständig verändern und erneutes *initiales* Codieren erfordern.

Das Ergebnis entwickelt sich nicht auf allen Ebenen im gleichen Tempo. Das bedeutet beispielsweise, dass einige Ergebnisteile bereits detailliert betrachtet werden können, während für andere Ergebnisteile erneut Daten erhoben werden müssen, ohne das zuvor sicher ist, ob sich Relevanz entfalten wird.

Software (in vorliegenden Fall MAXQDA) hilft dabei, die sich ständig wachsenden Konzepte und Erkenntnisse zu strukturieren und festzuhalten. Die Software bil-



### 6.3. Anmerkung zur späteren Darstellung der methodischen Herleitung der Ergebnisse

det nur den aktuellen *Ergebnis*stand ab und nicht den Weg zum Ergebnis. Der Prozess zum Ergebnis kann nur in wachsenden und sich entwickelnden Memos festgehalten werden. Erst nach Abschluss des Erkenntnisgewinns wird deutlich, welche Forschungsschritte relevant waren. Zur wissenschaftlichen Nachvollziehbarkeit und Glaubwürdigkeit ist es notwendig, den Weg zum Ergebnis im Nachhinein aus der Vielzahl der vorhandenen Daten erneut nachzeichnen zu können (siehe auch folgenden Abschnitt 7.1).

Durch die stringente Anwendung der GTM ist der Prozess selbst äußerst komplex und vielschichtig, die Ergebnisse sind jedoch linear zu den Daten rückführbar. Das heißt, jedes theoretische Ergebnis beruht auf einer Vielzahl von Daten (was die Definition von *Grounded-Theory* ausmacht). Den linearen Ergebnisstrang retrospektiv sichtbar zu machen, ist Teil des Ergebnisses und die Herausforderung in der GTM.

Alle weiteren methodischen Herleitungen in dieser Arbeit sind also ein im Nachhinein linearisierter, simplifizierter dargestellter Prozess. Diese Darstellung hilft die wissenschaftliche Nachvollziehbarkeit der Ergebnisse zu ermöglichen. Er stellt nur einen relativ kleinen Teil der tatsächlich stattgefundenen parallelen Handlungen und Schritte dar. Der dargestellte Weg zum Ergebnis folgt den Codierschritten Charmazs vom initialen Codieren hin zum fokussierten Codieren.



# Kapitel 7

## Ergebnisse

Nachdem die Methodik, ihre Anwendung und die Datenerhebung erläutert wurden, folgen nun die Ergebnisse. Die Ergebnisse unterteilen sich in ein Modell und Schlussfolgerungen bzw. induktive Hypothesen. Das Modell um das Konzept **KLARHEIT ZU GEWINNEN** beschreibt einen wesentlichen Prozess in Scrum-Teams und ist Kern und Grundlage der Ergebnisse.

Im vorherigen Kapitel 6 wurde das generelle methodische Vorgehen erläutert, dass zu vielen verschiedenen Zwischenergebnissen und zu Ergebnissen, die aus verschiedenen Gründen keinen Einzug in die vorliegende Arbeit erhalten haben, führte. Im nun folgenden Abschnitt 7.1 wird der konkrete methodische Erkenntnisgewinnprozess zum zentralen Modell dieser Arbeit erläutert.

In Abschnitt 7.2 wird erläutert, was das Modell und seine einzelnen Schritte theoretisch beinhaltet: Es spielt weniger eine Rolle *worüber* **KLARHEIT GEWONNEN** wird, da diese Faktoren von Team zu Team individuell verschieden sind. Das Konzept beschreibt einen Prozess innerhalb von Scrum-Teams zur Gewinnung von **KLARHEIT** auf verschiedenen Ebenen.

In den folgenden Unterabschnitten von 7.2 wird das Modell theoretisch tiefer und ausführlich beleuchtet (siehe farblich hervorgehobene Bereiche). In diesem Abschnitt erfolgt auch das Grounding durch Originaldaten als Beleg für das Modell. Es werden anhand von Episoden aus der Praxis die unterschiedlichen Fälle von geglückten und missglückten Bestrebungen **KLARHEIT ZU GEWINNEN** dargestellt, und welche Hypothesen sich daraus ableiten lassen (Abschnitt 7.3).

Abschließend werden die Ergebnisse, das Modell und die Schlussfolgerungen im Kontext verwandter Arbeiten diskutiert (Abschnitt 7.4).

### 7.1 Methodische Herleitung

In diesem Abschnitt wird die methodische Herleitung des Ergebnisses **KLARHEIT GEWINNEN** aus zwei Gründen erläutert:

1. Um die wissenschaftliche Nachvollziehbarkeit sicher zu stellen.
2. Um die Herausforderungen und Hindernisse in der Umsetzung der GTM aufzudecken und die gewählten Lösungswege transparent darzulegen.

Zwei zentrale Konzepte haben dabei zu dem Modell geführt **SELBSTREFLEXION** und **TEAMREFLEXION**. Die Entstehung der beiden Konzepte und den ersten Zusammenhang zum Modell wird nun erläutert.

### 7.1.1 Detaillierter Codierprozess des Konzeptes **Selbstreflexion**

Der erste relevante Schritt, der zum Ergebnis **KLARHEIT GEWINNEN** geführt hat, war das offene Codieren des Zitates Nummer 3 aus einem der ersten Interviews, das bereits in Abschnitt 6.2 als Beispiel für offenes Codieren verwendet wird (siehe Abbildung 7.1, links unten). Zu diesem Zitat wurde unter anderem der Handlung beschreibende induktive Code **SELBSTREFLEXION** vergeben (Ausführlich siehe Abschnitt 6.2), der später im Prozess ein eigenes Konzept wurde und zum Konzept **KLARHEIT GEWINNEN** weiterführte. Weitere Codings und die dazugehörigen, offenen Codes wurden unter dem zu diesem Zeitpunkt offenen Code **SELBSTREFLEXION** vorerst lose gesammelt (siehe Abbildung 7.1 unten Mitte).

Regelmäßig habe ich alle Codings, die zu einem Code gesammelt wurden, erneut gemeinsam bewertet dahingehend, ob einzelne Codings etwas zur Entwicklung des Konzeptes beitragen (oder zu einem anderen Konzept gehören), andere bisher nicht berücksichtigte Codings etwas zu diesem Konzept beitragen würden und welche offenen Fragen sich ergeben, die in den Daten erneut überprüft werden sollten (siehe auch *ständiges Vergleichen* Kapitel 4 und Abbildung 7.1). Während der Zusammenschau wurde ein erstes Memo angelegt, um die Zusammenhänge der Daten und dass sich daraus entwickelnde Verständnis für das gemeinsame Konzept **SELBSTREFLEXION** festzuhalten (Siehe Abbildung 7.1, oben).

Während dieses Schrittes wurde ein weiteres Coding, eine Erzählung eines Consultant aus der Dritte-Person-Perspektive geprüft (siehe Abbildung 7.1, unten rechts):

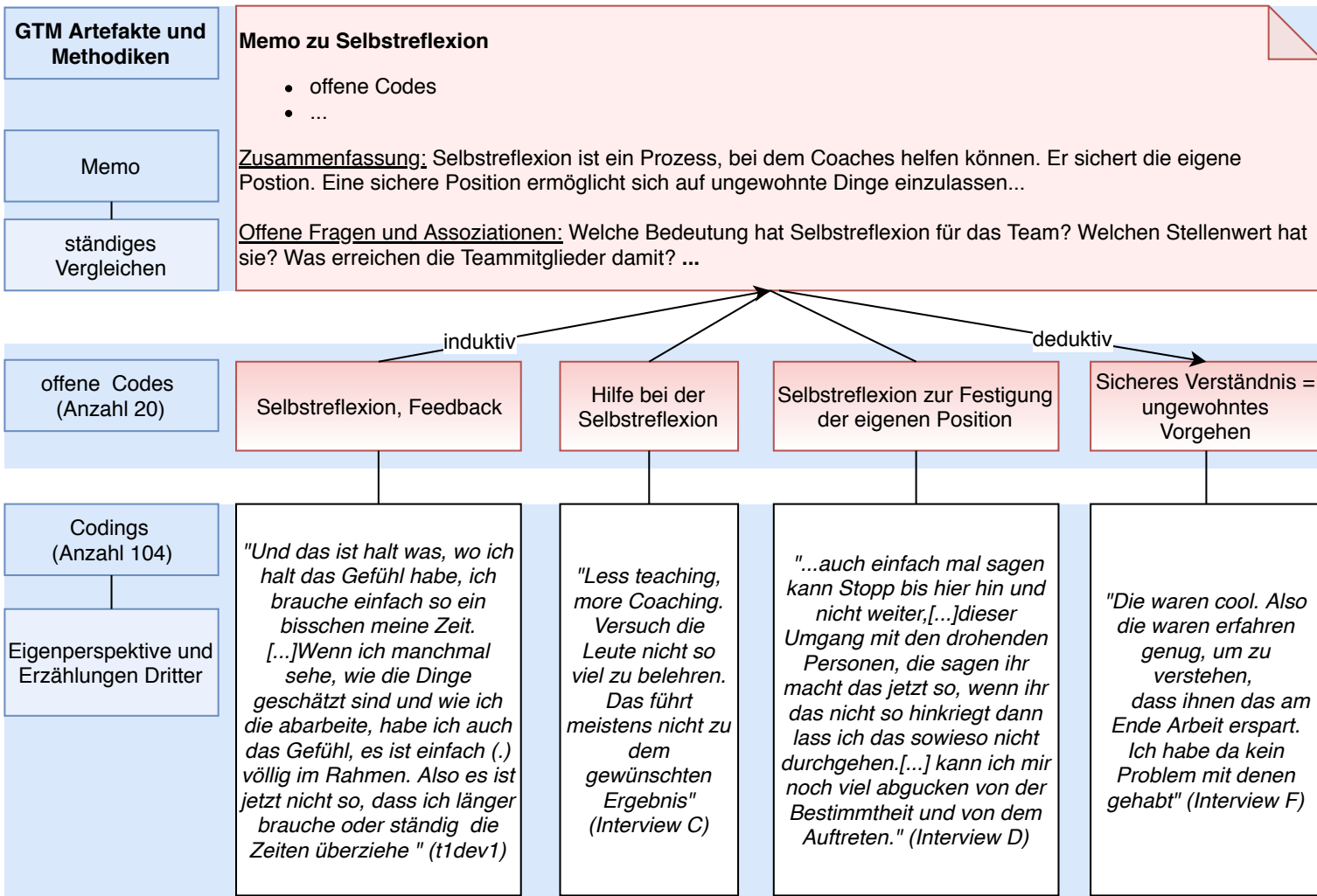
CNr. 5: I: "Und wie haben sich die Programmierer damit verhalten dazu, dass sie dann Papier Prototypen/?"

Experte F: "Die waren cool. Also die waren erfahren genug, um zu verstehen, dass ihnen das am Ende Arbeit erspart. Ich habe da kein Problem mit denen gehabt."

Zu diesem Zitat waren bisher die offenen Codes **CROSS-FUNKTIONAL**, **DOMÄNE**, **UNTERSCHIEDE** vergeben.

Um die hohe Relevanz des Zitats des Consultants besser verstehen zu können, muss der methodische und inhaltliche Kontext berücksichtigt werden. Der Consultant berichtete hier über ein cross-funktionales Team, bei dem die Entwickler\_innen einen Sprint lang nicht programmierten, sondern mit anderen Teammitgliedern des Teams Business-Prozesse simulierten und Papier Prototypen bauten. Im Forschungsprozess war bereits erkannt worden, dass sich Programmierer\_innen häufig schwer tun, andere Aufgaben als Programmieren als gleichwertig relevant wahrzunehmen und zu übernehmen. Daher wurde im Interview die offene Nachfrage gestellt wie sich die Programmierer\_innen in dieser Situation verhalten haben, um tiefere Daten zur Thematik zu erhalten (siehe auch *Theoretical Sampling* Kapitel 4). An dieser Stelle zeigt sich die Mächtigkeit der GT. Die Nachfrage im Interview war *offen* formuliert und eine unendliche Vielzahl von *möglichen* Erklärungen und Antworten wäre denkbar gewesen, wie beispielsweise: "Sie waren einfach bei jeder Aufgabe hoch motiviert", "die Teammitglieder der Business Prozesse und UXler waren besonders überzeugend, dass es

Abbildung 7.1: Grafische Darstellung der ersten methodischen Herleitungsschritte des Konzeptes SELBSTREFLEXION



wichtig ist“ oder *“sie haben es gemacht, weil ich es als Consultant gesagt habe“*. Durch die spontane unmittelbare Antwort wird deutlich, welches die nächstliegende Erklärung für das Verhalten der Programmierer\_innen im Gedankenkonzept des Consultants ist: *“Die waren cool.“* Ein solches Coding zu identifizieren, das implizite Gedankenkonzepte von Personen offenbart, ist das Ziel der GT und gleichzeitig die größte Herausforderung und erfordert äußerst sorgsames Arbeiten, Übung und Geduld sowohl bei der Datenerhebung als auch -auswertung. Ein solches Zitat kann für den Forschungsprozess besonders wertvoll sein, daher wurde es intensiv betrachtet und codiert.

Dieses Coding wurde schließlich auch zum Konzept **SELBSTREFLEXION** zugefügt und im Memo erneut offen codiert mit **SICHERES VERSTÄNDNIS DER EIGENEN POSITION UND FÄHIGKEITEN ERMÖGLICHT AUCH UNGEWOHNTE VORGEHEN**. Die Relevanz ergab sich bei diesem Zitat deduktiv im Vergleich wie im Folgenden dargestellt wird.

Die Programmierer\_innen waren *“cool“* und *“erfahren genug“*. Diese beiden Stichpunkte werden im offenen Codieren als **SICHERES VERSTÄNDNIS DER EIGENEN POSITION UND FÄHIGKEITEN** aufgebrochen. Während die Entwicklerin des Zitats Nummer 3 ein eher unsicheres Selbstbild hat, strahlen diese Programmierer\_innen zumindest nach außen hin etwas anderes aus. Daher erschien ein Vergleich vorerst sinnvoll (ohne dabei die Handlungen der Personen aus dem Fokus zu verlieren). Während einige Programmierer\_innen sich schwer tun, cross-funktional zusammenzuarbeiten und damit auch andere Aufgaben als das Programmieren selbst zu übernehmen, fiel es in diesem Beispiel den Programmierer\_innen leicht. Dieses Coding ist ein interessanter kleiner Baustein<sup>1</sup> bei der Erarbeitung des Konzeptes **SELBSTREFLEXION** unter der Berücksichtigung folgender Schritte:

Bei diesem Schritt des offenen Codierens ist es wichtig die Waage zu halten, wie weit die Daten interpretiert und damit aufgebrochen werden und was sie tatsächlich sagen (siehe auch Kapitel 6). Dieses Coding Nummer 5 und die dazugehörige Codes sagen *nichts* dazu aus, ob das **SICHERE VERSTÄNDNIS** (oder auch Selbstvertrauen oder Souveränität) durch Selbstreflexion entstanden ist. Der offene Code trifft ebenfalls keine Aussage dazu, worauf das sichere Selbstverständnis beruht und es wird auch keine hypothetische Aussage über potentielle vorhergehende **SELBSTREFLEXION** getroffen. Das Konzept hätte später auch umbenannt werden können, daher ist es zu diesem Zeitpunkt vorerst nicht relevant, ob tatsächlich Selbstreflexion stattgefunden hat. Das spätere Konzept geht schließlich über **SELBSTREFLEXION** hinaus und bezieht sowohl Folgen als auch Ursachen von Selbstreflexion mit ein und betrachtet den gesamten Handlungsprozess. Jedes einzelne Coding liefert genau einen kurzen klaren offenen Code und diesen Beitrag zum Konzept - nicht mehr und nicht weniger. Coding Nummer 3 (siehe Abbildung 7.1, unten links oder Seite 97) macht beispielsweise lediglich eine Aussage dazu, dass die Interviewte unsicher ist und selbstreflektiert, aber trifft keinerlei Aussage dazu wie sich das auf ihr Handeln auswirkt. Im späteren Ergebnis beschreiben die beiden Codings 5 und 3 zwei unterschiedliche Schritte im Prozess: Zum einen, dass Selbstreflexion grundsätzlich stattfindet. Zum anderen, welche Handlungen durch ein sicheres Selbstverständnis ermöglicht werden.

Diese zwei Bausteine plus 102 weitere Codings<sup>2</sup>, die ebenfalls in unterschiedlich

---

<sup>1</sup>man bedenke, wir befinden uns immer noch auf der Ebene des offenen kleinteiligen Codierens

<sup>2</sup>die hier aus Platz- und Anonymitätsgründen der Interviewten nicht alle dargelegt werden

notwendigem Maß aufgebrochen wurden in 19 kurze offene Codes und 20 ausführliche offene Codes im Memo<sup>3</sup>, ergeben beim Vergleich und in der Zusammenschau das Konzept **SELBSTREFLEXION**.

Bei der Erarbeitung des Konzeptes ergibt sich die Relevanz der Codings induktiv und deduktiv über den Vergleich zu anderen Codings und durch die *Tiefe* der Daten, die sich aus dem theoretischen Sampling und dem vorherigen Forschungsstand ergibt, beziehungsweise über die erneute vertiefende Nachfrage und die spontane Antwort. Das intensive offene Codieren ist vielfach Voraussetzung, um Zusammenhänge zu erkennen. In diesen detaillierten Beispielen der Codings entwickelte sich die Bedeutung der Codings für das Konzept **SELBSTREFLEXION** erst durch die Erhebung der Daten unter Berücksichtigung von theoretischem Sampling, das Aufbrechen der Daten durch offenes Codieren der Handlungen und den anschließenden ständigen Vergleich mit anderen Codings. Es beschreibt daher beispielhaft den komplexen und schwierigen Prozess und welche methodischen Teile der GT ineinandergreifen müssen.

Zusammengefasst wurde dieses erste Teilergebnis durch die Berücksichtigung wichtiger Prinzipien der GTM nach Charmaz erreicht:

- Nachfragen in der Datenerhebung bauten auf die vorherige Ergebnisse auf (Theoretisches Sampling).
- Die Codings wurden durch offenes Codieren aufgebrochen.
- Es wurden vorwiegend Handlungen codiert und nicht Personen Eigenschaften zugeordnet. Wodurch eine höhere Komplexität<sup>4</sup> im Ergebnis erreicht wurde.
- Es fand ein ständiger Vergleich statt.
- Ausführliche Überlegungen und Gedanken zu sich entwickelnden Konzepten wurden in strukturierten Memos langfristig entwickelt.

### 7.1.2 Detaillierter Codierprozess des Konzeptes **Klarheit gewinnen und die Herleitung des Modells**

Auf dieselbe Weise wurde auch das vorübergehende Konzept **TEAMFUNKTIONALITÄT** entwickelt. Dieser Name entstand spontan im Prozess bei der Beantwortung von Charmazs Frage „*Was beschreiben die Daten?*“ Antwort: „*Wie ein Team gut funktioniert (oder eben nicht)*“. Die offenen Codes in diesem Konzept beschreiben Handlungen, die dem Team helfen, eine positiv wahrgenommene Zusammenarbeit zu erreichen wie **FRAGEN STELLEN**, **EINGESTEHEN VON FEHLERN** oder **OFFENE KOMMUNIKATION** (siehe Abbildung 7.2). Dieses Konzept war inhaltlich sehr offen gehalten und daher umfangreich in den Daten zu finden (siehe Abbildung 7.2, links Anzahl Codings). Zusätzlich lässt es sich extern beobachten (im Gegensatz zu **SELBSTREFLEXION**) und ist

<sup>3</sup>Die Anzahl der Zeichen für Codes ist in der Software begrenzt. In Memos macht die Software keine maximalen Zeichenvorgabe und ausführlichere Gedanken zu den einzelnen Codings und ihre Zusammenhänge sind möglich

<sup>4</sup>Im konkreten Beispiel hat es ermöglicht zwei Codings in einen gemeinsamen Zusammenhang zu stellen und diesen ausdifferenzieren.

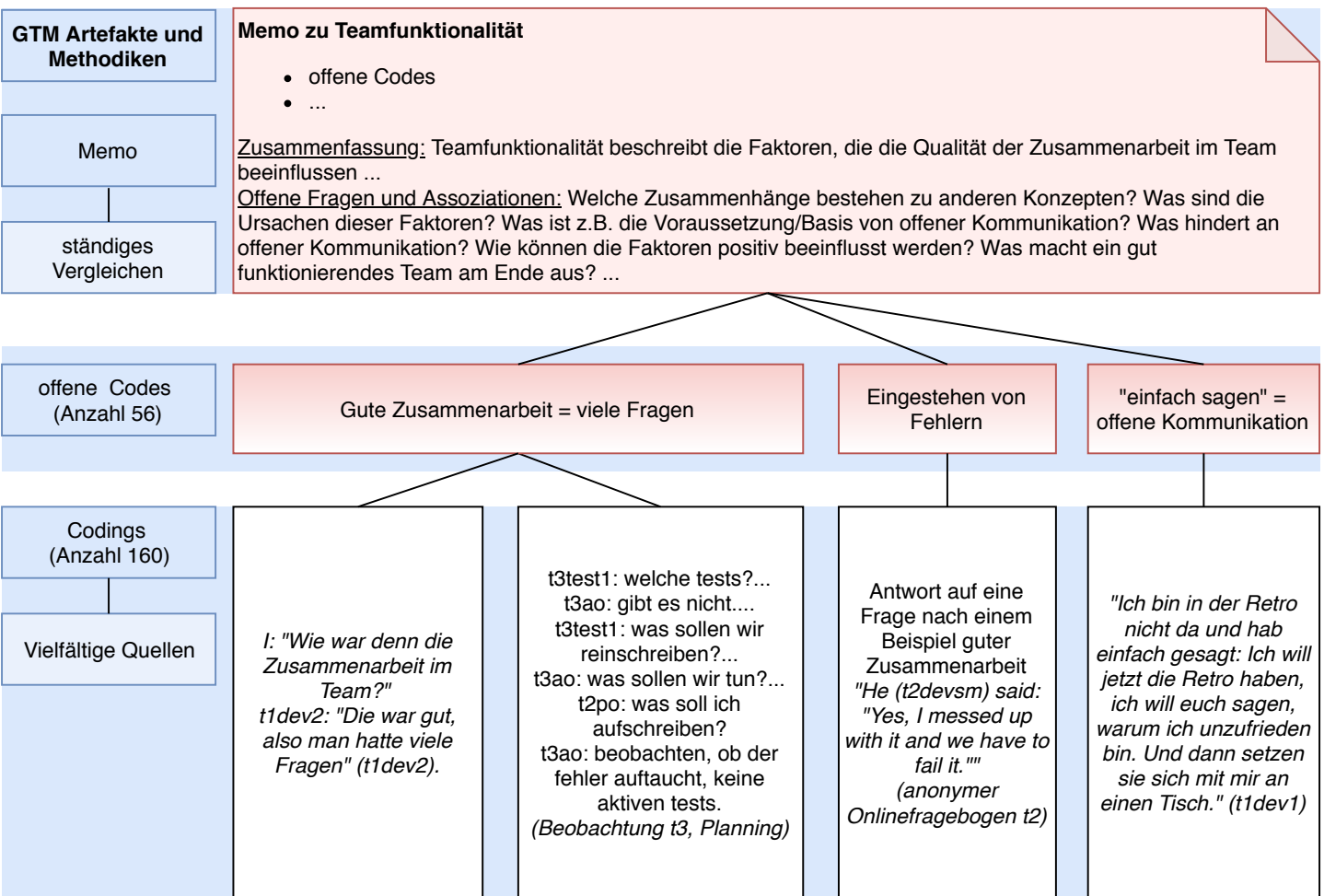


Abbildung 7.2: Grafische Darstellung der ersten methodischen Herleitungsschritte des Konzeptes **TEAMFUNKTIONALITÄT**



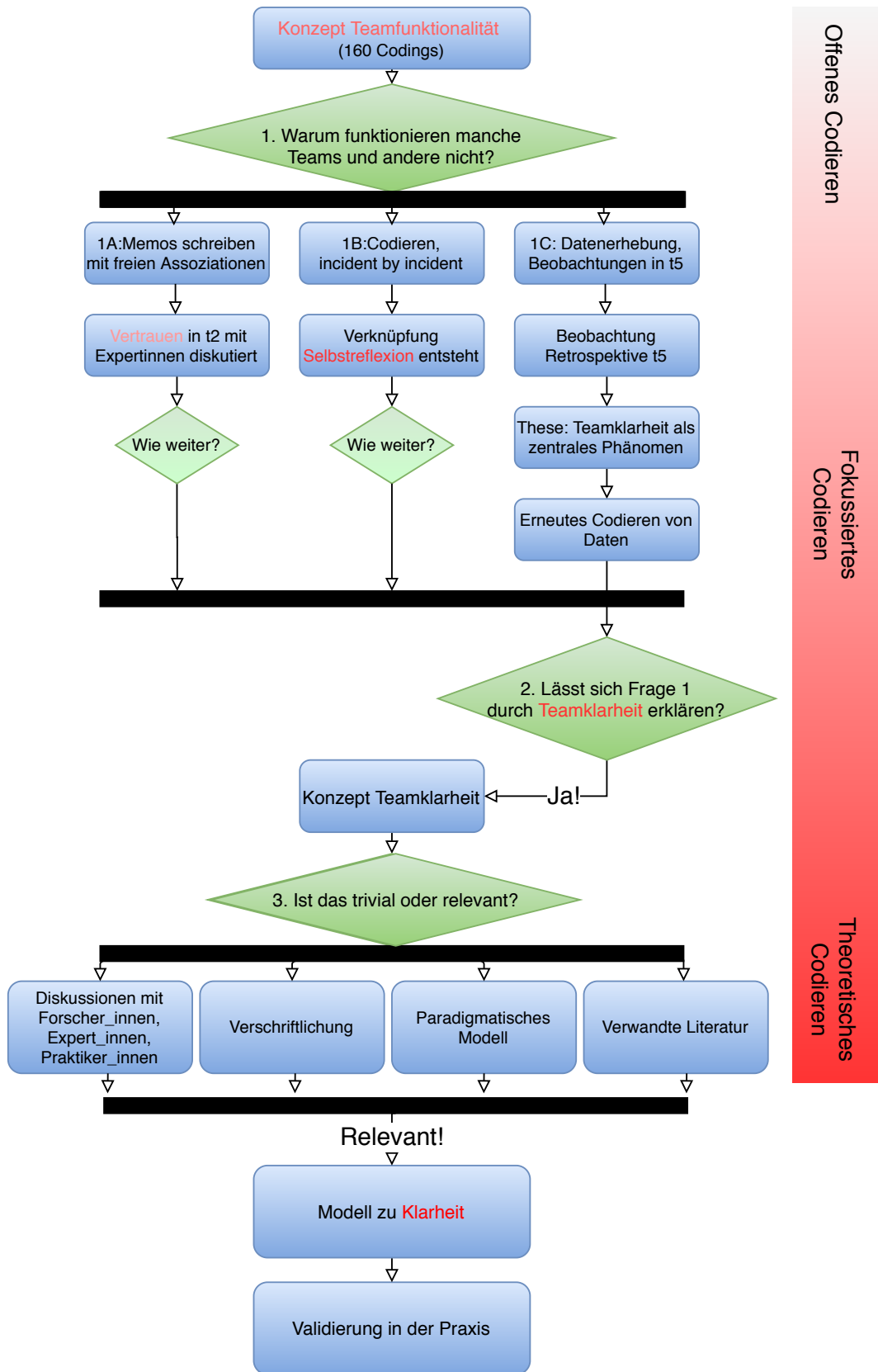


Abbildung 7.3: Vereinfachte, retrospektive, grafische Darstellung des Forschungsprozesses vom Konzept TEAMFUNKTIONALITÄT zum Modell. Erläuterungen siehe Abschnitt 7.1.2.

daher auch in unterschiedlichen Arten von Quellen zu finden (siehe Abbildung 7.2, untere Reihe).

Es bestand eine wachsende Datenlage mit einer Vielzahl offener Codes, die in Memos beschrieben und erfasst wurden. Die Daten aufzubrechen und Zusammenhänge zu erkennen im offenen Codieren, erforderte neben abstrahierendem Denken vor allem Übung und ist vorwiegend induktiv. Im nächsten Schritt in Richtung *fokussiertem* Codieren, ging es nun darum, abstrakter über die *dahinter liegenden Zusammenhänge* nachzudenken und Beziehungen zwischen den einzelnen Konzepten zu erkennen. Der Erkenntnisgewinn wurde abduktiver. Dies ist ein wesentlicher und schwieriger Schritt in der GT (siehe auch Kapitel 6). Der Codierprozess stagnierte vorerst und konnte mit verschiedenen Werkzeugen von Charmaz [Cha14] gelöst werden. Die folgende Herleitung ist in Abbildung 7.3 dargestellt. Die Erläuterungen zur Herleitung orientieren sich an den **drei Fragen in den Rauten**, die den Forschungsprozess begleitet haben. Der Forschungsprozess ist retrospektiv dargestellt. Auf die Fragen wären verschiedenen Antworten möglich gewesen. Dargestellt ist nur die im Prozess tatsächlich gefundene Antwort. Methodische Arbeitsschritte zur Beantwortung der Fragen sind in **den Rechtecken darunter** dargestellt. Der **Balken** an der rechten Seite dient lediglich als Orientierung im Codierprozess. Tatsächlich war auch in diesem Forschungsprozess der Ablauf nicht linear. Nach der 2. Fragen wurde beispielsweise erneut offen codiert wie im folgenden erläutert wird.

#### 1. Frage: Warum funktionieren einige Teams gut und andere nicht?

In einem Memo rückte die Frage in den Vordergrund, warum es einigen Teams gelingt **OFFEN ZU KOMMUNIZIEREN**, **FRAGEN ZU STELLEN** und **FEHLER EINZUGESTEHEN** und anderen Teams nicht. **Welche Ursache oder Faktoren steckte hinter einer guten Teamfunktionalität? Was war die Grundlage oder Voraussetzung für eine gute Zusammenarbeit?** (Siehe Abbildung 7.3, erste Frage, oben).

Im nächsten Schritt entwickelte sich das Ergebnis auf verschiedenen Ebenen durch viele parallel stattfindende Forschungsschritte (siehe Abbildung 7.3, Schritt 1A-1C).

Die bereits bestehenden Memos wurden um ausführlichere Abschnitte von freien Assoziationen und ersten Versuchen der theoretischen Erarbeitung ergänzt. Die Memos wurden nach offenen Codes, gegroundeten Zusammenhängen und offenen bisher unbelegten Fragen und Assoziationen getrennt, um sicher zwischen belegten und unbelegten Erkenntnissen unterscheiden zu können (siehe auch Abbildung 7.1, 7.2, Memo und Abbildung 7.3 Schritt 1A).

Eine mögliche deduktive These als Antwort zu dieser Frage: Ein gut funktionierendes Team vertraut sich. In einer mehrstündigen gemeinsamen Diskussion mit t2po und Interview D wurde die Bedeutung von (fehlendem) Vertrauen in t2 diskutiert (siehe Abbildung 7.3, zweiter Schritt 1A). Vertrauen ist ein relativ abstrakter Begriff, der sich induktiv in den Daten schwer finden lässt. Diese deduktive Überlegung als theoretisches Codieren führte nicht weiter. Sie wurde daher verworfen, um im Sinne Charmaz den Daten zu folgen. Die Frage: Warum kann **OFFENE KOMMUNIKATION** und alle weiteren positiven Handlungen stattfinden, blieb vorerst unbeantwortet und war ständiger Begleiter weiterer Forschungsstrategien.

An dieser Stelle wurde Charmaz Instruktion befolgt „*Incident by incident*“ zu co-

dieren und zu vergleichen, um Unterschiede aufzudecken (siehe Abschnitt 6.2.2 und Abbildung 7.3 Schritt 1B). Statt kleinerer Aussagen wurde eine ganze Episode aus einem Experteninterview codiert. Ein Coach berichtet davon einer PO bei der Selbstreflexion geholfen zu haben. Die PO trat daraufhin im Team selbstbewusster auf. Die Zusammenarbeit verbesserte sich. Durch dieses Coding entstand der erste Zusammenhang von SELBSTREFLEXION und TEAMFUNKTIONALITÄT. Die Selbstreflexion hat zu besserer Zusammenarbeit im Team geführt (siehe Abbildung 7.3, zweiter Schritt 1B).

Wiederum parallel fanden Datenerhebungen und Beobachtungen in Teams statt insbesondere in t5 (siehe Abbildung 7.3, Schritt 1C). T5 war zu diesem Zeitpunkt ein hoch funktionales Team. Während der Datenerhebung (Beobachtungen und Interviews) ging ich unter anderem den offenen Fragen aus dem Konzept TEAMFUNKTIONALITÄT nach und suchte nach Faktoren, die t5 so hochfunktional machten: War es die technische Expertise? Warum gelingt offene Kommunikation? Nach der Beobachtung einer Retrospektive, in der die gegenseitige Wertschätzung untereinander reflektiert wurde, kam die Idee auf: Es ist das Ausmaß an Klarheit, das die Teammitglieder übereinander haben, wieweit sie sich selbst und die anderen Teammitglieder im Arbeitskontext einschätzen können. (Genauerer dazu gleich folgend im Ergebnisteil.) Der Begriff TEAMKLARHEIT floss das erste Mal in ein Memo ein. Es war der Durchbruch im Prozess und der Begriff wurde deduktiv in den Daten angereichert (erneutes initiales Codieren und fokussiertes Codieren). Zitate wie das Folgende bekamen eine zentrale Rolle und bestätigten die bisherigen Erkenntnisse und die sich daraus entwickelnde zuvor teilweise deduktive These:<sup>5</sup> CNr. 6: *“Also da gab es auch mit mir als Scrum-Master [...] auch Konflikte, weil die gesagt haben, wozu brauchen wir denn einen Scrum-Master, wir brauchen doch keinen Babysitter (..) bis dann die Rollen so gefunden waren und das war dann eben nachdem die Rollen dann gefunden waren, war es auch irgendwie echt gut. Und von denen habe ich viel gelernt (lachend). Von denen habe ich nämlich gelernt, dass ich eine Klarheit darüber haben muss, was meine Rolle ist, sonst haben das nämlich andere auch nicht. Unabhängig davon ob man es ausspricht oder nicht. Muss man selber eine Klarheit haben was man tut.”* (Interview G)

## 2. Frage: Lassen sich die Erkenntnisse zu einem Konzept verknüpfen?

Es wurde geprüft, ob die bisherigen Forschungserkenntnisse aus Schritt 1A und 1B mit dem möglichen Konzept von TEAMKLARHEIT erklären lassen (siehe 7.3, 2. Frage). Selbstreflexion floss in das Modell mit ein wurde Voraussetzung für die Erreichung von TEAMKLARHEIT. Die Episoden (*incidents*, Schritt 1B) reicherten das Konzept an und ließen sich umgekehrt durch das Konzept erklären.

## 3. Frage: Ist das Konzept TEAMKLARHEIT trivial oder relevant?

So aufwendig dieser Erkenntnisgewinn und so aufregend der Durchbruch durch das Konzept TEAMKLARHEIT auch war, so trivial erschien es zunächst. Es stellte sich die Frage, ob dies ein relevantes und hilfreiches Konzept ist (siehe Abbildung 7.3, 3. Frage). Der Beantwortung der Frage wurde in vier parallelen Schritten nachgegangen.

<sup>5</sup>Die Hypothese, dass KLARHEIT auch Vertrauen fördert, ist naheliegend, aber wie zuvor beschrieben problematisch in den Daten zu eruieren. Klarheit ist im Vergleich zu Vertrauen ein pragmatischer und relativ leicht zu beobachtender Begriff.

Durch Reflexion mit anderen Forscher\_innen auf Konferenzen<sup>6</sup>, Praktiker\_innen und Kolleg\_innen, rückte die Bedeutung des Konzeptes immer wieder in den Mittelpunkt.

Die Erkenntnisse wurden verschriftlicht. Beim erneuten Zusammentragen der Daten, insbesondere kompletter Episoden, wurde die Reichhaltigkeit der Daten zu diesem Konzept sichtbar. Das theoretische und fokussierte Codieren habe ich für eine Konferenzbeitragseinreichung gemeinsam mit Kolleg\_innen und Praktiker\_innen fortgeführt.

Nach der Erarbeitung des Konzeptes wurde verwandte Literatur herangezogen und zum theoretischen Codieren verwendet. So haben insbesondere Hackmans Erkenntnisse das Konzept angereichert, präzisiert und validiert (siehe Abschnitt 7.4).

Abschließend wurde geprüft, ob das paradigmatische Modell Corbins und Strauss hilft, die Ergebnisse weiter zu systematisieren. Entsprechend Charmazs Vorgaben den Daten zu folgen, anstatt die Daten an eine Systematik anzupassen wurde ein eigenes Modell entwickelt.

**KLARHEIT GEWINNEN** entspricht Charmaz Anforderung Handlungen zu beschreiben, statt Labels zu vergeben. Der vorherige Konzeptname **TEAMKLARHEIT** wurde dem nicht gerecht. Nachdem und während der Entwicklung des Modells wurde es in der Praxis validiert (siehe Kapitel 8).

## 7.2 Modell zur Gewinnung von Klarheit

In diesem Teil wird beschrieben, was das Konzept **KLARHEIT GEWINNEN** beinhaltet:

- Es wird ein Modell beschrieben, auf welchen *Ebenen* Klarheit gewonnen wird, d.h. *wer* über *was* **KLARHEIT GEWINNT** bzw. daran scheitert.
- Es wird erläutert, *wie* und auf *welchen* Ebenen der Prozess zur **GEWINNUNG VON KLARHEIT** (erfolgreich) gefördert werden kann.
- Es werden die positiven und negativen Effekte von **GEWONNENER ODER NICHT GEWONNENER KLARHEIT** erläutert.

Zur ersten Orientierung und kurzen Einführung folgt nun eine *kurzer* Überblick über das Modell, bevor die detaillierte Beschreibung und das Grounding in den Unterabschnitten 7.2.1 bis 7.2.7 folgt.

Anhand der Abbildung 7.4 werden die Modellschritte von links nach rechts im Prozess des Modells erläutert und werden der Abbildung entsprechend farblich hinterlegt. In den weiteren Abschnitten werden die Bezüge zwischen Grafik und Text durch Zahl und Färbung wie beispielsweise **M2.1.1 Ausmaß** verdeutlicht. Die Nummerierung aus Abbildung 7.4 werden im Text mit einem vorangestellten **M-** für *Modell* referenziert, um Verwechslungen mit Kapitelnummerierungen zu vermeiden.

Tatsächlich stehen die einzelnen Schritte nicht, wie im Modell abgebildet, in einem stringent linearen Zusammenhang und sind auch nicht immer kausal. Das Modell ist

---

<sup>6</sup>Wie beispielsweise mit dem Forscher Lenberg, der ebenfalls den Begriff Clarity verwendet [LF18], allerdings in Bezug auf Normen, siehe 2.3.

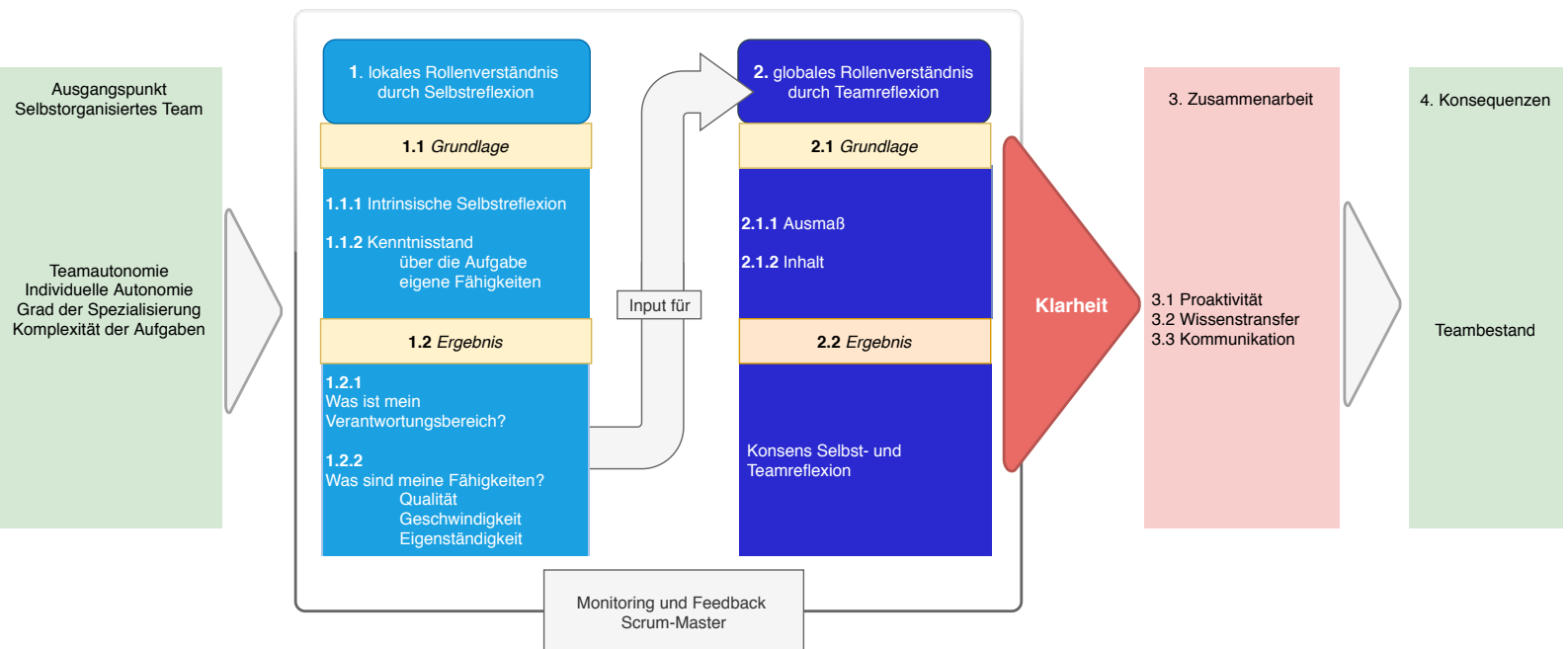


Abbildung 7.4: Ergebnis: Modell, um **KLARHEIT ZU GEWINNEN** und dessen Ausgangspunkt und Konsequenzen in agilen Softwareentwicklungsteams. Eine Einführung zum Modell siehe Seite 118 und die nachfolgenden Abschnitte 7.2.1 bis 7.2.7.

eine Vereinfachung, die ermöglicht, die komplexe Realität zu einem Teil zu systematisieren. Das Modell erhebt keinen Anspruch auf Vollständigkeit, da die Realität niemals auch nur ansatzweise vollständig abgebildet werden kann.

### Eine kurze einführende Beschreibung des Modells:

Der Prozess **KLARHEIT GEWINNEN** beginnt beim Ausgangspunkt (siehe Abbildung 7.4:SELBSTORGANISIERTES TEAM). Erst die Ursache der Selbstorganisation führt zu der Anforderung und der Notwendigkeit, im Team **KLARHEIT ZU GEWINNEN**. Durch den Transfer der Autonomie in das Team selbst, Entscheidungen über Arbeitsprozesse zu treffen (siehe auch Abbildung 2.1), steigen die Anforderungen an das Team derart, dass es notwendig wird, in einem erhöhten Ausmaß gegenseitige **KLARHEIT** über die Verantwortungsbereiche und Fähigkeiten der Teammitglieder **ZU GEWINNEN**. (Weitere Ausführungen siehe Abschnitt 7.2.1.)

Das Hauptaugenmerk liegt auf dem Prozess zur Gewinnung von **KLARHEIT**. **KLARHEIT** wird in zwei Schritten gewonnen. Jedes Teammitglied sollte selbst reflektieren und Rollenklarheit über seine Position im Team gewinnen (siehe Abbildung 7.4: 1. lokales Rollenverständnis durch Selbstreflexion. Weitere Ausführungen siehe Abschnitt 7.2.2).

Im nächsten Schritt ist es notwendig, sein Selbstbild in einer Teamreflexion mit dem Fremdbild abzugleichen und Erwartungen aneinander anhand der Reflexion angemessen anzupassen und einen Konsens zu finden (siehe Abbildung 7.4: 2. globales Rollenverständnis durch Teamreflexion. Weitere Ausführungen siehe Abschnitt 7.2.3).

Durch diese beiden Strategien (Selbst- und Teamreflexion) wird **KLARHEIT** gewonnen. Wie **KLARHEIT** entsteht, wird in den Abschnitten 7.2.2 und 7.2.3 erläutert. Welche Auswirkungen **KLARHEIT** oder fehlende **KLARHEIT** haben, ist in den Abschnitten 7.2.5 und 7.2.6 zu finden.

Das Ausmaß an **KLARHEIT** und der Konsens wirken sich auf die **ZUSAMMENARBEIT** aus (siehe Abbildung 7.4: 3. Zusammenarbeit). Je größer die **KLARHEIT** und der Konsens sind, umso positiver wird die Zusammenarbeit beeinflusst. Das heißt, ein hohes Maß an **KLARHEIT** kann zu *mehr* Wissenstransfer, *höherer* Proaktivität und *verbesserter* Kommunikation im Team führen. (Weitere Ausführungen siehe Abschnitt 7.2.5.)

Die Konsequenz der Qualität der Zusammenarbeit ist der Zusammenhalt im Team oder das Auseinanderbrechen (siehe Abbildung 7.4: 4. Konsequenz. Weitere Ausführungen siehe Abschnitt 7.2.6).

Während jedem dieser Schritte kann der/die Scrum-Master unterstützen und coachen (siehe Abbildung 7.4: graue Einrahmung der Schritte 1. und 2. und weitere Ausführungen in Abschnitt 7.2.7). Insbesondere die Selbstreflexion der Teammitglieder kann durch Feedback und Monitoring angeregt werden. Feedback und Monitoring können durch das Team selbst oder durch externe Personen angestoßen werden.

Das Modell für sich genommen ist deskriptiv d.h. es werden Teamprozesse neutral erklärt und dargestellt. Durch die Methodik der GT findet keine Bewertung der Daten und der Ergebnisse statt.

In der weiteren Beschreibung wird zur vereinfachten Darstellung der Zusammenhänge die Annahme vorausgesetzt, dass Teams danach streben, ein hohes Maß an Teamklarheit zu gewinnen. Ein hohes Maß an **KLARHEIT** stärkt die Zusammenarbeit im Team.

**KLARHEIT** zielt darauf, die gegenseitigen Erwartungshaltungen im Team zu klären. Dabei werden die Erwartungen an die Fähigkeiten einzelner Teammitglieder und die daraus abgeleitete Verantwortung geklärt. Das Verhältnis zwischen individueller Autonomie und Teamautonomie wird ausgehandelt. Welchen konkreten Nutzen das Modell in der Praxis hat, zeigt die Resonanz in der Validierung, siehe Kapitel 8.

Durch Selbst- und Teamreflexion **KLARHEIT** über seine eigene Rolle, Kompetenz und Erfahrung zu gewinnen, ist wichtiger für eine gute Zusammenarbeit als möglichst hohe Kompetenz, Erfahrung und Expertenwissen der Teammitglieder - so legt es das Modell nahe. Wie in den nun folgenden Abschnitten deutlich wird, kann fehlende Klarheit zum Auseinanderfallen des gesamten Teams führen.

Nach dieser ersten Erläuterung der allgemeinen Zusammenhänge des Modells werden nun die einzelnen Schritte im Modell detailliert erläutert.

Jeder Schritt im Modell wird in einem eigenen Unterabschnitt betrachtet. Die Unterabschnitte enthalten die **theoretische Ausarbeitung** und das ausführliche **Grounding** des jeweiligen Modellschrittes. In den Erläuterungen in den Unterabschnitten folgt in der Regel die theoretische Darstellung dem Grounding. Das Grounding, also der Beleg durch Originaldaten, dient

1. zur Vertiefung des Verständnisses des Modells **KLARHEIT GEWINNEN** und
2. der wissenschaftlichen Nachvollziehbarkeit.

Groundings sind im Text durch die Hervorhebung der *Originaldaten* zu erkennen, wie in Abschnitt 6.1 erläutert wurde.

Längere theoretische Ausführungen sind durch farblich hinterlegte Einschübe zu erkennen. Die farbliche Hervorhebung dient zur Orientierung beim Lesen. In diesen Teilen werden bisherige Erkenntnisse zum Modell zusammengefasst und die theoretische Beschreibung des Modells wird hervorgehoben. Teilweise findet in diesen Abschnitten auch eine Einordnung, also Wertung, statt. Die Wertung berücksichtigt dabei die Belegdichte der Daten und übersteigt somit erneut den deskriptiven Teil einer GT.

Zu Beginn werden zwei Episoden dargelegt, die den gesamten Prozess des Modells abbilden.

Die erste Episode besteht aus einem längeren (paraphrasierten) Zitat aus einem Interview, das eine Erzählung beinhaltet. Die zweite Episode ergibt sich aus verschiedenen Zitaten aus Interviews und Beobachtungen. Diese Episoden beschreiben *einen* konkreten Kontext zu jeder Dimension des Modells.

In der methodischen Herleitung 7.1 wird bereits eine Episode erwähnt, die wesentlich zur Erarbeitung des Konzeptes **KLARHEIT GEWINNEN** beigetragen hat. Diese

Episode wird nun im Detail beleuchtet und den einzelnen Schritten im Modell gegenübergestellt.

**Episode 2 (Consultant über PO).** CNr. 7: *“Ach, als ich mal eine junge Product-Ownerin gecoacht habe. War sehr engagiert und kam ziemlich frisch von der Uni und hatte also in dem Aushandeln der Produktteile, um die man sich kümmern sollte, halt irgendwas bekommen und traf nun auf dieses Team. Und dann hat sie gesagt: “Die haben ungefähr 30 Jahre Berufserfahrung Vorsprung vor mir. Wie soll ich denen jemals fachlich gegen die gegenüber treten können? Wenn die mir nur eine Frage stellen, bin ich doch total aufgeschmissen. Ich hab da keine Ahnung davon, wie die das machen”. [...] Ich hab mit der dann ganz eng zusammen gearbeitet. Und nach 3, 4 Monaten als sie das erste Mal in einem Sprint-Planning mit diesem Team zusammen saß und dann einer der Entwickler gesagt hat: “Das ist ja voll kompliziert. Wie soll ich denn das machen? Das kannst du mir jetzt mal gut erklären!”, hat die Product-Ownerin sich zurück gelehnt, am Stuhl festgehalten und hat gesagt: “Das ist nun wirklich nicht mein Problem”. [...] Als ich dieser Kollegin also geholfen hab, ihre Rolle als Product-Owner im Team so zu festigen, dass diese sehr erfahrenen Softwareentwickler sie dafür akzeptiert haben, dafür respektiert haben und sie auch ihre Arbeit haben machen lassen, das war eine große Freude für mich. [...] einerseits dazu gebracht hat, dass Dinge einfacher, besser und schneller funktionieren und andererseits die Menschen dabei zufriedener gearbeitet haben.” (Interview C)*

Der Scrum-Master hat bei diesem Teammitglied der PO **M1.1.1 Selbstreflexion** angeregt über ihren **M1.2 Verantwortungsbereich**, die (nicht) dazugehörenden **M1.2 Fähigkeiten** und die Grenzen aufgezeigt, indem er auf andere Teammitglieder wie Entwickler und ihre Verantwortungsbereiche und dazugehörige Fähigkeiten hinweist. Im Ergebnis wurde die PO **M1.1.2** selbstsicherer über ihre eigenen Kenntnisse und tatsächlichen Aufgaben. Im Planning fand indirekte **M2.1 Teamreflexion** über Verantwortungsbereiche der Teammitglieder statt, in der die PO ihr Ergebnis der Selbstreflexion kommunizierte: *“Das ist nicht mein Problem”*. Das ist nicht ihr Verantwortungsbereich, teilt sie damit mit und schafft **KLARHEIT**. Woraufhin sich die **M3. Zusammenarbeit** im Team laut Aussage des Consultant wesentlich verbessert hat.

**Episode 3 (Fehlende Reflexion).** *t2test3 war bereits lange im Unternehmen und wurde t2 als Tester zugeordnet. Als die interviewten Teammitglieder im Interview aufgefordert wurden, das Team aufzuzeichnen, wurde er beide Male vergessen. t2test1, die zuvor SM im Team war und versucht hat t2test3 im Team zu integrieren, sagt über ihn:*

CNr. 8: *“I feel like a babysitter. [...] we had just understood our roles, then he (t2test3) came into the team. We do not know his role, he does not know his role. I encourage him to talk, but he has a different personality: I try to find my place by myself; he wants somebody to do it for him.” (t2test1)*

Auch *t2dev1* versteht die Rolle von *t2test3* nicht, berichtet im Interview über ein Planning und sagt über ihn:

CNr. 9: *“Ja ich [t2test3] wollte das und das machen. Ok. Jo mach mal. Was hat das jetzt mit uns zu tun? Nichts. Gut. Warum bist du hier? Keine Ahnung. Warum stehst du in unserem Sprint-Board? Weiß ich nicht.” (t2dev1)*

*t2test3* erklärt im Interview, ihm fehle Feedback und das Team würde sich nicht für ihn interessieren:



CNr. 10: B: *“Woran machst du das fest, dass sie sich nicht so interessieren?”*

t2test3: *“Ja eben am Feedback was ich so erhalte. Nämlich gar keins.”*

B: *“Was sollten die denn für ein Feedback geben? Was würdest du dir denn wünschen?”*

t2test3: *“Es reicht ja, wenn sie mal eine Frage stellen.”*

T2dev1 erklärt, wie sich dieses ungeklärte Verständnis der Rolle auf die Zusammenarbeit auswirkt:

CNr. 11: *“Also mich persönlich hemmt das dann manchmal. Ich weiß nicht wie es den Anderen geht, aber ich hab schon oft den Eindruck, dass Dinge anders besprochen werden, weil er einfach/er kann nichts beitragen an der Stelle, wenn es dann an Details geht, wie man das intern macht. [...] Und ich würde von ihm einfach erwarten, dass er dann, wenn wir Features groomen, dass er dann auch einfach sagt: “Hör zu! wir haben hier auch ein Test-Tool. Wie sieht denn das aus? Habt ihr das berücksichtigt? Ich bräuchte die und die Zeichenkette da und da.” Aber das macht er auch nicht. Er bringt sich auch nicht ein. Im Scrum-Team da muss sich jeder einbringen. Und das macht er nicht. Er sitzt einfach nur da. Er ist da. Aber er ist auch nicht so unauffällig da, dass ich ihn ignorieren könnte.” (t2dev1)*

Die gegenseitigen Unklarheiten werden nicht aufgelöst. Als es in der Firma zu Stellenabbau kommt, wird t2test3 entlassen, als Einziger aus dem Team trotz langjähriger Firmenzugehörigkeit und der Anerkennung seines hohen Fachwissens im automatisierten Testen. In Gesprächen wird deutlich, dass es vor allem die Teamsituation war, die das Management zu dieser Entscheidung geleitet hat.

Es gab in diesem Team kein gemeinsames Verständnis über die **M1.2 Aufgaben und den Verantwortungsbereich** von t2test3. Während t2test1 das darauf zurückführt, dass t2test3 zu passiv sei und nicht **M1.1 selbst reflektiere**, bemängelt t2test3 das fehlende Feedback zu seinen Aufgaben. Es fehlt an **M2. Teamreflexion** über die Rolle von t2test3. Dadurch wird keine oder nur wenig **KLARHEIT** erreicht. Die Unklarheit wirkt sich auf die **M3. Zusammenarbeit** aus. T2dev1 fühlt sich gehemmt und hat den Eindruck die **M3.3 Teamkommunikation** sei nicht offen. In der **M4. Konsequenz** wird t2test3 entlassen.

Nach diesen beiden gegensätzlichen Beispielen, die jeweils das gesamte Modell widerspiegeln, werden im Folgendem die einzelnen Schritte des Modells in den Unterabschnitten genauer betrachtet. Die einzelnen Dimensionen der verschiedenen Schritte des Modells werden durch einzelne Zitate und Bruchstücke angereichert und theoretisch ausdifferenziert.

### 7.2.1 Ausgangspunkt:

#### Selbstorganisiertes Team

Dieser Schritt im Modell spiegelt den Stand vorheriger Forschungsarbeiten wider und knüpft daran an (siehe Kapitel 2 und die spätere Diskussion in Abschnitt 7.4.1).

In den Teams konnte beobachtet werden, dass ihre Selbstorganisation geprägt ist von den agilen Strukturen im Team, im Unternehmen und den individuellen Projektanforderungen. Das Maß agiler Transition oder die grundsätzliche Agilität von Teams wurden bereits erforscht und sind daher nicht Hauptaugenmerk dieser Arbeit (siehe Abschnitt 2.2).

Dieser Schritt des Modells entwickelte sich während des theoretischen Codierens mit verwandten Forschungsarbeiten (siehe Abbildung 7.3, nach der 3. Frage links). In den verwandten Arbeiten wird ein Spannungsfeld zwischen Eigen- und Teamperspektive auf verschiedenen Ebenen deutlich (siehe auch später in der Diskussion der verwandten Arbeiten 7.4.1). Diese Spannung wird in den verwandten Arbeiten als wesentliches Merkmal selbstorganisierter Teams dargestellt. Die individuelle Autonomie und Teamautonomie erhöhen sich in den Teams, Spannung zwischen diesen beiden Ebenen entsteht und die Anforderung damit umzugehen, wird größer.

Dieser Teil des Modells ist also weniger in den Daten als in der Literatur theoretisch begründet und greift auf bisherige Erkenntnisse zurück. Dieser Schritt ist insofern notwendig zur Erklärung, da er begründet, warum den weiteren Schritten im Modell eine wesentliche Bedeutung zukommt. Das Modell bietet Möglichkeiten, die Spannung zwischen den beiden Perspektiven (Eigen- und Teamperspektive) aufzulösen.

Die Dimensionen oder Attribute dieses Modellschrittes lassen sich subsumieren mit der Frage: Welche agile Reife haben das Unternehmen und das Team (entweder durch agile Transition oder - in jungen Unternehmen und Teams - von Beginn an) erreicht. Die Ausprägung der Dimensionen ist je nach Team individuell und bestimmt das Maß an Selbstorganisation.

Unabhängig von der Ausprägung hat (die Einführung der) Selbstorganisation *grundsätzlich* immer zur Folge, dass **KLARHEIT GEWONNEN** werden muss über die Verteilung und den Umgang der nun (in individueller Größenordnung) zugewonnenen Autonomie, um eine positive Zusammenarbeit zu ermöglichen.

## 7.2.2 1. Prozessschritt zur Gewinnung von Klarheit

### Lokales Rollenverständnis durch Selbstreflexion

Der erste Schritt zur **GEWINNUNG VON KLARHEIT** ist die **M1. Selbstreflexion** der Teammitglieder. Die **M1.1.1 intrinsische Selbstreflexion** ist dabei unterschiedlich ausgeprägt. Sie kann durch externes Feedback z.B. durch das Team oder durch den Scrum-Master angeregt werden. Dies gelingt aber nicht immer. In Episode 2 hilft der SM extrinsisch das unsichere Selbstbild über Verantwortungsbereiche und Aufgaben anzupassen. In Episode 3 gelingt es nicht laut Aussage der ehemaligen SM t2test1, Selbstreflexion anzuregen. t2dev1 beispielsweise sieht den agilen Transitionsprozess als große persönliche Herausforderung und betont die Notwendigkeit der Selbstreflexion:

CNr. 12: *“Aber das ist natürlich ein langer Prozess. Das ist einfach eine Umstellung im Denken, also eine mentale Umstellung. Es ist wahnsinnig schwierig. Da musst du schon sehr viel Selbstreflexion haben”* (Interview t2dev1)

Er erklärt, dass er dieser Herausforderung nur gewachsen war, indem er neue Wege gegangen ist, die nicht in sein Selbstverständnis als Mann passten:

CNr. 13: *“Und jetzt sag ich mal ein böses Wort, was man mit Männern vielleicht in nicht Verbindung bringt: Meditation. Das hilft. Das hilft einfach mal darüber nachzudenken, was stört dich jetzt so? Warum bist du jetzt so angepisst? ”* (t2dev1)

Auch t5sm1 hat in seinem Team Schwierigkeiten, ein Teammitglied zur Selbstreflexion anzuregen, wie er berichtet:

CNr. 14: *“Bei ihm [t6dev2] ist auch Selbstbild und Fremdbild ein bisschen anders. Also wenn ich ihm jetzt Feedback gegeben habe.. “Das ist nicht gut!” , sagt er: “Hä, achso das kann ich gar nicht einsehen. Ich dachte ich bin hier der Super-High-Performer” und er ist der Beste von allen. [...] Das ist halt das Problem und ich habe ihm auch Feedback gegeben diesbezüglich. Hol dir doch mal die Meinung von anderen noch mal ein, wenn du mir das nicht glaubst, dass meine Meinung jetzt exklusiv ist oder sowas, oder versuch mal das und das vielleicht zu verändern oder sowas. Im nächsten Gespräch war es dann wieder so: “Und? Hast du was getan” und er so: “Ach nee, so ist doch alles super.”” (t5sm1)*

Es gibt aber durchaus auch Teammitglieder, die proaktiv versuchen ihren Verantwortungsbereich S1 zu klären wie t2test3 (der selbst in Episode 3 Schwierigkeiten damit hat) über einen seiner neuen Kollegen berichtet:

CNr. 15: *“Ja er (t2test2) ist damals auf mich zu gekommen, als er angefangen hat wegen Test Automatisierung und wollte eben auch wissen, welche Aufgaben er übernehmen kann. Das haben wir jetzt auch gemacht. Er ist seit wenigen Monaten erst hier. Er macht sich ganz gut.” (t2test3)*

Punkt M1.1.1: Die Beispiele aus den Daten machen deutlich, dass es wichtig ist, im agilen Kontext eines selbstorganisierten Teams selbstreflektiert zu sein. Dabei beschreiben Teammitglieder es als Herausforderung. Scrum-Master versuchen, Selbstreflexion anzuregen und scheitern ggf. auch daran. Nicht selbstreflektierte Teammitglieder verursachen, wie in den folgenden Schritten noch sichtbar wird, Irritationen und Unruhe im Team oder können zum Bruch führen.

Ein Teammitglied reflektiert (im besten Falle), um ihr oder sein eigenes Rollenverständnis im Team zu klären. Das beinhaltet abzustecken, was ihr oder sein **M1.2.1 Verantwortungsbereich** ist bzw. für was sie oder er sich verantwortlich fühlt.

t1dev1 ist neu im Team und reflektiert viel über sich: CNr. 16: *“Man arbeitet eigentlich immer mit irgendwem an irgendwas. Dann war es halt irgendwann so, dass ich mir eine User-Story genommen habe [...] man hat halt seine eigene genommen und dann hat man sich die nächste Aktivität genommen und die nächste. Und dann hat man irgendwann gesagt: Kann da mal einer gucken? Ah ja, ich mach die noch fertig und dann gucke ich mal. Und irgendwann war sie dann fertig und dann dachtest du: Mensch, (.) fertig und ich kann jetzt ein Review eintragen und das Einzige, was wer Anderes macht ist das Reviewen. Also es ist dann so: Oh, das ist seine und man hat das alles alleine gemacht. ” (t1dev1)*

t1dev1 reflektiert hier über ihren **M1.2.1 Verantwortungsbereich**, ihre **M1.2.2 Fähigkeiten** und dabei insbesondere die **Qualität, Geschwindigkeit und Eigenständigkeit** ihrer Arbeit. Auch das bereits mehrfach erwähnte Coding Nummer 3 auf Seite 97 über die Geschwindigkeit ist von ihr.

M1.2 Teammitglieder reflektieren über ihre Fähigkeiten, die es ihnen ermöglichen, Verantwortung anzunehmen bzw. ihr gerecht zu werden. Über welche Fähigkeiten **KLARHEIT** gewonnen werden sollte, ist teamabhängig verschieden. Beispielsweise geht es darum, ob ein Teammitglied Arbeit in der vom Team geforderten Qualität liefern kann, wie schnell es sie liefern kann und wie eigenständig es arbeitet bzw. wie viel Hilfe es benötigt wie das Beispiel ebenfalls gezeigt hat.

Während das Modell eine lineare Darstellung ist, ist es in der Praxis viel mehr so, dass das Team Erwartungshaltungen in der Teamreflexion kommuniziert und die Teammitglieder über die an sie gestellten Erwartungen bezüglich ihrer Rolle (also Verantwortung und Fähigkeiten) selbst reflektieren.

Die Erwartungshaltung des Teams ist nicht, dass alle Fähigkeiten möglichst hoch ausgeprägt sind. Es geht lediglich darum, den eigenen Aufgaben- und Verantwortungsbereich zu kennen.

Wie unterschiedliche die Erwartungshaltung an Teammitglieder sind, machen folgende Beispiele deutlich:

CNr. 17: *“Dann gibt es noch einen der immer lächelt und freundlich ist, aber nicht so wirklich alles kapiert, aber er macht, sag ich mal, seine Aufgaben.”* (Interview E)

Und hier ein Beispiel, bei dem die Fähigkeit der Geschwindigkeit als zweitrangig eingestuft wird:

CNr. 18: *“Er ist zwar manchmal ein bisschen langsam [...] in der Implementierung. Es dauert ein bisschen länger bei ihm. Aber darum geht es ja nicht. Hauptsache es funktioniert am Ende. Mir ist ja egal, wie lange es dauert. Hauptsache es funktioniert. Es muss funktionieren.”* (t2dev1)

Auch Interviewpartner M ist der Ansicht, dass außergewöhnliche Fähigkeiten relativ sind:

CNr. 19: *“That’s people like Rxxx. Rxxx is not a genius. He doesn’t do amazing work and he doesn’t have a really high base but he is funny. He is really really funny. And I celebrate his sense of humour because that gives him something that nobody else in the team has. So for a little sad or little depressed or things didn’t go. Rxxx will get this healthy. Rxxx is a catalyst in that sense. No but he’s not the 007 architect. You understand the term? He’s not, he’s just a normal architect. And people appreciate that about Rxxx. Recently we had a round of promotions and Rxxx didn’t get promotion because the organisation I worked for unfortunately doesn’t see funny as a high value on their scale. Because I’m able to show Rxxx a lot of respect because I like him.”* (Interview M)

Im folgenden Beispiel wird besonders deutlich, dass **M1. Selbstreflexion** und das **M2. Finden von Übereinstimmung** wichtiger ist als die Höhe der Fähigkeiten:

CNr. 20: *“Also theoretisch hätte man ihm zur Probezeit schon sagen müssen. “I’m sorry t6dev1 hat nicht geklappt.” Jetzt haben wir ihn so ein bisschen durchgeschleppt, möchte ich vielleicht unfairer Weise sagen. Bei ihm ist das Ding, er ist total selbst reflektiert. Er weiß, dass er langsam lernt, das hat er auch schon selber gesagt, er weiß, dass sein Problem ist, dass er hier sehr viel neue Technologie kennenlernt und er halt noch ein bisschen altbacken ist. Das sieht man bei seinen HTML-Codes die er schreibt, wenn er eine Seite programmiert und die sieht dann wie aus den Neunzigern aus, aber er fand es gut. Er hat schon einen anderen Stand und der sagt das auch selber von sich so im Englischen: “It’s hard to learn about new tricks.” Das weiß er auch selber. Er weiß, dass er Probleme hat und wir versuchen ihn so ein bisschen in Richtung QA zu bringen und ihn da zu fokussieren auf einen Bereich, was er betreiben kann”* (t5sm)

Teammitglieder reflektieren über Fähigkeiten. Im Modell sind lediglich Beispiele für Fähigkeiten genannt, die umfangreicher gegroundet wurden. Während Fähigkeiten wie Eigenständigkeit und Geschwindigkeit häufiger in den Daten auftauchen und im Modell enthalten sind, sind Fähigkeiten wie Humor seltener genannt. Wie das Beispiel von Interviewpartner M zeigt, ist die Liste der Fähigkeiten und ihre Ausprägung nicht abgeschlossen und teamabhängig verschieden. Es geht darum, über die Fähigkeiten zu reflektieren, die für das Team relevant sind, da sie gebraucht werden. Das kann technische Fähigkeiten, aber auch soziale Fähigkeiten einschließen. Das Team hat Erwartungshaltungen zu bestimmten Eigenschaften und schätzt diese unterschiedlich wert. Dabei ist es aus Perspektive des Teams nicht wichtig wie schnell oder gut eine Aufgabe erledigt wird, sondern dass das Selbstverständnis des Teammitgliedes zu der Einschätzung des Teams ihrer/seiner Person passt.

Der folgende Teil zu M1.1.2 Kenntnisse ist nur geringer gegroundet und bietet ggf. Anknüpfungen für spätere Arbeiten.

Durch Reflexion über die eigene Rolle entsteht gegebenenfalls **M1.1.2 Selbstsicherheit**, wie in Episode 2 zu sehen ist oder im folgenden Zitat:

CNr. 21: *“Von denen [ein altes Team] habe ich nämlich gelernt, dass ich eine Klarheit darüber haben muss, was meine Rolle ist, sonst haben das nämlich Andere auch nicht. Unabhängig davon, ob man es ausspricht oder nicht, muss man selber eine Klarheit haben, was man tut.”* (t4sm2)

Unsicherheit wiederum entsteht dann, wenn keine oder nur wenig Kenntnis über die eigenen Fähigkeiten besteht, weil es beispielsweise keine Erfahrungswerte gibt. Die Genauigkeit der eigenen Einschätzungen und die Wahrscheinlichkeit, dass diese zutrifft sinkt und damit die Selbstsicherheit:

CNr. 22: *“Hardest to contribute for me, are situations when there is grooming for a new feature, where I cannot rely on my experience, because it is something totally new.”* (Anonyme Onlineumfrage t2)

Die **M1.1.2 Kenntnis** hängt ggf. auch davon ab, ob und wie viel Feedback vom Entwicklungsteam, Scrum-Master und anderen Personen als Input gegeben wird. Feedback hilft sich relativ zu anderen Teammitgliedern im Team zu verorten, inwieweit die Person im Wissens- und Erfahrungsstand den anderen Teammitgliedern voraus ist oder noch von Erfahrung lernen kann. Dabei ist der Zusammenhang hohe Fähigkeiten gleich hohe Selbstsicherheit nicht kausal wie das Coding 20 zeigt, in dem t6dev1 zu seinen veralteten Programmierkenntnissen selbstsicher steht. Auch eine Person, die z.B. sich selbst eher mit niedrigen Fähigkeiten einschätzt, kann selbstsicher damit umgehen.

### 7.2.3 2. Prozessschritt zur Gewinnung von Klarheit:

#### Globales Rollenverständnis durch Teamreflexion

Im nächsten Schritt muss das Ergebnis der Selbstreflexion mit dem M2.1 Team geteilt, verhandelt und im besten Falle eine M2.2 Übereinstimmung gefunden werden. Es bedarf der Reflexion im Team, das Feedback gibt welche Verantwortung es bei welchem Teammitglied sieht und welche Erwartungshaltungen damit an dessen Fähigkeiten gestellt werden. Wenn Übereinstimmung gefunden wurde, entsteht ein globales Rollenverständnis. Global bezieht sich hierbei lediglich auf ein teamweites Rollenverständnis.

Teamreflexion kann zu jeder Zeit stattfinden. Die Retrospektive ist im Scrum-Prozess dafür vorgesehen.

Daher zuerst ein musterhaftes Beispiel einer Teamreflexion zu einem Teammitglied während einer Retrospektive:

Ein ganzes Team  $t_3$  reflektiert sehr ausführlich über die Einarbeitung eines neuen Teammitglieds  $t_{3dev3}$ , der während der Reflexion selbst nicht anwesend ist. Im Team wird vorwiegend deutsch gesprochen.  $T_{3dev3}$  hat eine Sprachhürde, da er kein Muttersprachler ist. Während der Reflexion lässt der SM des Teams seine Erkenntnisse über die Wünsche von  $t_{3dev3}$ , die aus gemeinsamen Reflexionen über seine Rolle stammen, einfließen. Dieser Ausschnitt stammt aus einer Mitschrift einer Beobachtung und ist daher keine vollständige wortwörtliche Wiedergabe, sondern sinngemäß mitgeschrieben:

CNr. 23:  $t_{3sm}$  merkt an, dass sie nicht zu viel ohne die Anwesenheit  $t_{3dev3}$  diskutieren sollten.

$t_{3test1}$  sagt: Sie sollten aber einen Plan haben.

$t_{3sm}$  fasst die bisherige Diskussion zusammen: Redegeschwindigkeit und Vokabeln und Abkürzungen sind ein Problem. Was könnte noch Thema sein?

$t_{3ao1}$  soll von konkreter Erfahrung berichten. Obwohl er [ $t_{3dev3}$ ] länger gebraucht hat, war es sein Wunsch, den Testfall selbst zu bearbeiten, weil er etwas lernen möchte.

$t_{3test1}$  fragt  $t_{3dev2}$ : Hat er sich aktiv beklagt?

$t_{3dev2}$ : Nein

$t_{3sm}$  ergänzt:  $t_{3dev3}$  hat ihm in einem persönlichen Gespräch gesagt, er möchte Mehrwert für das Team leisten.

Frage von  $t_{3test1}$ : Wodurch wird er aus seiner Perspektive gebremst? Fachlich oder technisch?

$t_{3dev1}$ :  $t_{3dev3}$  macht vollständige Tasks. (Und leistet damit Mehrwert.)

$t_{3po}$ : Im nächsten Sprint gibt es gute Möglichkeiten, da alle auf der grünen Wiese anfangen.

Darüber herrscht Konsens.

$t_{3po}$ : Im nächsten Sprint sollten nicht so viele Tickets eingeplant werden, damit Pairprogramming möglich wäre.

$t_{3test1}$ : Stimmt zu, damit wäre auch die Sprachhürde durch 4-Augen geringer (bilateral).

$t_{3dev1}$ :  $t_{3dev3}$  könnte Vorbehalte haben gegen direkte Zusammenarbeit. Er will sich gerne erst alleine ein Bild machen. Dies ist meine Erfahrung bisher.

## 7.2. Modell zur GEWINNUNG VON KLARHEIT

t3ao1: Nach dem Refinement sollte man mit t3dev3 das Gespräch suchen, um Verständnisprobleme zu klären.

t3dev1: Habe t3dev3 das letzte Mal im Refinement abgewürgt, als er mir eine Frage stellen wollte und ihn gebeten, die Frage aufzuschreiben und später zu stellen, da ich selbst zuhören musste und mich konzentrieren, da Thema neu. t3dev3 hat hinterher die Frage nicht mehr gestellt.

t3sm: Bei dem neuen Thema t3dev3 ermutigen, direkt in die ganze Runde zu fragen. Das hält die Gruppe nicht auf, da alle unwissend sind.

t3test1: t3dev3 konkret darauf hinweisen, Fragen zu notieren.

t3sm steht am Flipchart und notiert.

t3sm: Wie wollen wir das konkret umsetzen und wer? Z.B. User-Stories vorher per Email schicken.

t3po: Muss schauen, ob das möglich ist, da die neuen User-Stories nicht so ausgereift sind wie die bisherigen, da neues Thema, grüne Wiese.

t3dev1: Behalten wir die doppelte Zeit für das Refinement/Planning?

t3test1: Das sollte eingeplant und berücksichtigt werden. (Für was?)

t3po: Diesen Sprint ja, danach mal schauen.

Maßnahmen werden festgehalten (siehe Foto)

t3test1 scherzhaft: Und t3dev2 holt Feedback bei t3dev3 ein (beim Mittagessen).

t3ao1: Wir sollten t3dev3 fragen!

t3sm: Habe ich gemacht und ich werde ihm auch von den heutigen Ergebnissen berichten. Außerdem bitte ich t3po und t3po2 im Refinement bei der Moderation bei t3dev3 aktiv nachzufragen.

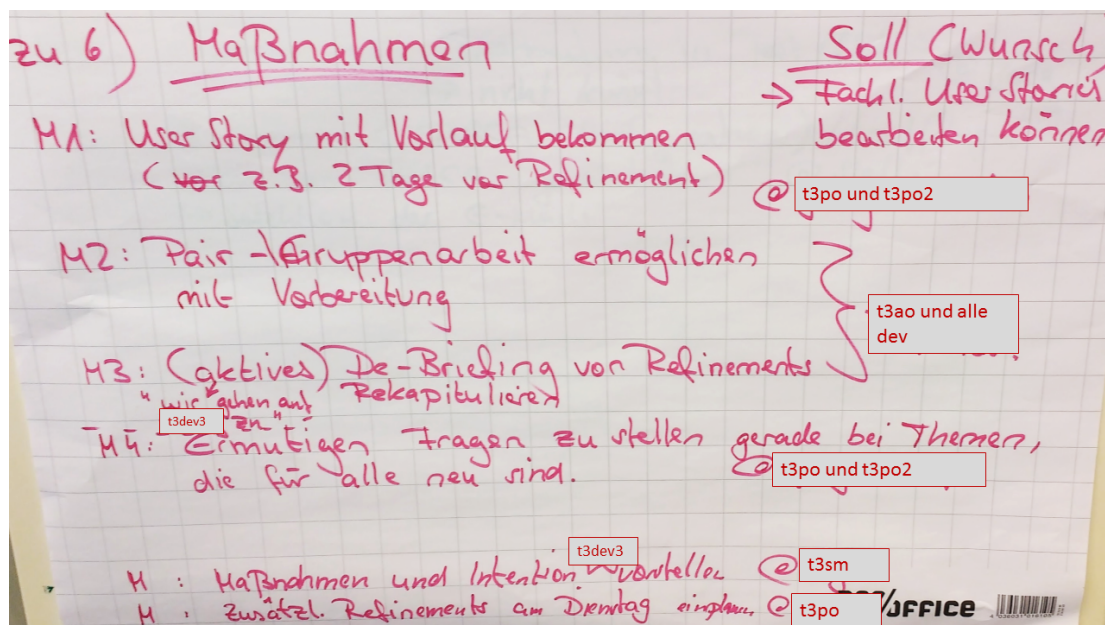


Abbildung 7.5: Foto des Protokolls der Retrospektive von t3 am 06.02.2017: Maßnahmen zur Einbindung von t3dev3. (Namen anonymisiert und durch Synonyme ersetzt)

In diesem Beispiel Coding Nummer 23 beteiligen sich viele verschiedenen Team-

mitglieder proaktiv an der Diskussion, das neue Teammitglied einzubinden und ihm zu ermöglichen, seinen **M1.2 Verantwortungsbereich und Fähigkeiten** zu erweitern. Die Maßnahmen werden, wie in Abbildung 7.5 zu sehen ist, festgehalten und zusätzlich durch  $t_{3sm}$  persönlich kommuniziert. Durch die Diskussion entsteht **KLARHEIT**, was das Teammitglied für Fähigkeiten hat und wie das Team ihm helfen könnte. Durch diese **KLARHEIT** können Maßnahmen entwickelt werden, um die Zusammenarbeit zu stärken.

Die Maßnahmen sind einzelnen Teammitgliedern zugeordnet und haben zum Ziel, die **M3. Zusammenarbeit** zu verbessern. Die **M3.3 Kommunikation** wird geöffnet für Fragen, die **M3.1 Proaktivität** wird angeregt mit der Aufforderung, Fragen zu stellen und **M3.2 Wissenstransfer** für alle zu ermöglichen.

Diese Verbesserung der Zusammenarbeit konnte nur erreicht werden, da zuvor **KLARHEIT** gewonnen wurde. Das Team hat darüber reflektiert, welche Fähigkeiten das Teammitglied hat (er möchte Zeit, um sich bevorzugt alleine einen Überblick verschaffen; er ist in der Lage, Fragen zu stellen; er hat eine Sprachhürde; er kann tiefen Diskussionen, in denen sich andere auskennen, nicht vollständig folgen). Das Team reflektiert auch über den Verantwortungsbereich von  $t_{3dev3}$  (er möchte Mehrwert schaffen). Durch diese **KLARHEIT** über seine Fähigkeiten und Verantwortungsbereiche können Maßnahmen entwickelt werden, die den Bedürfnissen von  $t_{3dev3}$  und den Bedürfnissen und der Erwartungshaltung des Teams entsprechen: Alle fangen beim nächsten Thema neu an, es ist mehr Zeit und Raum für Fragen,  $t_{3dev3}$  kann aktiv aufgefordert werden, seine Fragen einzubringen.

Dieses Beispiel steht im starken Kontrast zu der Einführungsperiode 3, in der keine **KLARHEIT** gewonnen wird. In der Einführungsperiode reflektieren weder  $t_{2test3}$  noch das Team über den Verantwortungsbereich und die Fähigkeiten von  $t_{2test3}$ . Dem Team ist völlig **UNKLAR**, welche Fähigkeiten und Verantwortungsbereiche  $t_{2test3}$  mitbringt. Sie gewinnen keine **KLARHEIT** darüber, welche Fähigkeiten  $t_{2test3}$  besitzt und für was er sich verantwortlich fühlt, um ihn besser einbinden zu können.  $t_{2dev1}$  fühlt sich durch die fehlende **KLARHEIT** gehemmt in der Kommunikation. Es herrscht gegenseitiges Schweigen. In sämtlichen beobachteten Scrum-Events hat  $t_{2test3}$  nur einmal etwas gesagt. Die Unklarheit hemmt alle positiven Folgen in der **Zusammenarbeit**: Es herrscht Passivität, es findet keine Kommunikation untereinander statt und keinerlei Wissenstransfer.

Fehlende **KLARHEIT** hemmt und bestehende **KLARHEIT** fördert damit **Zusammenarbeit**.

Während in diesem Beispiel die Klärung der Verantwortung explizit in der Retrospektive besprochen wird, finden die meisten in den Daten gefundenen Teamreflexionen implizit in anderen Scrum-Events und Situationen statt. Insbesondere im Planning wird die (neue) Autonomie selbstorganisierter Teams wahrgenommen und verhandelt, indem Aufgaben definiert und Verantwortungsbereiche geklärt werden. Das Entstehen für Verantwortung und Aufgabenbereiche spiegelt ggf. das Ergebnis der Selbstreflexion einzelner Teammitglieder wider. Das Team adressiert fachliche Fragen ggf. an einzelne Teammitglieder. Das ist indirektes Feedback der Teammitglieder darüber, wem sie welche Kompetenz zuweisen.

Im folgenden Beispiel wird die Reflexion über Verantwortungsbereiche implizit über einen Konflikt ausgetragen, der u.a. während des Refinements (Grooming) statt-



findet:

t2dev1 ist nicht einverstanden mit dem Verhalten eines Teammitgliedes t2dev2 und teilt ihm das im gemeinsamen Teammeeting dem Refinement deutlich mit:

CNr. 24: *“Und ich habe im Laufe der Jahre eine enorme Geduld entwickelt. Aber irgendwann ist auch da mal Ende. Das habe ich ihm [t2dev2] auch sehr deutlich zu spüren gegeben. Also jetzt nicht auf beleidigende Art, aber ich habe ihm schon sehr deutlich gesagt, was ich davon halte, dass er ...(3 sek) er aus meiner Sicht ziemlich viel Scheiße baut und ziemlich viel Unsinn macht und Leute nicht beteiligt. Also wirklich unter der Hand Dinge tut ..(2 sek) das wir im Grooming User-Stories besprechen und er uns dann sagt: “Ja habe ich doch schon längst fertig. Muss ich nur noch committen.” “Entschuldige, du hättest uns vielleicht beteiligen können an der Entscheidung! Du drückst uns jetzt deine Entscheidung rein und das ist irgendwie nicht so geil. Und wir als Team müssen ja nachher dafür geradestehen.” Und das habe ich ihm sehr deutlich zu verstehen gegeben, dass mir das überhaupt nicht gefällt. Und da war sehr lange Funkstille zwischen uns wegen der Hinsicht. Und letztendlich war es dann t2po die da mediat(unv.) hat am Ende. Aber seine Einstellung hat sich auch geändert zur Arbeit und wie man arbeitet. Und was auch erwartet wird von ihm innerhalb des Teams.”* (t2dev1)

T2dev1 regt mit seiner heftigen Reaktion eine **M2.1 gemeinsame Reflexion** über **M1.2 Verantwortungsbereiche** an. Die Reflexion ist im direkten Versuch nicht erfolgreich, sondern führt zu Kontaktabbruch. Die Aufnahme eines erneuten Gespräches und der Versuch **M2.2 Konsens** zu finden, entsteht durch Mediation durch t2po.

T2dev1 berichtet später im Interview, dass sich t2dev2 Verhalten nach der Teamreflexion verändert hat und gibt erste Hinweise auf eine verbesserte Zusammenarbeit bezüglich **M3. Kommunikation, Wissenstransfer und Proaktivität**:

CNr. 25: *“Er [t2dev2] trifft keine einsamen Cowboy-Entscheidungen mehr. Er versucht auch mal bei Gesprächen mit über Themen, von denen er keine Ahnung hat und die ihn eigentlich auch nicht interessieren, auch mal zuzuhören und auch mal was zu sagen. Er bringt seine Ideen ein, bevor er sie umsetzt.”* (interview t2dev1)

Das M2.1 Ausmaß und die M2.2 Übereinstimmung der Teamreflexion wirken sich wie ein Filter auf die Zusammenarbeit aus. Das bedeutet, je höher das Maß an **KLARHEIT** ist, umso besser ist die Zusammenarbeit im Hinblick auf Kommunikation, Fehlerkultur, Wissenstransfer und Proaktivität. In den Daten wurden verschiedene Beispiele gefunden von Fällen, in denen kein Konsens gefunden wurde. Das hatte verschiedene Gründe, z.B. den, dass eine Person nicht selbstreflektiert war (siehe auch vorheriger Abschnitt und Episode 3), weil keine Teamreflexion stattfand, Teamreflexion über die Zeit vernachlässigt wurde oder trotz Teamreflexion kein Konsens gefunden werden konnte.

Hierzu nun ein paar Beispiele aus den Daten:

Implizite oder explizite Teamreflexion führt im folgenden Beispiel nicht zu **M2.2 Übereinstimmung**:

CNr. 26: *“Das wird nicht immer versucht oder das gelingt nicht immer. Leider ist es dann bei anderen Problemen wieder aufgetreten, dass Sachen nicht angenommen wurden. Beispielsweise sollten Dashboards angepasst werden. [Zwei Teammitglieder] haben gesagt: “Okay das Dashboard, was du gemacht hast, das ist jetzt nicht so aussagekräftig, das sind halt leider keine so guten... versuch doch mal das, das, das!” Hätte sie auch als Lösung verstehen können.*

*Versuch doch mal das und das. Leider war sie da ein bisschen uneinsichtig und hat es nicht so richtig wahrgenommen und hat das nicht angenommen. Das Problem war schon auf persönlicher Ebene, weil sie von den beiden nichts annehmen wollte. Wäre jetzt jemand anders angekommen, hätte gesagt: "Guck mal so und so", dann hätte sie gesagt: "Ja Mensch ja, super danke für den Tipp." Das war das Problem. Das war schon so tief und auf zu persönlicher Ebene." (t5sm)*

In diesem Fall hat die angesprochene Person die Firma kurz darauf verlassen (Siehe **Punkt M4. Teambestand**).

Auch wenn keine **M2.2 Übereinstimmung** gefunden wird kann Teamreflexion die **M3. Zusammenarbeit** verbessern, wie das folgende Beispiel zeigt. t4dev3 strebt eine Führungsposition an, die ihm verwehrt wird, da sie gegen die agilen Prinzipien des Unternehmens wäre. Woraufhin t4dev3 anfängt, die Meetings des Teams zu boykottieren. t4dev2 spricht die Situation explizit in einer Retrospektive offen an:

*CNr. 27: "Also t4dev3 hat damals sehr offen gesagt, dass er das absichtlich macht und was er damit bezwecken will und was seine Taktik ist. Und dann habe ich so plötzlich gemacht: "Wie geht es euch denn damit, wie fühlt ihr euch?" Habe ich tatsächlich auch selber mitgemacht, habe dann auch gesagt wie es mir damit geht und (..) auch die Anderen haben da sehr offen drüber gesprochen, dass es denen halt schlecht damit geht und dass es sich nicht gut anfühlt, dass sie morgens nicht gern zur Arbeit kommen. Und dieses Aussprechen hat dann schon ganz viel gelöst(.). Also das hat es natürlich nicht weg gemacht, dass t4dev3 die Termine konterkariert hat und / und sabotiert hat. Hat aber dazu geführt, dass die Anderen (..) wesentlich ähm (..)halt nicht souveräner. Aber sie konnten es halt einordnen und haben es nicht mehr so persönlich genommen, sondern wussten was da ist und (..) das fand ich gut. Also auch als es klar war, dass t4dev3 geht, war das nicht so (öhh), sondern (..) ganz normaler Umgang " (Inteview t4sm2)*

Hier führt die **M2.1 Teamreflexion und der M2.2 fehlende Konsens** schließlich dazu, dass t4dev3 gebeten wird, die Firma zu verlassen (Siehe **Schritt M4. Teambestand**).

Teamreflexion kann auch in einem Kontext außerhalb des Unternehmens stattfinden.

t5dev1 und t5dev2 reflektierten z.B bei einem gemeinsamen Barbesuch, um die Erwartungen abzugleichen. Während des Interviews erklärt t5dev1, dass es gelingt. Zu einem späteren Zeitpunkt der Beobachtung des Teams wird klar, dass t5dev1 aufgehört hat, proaktiv auf t5dev2 zuzugehen, die ungeklärten Verhältnisse zu Unzufriedenheit geführt haben und t5dev1 in einer Streitsituation kündigt. Da beide Teammitglieder wichtige Schlüsselfiguren im Team waren, fällt das Team daraufhin auseinander.

Teamreflexion wurde in den Daten in ganz unterschiedlichen Situationen gefunden. Dabei ist es das Ziel, Konsens über die Fähigkeiten eines Teammitglieds und die daraus resultierende Erwartungshaltung des Teams an das Teammitglied zu finden. Gelingt es, Konsens zu bilden, verbessert sich die Zusammenarbeit. Gelingt es nicht, verschafft es zwar ggf. Erleichterung in der Situation, führte aber in einer Mehrzahl der beobachtenden Fälle zu einem Bruch im Team bis hin zum Teamzerfall.

### 7.2.4 Rollen

In den Modellschritten 1 und 2 ist ein Rollenverständnis enthalten. Das Verständnis, was eine **ROLLE** ist, wie sie im Team gefunden wird und welche Auswirkungen sie hat, wird im folgenden Abschnitt beschrieben. In meiner Arbeit werden **ROLLEN** neu beleuchtet und erarbeitet. Dabei sind die Definition und Wahrnehmung einer **ROLLE** und deren Bedeutung für das Team entscheidend. Das neue Konzept von **ROLLEN** wird dabei vom allgemeinen oder bisherigen Verständnis einer Rolle farblich differenziert. Der Unterschied zum bisherigen Verständnis von Rollen in agilen Softwareentwicklungsteams folgt in Abschnitt 7.4.3.

Das gegenseitige Verhandeln von Erwartungshaltungen lässt sich auch als das Verhandeln von Rollen definieren, wie das folgende, bereits erwähnte Zitat zeigt.

t4sm2 sagt sehr deutlich, dass es wichtig ist, über seine **M1.2.2. Fähigkeiten** und seinen **M1.2.1 Verantwortungsbereich KLARHEIT** zu gewinnen:

CNr. 28: *“Von denen [ein altes Team] habe ich nämlich gelernt, dass ich eine Klarheit darüber haben muss, was meine Rolle ist, sonst haben das nämlich Andere auch nicht. Unabhängig davon, ob man es ausspricht oder nicht, muss man selber eine Klarheit haben, was man tut.”* (Interview G/t4sm2)

Aus diesem Zitat und anderen Daten wie aus Abschnitt 7.2.3 lässt sich ein Rollenverständnis ableiten (siehe auch [BP19]). **ROLLE** wird definiert als:

1. Teammitglied A erkennt und akzeptiert seine **M1.2.1 Verantwortung** im Team,
2. besitzt die nötige **M1.2.2 Expertise**
3. und handelt danach.

Für das Verständnis dieser Definition ist es wichtig zu verstehen, dass durch Selbstreflexion eine einmalige und *individuelle* **ROLLE** erarbeitet und definiert wird. In einer vorangegangenen Publikation gemeinsam mit Lutz Prechelt wurde der Selbstreflexionsprozess als *Local Role Clarity* benannt [BP19].

Das Handeln im dritten Punkt der Definition ist (im besten Falle) Ergebnis der vorherigen Reflexion über Verantwortung und Fähigkeiten. Durch sein Handeln tritt Teammitglied A in den Austausch mit anderen Teammitgliedern, die daraufhin **M2.1 Feedback** geben können. Daraufhin kann gemeinsamer **M2.2 Konsens** gefunden werden (Globales Rollenverständnis), siehe vorheriger Abschnitt. Wenn ein Konsens über eine individuelle **ROLLE** eines Teammitglieds gefunden wird, entstehen ein globales Rollenverständnis und **ROLLENKLARHEIT** über die individuelle **ROLLE** von Teammitglied A. **ROLLENKLARHEIT** ist definiert als:

- Teammitglied A erkennt und akzeptiert seine Rolle (siehe Definition oben zu **ROLLE**, lokales Rollenverständnis ist hergestellt.)
- Das Team versteht und akzeptiert die **ROLLE** von Teammitglied A. Teammitglied A und das Team haben *dieselbe* Wahrnehmung und ein gemeinsames Verständnis der individuellen **ROLLE** von Teammitglied A.

Wenn das Team ein Verständnis für jede **ROLLE** aller Teammitglieder gefunden hat, also darüber reflektiert und ein Konsens gefunden wurde, ist für *jedes* Teammitglied

globale **ROLLENKLARHEIT** hergestellt. Das bedeutet, dass **KLARHEIT** im Team weitestgehend erreicht ist.

Eine **ROLLE** ist nur tragfähig und kann von Teammitgliedern wahrgenommen werden, wenn Konsens und damit **ROLLENKLARHEIT** über die **ROLLE** im Team entwickelt wurde. Das lokale, durch Selbstreflexion entwickelte Rollenverständnis kann von dem globalen Rollenverständnis abweichen, wie bei den Beispielen CNr. 14 und CNr. 27. In diesen Beispielen weicht das *lokale* Rollenverständnis von dem *globalen* Verständnis im Team ab und es kommt zu Konflikten. Es wird keine **ROLLENKLARHEIT** für das jeweilige Teammitglied erreicht. Damit ist **KLARHEIT** im Team nur unvollständig erreicht und es kommt zum Bruch im Team (**M3 Zusammenarbeit**) und dem Ausscheiden dieser beiden Teammitglieder (**M4 Konsequenzen**).

Der Prozess, ein gemeinsames Rollenverständnis zu erarbeiten und Konsens zu finden, wird auf unterschiedliche Weise in den Teams sichtbar.

In t3 wird im Beispiel CNr. 23 in der Retrospektive sehr explizit über die Erwartungshaltung an ein Teammitglied reflektiert und wie dieses Teammitglied unterstützt werden kann, seine **ROLLE** zu finden und zu erfüllen. t3sm hat in Gesprächen gemeinsam mit t3dev3 das individuelle lokale Rollenverständnis und damit die **ROLLE** von t3dev3 reflektiert und trägt das Ergebnis nun zur globalen Reflexion in das Team. Das Team ergänzt das Rollenverständnis von t3dev3 um die eigene Wahrnehmung, die auf der Wahrnehmung der Handlungen von und Kommunikation mit t3dev3 beruht (und damit per Rollendefinition auf dem Verständnis der individuellen **ROLLE** von t3dev3, siehe oben Punkt 3 *Handeln*). Es wird eine mögliche **ROLLE** und damit ein mögliches globales Rollenverständnis für t3dev3 diskutiert und erarbeitet. Es wird festgehalten, wie t3dev3 in seiner **ROLLE** unterstützt werden kann. t3sm will diese Absprachen anschließend mit t3dev3 besprechen, wodurch **ROLLENKLARHEIT** für t3dev3 hergestellt würde.

In t2 kommuniziert t2dev1 seine Erwartungshaltung an t2dev2 über einen Konflikt. Mit Hilfe anderer Teammitglieder wird der Konflikt gelöst, das Verständnis der **ROLLE** angepasst und mehr **ROLLENKLARHEIT** für die **ROLLE** von t2dev2 entwickelt. (Ob **ROLLENKLARHEIT** für die **ROLLE** von t2dev2 im gesamten Team herrscht und damit vollständige **ROLLENKLARHEIT** bezüglich t2dev2, kann aufgrund der Daten nicht erfasst werden.)

**ROLLEN** sind individuell für jedes Teammitglied und gelten im Zusammenhang der selbstorganisierten Zusammenarbeit in dem jeweiligen Team. Sie sind durchlässig und wandelbar und können im Arbeitsprozess stetig neu verhandelt werden, wie auch **KLARHEIT** neu gefunden werden kann.

### 7.2.5 Konsequenz:

#### Zusammenarbeit

Im vorherigen Abschnitt 7.2.3 wurde bereits u.a. der Zusammenhang zwischen Teamreflexion und ihre positiven Auswirkungen auf die Zusammenarbeit sichtbar (Beispiele Coding Nummer 23 oder 25). Ist grundsätzlich Übereinstimmung durch die Teamreflexion gefunden (oder zumindest Transparenz über die gegenseitige Erwartungshaltung) wirkt sich das positiv aus im Hinblick auf M3. Kommunikation, Wissenstransfer und Proaktivität (siehe Abbildung 7.4: 3. Schritt Zusammenarbeit).

In den Daten lässt sich erkennen, dass ein hohes Ausmaß an KLARHEIT zur positiven Wahrnehmung der Zusammenarbeit durch die Teammitglieder (in Interviews) führt und in Arbeitsbeobachtungen ebenfalls sichtbar wird.

Es gab eine Zeitspanne im Beobachtungszeitraum als t5 hochfunktional war und es ein hohes Maß an KLARHEIT gab, wie unter anderem t5dev1 im Interview berichtete, in Gesprächen mit den Scrum-Mastern des Teams und in Beobachtungen deutlich wurde. Zu diesem Zeitpunkt gab es eine Retrospektive, in der sich die Teammitglieder gegenseitig lobten. Im Folgenden gibt es einen Ausschnitt aus der Mitschrift. Da es zu diesem Zeitpunkt keine Probleme gab, wurde eine Übung "shower of appreciation" durchgeführt. Jedes Teammitglied wird dabei reihum von allen anderen Teammitgliedern gelobt. Im folgenden Ausschnitt schütten alle Teammitglieder ihr Lob für das neueste Teammitglied t5dev4 aus. Ihr Lob wurde stichwortartig mitgeschrieben.

CNr. 29: t5dev1 über t5dev4: Speed of learning new stuff. Curiosity to discover "Spring". Autonomy, asking for help.

t5dev2 über t5dev4: Focus on yourself, getting whole team forward! Good Quality of work. You have been helpful whenever I asked.

t5dev3 über t5dev4: Pair Programming is helpful and supports learning. Preparations. Take task seriously. You are the newest, but you are active getting to know the software. Initiative to bring "Spring" to the team. Discuss, challenging my mind.

t5dev4: "I like this Retro. It makes me smile."

In diesem kurzen Beispiel werden fast alle Ebenen des bisher vorgestellten Modells angesprochen. Die Teammitglieder erklären ihren M2.2 Konsens über die Art wie t5dev4 seine M1.2.1 Aufgaben wahrnimmt und seine M1.2.2 Fähigkeiten einbringt. Sie loben die Autonomie (die als Ausgangspunkt in das Team hineingetragen wird und nun angemessen von t5dev4 angenommen wird) und die Wahrnehmung von M1.2.1 Verantwortung ("taking task seriously"). Auch seine M1.2.2 Fähigkeiten bezüglich Geschwindigkeit ("Speed of learning new stuff"), Qualität ("Good Quality of work") und Eigenständigkeit ("Autonomy") werden hervorgehoben. Auf der Ebene der M3. Zusammenarbeit wird die M3.1 Proaktivität ("Curiosity", "asking for help", "Initiative to bring "Spring" to the team") von t5dev4 gelobt, die M3.3 offene Kommunikation, indem er Fragen stellt und damit M3.2 Wissenstransfer ermöglicht. Und letztlich die äußerst relevante Aussage, dass er sich auf sich selbst konzentrieren soll, um das ganze Team voranzubringen („Focus on yourself, getting the whole team forward!“). Hier wird also implizit Selbstreflexion, die eigene Position im Team zu finden, als wichtiger Schritt erkannt, um das Team insgesamt voranzubringen.

In einem anderen Beispiel berichtet t4sm2 über ein Team, das nach einer Storming-

Phase oder auch einer klaren Auseinandersetzung zur Erreichung von **KLARHEIT** zusammengefunden hat und fasst alle positiven Merkmale einer guten Zusammenarbeit in ihrem Zitat zusammen:

CNr. 30: *“Die hatten so ganz klar (/) Die wussten so gegenseitig wie sie sich aufeinander verlassen konnten. Die hatten so ganz (/)ich hätte jetzt beinah gesagt klar abgegrenzte Aufgabenfelder. Das ist natürlich nicht. Also die haben wirklich als Team gearbeitet, aber es war klar, wer so was kann. Der eine ist mehr so, wer was runterrockt, der andere ist mehr so der irgendwie voraus denkt und der andere kann mehr so das Diplomatische und irgendwie mit dem PO verhandeln und der Dritte ist mehr so: “Ja, ich renne mal los und hol den mal her”. Dann mal los und rede mal mit dem und hol den mal her. Das war aber so/ so unausgesprochen klar in diesem Team, wer was kann und wer welche Kompetenz übernimmt und die mussten das nicht groß absprechen, sondern das war / hat sich irgendwie so / so dynamisch ergeben und die hatten auch ein krasses Storming. Also ich habe die von Anfang an von dem/ ich habe sie quasi mit gegründet das Team und (/) die hatten krasses Storm, als die haben sich richtig in der Wolle gehabt. Das waren halt auch/ das waren, ach ist schon paar Jahre her, fünf oder sechs und das waren halt auch sechs so fünf Charaktere, sage ich mal. Ja das waren so sechs Alphamännchen in einem Team. Es war schon so hm (/) und die haben echt Sachen so runtergerockt und das war unglaublich, also was da raus kam und die hatten trotzdem immer Spaß dabei und es war professionell, aber nicht / aber trotzdem hat man auch über persönliches geredet. (/) [...] Man hat sich unterstützt ohne (/) ohne viele Worte, so das war irgendwie/ da musste man auch nicht hochrechnen. Ich hab das jetzt für dich gemacht, da musst du aber für mich und das war irgendwie echt ganz gut.” (t4sm2)*

Es wird deutlich, dass Teamreflexion sich in jedem Falle, auch wenn kein Konsens gefunden wurde, positiv auf die Zusammenarbeit im Team auswirkt. Sowohl positive als auch negative Beispiele wurden in den Daten gefunden. Inwiefern sich **KLARHEIT** immer auf alle Ebenen der Zusammenarbeit (Wissenstransfer, Kommunikation oder Proaktivität) auswirkt und in welchem Ausmaß, kann in den Daten nicht kausal oder quantitativ belegt werden.

## 7.2.6 Konsequenz:

### Teambestand

Die Güte der Zusammenarbeit sichert den Zusammenhalt und den Bestand des Teams und kann in seiner Konsequenz auch zum Zerfall des Teams führen (siehe Abbildung 7.4: Konsequenz, rechts).

In allen Teams (außer t1, das nur einmalig beobachtet wurde) gab es Teammitglieder, die die Firma freiwillig oder unfreiwillig verlassen haben. Neben individuellen Gründen hat fehlende **KLARHEIT** immer eine ausschlaggebende Rolle gespielt. Ein ausgeprägtes Beispiel war t5: Das Team war ein hochgelobtes, hochfunktionales Team. Es kam zu Unklarheiten zwischen t5dev1 und t5dev2, die irgendwann nicht mehr in Teamreflexionen geklärt wurden. T5dev1 reicht seine Kündigung ein. Ein halbes Jahr später hat sich das Team nicht von diesem Konflikt erholt, sondern drei weitere Teammitglieder haben das Team verlassen und bestätigten in Interviews, dass die Teamsituation ausschlaggebend war neben persönlichen Anreizen.

Auch Managemententscheidungen beruhen in den Daten auf fehlender **KLARHEIT**.

Der nicht selbstreflektierende Tester  $t_{2test1}$  aus der Episode 2 wird entlassen. Auch in  $t_5$  wird entschieden, dass das Teammitglied, das zwar niedrige Fähigkeiten hat, aber selbstreflektiert ist, bleiben darf (siehe Zitat 20). Ein anderes Teammitglied, das trotz mehrmaliger Anregung nicht über seine Fähigkeiten angemessen reflektiert, wird entlassen (siehe Zitat CNr. 14). Auch in dem Beispiel in dem das Fehlverhalten eines Teammitgliedes, der gerne eine Führungsposition inne hätte thematisiert wird, führt zu Entlassung des Teammitgliedes.

$T_{4sm2}$  hat ein sehr ungewöhnliches Vorgehen gewählt und damit das Prinzip des Modells umgekehrt.  $T_{4sm2}$  wählte den rückwärtigen Weg und lässt das Team in einer **M2.1 Teamreflexion** über seine eigene **M4. Zusammensetzung** entscheiden.

CNr. 31: *“Und die haben mich auch alle akzeptiert. Also das ist auch so etwas / ich bin nicht einfach reingegangen und habe gesagt: “Ich bin jetzt hier der neue Scrum-Master!”, weil die hatten vorher halt  $t_{4sm1}$  als Scrum-Master. Und ja da habe ich gesagt: “Überlegt euch das! Und wenn ihr euch entschieden habt, dann sagt ihr mir Bescheid.” Und war dann eben einstimmig, dass ich mitkommen kann [...] Ich glaube sonst kann man nicht wirken. Also wenn du da und hier 3 Leute hast, die dich nicht mögen oder mich nicht wollen, dann bringt es halt nichts.” ( $t_{4sm2}$ )*

Hier stellt die SM ihre Rolle grundsätzlich zur Diskussion: Braucht ihr mich und meine Fähigkeiten? Ich bin gerne dabei, wenn ihr sie braucht.

In der kleinen Menge von fünf Teams wurde eine erstaunlich hohe Zahl an Fällen beobachtet, in denen es durch fehlende **KLARHEIT** zum Bruch mit einzelnen Teammitgliedern kam bzw. in einem Fall ein ganzes Team zerbrach. Das lässt den Schluss zu, dass fehlende **KLARHEIT** eine Bedrohung für die Zusammenarbeit und den Teambestand ist. Allerdings kann im Rahmen der GT kein kausaler und immer gültiger Zusammenhang nachgewiesen werden.

### 7.2.7 Monitoring und Feedback

Monitoring und Feedback durch das Team oder den Scrum-Master kann zu jedem Zeitpunkt auftreten wie auch oben genannt beispielsweise während der Teamreflexion im Planning.

In den bisher genannten Beispielen sind es häufig die Scrum-Master, die Selbst- oder Teamreflexion anregen oder es zumindest versuchen. Andere Teammitglieder geben auch ihr Feedback, wenn auch manchmal in Form von Konflikten (siehe beispielsweise  $t_{2dev1}$ ). Wiederum andere Teammitglieder wie die PO aus  $t_2$  vermitteln und führen **KLARHEIT** herbei.

In den Daten wurde gegroundet, dass Monitoring und Feedback zu unterschiedlichen Zeitpunkten und von unterschiedlichen Personen stattfinden. Feedback und Monitoring unterstützen das Team darin, **KLARHEIT** zu gewinnen. Wie hoch die Effekte sind und wann es am effektivsten stattfinden sollte, wurde nicht abschließend validiert.

### 7.3 Ergebnis: Hypothesen

In Abschnitt 7.2 ist der Prozess zur Gewinnung von Klarheit - der methodisch aus der Praxis hergeleitet wurde - vereinfacht, linear und abstrakt in einem Modell dargestellt. Anhand von konkreten Codings wird deutlich, dass der Prozess in der Praxis weniger linear verläuft und vielen verschiedenen, komplexen Einflüssen ausgesetzt ist. Im folgenden Abschnitt werden die wichtigsten Erkenntnisse in Hypothesen zusammengefasst und erläutert. Während üblicherweise in der Forschung Hypothesen aufgestellt und dann geprüft werden, ergeben sich Hypothesen oder Theorien in der GTM durch den Prozess. Hypothesen sind somit vielmehr Schlussfolgerungen aus den Daten. Es sind induktive Hypothesen, die somit bereits belegt und teilweise bestätigt sind.

**Hypothese 1:** Selbstreflexion und proaktives Handeln ist notwendig, um **KLARHEIT** zu gewinnen und herausfordernd für die Teammitglieder in agilen Softwareentwicklungsteams.

Selbstorganisation im agilen Umfeld erfordert proaktives eigenständiges Handeln (Vergleich: CNr. 15) und Selbstreflexion. Teammitglieder fordern proaktives Handeln von einander ein (Vergleich: Episode 3, CNr. 24). Selbstreflexion ist dafür ein wichtiger erster Schritt. Das beinhaltet eigenständig seine Verantwortungsbereiche abzustecken und sich im Team einzubringen (Vergleich CNr.: 6).

Für einige Teammitglieder ist Selbstreflexion eine große Herausforderung (Vergleich: Episode 3, CNr. 12, 13). Es gibt Teammitglieder, die es trotz Anregung von außen es nicht schaffen, über sich selbst und ihre Verantwortungsbereiche zu reflektieren (Vergleich: CNr. 14). Anderen hilft der regelmäßige Austausch im 4-Augen-Gespräch mit dem Scrum-Master (Vergleich: Episode 2), das implizite und explizite Feedback in verschiedenen Scrum-Events (Vergleich: CNr. 24) oder individuelle Bewältigungsstrategien wie Meditation (Vergleich: CNr. 13) oder eine Auszeit (Beispiel: t2dev1 hat nach der agilen Einführung eine Auszeit von mehreren Monaten beantragt).

Die Erwartungshaltungen wie viel Feedback benötigt wird und wie viel eigenständige Selbstreflexion notwendig ist, sind unterschiedlich (Vergleich: Episode 3).

**Hypothese 2:** **KLARHEIT** auf eigener *und* auf Teamebene fördert eine positive Zusammenarbeit.

Selbstreflexion allein reicht nicht aus. Es können Diskrepanzen zwischen Selbst- und Fremdbild bestehen (Vergleich: CNr. 6, 14, 27), die durch Teamreflexion aufgedeckt werden müssen. Das kann implizit oder explizit zum Beispiel während der Retrospektive oder des Plannings stattfinden. In der Teamreflexion können Erwartungshaltungen bezüglich Verantwortungsbereich und Fähigkeiten der Teammitglieder untereinander diskutiert werden. Es entsteht ein gemeinsames Rollenverständnis. Auch wenn kein Konsens gefunden wird, kann die offene Diskussion sich positiv auswirken.

Gute Zusammenarbeit ist lediglich und vor allem ein Ergebnis der guten Selbst- und Teamreflexion wie viele Beispiele gezeigt haben (Vergleich CNr. 25, 23, 29, 30, 31).

**Hypothese 3:** Gegenseitiger Konsens über die Fähigkeiten der Teammitglieder ist wichtiger für den Fortbestand des Teams, als die Fähigkeiten selbst.



Selbstreflexion und ein gemeinsamer Konsens im Team darüber welche Fähigkeiten ein Teammitglied hat, ist wichtiger, als die Höhe und die Art der Fähigkeiten (Vergleich CNr. 17, 18, 19, 20, 23). Die Fähigkeiten, die in einem Team für nützlich und wichtig erachtet werden, sind sehr unterschiedlich. Dazu gehören soziale Fähigkeiten oder auch beispielsweise Humor. Fachliche und technische Fähigkeiten sind nicht primär ausschlaggebend.

Während der Beobachtungen und aus Episoden aus Erzählungen wird deutlich, dass bei fehlender **KLARHEIT** die Tendenz besteht, dass Teammitglieder das Team verlassen. In keinem der Fälle, die es in allen untersuchten Teams gab, lag es (ausschließlich) an fehlender fachlicher Expertise. Es gab immer auch einen Bezug zu fehlender **KLARHEIT** und der daraus entstandenen Konflikte oder Unzufriedenheit. In einem Fall zerbrach das ganze Team (t5).

**Hypothese 4:** Scrum-Master und andere Teammitglieder können unterstützen **KLARHEIT ZU GEWINNEN**.

Scrum-Master und andere Teammitglieder können auf beiden Ebenen (Eigen- und Fremdwahrnehmung) unterstützen (Vergleich 24, 23, 27), indem sie ausreichend Feedback geben. Dabei ist es erfolgreicher, wenn Reflexion auf beiden Ebenen gefördert wird.

**Hypothese 5:** Trotz aller Unterstützung kann der Prozess zu jedem Zeitpunkt scheitern insbesondere an persönlichen Sympathien und Emotionen.

Der Prozess ist anfällig für Emotionen, die aus Konflikten oder gegenseitigen Abneigungen herrühren. Wenn Personen kein Feedback mehr annehmen können oder wollen oder aufhören, sich aktiv in den Prozess einzubringen, kann er leicht zu jedem Zeitpunkt scheitern (Vergleich: CNr. 24, 26, 27).

## 7.4 Diskussion der Ergebnisse und des Forschungsbeitrages **Klarheit gewinnen im Kontext der verwandten Arbeiten**

Im folgenden Teil werden vorwiegend inhaltliche Gemeinsamkeiten und Unterschiede verwandter Arbeiten, die ausführlich in Kapitel 2 dargestellt wurden, und den vorgestellten Ergebnissen aus Abschnitt 7.2 diskutiert. Es wird erläutert, inwiefern die Ergebnisse sich von verwandten Arbeiten abgrenzen lassen, diese ergänzen und welchen Wert die Ergebnisse im Vergleich ergeben. Um einzelne Argumentationsketten zu untermauern, werden gelegentlich auch weitere, bisher nicht oder nur kurz eingeführte Forschungsarbeiten diskutiert.

Im folgenden Abschnitt werden die verwandten Arbeiten gemeinsam diskutiert. Textabschnitte, in denen ich darüber hinaus meine Ergebnisse im Kontext verwandter Arbeiten diskutiere sind grau hervorgehoben, um die Orientierung für die Lesenden zu erleichtern. Bei den kurzen Diskussionen der Hypothesen in Abschnitt 7.3 wird auf diese Hervorhebung verzichtet.

Es wird gezeigt, dass sich die Ergebnisse auf ein Geflecht aus verwandten Arbeiten unterschiedlicher Strömungen stützen können. Unter Berücksichtigung der Spezifitäten von selbstorganisierten Softwareentwicklungsteams werden sie zu einem bisher einzigartigen Modell verdichtet und erweitert. Das ist der Mehrwert dieser Arbeit.

Zuerst wird das erarbeitete Modell schrittweise diskutiert. Im Anschluss werden die induktiven Hypothesen aus Abschnitt 7.3 diskutiert.

#### 7.4.1 Ausgangspunkt: Selbstorganisierte Teams

Hoda hat in ihrer Arbeit erkannt, dass agile Softwareentwicklungsteams durch ihre Selbstorganisation geprägt und definiert werden [Hod11]. In den verschiedenen verwandten Arbeiten wie Hackman [Hac86], Hoda [Hod11], Magpili und Pazos [MP18] oder Moe u.a. [MDD08] werden selbstorganisierte Teams diskutiert, manchmal definiert und manchmal eine Definition stillschweigend angenommen oder vorausgesetzt.

Gemeinsam haben die Definitionen Folgendes: Teamdynamiken in selbstorganisierten Teams verändern sich durch den Zuwachs an Autonomie. Wie sich diese Dynamiken verändern wird jeweils im Hinblick auf Schwerpunkte und Blickwinkel unterschiedlich betrachtet und bewertet.

Hackman nimmt eine hilfreiche (Teil-)Definition selbstorganisierter Teams (performing units) vor, die auch für das Verständnis anderer Arbeiten interessant ist und bereits erwähnt wurde (siehe Abschnitt 2.3.2.2):

I will speak of performing units as if they consisted of several individuals working interdependently on a common task [Hac86, S.91]

Selbstorganisierte Teams bestehen damit aus Individuen, die in *Abhängigkeit* voneinander an einer *gemeinsamen* Aufgabe arbeiten. Mit dieser Definition werden die Grenzen und Freiheiten der Autonomie in selbstorganisierten Teams abgesteckt, die sich in den weiteren aktuelleren Arbeiten aus dem Software Engineering wiederfinden:

Hodas *Balancing Acts* [HNM10a] lassen sich hier einsortieren. Sie erkannte, dass Teammitglieder eine Balance zwischen *Freiheit und Verantwortung* finden müssen. Sie sind für sich selbst in der Aufgabenabarbeitung und parallel für die Erreichung der Teamziele verantwortlich.

Moe u.a. [MDD08] sehen darin eine große Hürde für Softwareentwicklungsteams. In Softwareentwicklungsteams sind die Individuen zumeist hoch spezialisiert. An einer gemeinsamen hochkomplexen Aufgabe zu arbeiten, mit individuellen Spezialisierungen führt zu Problemen:

[...] the problem with "highly specialized skills and corresponding division of work" was thoroughly discussed, and the team decided to focus on this challenge.[MDD08, S.83]

Moes u.a. Erkenntnis spiegelt sich in Hodas *Balancing Act* zwischen Crossfunktionalität und Spezialisierung wider.

In Moes u.a. [MDD09] Arbeiten ist zu erkennen, dass die Neuerung für die Teams durch Selbstorganisation darin besteht, Entscheidungen zu dezentralisieren, beziehungsweise mit dem Team zu teilen, obwohl sich die Teammitglieder vorwiegend für die eigenen Aufgaben verantwortlich fühlen. Es entsteht für das Team mehr gemeinsame Entscheidungsautonomie, während die Verantwortung oder Autonomie für die Abarbeitung der Aufgaben bei einzelnen Teammitgliedern individuell liegt.

Because developers usually worked alone on a module, they often created individual plans. Developers got full control over their own schedule and task implementation, which resulted in even more individual autonomy. [MDD09, S. 22]

In Hackmans Definition ist also schon der Autonomiekonflikt zwischen individueller und Team-Autonomie inhärent. Moe u.a. [MDD09] und Hoda [HNM10a] erkennen, dass dieses Phänomen in agilen Softwareentwicklungsteams mit hoher Spezialisierung und komplexen Aufgaben besonders hervortritt. Teammitglieder sind gefordert, ihre persönliche Entwicklung bzw. Spezialisierung gegenüber der Erreichung von gemeinsamen Teamzielen, also einer gemeinsamen Aufgabe, abzuwägen.

Meine Arbeit teilt Hackmans Definition und erkennt auch, dass die gemeinsame Verantwortung bei geteilter Aufgabenstellung für spezialisierte Softwareentwicklungsteams eine Herausforderung ist. Neu ist die Einordnung und die Perspektive in meiner Arbeit: Der Zugewinn an Autonomie durch Selbstorganisation und der damit einhergehende Konflikt zwischen Selbst- und Team-Autonomie wird nicht als Hürde, sondern neutral als Ausgangspunkt für agile Softwareentwicklungsteams betrachtet. Der Autonomiekonflikt ist eine Voraussetzung für den weiteren Prozess, bzw. macht den im Modell dargestellten Prozess erst notwendig. Die Ursache und welcher Prozess dadurch angestoßen wird, wurde erkannt und weiter ausformuliert. Selbstorganisation bildet keine unbedingte Hürde, sondern führt zu ausgeprägter und spezieller Teamdynamik, die im Modell betrachtet werden.

### 7.4.2 Klarheit gewinnen

Das Konzept **KLARHEIT GEWINNEN** ist in zwei Schritte eingeteilt: in die Selbst- und Teamreflexion. Die Gegenüberstellung von *Selbst* und *Team* findet sich in vielen Arbeiten wieder (siehe vorherigen Abschnitt) und auch der darin potentiell bestehende Konflikt wird beschrieben.

Im *Behavioral Software Engineering* (siehe Abschnitt 2.1 und [LFW15]) ist diese Unterscheidung Teil der Systematik zur Einordnung der Forschungsarbeiten: Es wird zwischen Individuen, Gruppen und Organisation unterschieden.

Bezüglich der Motivation werden Unterschiede und Spannungsverhältnisse zwischen Eigen- und Teammotivation erkannt [dF12] (siehe auch Abschnitt 2.3.3). Laut Langfred [Lan00] stehen Selbst- und Teamautonomie in selbstorganisierten Teams im Widerspruch zueinander. Moe u.a. [MDD09] erkennen diesen Autonomiekonflikt in selbstorganisierten Softwareentwicklungsteams (siehe auch vorherigen Abschnitt). Hodas „*Balancing acts*“ [HNM10a] sind Teil dieses Autonomiekonfliktes. Jeder einzelne Balanceakt spiegelt den Konflikt zwischen Selbst- und Teamzielen wider.

Das Modell dieser Arbeit besteht aus zwei Schritten, der Selbst- und Teamreflexion, und greift so den Konflikt zwischen diesen beiden Perspektiven auf. Das Modell erkennt aber nicht nur den Konflikt oder die Spannungen zwischen Eigen- und Teamperspektive, sondern bietet zusätzlich eine mögliche Lösung an. Durch den Prozess **KLARHEIT ZU GEWINNEN** wird im besten Falle, dieser Konflikt gelöst oder zumindest transparent gemacht. Die zwei Schritte zur Gewinnung von Klarheit berücksichtigen beide Perspektiven. Ziel ist es beide Perspektiven zueinander zu bringen und möglicherweise einen Konsens zu finden.

Der Prozess wird im Modell außerdem durch den Ausgangspunkt und die möglichen positiven Folgen (gute Zusammenarbeit) und negative Folgen (Auseinanderbrechen des Teams) ergänzt.

Die Aufteilung zwischen Eigen- und Teamperspektive wird auf einer höheren Metaebene vorgenommen und die einzelnen Punkte zur Reflexion sind als offene Liste zu verstehen. Während der Selbstreflexion kann die eigene Motivation beleuchtet werden genauso wie die eigene Autonomie oder Fähigkeiten, die dann im zweiten Schritt mit der Teamperspektive abgeglichen werden. Dadurch werden, wie bereits erwähnt, Spannungen und Konflikte auf diesen Ebene abgebaut. Für diese von den verwandten Arbeiten als zentrale Hürde dargestellte Problematik, Team vs. Selbst auf Autonomie oder Motivationsebene, wird eine mögliche Lösung durch **GEWINNUNG VON KLARHEIT** angeboten.

### 7.4.3 Rollen

Diese zweischrittige Reflexion beinhaltet auch die Reflexion und Erarbeitung von Rollen im Team.

Das Rollenverständnis, das in dieser Arbeit entwickelt wurde, weicht vom Rollenverständnis, wie es beispielsweise im Scrum-Guide [SS17b] oder in anderen Modellen im Software Engineering [LL13] vertreten wird, ab. Es werden keine präskriptiven, also vordefinierten und zu erfüllenden Rollen an das Team herangetragen: Im V-Modell XT gibt es beispielsweise mehr als 30 präskriptive Rollen. Dabei kann eine Person mehrere Rollen wahrnehmen; es können auch mehrere Personen eine Rolle erfüllen [LL13]. In meiner Arbeit ist das Vorgehen zur Definition von Rollen abweichend: Teams entwickeln ein gemeinsames Rollenverständnis im Prozess zur Gewinnung von **KLARHEIT**. Im Ergebnis, falls Konsens gefunden wird, kann jedes Teammitglied eine individuelle **ROLLE** wahrnehmen. Diese **ROLLEN** sind deskriptiv, nicht präskriptiv, denn sie sind flüchtig und können sich jederzeit ändern.

In einer früheren Arbeit [BP19] definieren wir Rolle durch drei Kriterien: Verantwortlichkeit klären, Fähigkeiten besitzen und danach zu handeln. Dieses Rollenverständnis ist in das Modell eingeflossen und ausdifferenziert worden. Die Verantwortlichkeit und Fähigkeiten sind in der Selbstreflexion zu finden. Sich auf ein gemeinsames Handeln zu einigen findet in der Teamreflexion statt. Hackman [Hac86] definiert drei Prozesskriterien (Engagement, Wissen und Fähigkeiten und Arbeitsstrategien, siehe S. 45), die den drei definierten Rollenkriterien ähneln. Das Rollenverständnis oder auch die Prozesskriterien beeinflussen die spätere Zusammenarbeit. Beide Arbeiten haben unabhängig voneinander erkannt, dass die drei Punkte: proaktiv Verantwortung klären, Fähigkeiten besitzen und danach handeln zu können, einen wesentlichen Einfluss auf eine erfolgreiche Zusammenarbeit ausüben. Die vorliegende Arbeit bettet diese Erkenntnisse in ein Modell ein und verknüpft es mit bisherigen Erkenntnissen zu selbstorganisierten Softwareentwicklungsteams.

Magpili und Pazos [MP18] identifizieren in ihrem Literaturreview Faktoren, die die Leistung selbstorganisierter Teams beeinflussen, unter anderem individuelle Autonomie, Rolle und Fähigkeiten. Alle drei Aspekte werden im Modell aufgegriffen. Der Zuwachs an Autonomie ist der Ausgangspunkt. Fähigkeiten sind Teil der Selbstreflexion und der Rolle.

Ausgehend vom Begriff des Rollenverständnisses lassen sich noch weitere verwandte Arbeiten heranziehen. In diesen verwandten Arbeiten wird von *Rolle* gesprochen, das Verständnis unterscheidet sich jedoch häufig.

Das Rollenverständnis in der Psychologie und Anthropologie wird in den oben genannten Beispielen hierarchisch gedacht [JBP03] [Sla55]. Teammitglieder werden beispielsweise aufgefordert zu Forschungszwecken, alle Teammitglieder hierarchisch anzuordnen von der besten bis zur schlechtesten Führungsperson. Ein Verständnis, das weder den agilen Grundgedanken [BBv<sup>+</sup>01] noch den in dieser Arbeit verwendete Begriff von Rollen widerspiegelt und daher nur bedingt berücksichtigt wird. Die Arbeiten aus der Psychologie und Anthropologie unterstützen jedoch die Aussage Hackmans und der vorliegenden Arbeit, dass die Höhe des Konsens über die Teamrollen Einfluss auf die Teameffektivität hat [JBP03] [Sla55].

In Johnson u.a. [JBP03] wird darauf hingewiesen, dass informelle und emergente

Rollen die formelle Gruppenstruktur ablösen. Johnson u.a. ergänzen auch, dass es schwierig ist alle unterschiedlichen Rollen abgesehen von den klassischen Beispielen wie "leader" und "follower" abzubilden. Grundsätzlich ist es ein komplexes Zusammenspiel verschiedener Rollen, die die Zusammenarbeit bestimmen und wenig vorhersehbar machen [JBP03].

Das Rollenverständnis beeinflusst also die Zusammenarbeit und Rollen lassen sich nicht endlich abbilden. Das greift die vorliegende Arbeit auf: Das Modell zeigt, dass das Rollenverständnis bzw. Selbst- und Teamreflexion direkt die Zusammenarbeit beeinflusst. Das offene Rollenverständnis im Modell beinhaltet unendliche Möglichkeiten, fließende Rollen zu definieren und gibt gleichzeitig Sicherheit, indem es eine Definition von Rollen auf Metaebene zur Verfügung stellt. Das Ziel sind nicht vorgefertigte oder formelle Rollen, sondern individuelle Rollen und deren Verantwortlichkeiten in selbstorganisierten Softwareentwicklungsteams. Durch die Reflexion werden diese Rollen erfahr- und diskutierbar. Hoda [Hod11] [HNM13] stellt in ihrer Arbeit Rollen in Softwareentwicklungsteams vor, erklärt deren Handeln und wie sie im Team zusammenwirken. In ihrem Rollenverständnis sind diese Rollen durchlässig, spontan und situationsabhängig. Dieses Verständnis teilt meine Arbeit. Allerdings gehe ich noch weiter und benenne keine expliziten Rollen wie Mentor\_in mehr, sondern treffe die Aussage, dass es eine unendliche Vielfalt an möglichen Rollen gibt und dass nicht die Ausdifferenzierung der Rolle die Zusammenarbeit prägt, sondern das Ausmaß an **KLARHEIT**, das darüber gewonnen wird. Eine Rolle kann nur ausgeführt werden insoweit wie das Team darüber ein Einverständnis gewonnen hat. Ist das Einverständnis zur Ausübung der Rolle nur einseitig, die Rolle nicht klar verstanden worden oder entspricht nicht den Erwartungen, hat das negative Folgen für das Team.

Ein gutes Beispiel, wie die vorliegende Arbeit an andere Arbeiten anschließt und weiterführt, zeigt sich, in dem sich das Modell auf Originalzitate anderer Arbeiten anwenden lässt. Ein\_e Entwickler\_in erklärt bei Moe u.a.:

*"They usually tell me what to do. I think I possess more knowledge than they realize and that I get too simple tasks. I'm also missing a good overview of what we are going to deliver, because many of the discussions regarding the design were done without me. It's probably too expensive if everyone to participate in every meeting, but for me it is difficult to see dependencies between the modules."* [MDD09, S. 24]

In diesem Beispiel fehlt Klarheit. Der oder die Entwickler\_in hat nicht dieselbe Vorstellung über die eigenen Fähigkeiten wie das Team. Selbstreflexion scheint stattgefunden zu haben. Teamreflexion oder Konsens fehlt.

#### 7.4.3.1 Selbstreflexion

In mehreren verwandten Arbeiten wird die Bedeutung der Fähigkeiten der einzelnen Teammitglieder und die Selbstreflexion über ihre Fähigkeiten als Einflussfaktor auf die Leistung von Teams identifiziert. Hackman [Hac86] benennt als erstes Anzeichen selbstorganisierter Teams, dass Personen Verantwortung übernehmen für ihr

Handeln. Außerdem braucht es entsprechende Fähigkeiten und Wissen. Die Aufgabenstellung und die Fähigkeiten sollten im Einklang zueinander stehen, um dauerhafte Überforderung und Unzufriedenheit zu verhindern. Magpili und Pazos [MP18] erkennen ebenfalls in ihrem Review, dass es wichtig ist, Selbstreflexion, -kritik und -verbesserung anzuregen, um die Autonomie des Teams weiterzuentwickeln.

Das Modell greift auf, dass es zuerst (im besten Falle intrinsischer) Selbstreflexion bedarf, um zu erkennen, ob Fähigkeiten und Aufgaben zueinander passen. Sie ist der erste Schritt, um die Zusammenarbeit in agilen Teams positiv beeinflussen zu können.

Dass bei der Selbstreflexion nicht nur technische, sondern auch soziale Fähigkeiten relevant sind, wird ebenfalls in mehreren Arbeiten erkannt [Hac86] [JBP03]. In der vorliegenden Arbeit wurden in den Daten Reflexionen zu verschiedenen Eigenschaften und Fähigkeiten wie Geschwindigkeit und Wissen gefunden. Interessant ist, dass in einem Zitat der Aspekt des Humors auftaucht, der vom Team geschätzt wird. Während in Softwareentwicklungsteams diese Bedeutung bisher nicht oder nur wenig erforscht ist<sup>7</sup>, sagt Johnson u.a. [JBP03], dass Humor in extrem isolierten Teams eine der wichtigsten Fähigkeiten ist. Sie führt zu Zusammenhalt und Erleichterungen in schwierigen Situationen. Für Softwareentwicklungsteams lässt sich daraus mindestens ableiten, dass es weitere Fähigkeiten gibt, die für anspruchsvolle, häufig mit Stress verbundene Zusammenarbeit an hochkomplexen Problemen relevant und nicht zu unterschätzen sind.

Das in der vorliegenden Arbeit entwickelte Modell fordert von den Teams, über benötigte Fähigkeiten selbst zu reflektieren und lässt die Liste der Fähigkeiten offen. Die Beispiele zeigen, dass eine offene Liste richtig ist, da die Bandbreite der gewünschten und benötigten Fähigkeiten in den Praxisbeispielen hoch ist. Gleichzeitig zeigt sich, dass die Teams durchaus in der Lage sein können ggf. auch weniger naheliegende Fähigkeiten wie Humor selbst zu identifizieren. Damit leistet das Modell einen wichtigen Beitrag zur Forschung und Praxis agiler Softwareentwicklungsteams wie im nächsten Kapitel 8 noch deutlich wird. Das Modell benennt Fähigkeiten und ihre Auswirkungen. Gleichzeitig wird die Verantwortung, benötigte individuellen Fähigkeiten für das jeweilige Team zu identifizieren, den Teams selbstorganisiert überlassen.

##### 7.4.3.2 Teamreflexion

Es ist das zweite Anzeichen selbstorganisierter Teams, proaktiv Feedback über die eigene Leistung einzufordern (Siehe S. 44). Teamreflexion kann im Sinne Hackmans [Hac86] als eine Art gegenseitige Kontrolle verstanden werden. Durch Teamreflexion wird Regulation der Teammitglieder ausgeübt. Es werden Erwartungen ausgetauscht und angepasst. Positive Effekte wie Wissenszuwachs (im Modell eine Folge) können laut Hackman nur durch gegenseitiges Feedback erreicht werden. Bei Magpili und Pazos [MP18] ist gegenseitige Kontrolle ein Faktor auf Teamebene, der die Leistung von selbstorganisierten Teams beeinflusst. Hackman erklärt, dass es wichtig ist, an die Teams Erwartungen zu kommunizieren, um die Arbeitsstrategien regelmäßig anzupassen. In agilen Teams werden Erwartungen zu Arbeitsstrategien im Team selbst

<sup>7</sup>Zumindest soweit mir bekannt. Humor taucht in aktuellen Arbeiten zu Fähigkeiten wie beispielsweise [FS00] oder Literatur für die Praxis [VSM14] nicht auf.

und nicht von außenstehenden Manager\_innen kommuniziert. Teamreflexion trägt zum Abgleich gegenseitiger Erwartungen, Kontrolle und zur Klärung der Rollen bei und wirkt sich damit positiv auf die Teamleistung aus (Vergleich auch Modell Abbildung 7.4, S. 117).

Johnson u.a. [JBP03] bewerten ein globales Verständnis der Rollen positiv, auch wenn zur Erreichung instabile Phasen im Team notwendig sind. Es ist positiv, wenn das formale und informelle Rollenverständnis kohärent ist [JBP03]. Abramis Erkenntnisse [Abr94] zeigen ebenfalls, dass Rollenunklarheit negativen Einfluss auf das Team hat. In Bunderson u.a.[BB10] gibt es Indizien, die sich auf selbstorganisierte Teams anwenden lassen, dass Struktur und Rollenklarheit sich positiv auf psychologische Sicherheit und Leistung auswirkt. Dönmez und Grote [DG18] stellen fest, dass Rollen helfen mit Unsicherheiten in der Softwareentwicklung umzugehen. Koenigswieser [Kön06] erklärt, dass Reflexion auf Teamebene gerade in Umbruchphasen wichtig sind, da sie zur Orientierung und Erleichterung führt. Es ist es wichtig, auch Tabus anzusprechen

Durch Teamreflexion wird das globale Rollenverständnis reflektiert und im besten Fall ein Konsens gefunden, so das Ergebnis der vorliegenden Arbeit. Dass Teamreflexion positive Folgen hat, erklärt auch Hackman (s.o.). Wie vielfältig positiv sich ein kohärentes Rollenverständnis auswirkt, zeigen auch die Arbeiten aus der Psychologie. Obwohl das Grounding zu den konkreten Folgen im Modell bisher nicht vollständig theoretisch gesättigt war, bestätigen andere Arbeiten damit die vielfältigen positiven Effekte.

Babb u.a. [BHN14] strukturieren die Möglichkeiten zur Reflexion in agilen Softwareentwicklungsteams sehr genau. Sie erkennen, dass nicht nur die Retrospektive Möglichkeiten zur Reflexion bietet, sondern auch agile Praktiken wie Plannings sich für *Reflection in Action* eignen. In den angeregten Reflexionen von Babb u.a. geht es vor allem um Projekthinhalte und die technischen Wissensressourcen des Teams. Babb u.a. sehen auch, dass die Teammitglieder über ihre Fähigkeiten reflektieren.

Die Situationen der Reflexion in Babbs u.a. Reflexionsmodell sollten von Teams genutzt werden, um weitere relevante Faktoren aus dem von mir entwickelten Modell zu reflektieren. Wie das Modell zeigt, sollte in den agilen Praktiken über die Rolle der einzelnen Teammitglieder reflektiert werden, welche Verantwortung die einzelnen Teammitglieder unter Berücksichtigung ihrer Fähigkeiten wahrnehmen wollen und können. Der Umgang mit dem Zugewinn der Autonomie sollte ebenfalls reflektiert werden. Unter Berücksichtigung der Erkenntnisse, die ein Team über seine Wissensressourcen und die Projektanforderungen gewonnen hat durch Reflexion, sollte in einem nächsten Schritt geschlussfolgert werden, welche Folgen und Bedeutung es für das Team und jede Einzelne bei der Abarbeitung der Aufgaben hat. Wer übernimmt welche Verantwortung bzw. Rolle? Hier schließt die vorliegende Arbeit an bisherige an und erweitert sie sinnvoll.



Während der Teamreflexion im Modell, die während verschiedener agiler Praktiken stattfinden kann, wird der Umgang mit der zugewonnenen Autonomie reflektiert, Konflikte zwischen Eigen- und Teaminteressen offen gelegt, Rollen (Fähigkeiten und Verantwortung) reflektiert und dadurch die Wahrnehmung der Rollen im Team kohärenter. Dieser Schritt greift also in vielerlei Hinsicht bisherige relevante Forschungsergebnisse auf und ist wesentlich für die positive Zusammenarbeit im Team.

#### 7.4.4 Zusammenarbeit

Bereits in den vorherigen Abschnitten wird deutlich, dass mehrere verwandte Arbeiten wie die Hackmans [Hac86] oder Johnsons [dJHD<sup>+</sup>13] einen Zusammenhang zwischen Rollen oder Prozessstrukturen und einer guten Zusammenarbeit herstellen.

Johnson [JBP03] erklärt, dass erfolgreiche Teams eine hohes Maß an Rollenklarheit haben. Dabei sind die Rollen heterogen verteilt, um Konflikte bei überschneidender Rollenverteilung zu minimieren. Gute Kommunikation beruht dabei auf heterogenen klaren Rollen.

Hackman geht davon aus, dass die drei Prozesskriterien, die in der vorliegenden Arbeit Rollen definieren (siehe Abschnitt 7.4.3.1), die Effektivität beeinflussen. Hackman bemisst Effektivität am Output und daran wie nachhaltig eine Teamkultur geschaffen wird, in der sich alle Teammitglieder weiterentwickeln können.

Lenberg und Feldt [LF18] stellen einen Zusammenhang von Klarheit über Teamnormen und verbesserter Leistung, Zufriedenheit und psychologischer Sicherheit her.

Hochfunktionale Teams werden in der Literatur auch als *gelled* Teams bezeichnet [WB07] [DL13]. Die Teams zeichnen sich dadurch aus, dass Teammitglieder gerne und ausgesprochen erfolgreich zusammenarbeiten. Je mehr **KLARHEIT** in einem Team gewonnen wird, um so näher kommen Teams diesem Ideal.

Dass Klarheit über Rollen, Prozesse oder Normen sich auf die Zusammenarbeit auswirkt, wurde durch die Analyse der Daten ebenfalls erkannt und spiegelt sich im Modell wider. Die Qualität der Zusammenarbeit wird als eine Folge von (un-) geklärten Rollen bzw. Reflexion definiert. In der vorliegenden Arbeit wurden also die wichtigsten Kriterien und ihre Zusammenhänge erkannt, bestätigt und sind in das Modell für selbstorganisierte Softwareentwicklungsteams eingeflossen.

#### 7.4.5 Konsequenz: Teambestand

Hackman definiert Effektivität unter anderem durch das Ausmaß inwiefern die Gruppe auch zukünftig erfolgreich und gern zusammenarbeitet.

Nach dieser Definition ist der Teambestand im vorliegenden Modell ein Teil von Effektivität. Effektivität wird an den sozialen Fähigkeiten der Gruppen gemessen. Ein wichtiger Punkt, um die Relevanz des vorliegenden Modells erneut zu betonen.

Auch Johnson [JBP03] erklärt, dass in Folge guter Teamreflexion der Zusammenhalt des Teams gestärkt wird. Es geht vor allem darum, akzeptables Verhalten im Team zu definieren und Normen zu schaffen. Ein gemeinsames Verständnis stärkt den Teamzusammenhalt.

#### 7.4.6 Diskussion der verwandten Arbeiten zu den Hypothesen aus Abschnitt 7.3

##### 7.4.6.1 Hypothese 1: Selbstreflexion und proaktives Handeln ist notwendig, um Klarheit zu gewinnen und herausfordernd für die Teammitglieder in agilen Softwareentwicklungsteams.

Wie wichtig Selbstreflexion zur Zusammenarbeit ist, wird in verschiedenen Arbeiten deutlich (siehe Abschnitt 7.4.3.1).

Hackmans fünf Anzeichen von selbstorganisierten Gruppen unterstützen die Relevanz von proaktiver Selbstreflexion. Die fünf Anzeichen sind: sich persönlich verantwortlich fühlen, sich selbst kontrollieren und regulieren, Unterstützung einfordern und Hilfe anbieten.

In Heintel (ergänzende Literatur zur Teamdynamiken) ist folgendes Zitat zu finden:

Man kann es so sagen: Ohne Selbstbeobachtung, Selbstbeschreibung, Selbstthematisierung keine Selbsterforschung, ohne Selbsterforschung keine Selbsterkenntnis und kein Selbstbewusstsein, ohne Selbstbewusstsein keine Selbstbestimmung (griech. „Autonomie“), ohne Selbstbestimmung keine Selbststeuerung. [Hei06, S. 18]

Heintel stellt einen Zusammenhang zwischen Selbstreflexion und Autonomie her. Er betont damit, dass Selbstreflexion notwendig ist, um die eigene Autonomie wahrzunehmen. Im Modell also der erste Schritt zur Gewinnung von KLARHEIT. Der Zugang und der Umgang mit Autonomie ist ein zentrales Element selbstorganisierter Softwareentwicklungsteams und von besonderer Bedeutung.

Prior u.a. haben festgestellt, wie schwierig Selbstreflexion sein kann [PFL16] und fordern dies bereits von ihren Studierenden ein.

Durch die verwandten Arbeiten wird also die Schlussfolgerung, dass Selbstreflexion in selbstorganisierten Teams nötig und gleichzeitig schwierig ist, untermauert.

##### 7.4.6.2 Hypothese 2: Klarheit auf eigener und auf Teamebene fördert eine positive Zusammenarbeit.

Die Relevanz der Selbst- und Teamreflexion in verwandten Arbeiten wurde zuvor (siehe Abschnitte 7.4.3.1 und 7.4.3.2) ausführlich erklärt: Sie führt zu kohärenten Rollen, verbesserter Zusammenarbeit, löst Konflikte und ermöglicht die Diskussion über den Umgang mit zugewonnener Autonomie und den daraus entstehenden möglichen Konflikten zwischen Eigen- und Teaminteressen. Positive Effekte können nur entstehen, wenn beide Ebenen, sowohl Selbst- als auch Teamreflexion erreicht werden. Die Erreichung eines Konsens ist dabei nicht zwingend nötig. Erleichterung schafft auch schon Transparenz und damit KLARHEIT durch Reflexion.

##### 7.4.6.3 Hypothese 3: Gegenseitiger Konsens über die Fähigkeiten der Teammitglieder ist wichtiger als die Fähigkeiten selbst.

Ein gemeinsames Verständnis über informelle und formelle Rollen und die dazugehörigen Fähigkeiten erhöht die Leistungsfähigkeit in Teams [JBP03]. Wie wichtig dabei

Fähigkeiten außerhalb der Kernfähigkeiten wie Humor sein können, erklärt Johnson [JBP03] und in Abschnitt 7.4.3.1 wird deutlich, dass es auch von mindestens einem selbstorganisierten Softwareentwicklungsteam erkannt wurde. In einem Beispiel von Hoda siehe Abschnitt 2.3.5, wurde ein Teammitglied versetzt, obwohl es wertvolle Erfahrung hatte. Die Person hat durch übergriffige Entscheidungen die Selbstorganisation untergraben. Für die Selbstorganisation agiler Teams ist Konsens über die Fähigkeiten wichtiger als die Höhe oder Art der Fähigkeiten.

#### 7.4.6.4 Hypothese 4: Scrum-Master und andere Teammitglieder können unterstützen Klarheit zu gewinnen.

In den Forschungsarbeiten [Hac86] [Hei06] [JBP03] [SBB<sup>+</sup>09] wird deutlich, dass die Vorhersehbarkeit von Teamverhalten an Grenzen stößt. Daraus könnte man schlussfolgern, dass auch die Arbeit von Coaches und Scrum-Mastern an Grenzen stoßen muss. Nichtsdestotrotz, werden in den verwandten Arbeiten verschiedene Anforderungen an Coaches wie Scrum-Master gestellt. Magpili und Pazos erkennen in ihrem Review, dass es Aufgabe von Führungspersonen in selbstorganisierten Teams ist zu coachen, Selbstreflexion anzuregen und hilfreiche Teamnormen zu entwickeln (siehe Abschnitt 2.3.1). Hackman erklärt, dass ein Coach unterstützen kann bezüglich Engagement (Koordination der Aufgaben), Wissenszuwachs, kreativer Prozesse und der Passung zwischen Person und Aufgabe (siehe Abschnitt 2.3.2).

Andriyani [AHA17] erkennt, wie schwierig es für Teams ist höhere Reflexionsebenen zu erreichen. Bradley u.a. weisen darauf hin, dass es unterschiedliche Ergebnisse gibt, ob interpersonelles Eingreifen zur Teamentwicklung sich positiv auf die Teamleistung auswirkt [BWM03]. In der Metaanalyse von Salas u.a. [SRMD99] über den Effekt von Teaminterventionen auf Leistung wird ebenfalls klar, dass es widersprüchliche Ergebnisse gibt. Lediglich *role clarification* wirkt sich durchweg positiv auf die Teamleistung aus. Wobei unter *role clarification* ein gegenseitiger Austausch über die individuellen Verpflichtungen im Team verstanden wird.

Die Aufgaben, die in den verschiedenen verwandten Arbeiten möglichen Coaches oder Führungspersonen zugeschrieben werden, lassen sich zum Konzept **KLARHEIT GEWINNEN** im Modell zuordnen. Durch den Prozess im Modell werden Teamnormen und Rollen erarbeitet und Aufgaben und Wissenszuwachs koordiniert. Unterstützendes Eingreifen bezüglich der Rollenklarheit wirkt sich positiv aus. Im Hinblick auf das Modell bedeutet das ein Coach bei den verschiedenen Schritten der Selbst- und Teamreflexion unterstützen kann und so die Zusammenarbeit fördern kann. Im Falle selbstorganisierter Softwareentwicklungsteams ist das Aufgabe von Scrum-Mastern. Wie herausfordernd und wichtig diese Aufgabe ist, zeigen Andriyanis Ergebnisse.

#### 7.4.6.5 Hypothese 5: Trotz aller Unterstützung kann der Prozess zu jedem Zeitpunkt scheitern insbesondere an persönlichen Sympathien und Emotionen.

Die Unvorhersehbarkeit (siehe oben) macht es unmöglich, den Prozess vollends zu steuern und somit kann er stets scheitern. Insbesondere Gefühle einzugestehen ist eine Hürde in der Reflexion und nicht einfach [Kön06]. Verändern sich Beziehungen durch Sympathien hat das Folgen für die gesamte Gruppe [Hei06]. Das Modell

wird daher an seine Grenzen stoßen, die durch Emotionen und menschliche Beziehungen geprägt sind<sup>8</sup>. Es kann lediglich versucht werden, in einem angemessenen Rahmen Transparenz und damit Erleichterung zu schaffen. Bunderson und Boumgarden [BB10] stellen fest, dass Struktur z.B. ein gegenseitiges Rollenverständnis sich positiv auf Leistung, psychologische Sicherheit und Konfliktfreiheit auswirkt. Somit kann das Modell vielleicht dazu beitragen, ein stabiles Arbeitsumfeld zu schaffen, das einen positiven Umgang mit Menschen und ihren Emotionen ermöglicht (siehe auch folgenden Abschnitt zur Validierung 8.1).

---

<sup>8</sup>Und damit auch Grenzen, die gewahrt werden sollten und sich niemals, auch nicht in der Softwareentwicklung vernachlässigen oder prozessieren lassen.

## Kapitel 8

# Validierung

Did you conduct a reliability check; i.e., have your analysis reviewed by someone else. If so, who, how, what did they find and what changes resulted? Describe their expertise. [SRF16]

Dies ist eine Forderung von Stol u.a. für eine angemessenen GTM. Hoda beschreibt, dass GTM häufig dafür kritisiert wird keine explizite Validierungsphase zu beinhalten [HNM12]. Sie sieht jedoch in mehreren Aktivitäten während des Forschungsprozesses wichtige Validierungsprozesse, die helfen die Entwicklung einer GT voranzutreiben und zu unterstützen. Sie hat beispielsweise ihre Ergebnisse mehreren Teams vorgestellt und diskutieren lassen. Auch in der Diskussion mit anderen Forscher\_innen werden die Ergebnisse erneut beleuchtet und damit stückweise validiert, so Hoda.

Als eine weitere Quelle der Validierung sieht Hoda den nachträglichen Literaturreview. Inwiefern die eigenen Ergebnisse an bisherige Erkenntnisse anschließen können, ist ein Indiz zur Validierung des Prozesses, so Hoda.

Wie das von mir entwickelte Modell an bisherige Literatur anschließt, habe ich in Abschnitt 7.4 dargelegt. Dabei ist hervorzuheben, dass Forschungsergebnisse aus anderen Forschungsrichtungen (R2) mit anderen Methodiken (G1-Methode) und anderer Daten (G2-Daten) zu ähnlichen Ergebnissen kommen (beispielsweise Hackman siehe Abschnitt 2.3.2 und 7.4). Damit kann angenommen werden, dass ein Teil meiner Ergebnisse sich verallgemeinern und sich auf andere Gebiete übertragen lässt. Mein Forschungsprozess hat zu Ergebnissen geführt, die in anderen Forschungsbereichen bereits validiert sind (teilweise auch quantitativ siehe beispielsweise [LF18] oder [BB10]).

Charmaz GTM beinhaltet wie andere GTMs keine explizite Validierungsphase. Allerdings ist die Resonanz aus der Praxis ein wichtiges Evaluationskriterium für eine GT.

Da Charmaz GTM besonderen Wert auf Transparenz legt, Stol u.a. dies ebenfalls fordern und es in bisherigen Veröffentlichungen wie Hoda [Hod11] lediglich im Forschungsprozess enthalten aber nicht detailliert dargelegt wird, werde ich im folgenden Abschnitt den Validierungsprozess genau beleuchten.

Während des gesamten Forschungsprozesses gab es unterschiedliche Arten der Validierung. Die Validierungen unterscheiden sich in der Art und dem Zeitpunkt der Erhebung.

Zum einen gab es Validierungen, der sich entwickelnden Ergebnisse durch die beforchten Teams selbst, vor allem während und auch nach dem Forschungsprozess (also ähnlich zu dem Vorgehen Hodas [HNM12]).

Zum anderen und darüber hinaus gab es während und nach dem Forschungsprozess Validierungen durch Präsentationen und Diskussionen außerhalb der beforchten Teams. Dazu gehörte kollegiale Beratung zu Teamsituationen während Meet-ups, Diskussionen der Teamsituation mit externen Beraterinnen, Forschungsgruppenaustausch oder Präsentationen auf Konferenzen und Tagungen. Eine Auswahl der wichtigsten Validierungen fasst Tabelle 8.1 zusammen. Dabei meint "nach dem Forschungsprozess", dass die gesamte Erhebung und Forschung bereits weitestgehend abgeschlossen war.

## 8.1 Validierung der Ergebnisse während des Forschungsprozesses

Während des Forschungsprozesses wurden die sich entwickelnden Ergebnisse in den jeweiligen Teams präsentiert und diskutiert (ausgenommen  $t_1$ ). Die Diskussionen dienten der Validierung des bisherigen Verständnissen, der erfassten Daten und den daraus resultierenden Zwischenergebnissen. Die Resultate der Diskussionen wurden verwendet, um die Forschungsergebnisse weiterzuentwickeln und zu spezifizieren. Alle vier Teams waren hochinteressiert an den (Zwischen-) Ergebnissen, bewerteten die Ergebnisse als ausgesprochen hilfreich für die Teamentwicklung und sie führten dazu anschließend angeregte Diskussionen über mögliche Konsequenzen in der Zusammenarbeit oder neue Vorgehensweisen. In allen vier Teams war die Validierung somit ausgesprochen ertrag- und erfolgreich.

### 8.1.1 Validierung in $t_2$

In  $t_2$  bestand die längste Zusammenarbeit. Es gab bereits während der Erhebung der Daten zwei mehrstündige Diskussionen. In einem Treffen wurden die vorläufigen Ergebnisse mit der PO des Teams und einer externen agilen Beraterin ausführlich diskutiert und evaluiert. In einem weiteren Treffen wurde die Teamsituation in einer kollegialen Beratung mit der PO des Teams und weiteren agilen Expertinnen diskutiert und die Zwischenergebnisse validiert.

Nach den ersten Zwischenergebnissen gab es einen anonymen Online-Fragebogen für alle Teammitglieder. In dem Fragebogen ging es darum, über sich selbst und den Beitrag anderer aus dem Team zu reflektieren und Auskunft zu geben. Er spiegelt also die Selbstreflexion und ein erster Schritt zur Teamreflexion des späteren Modells wider.

Nach der Erhebung der Daten habe ich außerdem eine Retrospektive in  $t_2$  durchgeführt, um die Zwischenergebnisse der Forschung zu validieren. Zu diesem Zeitpunkt der Forschung war das Modell noch nicht entwickelt. Bemerkenswerterweise im Sinne der Methodik auch konsequenterweise, sind in der Retrospektive die späteren Schritte des Modells bereits erkennbar. Bei der Erhebung der Daten entwickelte sich die Hypothese, dass fehlende Klarheit über die eigenen Verantwortlichkeiten und

Tabelle 8.1: Validierungen der Forschungsergebnisse

	Während des Forschungsprozesses	Nach dem Forschungsprozess
t1	-	
t2	Präsentation, Online-Fragebogen, Retrospektive	
t3	Präsentation und Diskussion	
t4	Gruppendiskussion	
t5	Gespräch SM, Präsentation, Gruppendiskussion	Interviews t5dev2, t5dev3 und t5dev4
Externe		
	t2: kollegiale Beratung Meet-up, Gruppendiskussion mit externer Beraterin	Meet-Up Expertinnen 24.08.2018
	Diskussion der Zwischenergebnisse mit Judith Andresen, Expertin und Autorin u.a. von [And14]	Pilotpräsentation Forschungsgruppe und Expertin
	Poster auf Konferenz [Bar18] und Workshop auf Fachtag zum selben Thema 2018 <sup>a</sup>	Agile piano Meet-up 28.05.2018
	Vortrag und Diskussion auf CHASE 2018 [BP18]	

<sup>a</sup><https://www.gender-wissen-informatik.de/Fachtage/Berlin>, geprüft November 2019

die Verantwortlichkeiten des Teams, die mit Selbstorganisation einhergehen, zu Problemen in der Zusammenarbeit führen. Es wurden drei offene weitgefaste Fragen an das Team gestellt (siehe auch Abbildung 8.1, Spalte 1-3):

- What is the purpose of self-organization?
- What is required from individuals?
- What is required from the team?

Alle Teammitglieder hatten Zeit, diese Fragen an einem digitalen Board für sich zu beantworten (das Team war verteilt). Anschließend wurden alle Antworten sichtbar geschaltet und im gesamten Team diskutiert. Nach der Diskussion wurden wichtige Themen bewertet (siehe Abbildung 8.1, Spalte 4). Zu zwei wichtigen Themen wurden durch offene Fragen Diskussionen zu möglichen Verbesserungsmaßnahmen angeregt (siehe Abbildung 8.1, Spalte 5 und 6). In dem digitalen Board sind einige Antworten und Hinweise auf das spätere Modell zu finden: In Spalte 1 sind bereits Verweise auf die späteren Ausgangspunkt des Modells zu finden. Aussagen wie *„to give responsibility to the team to let the team decide how to work“* und *„Self organization gives team power to decide and responsibility at the same time“* spiegeln die erhöhte Teamautonomie wider, die das Team durch Selbstorganisation erlangt.

Die ersten beiden Aussagen in Spalte 2 *„being active instead of waiting for something to happen“* und *„be pro-active and be ready to take responsibility and share own difficulties“* betonen wie wichtig proaktives Handeln sind und eine gemeinsame Verantwortung, was sich im Rollenverständnis wiederfindet.

Die Aussage in der dritten Spalte *„Have same understanding of the way we want to work together“* kann als erster Verweis auf ein globales Rollenverständnis verstanden werden.

### 8.1.2 Validierung in $t_3$

In  $t_3$  wurden die teambezogenen Zwischenergebnisse dem gesamten Team präsentiert und durch das Team in einer Diskussion validiert.

### 8.1.3 Validierung in $t_4$

In  $t_4$  wurden die Ergebnisse ähnlich wie in  $t_2$  mit offenen Fragen beleuchtet. Diesem Team fehlte Klarheit über ihre crossfunktionale Zusammenarbeit und die damit einhergehende Verantwortung bzw. Rollen. Ich habe zwei offene Fragen gestellt: Was ist der Sinn und Zweck von Crossfunktionalität...

- für dich?
- für das Team?

Das offene Vorgehen bei der Validierung wie in  $t_2$  und  $t_4$  hat einige Vorteile. Die offenen Fragen verhindern ein Forcing der Ergebnisse. Das heißt die Teilnehmer\_innen wurden bei der Beantwortung der Fragen in keine Richtung gedrängt. Gleichzeitig brachte dieses Vorgehen in den konkreten Fällen hervor, dass tatsächlich



Set the context of the retrospective here...

What is the purpose of Self-Organization	What is required from Individuals	What is required from the Team	For Discussion	What could I do to support Trust/Solidarity	What would make me feel more self-motivated
have more efficient time planning	being active instead of waiting for something to happen	be able to come to an agreement and to stick to this agreement also if it's not in accordance with own way of thinking	Autonomy	be fair, honest (but not mean) and supportive	more optimistic atmosphere in the team
to give responsibility to the team to let the team decide how to work	be pro-active and be ready to take responsibility and share own difficulties	trust	Same level of knowledge	encourage discussions, ask more questions if I feel something is not clarified until the end	consider mistakes as feedback and be happy to learn from them, be ready for experiments
to give us impression that we can decide about our work on our own - to personalize us with our tasks and thus make us more effective/motivated	sharing ideas, problems, concerns instead of trying to solve everything on your own.	trust understanding acceptance	Self motivation	Support whatever decision the team has come up with as if it was my own ... "we have decided ..."	more time for software design before implementation
To find the right mixture of motivated people and "experts"	self motivation trust to other teammembers ability to express opinions openly	to quickly come up with solutions with keeping most of the team happy (because it's impossible to make everyone happy)	Trust/Solidarity	Tell a colleague how much I appreciate his/her work and that I am happy to have him/her in the team.	Knowing others would help out and participate
Self-organization gives team power to decide and responsibility at the same time	Self motivation, as there is no (or less) control from a superior. Trust in your colleagues Courage to speak up Interest in other peoples' opinions Ability to compromise	Have the same understanding of the way we want to work together		appreciate each other's contribution to the project	more silence around
to distribute work in a more effective way	to take responsibility and not hide behind team/PO/decision maker	A focus on understandability and extensability		be friendly and kind even if you are in a bad mood and do not offend someone else	have user stories I feel able to work on in stand up; agree on/commit to individual goals (e.g. finish this task)
fast decisions without having to consult a leader / boss	A willingness to participate on all topics and help as best we can			have trust in myself first, maybe by being more satisfied with quality of my work or maybe by increase of contribution	not working on dummy topics
people decide who know best about the matter in question	be motivated and fair stick to agreements be open to give and receive positive and negative feedback				better atmosphere in the office
equal workload					
Lena: Solidarity					
Lena: Autonomy					

Abbildung 8.1: Digitales Board der Retrospektive von t2 am 19.10.2017 zur Validierung der Forschungszwischenergebnisse

Klarheit über sich selbst und im Team in Bezug auf Verantwortlichkeiten fehlte. Beide Teams empfanden dieses Vorgehen als ausgesprochen nützlich und Augen öffnend. Zum einen wurden die Teammitglieder angeregt, unabhängig voneinander über ihre eigenen Verantwortungsbereiche zu reflektieren. Zum anderen wurden im zweiten Schritt, beim Teilen der Reflexionen, die unterschiedlichen Wahrnehmungen offen gelegt und anschließend diskutiert. Insbesondere die persönlichen Verantwortlichkeitsgrenzen und die damit einhergehende Erwartungshaltung an Andere wurde intensiv diskutiert.

#### 8.1.4 Validierung in $t_5$

In  $t_5$  und  $t_6$  baten die Scrum-Master darum, die Ergebnisse zuerst präsentiert zu bekommen, um einen Eindruck über die Qualität gewinnen zu können. Nach der Präsentation der teamspezifischen und darüber hinaus entwickelten Forschungsergebnisse bewerteten sie die Ergebnisse als hochrelevant und beide Scrum-Master diskutierten noch lange untereinander und mit mir darüber. In einem weiteren Termin durften ich die Ergebnisse dem gesamten Teams vorstellen. Die Präsentation und Diskussion wurde aufgezeichnet. Es folgte eine ausführliche und angeregte Diskussion, die über das Meeting und die Aufzeichnung hinaus ging. Beim nächsten Treffen einige Wochen später stellte sich heraus, dass keine konkreten Maßnahmen umgesetzt werden konnten. Woran die Umsetzung scheiterte, habe ich nicht untersucht. Grundsätzlich ist ein Forschungs- zu Praxistransfer häufig schwierig. Auch im üblichen Scrum-Prozess werden Verbesserungsvorhaben, die in Retrospektiven festgehalten wurden, häufig nur im geringen Maße umgesetzt. Ob es also am Arbeitsalltag, weiteren Hürden im Team oder fehlenden konkreten Ansätzen lag, lässt sich nicht mehr überprüfen. Es wäre ein spannendes Feld für zukünftige Forschung.

## 8.2 Validierung der finalen Ergebnisse

Eine abschließende Validierung ist in bisherigen Arbeiten nicht explizit zu finden ([SRF16] [Hod11]), da sie von der GTM nicht explizit gefordert wird. Es besteht jedoch die Forderung das Forschungsergebnisse einen Nutzen für die Lebensrealität von agilen Softwareentwicklungsteams haben sollen [Ral18]. Charmaz fordert als ein Evaluationskriterium folgende Frage zu beantworten (siehe auch Abschnitt 4.2.9):

Does your grounded theory make sense to your participants or people who share their circumstances? Does your analysis offer them deeper insights about their lives and worlds? [Cha14, S.337f]

Am Ende meines Forschungsprozesses bestand immer noch die Frage, ob das Modell trivial oder relevant ist (siehe auch Abbildung 7.3). Aus diesem Grund habe ich das Modell mehrfach verschiedenen Expert\_innengruppen vorgestellt. Zwei wichtige längere Vorstellungen und Diskussionen mit Personen aus der Praxis möchte ich hier darlegen. Dabei bestand durchaus die Möglichkeit, dass die Praxis keinen Zugang zu dem Modell finden würde oder es als nicht hilfreich oder relevant einstufen würde. Zum Ende des Forschungsprozesses wurde das Modell in einem kleineren Rahmen

vom Expert\_innen diskutiert. Nach Beendigungen der Forschung, wurde das Modell in einem großen öffentlichen Rahmen Praktiker\_innen und Expert\_innen vorgestellt, diskutiert und seine Nützlichkeit evaluiert. Damit geht die Validierung meiner Arbeit weit über die mir bekannten Validierungen bestehende GTs hinaus. Beide Ereignisse werden im nun folgenden Abschnitt transparent dargelegt.

Die finalen Ergebnisse, also das Modell habe ich einer agilen Expertinnenrunde vorgestellt. Die Expertinnenrunde bestand aus einer agilen Beraterin einer bekannten Agentur aus Berlin, einer PO, einer Expertin aus dem Bereich Personalentwicklung und agile Transformation in einem großen Unternehmen und einer Selbstständigen mit Berufserfahrung im agilen Umfeld.

Es entwickelte sich direkt eine angeregte Diskussion. Alle Expertinnen fingen direkt an, das Modell auf praktische Erfahrungen und Erlebnisse in Teams anzuwenden und die jeweiligen Situationen neu zu reflektieren. Daraus entstand eine Diskussion welche Vorteile das Modell gegenüber der alleinigen Fokussierung auf Expertise bietet. Wenn sich ein Team über die Expertise einzelner Mitglieder definiert, entstehen Grenzen und egozentrische Arbeitsweisen. Während mit dem Modell und dem Ziel Klarheit zu gewinnen, laut der Expertinnen voraussichtlich Vertrauen und Zusammenhalt gefördert werden könnte. Die Diskussion wurde durch eine Teilnehmerin visuell protokolliert siehe Abbildung 8.2.

Aus dieser Validierung lässt sich ableiten, dass die Anwendung des Modells auf die tatsächliche Praxis den Expertinnen ausgesprochen leicht fiel und sie es für relevant erachteten. Das Modell war verständlich und bot einen ausgeprägten Mehrwert. Die Expertinnen konnten bisherige schwierige Teamsituationen aus der Praxis neu reflektieren und einordnen. Ein Perspektivwechsel von der Fokussierung auf Expertise hin zum Modell und der Gewinnung von **KLARHEIT** hätte aus ihrer Sicht viele positive Folgen für Teams. Zu den positiven Folgen gehört u.a. die Stärkung von Vertrauen, Raum für alle und gegenseitige Wertschätzung.

Die agilen Expertinnen brachten noch zwei wichtige Fragen auf. Zum einen ist eine regelmäßige Reflexion herausfordernd und braucht Ressourcen. Wie viel Zeit und Ressourcen sollten also zur Gewinnung von Klarheit verwendet werden, um ein ausreichendes Maß an Klarheit zu gewinnen? Zum anderen wurde hinterfragt in welchem Maß, implizites Wissen explizit gemacht werden sollte, ohne dem Team oder Individuen zu schaden. Es müssen Grenzen zwischen Privatem und Arbeitsbereich gewahrt werden. Als Beispiel wurde genannt, dass an Teammitgliedern mit herausragender Expertise auch hohe Erwartungen gestellt werden. Ob sie diesen Erwartungen bezüglich ihrer Verantwortung gerecht werden können wird auch bedingt durch ihre persönliche Situation, wie Krankheit oder Belastung im privaten Umfeld. Persönliche Situationen möchten Teammitglieder nicht zwingend teilen und eine Grenze im Hinblick auf den Umfang der **GEWINNUNG VON KLARHEIT** muss gewahrt werden.

Eine abschließende Validierung fand zum Ende der Forschungsarbeit auf dem *agile Piano Meet-up* bei der Firma *Aperto - An IBM Company* statt. Das Meet-up findet regelmäßig in Berlin mit ausgewählten Redner\_innen statt. Beim Meet-up einen Monat zuvor hatte Jeff Sutherland, ein Erfinder der Scrum-Methodik gesprochen.

Das Meet-up zur Validierung der Forschungsarbeit trug den Titel "Role deficiencies can wreck agile teams" und fand am 28.05.2019 statt. Es kamen über 30 Personen aus verschiedenen Unternehmen mit agilen Arbeitsweisen zusammen. Das Meet-up



Abbildung 8.2: Diskussionsergebnisse der Expertinnenrunde zum Modell am 24.08.2018

dauerte 3 Stunden.

Zum Einstieg haben alle Teilnehmer\_innen an einer kurzen Übung zur impliziten Reflexion des eigenen Teams teilgenommen (die Übung wird in Abschnitt 8.4 genauer erläutert). Anschließend habe ich die Ergebnisse in einer ca. 20 minütigen Präsentation vorgestellt. Das Modell wurde dabei inhaltlich auf Rollen reduziert, um ein schnelles Verständnis zu ermöglichen (siehe Abbildung 8.3).

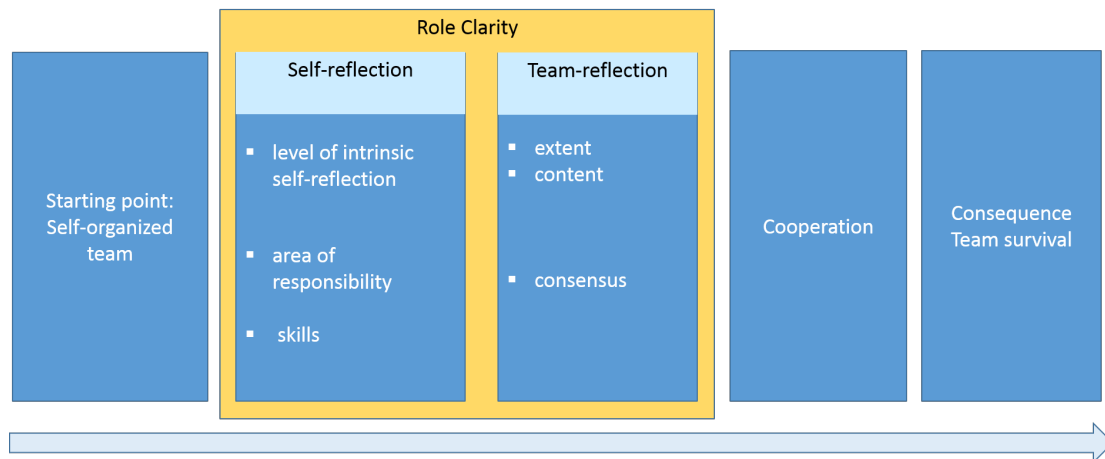


Abbildung 8.3: Reduziertes Modell aus der Präsentation zum Meet-up am 28.05.2019

Das Modell wurde mit Praxisbeispielen greifbar dargestellt und mit Bildern aus der populären Medienkultur hinterlegt. Damit sollte ein Bezug zum Umfeld der Teilnehmenden hergestellt und die Aufmerksamkeit und Abstraktionsfähigkeit unterstützt werden.

Anschließend teilten sich die Teilnehmenden in fünf Arbeitsgruppen mit mindestens fünf Teilnehmenden auf. Sie erhielten ein ausgedrucktes Modell, ein Flipchart-Papier und folgende Fragen:

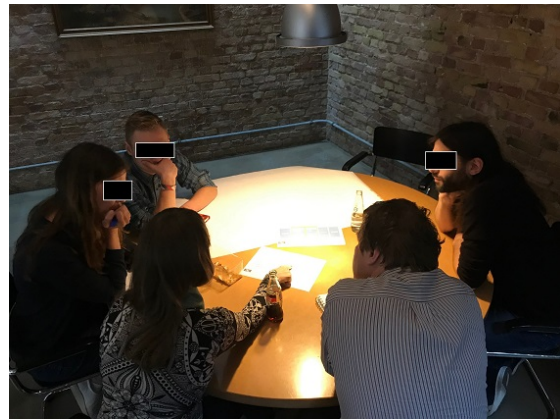
1. Please describe one critical situation in one team.
2. Where is Role Clarity likely missing? (Self-reflection? Team Reflection? Both?) Mark it on the model.
3. What tool could help in that situation to gain clarity and/or find consensus?
4. Was the model helpful? Why?
5. Any more ideas? Feedback?

Alle fünf Gruppen diskutierten in den angesetzten 20 Minuten angeregt miteinander. Die Ergebnisse der Gruppenarbeit wurden anschließend im Plenum präsentiert und diskutiert (siehe Abbildungen 8.4). In allen Gruppen wurden schwierige Teamsituationen anhand des Modells neu beleuchtet und reflektiert. In einigen Teams wurden konkrete Maßnahmen hergeleitet (siehe auch Abschnitt 8.4). Das Modell wurde von allen verstanden und konnte erfolgreich auf konkrete Teamsituationen angewendet werden. Im Plenum bewerteten alle Teams das Modell als hilfreich.

Abbildung 8.4: Bilder des Meet-ups zur Validierung am 28.05.2019



Präsentation des Modells



Anschließende Gruppenarbeit



Präsentation der Gruppenarbeit im Plenum 1



Präsentation der Gruppenarbeit im Plenum 2

Nach der Gruppenarbeit fand weiterer informeller Austausch statt. Beim Hinausgehen konnten die Teilnehmenden Feedback mit Klebezetteln an eine Wand hängen. Im Nachgang habe ich über Slack und LinkedIn versucht, weiteres Feedback einzuholen.

Über LinkedIn kam wenig später eine Rückmeldung, dass das Modell bei der Zusammenarbeit in einem Team angewendet wird („Ich fand das Konzept für mich relevant und übernehme aktuell ein paar Aspekte davon in meiner Arbeit mit einem Team. [...]“)

Eine Teilnehmerin hätte sich konkretere Handlungsanweisungen gewünscht, insbesondere für die Zeitpunkte der Reflexionen.

Der Organisator des Meet-ups fand die Ergebnisse so wertvoll und herausragend, dass er darum bat, sie zu einem Pattern im Rahmen eines Projektes von Jeff Sutherland zu verarbeiten<sup>1</sup>. Leider konnte aus Zeitgründen keine Umsetzung erfolgen.

Das Feedback zum Inhalt auf den Klebezetteln war durchweg positiv (siehe Abbildung 8.5, Roter Klebezettel zu „Can Improve“: „Try an ice-breaker or some networking exercise to bring ppl together as they come in“). Ein Feedback soll hier beispielhaft besprochen werden, da es die Qualität der Methodik unterstreicht (siehe auch Deckblatt):

Sometimes, stating what is quite obvious is a real eye-opener. Very interesting!

Dieses Zitat unterstreicht, dass das Modell implizites Wissen explizit darstellt und Prozesse und Mechanismen offenbart. Damit wurde das Ziel der Grounded-Theory erreicht. Es unterstreicht wie wertvoll die Offenlegung der tiefen Strukturen eines Prozesses sein kann. Das macht die Qualität einer Grounded-Theory aus.

Ein weiteres, nicht auf Klebezetteln dokumentiertes Feedback einer Person, die von sich aus sagt, dass sie eher forschungskritisch ist, war dass sich die langjährigen Forschungsarbeit gelohnt hätte und alle ihre Arbeitskolleg\_innen davon erfahren sollten, um davon zu profitieren.

Aus der Validierung lässt sich ableiten, dass die Ergebnisse für die Praxis von hoher Bedeutung sind. Vielen der Teilnehmenden aus unterschiedlichen Unternehmen half das Modell, eine neue und hilfreiche Perspektive auf ihr Team zu erlangen.

## 8.3 Relevanz der Ergebnisse für die Praxis

Die Forschungsergebnisse wurden in der Praxis in Form von Austausch und Diskussionen (siehe vorherigen Abschnitt) validiert. Die Praktiker\_innen waren in der Lage das Modell anzuwenden und neue Erkenntnisse für sich zu generieren.

Praktiker\_innen bewerteten das Modell aus folgenden Gründen als relevant und hilfreich:

**Greifbar machen:** Der Prozess wird benannt und dadurch für Scrum-Teams erstmalig bewusst verhandel- und steuerbar. Durch die detaillierte Darlegung des Prozesses wird er sichtbar und besprechbar im Team und mit Einzelnen.

**Bewusste Steuerung:** Wenn der Prozess und die Faktoren greifbar sind, kann gezielt und bewusst, statt intuitiv gesteuert werden.

---

<sup>1</sup>siehe auch ScrumPlop.org

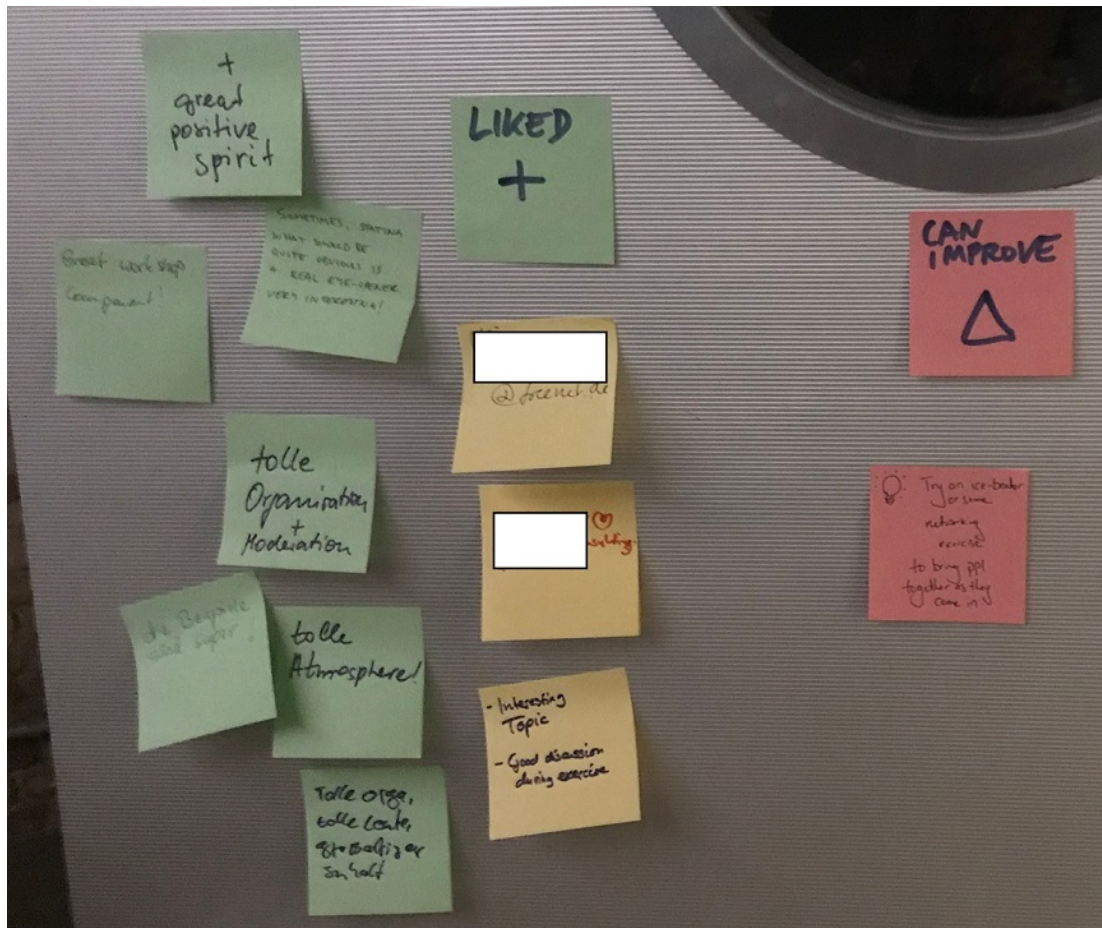


Abbildung 8.5: Feedback nach dem Meet-up am 28.05.2018, leider in schlechter Bildqualität. Hier eine Auswahl von links nach rechts: „Great Workshop Competent!“ (ganz links); „Die Beispiele sind super!“ (u.l.); „großartiger Inhalt“ (grüner Klebezettel ganz unten); „Good discussion during exercise“ (gelber Zettel ganz unten). Einige Teilnehmenden haben ihre Email-Adresse hinterlassen, sie wurden anonymisiert.



Die Praktiker\_innen erhofften sich folgende mögliche Vorteile aus der Anwendung eines solchen Modells in der Praxis:

**Motivation:** Zu Verstehen welche Vorteile es bringt, Selbst- und Teamreflektion zu Erlangung von **TEAM KLARHEIT** anzuregen, erhöht die Motivation sich in den Prozess einzubringen. Das Verständnis für den stetig veränderbaren Prozess erhöht die Motivation, sich aktiv zu beteiligen und Einfluss zu nehmen.

**Vertrauen:** Durch den Prozess des gemeinsamen Reflektierens und der **GEWINNUNG VON KLARHEIT** wird das Vertrauen gestärkt. Gegenseitige Wertschätzung kann gefördert werden.

**Psychologische Sicherheit:** Durch gewonnene Klarheit kann auch Sicherheit gewonnen werden, für sich selbst und das Team.

## 8.4 Hilfreiche Werkzeuge aus dem agilen Umfeld zur Reflexion und der Gewinnung von Klarheit

In diesem Abschnitt werden bereits vorhandene und mögliche neue Methoden -im agilen Umfeld Werkzeuge genannt- vorgestellt und der Nutzen im Hinblick auf das Modell reflektiert. Die Werkzeuge wurden von Expert\_innen empfohlen, nachdem sie das Modell kennengelernt haben und sie daraufhin über ihre Teamsituation und die möglichen hilfreichen Werkzeuge diskutiert haben. Ein Werkzeug, die Teamaufzeichnung, wurde von mir während der Forschungsarbeit entwickelt und erprobt. Die Werkzeuge greifen häufig mehrere Ebenen des Modells auf. Welches Werkzeug zu welcher Situation und welchem Team passt, muss individuell entschieden werden. Ein einfaches Ursache-Wirkung-Prinzip greift hier zu kurz. Es geht darum, erste Ideen um einen Reflexionsprozess im Sinne des Modells anzuregen, festzuhalten.

**Rollenkartenspiel** unterstützt die Reflexion über die Vor- und Nachteile einzelner Teamrollen und damit -verantwortlichkeiten: <http://agilesuperleaders.info/>, geprüft Juni 2020. Das Rollenkartenspiel könnte um Hodas Rollen siehe Abschnitt 2.3.5 ergänzt werden.

**De Bonos 6 Hüte** stärkt die Empathie für andere Personen und Verhaltensweisen und ermöglicht Reflexion über die Vor- und Nachteile einzelner Rollen. Hilft Klarheit zu gewinnen: [https://retrospectivewiki.org/index.php?title=6\\_Thinking\\_Hats\\_Retrospective](https://retrospectivewiki.org/index.php?title=6_Thinking_Hats_Retrospective), geprüft Januar 2019

**Jobtitel Spiel** unterstützt den Hauptaspekt seine eigene Rolle zu formulieren und anschließend zu diskutieren, ob das Team die Einschätzung teilt. <http://tastycupcakes.org/2016/01/the-job-title-game/>, geprüft Januar 2019

**Lego serious play:** Alle Teammitglieder bauen eine metaphorische Teamszene und erklären sie dem Team mit der Beantwortung der Fragen:

- Wer steht wo? (Teamreflexion)
- Wer macht was und warum? (Teamreflexion)

- Was mache ich selbst? (Selbstreflexion)

Es führt zu mehr Offenheit in dritter Person über eine metaphorische Situation zu sprechen. Es werden verdeckte Teamprobleme dadurch offenbart.

**Theaterstück schreiben:** Ein ähnliches Werkzeug ist es, ein Theaterskript für das Team zu schreiben. Alle Teammitglieder bekommen eine Rollenbeschreibung im Theaterstück und eine typische Theaterszene wird verfasst. Das soll die Reflexion der Rollen und der Situation dienen.

**Feedback:** Eine Teilnehmerin (Teilnehmerin) des Meet-ups berichtete, dass sie, um das Maß an Klarheit zu eruieren, regelmäßig Feedback einholt vom Team. Sie verschickt eine Email in der sie allen Teammitgliedern zwei Fragen stellt:

- Was macht Teilnehmerin wohl den ganzen Tag?
- Was sollte Teilnehmerin eigentlich tun?

Damit ermöglicht sie Teamreflexion zu ihrer Rolle. Gleichzeitig ermöglicht es ihr zu prüfen, ob sie mit ihrer eigenen Wahrnehmung dieses Verständnis teilt.

**Nicht-Rolle:** Die Perspektive umzukehren hilft, Grenzen besser wahrzunehmen und schafft mehr Offenheit. Daher kann es hilfreich sein im Team eine Selbst- und Teamreflexion anzuregen, was nicht Teil der eigenen und der Rolle anderer ist.

**Teamzeichnung:** In den Interviews, bei einem Vortrag auf einer Fachtagung und beim Meet-up zur Validierung, wurde folgendes Werkzeug von mir entwickelt und validiert: Die Teilnehmenden wurden aufgefordert ihr Team zu zeichnen. In den Interviews wurde dazu gesagt, dass die Zeichnung dazu dienen soll, dass ich im Interview Erläuterungen zum Team besser folgen kann. Mehr wurde dazu nicht gesagt. Die Zeichnungen unterscheiden sich dabei von Person zu Person (siehe auch Abbildung 8.6). Alle Zeichnungen beinhalteten jedoch mindestens ein strukturgebendes Element für das Team wie beispielsweise Scrumrollenbezeichnungen, Zugehörigkeiten zur Firma oder Junior und Senior Status. Diese unbewusst verwendeten strukturgebenden Elemente geben einen Hinweis darauf, nach welchen Rollenmerkmalen im Team unterschieden wird. Über diese Grenzen der Rollen zu sprechen und zu reflektieren schafft **KLARHEIT**. In Abbildung 8.6 wurden für das Meet-up original Teamzeichnungen aus den Interviews abgezeichnet und durch ausschließlich weibliche Namen pseudonymisiert. In diesen Beispielen wird nach der funktionalen Rolle unterschieden, nach introvertiertem und extrovertiertem Verhalten und nach der Sitzordnung (die sich im darauffolgenden Interview bei der Abarbeitung von Aufgaben als relevant herausstellte und damit ein wichtiges strukturgebendes Element in diesem Team war).

**Hackmans Fragen** Hackman erklärt, dass es wichtig ist nach den Prozessindikatoren zu fragen, um selbstorganisierte Teams effektiver arbeiten zu lassen (siehe auch Abschnitt 2.3.2): Ist es ein Engagement-, Talent- oder Strategieproblem, das das Team behindert? Die ersten beiden Fragen nach Engagement und Talent lassen

sich auf das Modell anwenden und schaffen **KLARHEIT** über die Verantwortlichkeiten im Team. Mithilfe von Hackmans Fragen kann ein Zugang zum Modell gefunden werden.

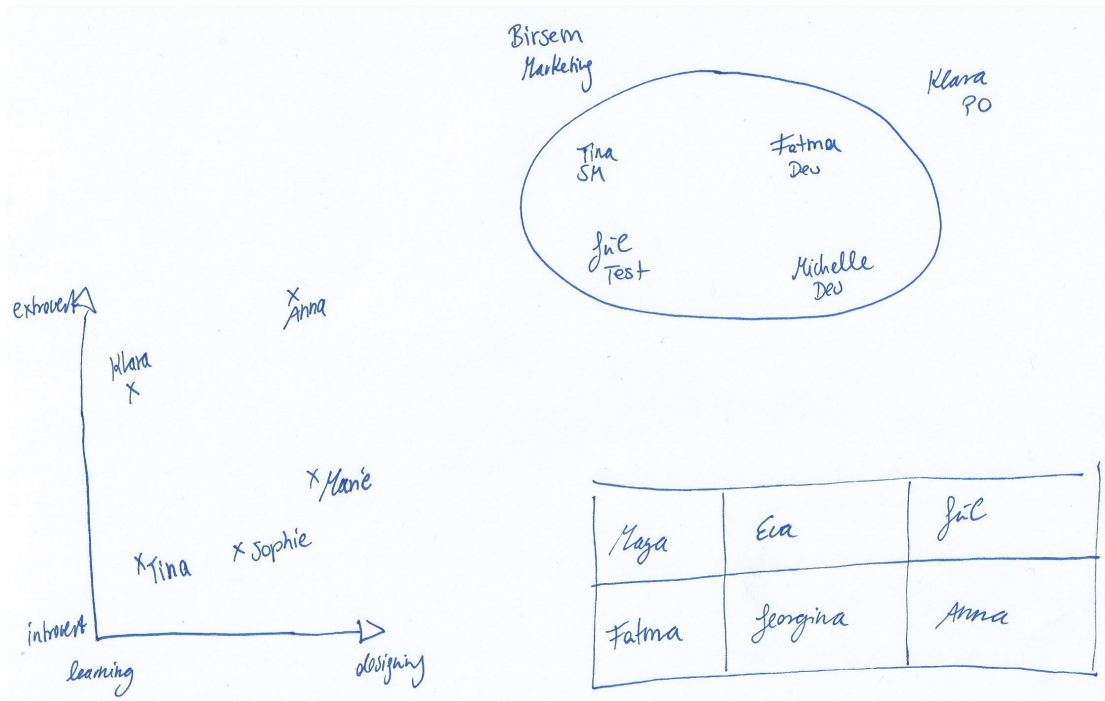


Abbildung 8.6: Anonymisierte Beispiele von Teamzeichnungen aus Interviews mit jeweils unterschiedlichen strukturgebenden Elementen: o.r. funktionale Rolle, u.l. Introvertiert/extrovertiert und learning/designing, u.r. Sitzordnung



## Kapitel 9

# Evaluierung und Grenzen des Modells

In diesem Abschnitt wird geprüft, inwieweit das Modell Evaluationskriterien standhält.

Um zu prüfen ob das Modell und die methodische Herleitung die zu ihm geführt hat, Charmazs Evaluationskriterien standhält, ist zuerst zu prüfen ob *theoretische Sättigung* erreicht wurde. Charmaz trifft die Aussage, dass die Menge der Daten nicht ausschlaggebend ist, sondern vielmehr die Qualität der Daten und die Auswertung. Als Hilfestellung stellt Charmaz stattdessen Fragen (siehe Abschnitt 4.2.5).

In Kapitel 6 zur Anwendung der Methode und insbesondere in Abschnitt 7.1 wurde in der methodischen Herleitung ausführlich und transparent dargelegt und damit Charmaz fragen ausführlich beantwortet:

- wie umfanglich Daten verglichen wurden,
- wie sie konzeptuell in einen Sinnzusammenhang gestellt wurden,
- wohin sie den Forschungsprozess geführt haben,
- welche theoretischen Kategorien entwickelt wurden,
- welche Richtungen dadurch entstanden und
- in welche Beziehungen sie gestellt wurden.

Es wurde den Daten gefolgt, die ursprüngliche Forschungsfrage verlassen und eine dem Forschungsfeld entsprechende Forschungsfrage entwickelt (siehe Kapitel 1). Daten wurden kleinteilig codiert und Codings ständig verglichen (siehe Kapitel 6). Konzepte wurden beständig entwickelt und verglichen. Es entstanden Konzepte wie **SELBSTREFLEXION** und **TEAMFUNKTIONALITÄT**, deren Sinnzusammenhang aufgedeckt werden konnte, wodurch sich die neue theoretische Kategorie **KLARHEIT GEWINNEN** entwickelte. Durch diese zentrale Kategorie konnten bisherige und neue Erkenntnisse in eine Beziehung gestellt werden. Das Modell, als Ergebnis dieser Arbeit entstand.

Nicht alle Teile des Modells haben theoretisch vollständig Sättigung erreicht. In Abschnitt 7.2 in den farblich hinterlegten Erläuterungen werden die unterschiedlichen

Sättigungsgrade genannt. Während die zentralen Konzepte **LOKALES** und **GLOBALES ROLLENVERSTÄNDNIS** durch Selbst- und Teamreflexion als gesättigt betrachtet werden können, sind einzelne Unterschritte nur teilweise gesättigt. Die Liste der Fähigkeiten ist nicht abgeschlossen. Der Kenntnisstand und sein Zusammenhang zu Selbstsicherheit ist nicht vollständig aufgeklärt. Dass ein klarer Zusammenhang zwischen Teamklarheit und Zusammenarbeit und Teambestand besteht, konnte zwar gezeigt werden, die Häufigkeit und das Ausmaß wurden aber nicht abschließend bestimmt.

Die Antworten auf Charmaz Fragen zur Sättigung werden transparent in Abschnitt 7.2 beantwortet und es kann die Annahme getroffen werden, dass theoretische Sättigung für die zentralen Elemente erreicht wurde.

Zu den weiteren Evaluationskriterien, die Charmaz vorschlägt und dazugehörige Fragen formuliert (siehe Abschnitt 4.2.9):

### **Resonanz und Nutzen**

Das Modell wurde Praktiker\_innen vorgestellt und validiert (siehe Kapitel 8). Damit geht die Arbeit über die geforderten Schritte einer GTM hinaus, um den Beleg der Resonanz und des Nutzen zu erbringen.

Die Praktiker\_innen der Abschlussvalidierung waren nicht Teil der vorherigen Forschung, nicht in die Erarbeitung des Modells eingebunden und kannten das Modell vorher nicht.

Nach einer kurzen Einführung, konnten sie das Modell auf alltägliche Situationen anwenden und ihr Verständnis für Teamprozesse erweitern. Sie waren direkt in der Lage weitere Ideen zur Implementierung in der Praxis zu sammeln. Außerdem haben sie direkt einen möglichen Nutzen für die Praxis der Anwendung des Modells assoziiert (siehe Abschnitt 8.3). Die Resonanz in der Praxis war damit ausgesprochen positiv. Zukünftige Forschung kann an diese Ergebnisse anschließen.

Es kann festgehalten werden, dass durch das methodische Vorgehen Stols u.a. [SRF16] Forderungen entsprechend transparent dargestellt wurden, ein sinnvolles Modell entwickelt wurde, das Nutzen im Alltag von Softwareentwickler\_innen entfaltet, wie Ralph es fordert [Ral18]. Das gelingt, weil das Modell komplexe, umfangreiche Zusammenhänge für Praktiker\_innen greifbar und nachvollziehbar darstellt.

### **Glaubwürdigkeit**

Durch die Datenerhebung wurde enge Vertrautheit mit dem Forschungsfeld und der Umgebung erreicht. Es wurden Interviews, Beobachtungen und Validierungen in der Praxis durchgeführt (siehe Kapitel Datenerhebung 5 und 8). In Abschnitt 7.1 und Kapitel 6 wurde dargelegt, welche systematischen Vergleiche zwischen den Beobachtungen, Daten und Kategorien hergestellt wurden. Das Modell deckt einen großen Umfang der empirischen Daten ab. Die Kategorien sind durch das Modell eng miteinander verknüpft (siehe Kapitel 7). In Abschnitt 7.2 werden die Zusammenhänge zwischen den Daten und dem Modell dargelegt. Die Lesenden können durch die hohe Transparenz des Forschungsprozesses eigenständig das Vorgehen und die Argumentationskette prüfen.

## **Originalität**

Das Modell umfasst mit **KLARHEIT GEWINNEN** ein eigenständiges und neues Konzept. Es bietet eine neue Perspektive für agile Softwareentwicklungsteams und stellt die Prozesse in einen neuen Zusammenhang. Das Phänomen **KLARHEIT GEWINNEN** wird erkannt und in das Zentrum des Modells gestellt. Selbstorganisation, die in bisherigen Arbeiten das zentrale Element war [Hod11] [MDD08], ist Ausgangspunkt und nicht Zentrum des Modells. Damit geht das Modell über bisherige Forschung hinaus, greift die bisher erkannten Hürden und Problematiken auf, setzt sie auf den Ausgangspunkt und knüpft daran an, um einen Lösungsprozess aufzuzeigen.





# Kapitel 10

## Fazit und Ausblick

In Kapitel 3 wurden folgende Forschungsfragen gestellt, die ich mit dieser Arbeit beantworten wollte:

1. Wie wäre es möglich, die bisherigen Erkenntnisse verwandter Arbeiten zu verbinden unter der besonderen Berücksichtigung menschlicher Aspekte?
2. Welche weiteren relevanten Phänomene oder Konzepte gibt es unter besonderer Berücksichtigung menschlicher Aspekte?
3. Können die Forschungserkenntnisse zu menschlichen Aspekten in agilen Softwareentwicklungsteams in einem Kontext dargelegt werden, der möglicherweise Softwareentwicklungsteams in der Praxis helfen kann, die Hürden der selbstorganisierten Zusammenarbeit zu erkennen und zu überwinden?

Wie diese drei Fragen in der Arbeit beantwortet wurden, fasse ich nachfolgend zusammen:

1. Verwandte Arbeiten erkennen ein Problem in der Vereinbarkeit von Teamzielen und individuellen Zielen, beziehen sich auf konkrete Methodiken und Praktiken im agilen Arbeitsumfeld oder beschreiben immer komplexer und detaillierter einzelne Phänomene (siehe Kapitel 2). Wie sich in der Diskussion in Kapitel 3 herausstellt, fehlt ein Modell, das menschliche Aspekte in den Mittelpunkt stellt und einen angemessenen Grad an Abstraktion schafft, um Nutzen entfalten zu können. Das Modell, das den Forschungsbeitrag dieser Arbeit darstellt, wurde durch die GTM Charmaz entwickelt. Das Modell beantwortet auch die zwei weiterführenden Fragen aus Kapitel 3 und beinhaltet eine Rollendefinition und erste Hinweise für relevante und nicht relevante Fähigkeiten einzelner Teammitglieder, um die Zusammenarbeit zu stärken.
2. Einzigartig an dem Modell ist, dass Beziehungen zwischen bestehenden und neuen Erkenntnissen erkannt und benannt wurden (siehe u.a. Abschnitt 3.1 und 7.4). Das gelingt, indem das neue Konzept **KLARHEIT GEWINNEN** als zentrales Konzept erkannt und ausdefiniert wurde. Selbstorganisation ist der Ausgangspunkt und nicht mehr das zentrale Konzept. Dadurch werden die Perspektive verändert, neue Beziehungen hergestellt und neue Eingriffsmöglichkeiten geschaffen.

3. Die Relevanz dieses einzigartigen Modells entfaltet sich durch die Validierung in der Praxis (siehe Kapitel 8). Praktiker\_innen fällt es leicht, Zugang zu dem Modell zu finden und es auf ihre Teamsituationen anzuwenden. Sie erkennen ein großes Potential, durch diese Perspektive die Zusammenarbeit zu stärken und auf neue Weise reflektieren und lenken zu können. Damit erfüllt es nicht nur Ralphs [Ral18] Forderung nach lebensrealitätsnaher, hilfreicher Forschung, sondern auch Charmaz' Evaluationskriterien (siehe Abschnitt 9). Der Nutzen der GTM wird unterstrichen und damit die konsequente, richtige und herausfordernde Umsetzung der GTM belegt. Die Arbeit zeigt damit, wie durch konsequente Anwendung der komplexen GTM nach Charmaz ein einzigartiges Modell entwickelt werden konnte, das durch seine augenscheinliche Einfachheit besticht und gleichzeitig den relevanten Kern komplexer Vorgänge in selbstorganisierten Softwareentwicklungsteams erstmalig identifiziert und beschreibt. Die augenscheinliche Einfachheit wurde durch ein hohes Maß an Abstraktion erreicht. Der Prozess der Abstraktion wurde konsequent durch GTM bestritten. Durch einen umfänglichen, zeitlich langen und von Vertrauen geprägten Zugang zu verschiedenen Teams konnten hilfreiche, ungefilterte und vielfältige Daten erhoben werden, die eine ausgeprägte und einzigartige GTM-Analyse ermöglichten. Durch eine beständige Verbindung zu den erhobenen Daten, die tief durchdrungen wurden, blieb das Modell trotz Abstraktion nah an der Praxis und für diese relevant und nicht rein theoretisch abstrakt: Ein Balanceakt der GTM, der gemeistert wurde.

Ich erhoffe mir durch die transparent und ausgesprochen detaillierte Darstellung der Anwendung der GTM nach Charmaz das Interesse von Forschenden an der Methodik Charmaz' zu fördern. Die konkrete Darstellung der Arbeitsweise und der Hürden im Forschungsprozess soll anderen Forschenden helfen, Lösungen und Ideen für ihre eigene Forschung zu finden und die praktische Umsetzung erleichtern.

Zukünftige Arbeiten können auf vielfältige Weise an meine Arbeit anschließen. In der Validierung wurden die Praxisrelevanz bestätigt und erste Diskussionen angeregt. Die nachhaltige Verankerung von Werkzeugen oder dem hier generierten Wissen war durch einmaliges Vorstellen der Ergebnisse nicht realisierbar. Ein weiterführender Schritt könnte sein zu erforschen, wie das hier entwickelte Modell in Teams zu regelmäßiger und nachhaltiger Anwendung kommen könnte. Erste Ansatzpunkte bieten dabei die vorgestellten praktischen Übungen der agilen Expert\_innen aus Abschnitt 8.4 oder Reflexionsmodelle wie von Babb u. a. [BHN14].

Die Praktiker\_innen assoziieren auch einen möglichen positiven Nutzen des Modells wie die Stärkung von Vertrauen oder psychologische Sicherheit (siehe Abschnitt 8.3). Hier anzuschließen und angemessene Forschungsmethoden zur Evaluierung zu entwickeln, wäre ein weiteres interessantes Forschungsfeld.

In der Einleitung zu Kapitel 7 erwähne ich bereits, dass die vorliegende Arbeit lediglich einen Ausschnitt der Erkenntnisse aus den gewonnenen Daten wiedergibt. Die Datenauswertung ist komplex. Das hier beschriebene Ergebnis ist nur ein kleiner Ausschnitt aus der Gesamtdatenmenge der entstandenen Konzepte. Ich habe mich entschieden, mich auf das zentrale Modell zu beschränken mit dem Ziel, das Modell möglichst detailreich und den Forschungsprozess ausführlich darzustellen.

Es gab aber noch weitere Themenpunkte, die spannend und reichhaltig gewesen wären und die nun, da das Konzept **KLARHEIT GEWINNEN** beleuchtet wurde, relevant sein könnten:

- Entscheidungen: Wie kann die Entscheidungsfindung durch **KLARHEIT** positiv beeinflusst werden?
- Wie kann die Neigung zur cross-funktionalen Zusammenarbeit durch **KLARHEIT** gefördert werden (siehe auch [BP18])?
- Welche Fähigkeiten sind im Teams besonders wichtig? Welche Rolle spielt beispielsweise Humor?

Im Modell selbst könnte validiert werden, wie groß der Zusammenhang zwischen **KLARHEIT**, **ZUSAMMENARBEIT** und **TEAMBESTAND** ist. Zum Beispiel: Welche Verhaltensweisen und fehlende **KLARHEIT** sind besonders schädlich für ein Team?

Im Zusammenhang mit der verwandten Literatur und bisherigen Forschung könnten folgende Fragen weiter verfolgt werden:

- Unter Berücksichtigung Hackmans Definition von erfolgreichen Teams (u.a. inwiefern die Teammitglieder nachhaltig gerne weiter miteinander arbeiten möchten) könnte geprüft werden, ob das Modell zum Erfolg von agilen Softwareentwicklungsteams beiträgt.
- Wie wirkt sich **KLARHEIT** auf die Motivation aus? Lässt sich das Konzept mit bestehenden Modellen zu Motivation verbinden (siehe Abschnitt 2.3.3)?
- Wie kann das Wohlbefinden von Teammitgliedern gestärkt werden (ggf. mit Hilfe des Modells)?
- Die Rollendefinition dieser Arbeit könnte dahingehend validiert werden, ob sie hilft, positive Effekte zu fördern, wie sie in anderen Forschungsarbeiten zu Rollen bereits erkannt wurden [JBP03].

Eine ganz grundsätzliche weiterführende Frage wäre, inwieweit das Modell auf selbstorganisierte Teams außerhalb der IT angewandt werden könnte.

Die Forschung mit GTM nach Charmaz ermöglicht trotz ihrer Hürden und Komplexität hilfreiche Erkenntnisse - nicht nur für agile Softwareentwicklungsteams.



# Literaturverzeichnis

- [Abr94] ABRAMIS, DAVID J.: *Work Role Ambiguity, Job Satisfaction, and Job Performance: Meta-Analyses and Review*. Psychological Reports, 75(3\_suppl):1411–1433, 1994.
- [AE14] ADZIC, GOJKO und DAVID EVANS: *Fifty quick ideas to improve your user stories*. Neuri Consulting LLP, London, 15 October 2014.
- [AFM<sup>+</sup>11] ABRAHAMSSON, PEKKA, ILENIA FRONZA, RAIMUND MOSER, JELENA VLASENKO und WITOLD PEDRYCZ: *Predicting Development Effort from User Stories*. In: *2011 International Symposium on Empirical Software Engineering and Measurement*, Seiten 400–403. IEEE, 2011.
- [AGJ09] ACUÑA, SILVIA T., MARTA GÓMEZ und NATALIA JURISTO: *How do personality, team processes and task characteristics relate to job satisfaction and software quality?* Information and Software Technology, 51(3):627–639, 2009.
- [AHA17] ANDRIYANI, YANTI, RASHINA HODA und ROBERT AMOR: *Reflection in Agile Retrospectives*. In: BAUMEISTER, HUBERT, HORST LICHTER und MATTHIAS RIEBISCH (Herausgeber): *Agile Processes in Software Engineering and Extreme Programming*, Seiten 3–19. Springer International Publishing, Cham, 2017.
- [AHS<sup>+</sup>05] ABDULLAH, NIK NAILAH BINTI, SHINICHI HONIDEN, HELEN SHARP, BASHAR NUSEIBEH und DAVID NOTKIN: *Communication patterns of agile requirements engineering*. In: *Proceedings of the 2005 workshop on Human and social factors of software engineering*, Seiten 1–4, New York, NY, 2005. ACM.
- [All03] ALLAN, GEORGE: *A critique of using grounded theory as a research method*. Electronic Journal of Business Research Methods, 2(1):1–10, 2003.
- [And14] ANDRESEN, JUDITH: *Retrospektiven in agilen Projekten: Ablauf, Regeln und Methodenbausteine*. Hanser, München, 2014.
- [Bad13] BADREDDIN, OMAR: *Thematic Review and Analysis of Grounded Theory Application in Software Engineering*. Advances in Software Engineering, 2013(3):1–9, 2013.

- [Bar18] BARKE, HELENA: *How many story points for diversity?* In: MARSDEN, NICOLA, VOLKER WULF, JENNIFER RODE und ANNE WEIBERT (Herausgeber): *Proceedings of the 4th Conference on Gender & IT - GenderIT '18*, Seiten 221–223, New York, New York, USA, 2018. ACM Press.
- [BB10] BUNDERSON, J. STUART und PETER BOUMGARDEN: *Structure and Learning in Self-Managed Teams: Why “Bureaucratic” Teams Can Be Better Learners*. *Organization Science*, 21(3):609–624, 2010.
- [BBH<sup>+</sup>08] BEECHAM, SARAH, NATHAN BADDOO, TRACY HALL, HUGH ROBINSON und HELEN SHARP: *Motivation in Software Engineering: A systematic literature review*. *Information and Software Technology*, 50(9-10):860–878, 2008.
- [BBv<sup>+</sup>01] BECK, KENT, MIKE BEEDLE, ARIE VAN BENNEKUM, ALISTAIR COCKBURN, WARD CUNNINGHAM, MARTIN FOWLER, JAMES GRENNING, JIM HIGHSMITH, ANDREW HUNT, RON JEFFRIES, JON KERN, BRIAN MARICK, ROBERT C. MARTIN, STEVE MELLOR, KEN SCHWABER, JEFF SUTHERLAND und DAVE THOMAS: *Manifesto for Agile Software Development*, 2001.
- [BC07] BRYANT, ANTONY und KATHY CHARMAZ (Herausgeber): *The SAGE handbook of grounded theory*. SAGE, Los Angeles, 2007.
- [BGGM14] BERGMANN, RALPH, SARAH GESSINGER, SEBASTIAN GÖRG und GILBERT MÜLLER: *The Collaborative Agile Knowledge Engine CAKE*. In: GOGGINS, SEAN, ISA JAHNKE, DAVID W. McDONALD und PERNILLE BJØRN (Herausgeber): *Group '14*, Seiten 281–284, New York, NY, 2014. Association for Computing Machinery.
- [BHN14] BABB, JEFFREY, RASHINA HODA und JACOB NORBJERG: *Embedding Reflection and Learning into Agile Software Development*. *IEEE Software*, 31(4):51–57, 2014.
- [Bib18] BIBIK, ILYA: *How to Kill the Scrum Monster: Quick Start to Agile Scrum Methodology and the Scrum Master Role*. Springer, 2018.
- [BP18] BARKE, HELENA und LUTZ PRECHELT: *Some reasons why actual cross-fertilization in cross-functional agile teams is difficult*. In: SHARP, HELEN, CLEIDSON R. B. DE SOUZA, DANIEL GRAZIOTIN, MEIRA LEVY und DAVID SOCHA (Herausgeber): *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '18*, Seiten 97–103, New York, New York, USA, 2018. ACM Press.
- [BP19] BARKE, HELENA und LUTZ PRECHELT: *Role clarity deficiencies can wreck agile teams*. *PeerJ Computer Science*, 5:e241:1–20, 2019.
- [BSFK16] BARKE, HELENA, JULIANE SIEGERIS, JÖRN FREIHEIT und DAGMAR KREFTING (Herausgeber): *Gender und IT-Projekte: Neue Wege zu digitaler Teilhabe*. Budrich UniPress Ltd, Opladen and Berlin and Toronto, 2016.

- [BSN14] BARKE, HELENA, JULIANE SIEGERIS und STEFANIE NORDMANN: *Gender all around! A practical and holistic approach towards recruiting and retaining women in the field of IT*. In: WULF, VOLKER (Herausgeber): *GenderIT 2014*. ACM Digital Library and European Society for Socially Embedded Technologies, New York, NY and Siegen, 2014.
- [BWM03] BRADLEY, JOHN, BARBARA JO WHITE und BRIAN E. MENNECKE: *Teams and Tasks: A Temporal Framework for the Effects of Interpersonal Interventions on Team Performance*. *Small Group Research*, 34(3):353–387, 2003.
- [CdC15] CRUZ, SHIRLEY, FABIO Q.B. DA SILVA und LUIZ FERNANDO CAPRETZ: *Forty years of research on personality in software engineering: A mapping study*. *Computers in Human Behavior*, 46:94–113, 2015.
- [Cha11] CHARMAZ, KATHY C.: *Den Standpunkt verändern: Methoden der konstruktivistischen Grounded Theory*. In: MEY, GÜNTER und KATJA MRUCK (Herausgeber): *Grounded Theory Reader*. Springer Fachmedien, Wiesbaden, 2011.
- [Cha14] CHARMAZ, KATHY: *Constructing grounded theory*. Introducing qualitative methods. SAGE, Los Angeles and London and New Delhi and Singapore and Washington DC, 2nd edition Auflage, 2014.
- [CMD16] CRUZES, DANIELA S., NILS B. MOE und TORE DYBA: *Communication between Developers and Testers in Distributed Continuous Agile Testing*. In: *11th IEEE International Conference on Global Software Engineering*, Seiten 59–68, Piscataway, NJ, 2016. IEEE.
- [Coh12] COHN, MIKE: *Agile estimating and planning*. Robert C. Martin series. Prentice Hall PTR, Upper Saddle River, NJ, 12. print Auflage, 2012.
- [Coh13] COHN, MIKE: *User stories applied: For agile software development*. Addison-Wesley signature series. Addison-Wesley, Boston, Mass., 18. print Auflage, 2013.
- [CSMU13] CHAMBERLAIN-SALAUN, JENNIFER, JANE MILLS und KIM USHER: *Linking Symbolic Interactionism and Grounded Theory Methods in a Research Design*. *SAGE Open*, 3(3):215824401350575, 2013.
- [CVR15] CAPRETZ, LUIZ FERNANDO, DANIEL VARONA und ARIF RAZA: *Influence of personality types in software tasks choices*. *Computers in Human Behavior*, 52:373–378, 2015.
- [DCP12] DRURY, MEGHANN, KIERAN CONBOY und KEN POWER: *Obstacles to decision making in Agile software development teams*. *Journal of Systems and Software*, 85(6):1239–1254, 2012.
- [DD08] DYBÅ, TORE und TORGEIR DINGSØYR: *Empirical studies of agile software development: A systematic review*. *Information and Software Technology*, 50(9):833–859, 2008.

- [DD12] DYBÅ, TORE und TORGEIR DINGSØYR: *Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies*. In: CHARP, HELEN (Herausgeber): *2012 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, Seiten 27–29, Piscataway, NJ, 2012. IEEE.
- [dF12] DA SILVA, FABIO Q.B. und A. CÉSAR C. FRANÇA: *Towards understanding the underlying structure of motivational factors for software engineers to guide the definition of motivational programs*. *Journal of Systems and Software*, 85(2):216–226, 2012.
- [DG04] DOUGLAS, CEASAR und WILLIAM L. GARDNER: *Transition to self-directed work teams: implications of transition time and self-monitoring for managers' use of influence tactics*. *Journal of Organizational Behavior*, 25(1):47–65, 2004.
- [DG18] DÖNMEZ, DENNIZ und GUDELA GROTE: *Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities*. *Information and Software Technology*, 93:94–111, 2018.
- [dJHD<sup>+</sup>13] JOHNSON, MICHAEL D., JOHN R. HOLLENBECK, D. SCOTT DERUE, CHRISTOPHER M. BARNES und DUSTIN JUNDT: *Functional versus dysfunctional team change: Problem diagnosis and structural feedback for self-managed teams*. *Organizational Behavior and Human Decision Processes*, 122(1):1–11, 2013.
- [DL12] DERBY, ESTHER und DIANA LARSEN: *Agile retrospectives: Making good teams great*. *The pragmatic programmers*. The Pragmatic Bookshelf, Dallas, Texas and Raleigh, North Carolina, Bookversion: P7.0 Auflage, August 2012.
- [DL13] DEMARCO, TOM und TIMOTHY R. LISTER: *Peopleware: Productive projects and teams*. Addison-Wesley, Upper Saddle River, NJ, 3rd ed. Auflage, 2013.
- [DMG14] DYBÅ, TORE, NEIL MAIDEN und ROBERT GLASS: *The Reflective Software Engineer: Reflective Practice*. *IEEE Software*, 31(4):32–36, 2014.
- [Fau00] FAULKNER, WENDY: *The Power and the Pleasure? A Research Agenda for "Making Gender Stick" to Engineers*. *Science, Technology, & Human Values*, 25(1):87–119, 2000.
- [Flo92] FLOYD, CHRISTIANE: *Software Development as Reality Construction*. In: FLOYD, CHRISTIANE, HEINZ ZÜLLIGHOVEN, REINHARD BUDDE und REINHARD KEIL-SLAWIK (Herausgeber): *Software Development and Reality Construction*, Seiten 86–100. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [FS00] FARAJ, SAMER und LEE SPROULL: *Coordinating Expertise in Software Development Teams*. *Management Science*, 46(12):1554–1568, 2000.



- [FW04] FAULSTICH-WIELAND, HANNELORE: *Doing Gender*. In: GLASER, EDITH, DORLE KLIKA und ANNEDORE PRENGEL (Herausgeber): *Handbuch Gender und Erziehungswissenschaften*, Seiten 175–191. Klinkhardt, Bad Heilbrunn / Obb., 2004.
- [GGGM19] GUTIERREZ, GEMA, JAVIER GARZAS, MARIA TERESA GONZALEZ DE LENA und JAVIER M. MOGUERZA: *Self-Managing: An Empirical Study of the Practice in Agile Teams*. IEEE Software, 36(1):23–27, 2019.
- [GH15] GOLL, JOACHIM und DANIEL HOMMEL: *Mit Scrum zum gewünschten System*. Springer Vieweg, Wiesbaden, 2015.
- [Gla78] GLASER, BARNEY G.: *Theoretical sensitivity*. Advances in the methodology of grounded theory. Soc. Pr, Mill Valley, Calif., 1978.
- [Gla92] GLASER, BARNEY G.: *Basics of grounded theory analysis: Emergence vs forcing*. Sociology Press, Mill Valley, Calif., 2. print Auflage, 1992.
- [Gla05] GLASER, BARNEY G.: *The grounded theory perspective III: Theoretical coding*. Sociology Press, Mill Valley, Calif., 2005.
- [Gla12] GLASER, BARNEY G.: *Constructivist Grounded Theory?* Grounded Theory Review: An International Journal, 11(1), 2012.
- [Glo13] GLOGER, BORIS: *Scrum: Produkte zuverlässig und schnell entwickeln*. Carl Hanser Verlag GmbH & Co. KG, 2013.
- [GMD11] GULLIKSEN STRAY, VIKTORIA, NILS BREDE MOE und TORGEIR DINGSØYR: *Challenges to Teamwork: A Multiple Case Study of Two Agile Teams*. In: SILLITTI, ALBERTO, ORIT HAZZAN, EMILY BACHE und XAVIER ALBALADEJO (Herausgeber): *Agile Processes in Software Engineering and Extreme Programming*, Band 77 der Reihe *Lecture Notes in Business Information Processing*, Seiten 146–161. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [GR14] GLOGER, BORIS und DIETER RÖSNER: *Selbstorganisation braucht Führung: Die einfachen Geheimnisse agilen Managements*. Hanser, München, 2014.
- [GS67] GLASER, BARNEY G. und ANSELM STRAUSS: *The discovery of grounded theory: Strategies for qualitative research*. Aldine de Gruyter, New York, 1967.
- [GWA15] GRAZIOTIN, DANIEL, XIAOFENG WANG und PEKKA ABRAHAMSSON: *How do you feel, developer? An explanatory theory of the impact of affects on programming performance*. PeerJ Computer Science, 1(1):e18, 2015.
- [Hac80] HACKMAN, J. RICHARD: *Work redesign and motivation*. Professional Psychology, 11(3):445–455, 1980.
- [Hac86] HACKMAN, J. RICHARD: *The psychology of self-management in organizations*. In: PALLAK, MICHAEL S. und ROBERT O. PERLOFF (Herausgeber):

- Psychology and work: Productivity, change, and employment*, Seiten 89–136. American Psychological Association, Washington, 1986.
- [Hac90] HACKMAN, J. RICHARD (Herausgeber): *Groups that work (and those that don't): Creating conditions for effective teamwork*. The Jossey-Bass management series. Jossey-Bass Publ, San Francisco, 1. ed., 9. printing Auflage, 1990.
- [Hac03] HACKMAN, J. RICHARD: *Learning more by crossing levels: Evidence from airplanes, hospitals, and orchestras*. *Journal of Organizational Behavior*, 24(8):905–922, 2003.
- [Hei06] HEINTEL, PETER (Herausgeber): *Betrifft: TEAM: Dynamische Prozesse in Gruppen*, Band 4 der Reihe *Schriften zur Gruppen- und Organisationsdynamik*. VS Verlag für Sozialwissenschaften | GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 1. Auflage, 2006.
- [HG01] HOEGL, MARTIN und HANS GEORG GEMUENDEN: *Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence*. *Organization Science*, 12(4):435–449, 2001.
- [HK15] HEINTEL, PETER und EWALD E. KRAINZ: *Projektmanagement: Hierarchiekrise, Systemabwehr, Komplexitätsbewältigung*. Springer Gabler, Wiesbaden, 6. Aufl. Auflage, 2015.
- [HN17] HODA, RASHINA und JAMES NOBLE: *Becoming Agile: A Grounded Theory of Agile Transitions in Practice*. In: UCHITEL, SEBASTIAN (Herausgeber): *Proceedings of the 39th International Conference on Software Engineering, ICSE '17*, Seiten 141–151, Piscataway, NJ, USA, 2017. IEEE Press.
- [HNM10a] HODA, RASHINA, JAMES NOBLE und STUART MARSHALL: *Balancing acts*. In: DITTRICH, YVONNE, CLEIDSON DE SOUZA, MIKKO KORPELA, HELEN SHARP, JANICE SINGER und HEIKE WINSHIERS-THEOPHILUS (Herausgeber): *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '10*, Seiten 5–12, New York, New York, USA, 2010. ACM Press.
- [HNM10b] HODA, RASHINA, JAMES NOBLE und STUART MARSHALL: *Using grounded theory to study the human aspects of software engineering*. In: *Human Aspects of Software Engineering on - HAoSE '10*, Seite 1, New York, New York, USA, 2010. ACM Press.
- [HNM11] HODA, RASHINA, JAMES NOBLE und STUART MARSHALL: *The impact of inadequate customer collaboration on self-organizing Agile teams*. *Information and Software Technology*, 53(5):521–534, 2011.
- [HNM12] HODA, RASHINA, JAMES NOBLE und STUART MARSHALL: *Developing a grounded theory to explain the practices of self-organizing Agile teams*. *Empirical Software Engineering*, 17(6):609–639, 2012.

- [HNM13] HODA, RASHINA, JAMES NOBLE und STUART MARSHALL: *Self-Organizing Roles on Agile Software Development Teams*. IEEE Transactions on Software Engineering, 39(3):422–444, 2013.
- [Hod11] HODA, RASHINA: *Self-Organizing Agile Teams: A Grounded Theory*. Doktorarbeit, Victoria University of Wellington, 2011.
- [HRH13a] HUMMEL, MARKUS, CHRISTOPH ROSENKRANZ und ROLAND HOLTEN: *Die Bedeutung von Kommunikation bei der agilen Systementwicklung*. WIRTSCHAFTSINFORMATIK, 55(5):347–360, 2013.
- [HRH13b] HUMMEL, MARKUS, CHRISTOPH ROSENKRANZ und ROLAND HOLTEN: *The Role of Communication in Agile Systems Development*. Business & Information Systems Engineering, 5(5):343–355, 2013.
- [HSGT17] HODA, RASHINA, NORSAREMAH SALLEH, JOHN GRUNDY und HUI MIEN TEE: *Systematic literature reviews in agile software development: A tertiary study*. Information and Software Technology, 85(Supplement C):60–70, 2017.
- [HWAB17] HACKL, BENEDIKT, MARC WAGNER, LARS ATTMER und DOMINIK BAUMANN: *New Work: Auf dem Weg zur neuen Arbeitswelt: Management-Impulse, Praxisbeispiele, Studien*. Springer Gabler, Wiesbaden, 2017.
- [JBP03] JOHNSON, JEFFREY C., JAMES S. BOSTER und LAWRENCE A. PALINKAS: *Social roles and the evolution of networks in extreme and isolated environments*. The Journal of mathematical sociology, 27(2-3):89–121, 2003.
- [JJ71] JOHN M. ROBERTS und JAMES O. WICKE: *Flying and expressive self-testing: An exploratory consideration*. Naval War College Review, 23(5):67–80, 1971.
- [JQJ72] JOHN M. ROBERTS, QUENTIN S. MEEKER und JAMES C. ALLER: *Action styles and management game performance: An exploratory consideration*. Naval War College Review, 24(10):65–81, 1972.
- [JTG80] JOHN M. ROBERTS, THOMAS V. GOLDRER und GARRY E. CHICK: *Judgment, Oversight, and Skill: A Cultural Analysis of P-3 Pilot Error*. Human Organization, (39), 1980.
- [JZA<sup>+</sup>13] JAVDANI GANDOMANI, TAGHI, HAZURA ZULZALIL, ABDUL AZIM ABDUL GHANI, ABU BAKAR MD SULTAN und KHAIRONI SHARIF: *How Grounded Theory can facilitate research studies in context of Agile software development*. Science International (Lahore), Special Issue - Agile Symposium, Malaysia, 25(4):1131–1136, 2013.
- [JZA<sup>+</sup>14] JAVDANI GANDOMANI, TAGHI, HAZURA ZULZALIL, ABDUL AZIM ABDUL GHANI, ABU BAKAR MD. SULTAN und KHAIRONI YATIM SHAIRF: *Exploring Facilitators of Transition and Adoption to Agile Methods: A Grounded Theory Study*. Journal of Software, 9(7), 2014.

- [KA03] KAHKONEN, TUOMO und PEKKA ABRAHAMSSON: *Digging into the fundamentals of extreme programming building the theoretical base for agile methods*. In: HOFER, CHRISTIAN, GERHARD CHROUST und IVICA CRNKOVIC (Herausgeber): *Proceedings 29th Euromicro Conference*, Seiten 273–280, Los Alamitos, Calif, 2003. IEEE Computer Society.
- [KC69] KLEIN, MICHAEL und G. CHRISTIANSEN: *Group composition, group structure, and group effectiveness of basketball team*. In: JOHN W. LOY und GERALD S. KENYON (Herausgeber): *Sport, culture, and society : a reader on the sociology of sport*, Seiten 397–408. Macmillan, London, 1969.
- [KDS<sup>+</sup>09] KLEIN, CAMERON, DEBORAH DIAZGRANADOS, EDUARDO SALAS, HUY LE, C. SHAWN BURKE, REBECCA LYONS und GERALD F. GOODWIN: *Does Team Building Work?* *Small Group Research*, 40(2):181–222, 2009.
- [KIK<sup>+</sup>10] KARHATSU, HENRI, MARKO IKONEN, PETRI KETTUNEN, FABIAN FAGERHOLM und PEKKA ABRAHAMSSON: *Building blocks for self-organizing software development teams a framework model and empirical pilot study*. In: KETTANI, HOUSSAIN (Herausgeber): *2nd International Conference on Software Technology and Engineering (ICSTE), 2010*, Piscataway, NJ, 2010. IEEE.
- [KMAB18] KROPP, MARTIN, ANDREAS MEIER, CRAIG ANSLOW und ROBERT BIDDLE: *Satisfaction, Practices, and Influences in Agile Software Development*. In: RAINER, AUSTEN, STEPHEN G. MACDONELL und JACKY KEUNG (Herausgeber): *EASE 2018*, ICPS: ACM international conference proceeding series, Seiten 112–121, New York, New York, 2018. The Association for Computing Machinery.
- [Kom17] KOMUS, AYELT: *Studie Status Quo Agile - Verbreitung und Nutzen agiler Methoden*, 2017.
- [Kön06] KÖNIGSWIESER, ROSWITA: *Reflexion als Sprungbrett*. In: HEINTEL, PETER (Herausgeber): *Betrifft: TEAM*, Schriften zur Gruppen- und Organisationsdynamik, Seiten 69–79. VS Verlag für Sozialwissenschaften | GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2006.
- [Kra06] KRAINZ, EWALD E.: *Gruppendynamik als Wissenschaft*. In: HEINTEL, PETER (Herausgeber): *Betrifft: TEAM*, Schriften zur Gruppen- und Organisationsdynamik, Seiten 7–28. VS Verlag für Sozialwissenschaften | GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2006.
- [KWF13] KÖNIGSWIESER, ROSWITA, RUDOLF WIMMER und SIMON FRITZ: *Back to the roots? Die neue Aktualität der ("systemischen") Gruppendynamik*. *Organisationsentwicklung : Zeitschrift für Unternehmensentwicklung und Change Management*, 32(1):65–73, 2013.
- [Lan00] LANGFRED, CLAUS W.: *The paradox of self-management: individual and group autonomy in work groups*. *Journal of Organizational Behavior*, 21(5):563–585, 2000.

- [LDvB16] LUCASSEN, GARM, FABIANO DALPIAZ, JAN MARTIJN E. M. VAN DER WERF und SJAACK BRINKKEMPER: *The Use and Effectiveness of User Stories in Practice*. In: DANEVA, MAYA und OSCAR PASTOR (Herausgeber): *Requirements Engineering: Foundation for Software Quality*, Band 9619 der Reihe *Lecture Notes in Computer Science*, Seiten 205–222. Springer International Publishing, Cham, 2016.
- [LdWRJ05] LEACH, DESMOND J., TOBY D. WALL, STEVEN G. ROGELBERG und PAUL R. JACKSON: *Team Autonomy, Performance, and Member Job Strain: Uncovering the Teamwork KSA Link*. *Applied Psychology*, 54(1):1–24, 2005.
- [LF18] LENBERG, PER und ROBERT FELDT: *Psychological Safety and Norm Clarity in Software Engineering Teams*. In: SHARP, HELEN, CLEIDSON R. B. DE SOUZA, DANIEL GRAZIOTIN, MEIRA LEVY und DAVID SOCHA (Herausgeber): *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '18*, New York, New York, USA, 2018. ACM Press.
- [LFW15] LENBERG, PER, ROBERT FELDT und LARS GÖRAN WALLGREN: *Behavioral software engineering: A definition and systematic literature review*. *Journal of Systems and Software*, 107:15–37, 2015.
- [Lin13] LIN, JUN: *Context-aware task allocation for distributed agile team*. In: DENNEY, EWEN (Herausgeber): *IEEE/ACM 28th International Conference on Automated Software Engineering (ASE), 2013*, Seiten 758–761, Piscataway, NJ, 2013. IEEE.
- [LL13] LUDEWIG, JOCHEN und HORST LICHTER: *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. dpunkt.verl., Heidelberg, 3., korrigierte Aufl. Auflage, 2013.
- [MAD12] MOE, NILS BREDE, AYBÜKE AURUM und TORE DYBÅ: *Challenges of shared decision-making: A multiple case study of agile software development*. *Information and Software Technology*, 54(8):853–865, 2012.
- [Mas10] MASON, MARK: *Sample Size and Saturation in PhD Studies Using Qualitative Interviews*. *Forum: Qualitative Social Research*, 11(3), 2010.
- [Max18] MAXIMINI, DOMINIK: *The Scrum Culture: Introducing Agile Methods in Organizations*. *Management for Professionals*. Springer International Publishing, Cham, 2nd ed. 2018 Auflage, 2018.
- [MB09] MCAVOY, JOHN und TOM BUTLER: *A Failure to Learn in a Software Development Team: The Unsuccessful Introduction of an Agile Method*. In: BARRY, CHRIS (Herausgeber): *Information systems development*, Band 21, Seiten 1–13. Springer, New York, NY, 2009.
- [McK16] MCKENNA, DAVE: *The art of scrum: How scrum masters bind dev teams and unleash agility*. Apress and CA Press, New York and Aliquippa, Pennsylvania, 2016.

- [MD08] MOE, NILS BREDE und TORGEIR DINGSØYR: *Scrum and Team Effectiveness: Theory and Practice*. In: ABRAHAMSSON, PEKKA, RICHARD BASKERVILLE, KIERAN CONBOY, BRIAN FITZGERALD, LORRAINE MORGAN und XIAOFENG WANG (Herausgeber): *Agile Processes in Software Engineering and Extreme Programming*, Band 9 der Reihe *Lecture Notes in Business Information Processing*, Seiten 11–20. Springer, Berlin and Heidelberg, 2008.
- [MDD08] MOE, NILS BREDE, TORGEIR DINGSØYR und TORE DYBÅ: *Understanding Self-Organizing Teams in Agile Software Development*. In: HUSSAIN, FAROOKH KHADEER (Herausgeber): *19th Australian Conference on Software Engineering, 2008*, Seiten 76–85, Los Alamitos, Calif., 2008. IEEE Computer Soc.
- [MDD09] MOE, NILS BREDE, TORGEIR DINGSØYR und TORE DYBÅ: *Overcoming Barriers to Self-Management in Software Teams*. *IEEE Software*, 26(6):20–26, 2009.
- [MDD10] MOE, NILS BREDE, TORGEIR DINGSØYR und TORE DYBÅ: *A teamwork model for understanding an agile team: A case study of a Scrum project*. *TAICPART 2008*, 52(5):480–491, 2010.
- [MDR09] MOE, NILS BREDE, TORGEIR DINGSØYR und EMIL A. RØYRVIK: *Putting Agile Teamwork to the Test – An Preliminary Instrument for Empirically Assessing and Improving Agile Software Development*. In: AALST, WILL, PEKKA ABRAHAMSSON, MICHELE MARCHESI, FRANK MAURER, JOHN MYLOPOULOS, NORMAN M. SADEH, MICHAEL J. SHAW und CLEMENS SZYPERSKI (Herausgeber): *Agile Processes in Software Engineering and Extreme Programming*, Band 31 der Reihe *Lecture Notes in Business Information Processing*, Seiten 114–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [Mey14] MEYER, BERTRAND: *Agile! The good, the hype and the ugly*. Springer, Switzerland, 2014.
- [MF016] MOE, NILS BREDE, TOR ERLEND FAEGRI, DANIELA S. CRUZES und JAN EDVARD FAUGSTAD: *Enabling Knowledge Sharing in Agile Virtual Teams*. In: *11th IEEE International Conference on Global Software Engineering*, Seiten 29–33, Piscataway, NJ, 2016. IEEE.
- [MM10] MEY, GÜNTER und KATJA MRUCK: *Grounded-Theory-Methodologie*. In: MEY, GÜNTER und KATJA MRUCK (Herausgeber): *Handbuch Qualitative Forschung in der Psychologie*, Seiten 614–626. VS Verlag für Sozialwissenschaften, Wiesbaden, 2010.
- [MM11] MEY, GÜNTER und KATJA MRUCK (Herausgeber): *Grounded Theory Reader*. Springer Fachmedien, Wiesbaden, 2011.
- [MMRG08] MATHIEU, JOHN, M. TRAVIS MAYNARD, TAMMY RAPP und LUCY GILSON: *Team Effectiveness 1997-2007: A Review of Recent Advancements and a Glimpse Into the Future*. *Journal of Management*, 34(3):410–476, 2008.

- [MP18] MAGPILI, NINA CRISTINA und PILAR PAZOS: *Self-Managing Team Performance: A Systematic Review of Multilevel Input Factors*. *Small Group Research*, 49(1):3–33, 2018.
- [Nya19] NYAMSI, ERIC A.: *Projektmanagement mit Scrum*. Springer Fachmedien Wiesbaden, Wiesbaden, 2019.
- [O’C10] O’CONNOR, CHRISTOPHER P.: *Better planning via tasking as a team*. In: COOK, WILLIAM R. (Herausgeber): *SPLASH’10*, Seite 305, New York, NY, 2010. ACM Press.
- [PEH15] PATTON, JEFF, PETER ECONOMY und PETRA HILDEBRANDT: *User story mapping: Die Technik für besseres Nutzerverständnis in der agilen Produktentwicklung*. O’Reilly, Beijing, Dt. Ausg., 1. Aufl. Auflage, 2015.
- [PFL16] PRIOR, JULIA, SAMUEL FERGUSON und JOHN LEANEY: *Reflection is hard*. In: UNKNOWN (Herausgeber): *Proceedings of the Australasian Computer Science Week Multiconference on - ACSW ’16*, Seiten 1–8, New York, New York, USA, 2016. ACM Press.
- [PHO<sup>+</sup>08] PIKKARAINEN, MINNA, JUKKA HAIKARA, SALO OUTI, PEKKA ABRAHAMSSON und JARI STILL: *The impact of agile practices on communication in software development*. *Empirical Software Engineering*, 13(3):303–337, 2008.
- [Ral18] RALPH, PAUL: *The two paradigms of software development research*. *Science of Computer Programming*, 156:68–89, 2018.
- [RPSG<sup>+</sup>17] REITER-PALMON, RONI, TANMAY SINHA, JOSETTE GEVERS, JEAN-MARC ODOBEZ und GUALTIERO VOLPE: *Theories and Models of Teams and Groups*. *Small Group Research*, 48(5):544–567, 2017.
- [RVMQNR18] RAMOS-VILLAGRASA, PEDRO J., PEDRO MARQUES-QUINTEIRO, JOSÉ NAVARRO und RAMÓN RICO: *Teams as Complex Adaptive Systems: Reviewing 17 Years of Research*. *Small Group Research*, 49(2):135–176, 2018.
- [Sal13] SALINGER, STEPHAN: *Ein Rahmenwerk für die qualitative Analyse der Paarprogrammierung*. Doktorarbeit, Freie Universität Berlin, 2013.
- [SB19] SIEGERIS, JULIANE und HELENA BARKE: *Agile for agile-new ideas for the transformation of student projects*. In: LINSSEN, OLIVER, MARTIN MIKUSZ, ALEXANDER VOLLAND, ENES YIGITBAS, MARTIN ENGSTLER, MASUD FAZAL-BAQAIE und MARCO KUHRMANN (Herausgeber): *GI-Edition Lecture Notes in Informatics*, Band P-298, Seiten 151–164, 2019.
- [SBB<sup>+</sup>09] SHARP, HELEN, NATHAN BADDOO, SARAH BEECHAM, TRACY HALL und HUGH ROBINSON: *Models of motivation in software engineering*. *Information and Software Technology*, 51(1):219–233, 2009.
- [SC94] STRAUSS, ANSELM und JULIET CORBIN: *Grounded Theory Methodology: An Overview*. In: DENZIN, N. und Y. LINCOLN (Herausgeber): *Handbook of Qualitative Research*. SAGE, 1994.

- [SC96] STRAUSS, ANSELM und JULIET CORBIN: *Grundlagen Qualitativer Sozialforschung*. Beltz and Psychologie Verlags Union, Weinheim, 1996.
- [SC98] STRAUSS, ANSELM L. und JULIET M. CORBIN: *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publ, Thousand Oaks, 2. ed. Auflage, 1998.
- [SC10] STRAUSS, ANSELM und JULIET CORBIN: *Grounded theory: Grundlagen qualitativer Sozialforschung*. Beltz, Weinheim, Unveränd. Nachdr. der letzten Aufl. Auflage, 2010.
- [Sch83] SCHÖN, DONALD A.: *The reflective practitioner: How professionals think in action*. Basic Book, New York, 1983.
- [Sch10] SCHÖSSLER, FRANZISKA: *Einführung in die Gender Studies*. Akademie Studienbücher Literaturwissenschaft. Akademie-Verlag and DE GRUYTER, Berlin, 2010.
- [Sla55] SLATER, PHILIP E.: *Role Differentiation in Small Groups*. American Sociological Review, 20(3):300, 1955.
- [Smi66] SMITH, W. M.: *Observations over the lifetime of a small isolated group: structure, danger, boredom, and vision*. Psychological Reports, 19(2):475–514, 1966.
- [SRF16] STOL, KLAAS-JAN, PAUL RALPH und BRIAN FITZGERALD: *Grounded theory in software engineering research*. In: DILLON, LAURA, WILLEM VISSER und LAURIE WILLIAMS (Herausgeber): *ICSE'16*, Seiten 120–131, [New York], 2016. ACM, Association for Computing Machinery.
- [SRMD99] SALAS, EDUARDO, DREW ROZELL, BRIAN MULLEN und JAMES E. DRISKELL: *The Effect of Team Building on Performance*. Small Group Research, 30(3):309–329, 1999.
- [SS17a] SCHWABER, KEN und JEFF SUTHERLAND: *Der Scrum Guide: Der gültige Leitfaden für Scrum: Die Spielregeln*, 2017.
- [SS17b] SUTHERLAND, JEFF und KEN SCHWABER: *Der Scrum Guide: Der gültige Leitfaden für Scrum: Die Spielregeln*, 2017.
- [SSB18] SIEGERIS, JULIANE, RUTH STEINSEIFER und HELENA BARKE: *Agile redesign of student projects in a women-only degree course*. In: MARSDEN, NICOLA, VOLKER WULF, JENNIFER RODE und ANNE WEIBERT (Herausgeber): *Proceedings of the 4th Conference on Gender & IT - GenderIT '18*, Seiten 19–28, New York, New York, USA, 2018. ACM Press.
- [Ste06] STEWART, GREG L.: *A Meta-Analytic Review of Relationships Between Team Design Features and Team Performance*. Journal of Management, 32(1):29–55, 2006.



- [TPPD02] THOMS, PEG, JEFFREY K. PINTO, DIANE H. PARENTE und VANESSA URCH DRUSKAT: *Adaptation to Self-Managing Work Teams*. *Small Group Research*, 33(1):3–31, 2002.
- [Uko06] UKOWITZ, MARTINA: *Roswita Königswieser & Martin Hillebrand (2004). Einführung in die systemische Organisationsberatung*. *Gruppendynamik und Organisationsberatung*, 37(2):241–247, 2006.
- [UMWB14] USMAN, MUHAMMAD, EMILIA MENDES, FRANCILA WEIDT und RICARDO BRITTO: *Effort estimation in agile software development*. In: WAGNER, STEFAN und MASSIMILIANO DI PENTA (Herausgeber): *PROMISE 2014*, ACM International Conference Proceedings Series, Seiten 82–91, New York, 2014. ACM.
- [vBK10] VAN DER VEGT, GERBEN S., STUART BUNDERSON und BEN KUIPERS: *Why Turnover Matters in Self-Managing Work Teams: Learning, Social Integration, and Task Flexibility*. *Journal of Management*, 36(5):1168–1191, 2010.
- [VSM14] VIGENSCHOW, UWE, BJÖRN SCHNEIDER und INES MEYROSE: *Soft Skills für Softwareentwickler: Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle*. dpunkt, s.l., 3. Aufl. Auflage, 2014.
- [WB07] WHITWORTH, ELIZABETH und ROBERT BIDDLE: *The Social Nature of Agile Teams*. In: ECKSTEIN, JUTTA (Herausgeber): *Agile 2007*, Seiten 26–36, Los Alamitos, Calif., 2007. IEEE Computer Soc.
- [WBW91] WELLINS, RICHARD S., WILLIAM C. BYHAM und JEANNE M. WILSON: *Empowered teams: Creating self-directed work groups that improve quality, productivity, and participation*. The Jossey-Bass management series. Jossey-Bass, San Francisco, Calif., 1st ed. Auflage, 1991.
- [Wei71] WEINBERG, GERALD M.: *The psychology of computer programming*. Computer science series. Van Nostrand Reinhold, New York, 1971.
- [WHL05] WAGEMAN, RUTH, J. RICHARD HACKMAN und ERIN LEHMAN: *Team Diagnostic Survey*. *The Journal of Applied Behavioral Science*, 41(4):373–398, 2005.
- [WM17] WIRDEMANN, RALF und JOHANNES MAINUSCH: *Scrum mit User Stories*. Hanser, München, 3., erweiterte Auflage Auflage, 2017.
- [YP14] YU, XIAODAN und STACIE PETTER: *Understanding agile software development practices using shared mental models theory*. *Information and Software Technology*, 56(8):911–921, 2014.

*Literaturverzeichnis*

## Anhang A

# Interviewleitfaden

Der Interviewleitfaden hat sich über die Zeit weiterentwickelt (siehe Abschnitt 5.1.2). Der hier abgebildete Interviewleitfaden wurde am häufigsten verwendet. Der Interviewleitfaden war für Teammitglieder und Expert\_innen unterschiedlich. Alle Interviewten wurden gebeten eine Einverständniserklärung zu unterschreiben und ein Datenblatt zu ihrer Person auszufüllen (dass sich noch an Diversity-Merkmalen orientierte). Von allen Unterlagen gab es auch eine englischsprachige Version.

## Kapitel A. Interviewleitfaden

### Einführung:

- Helena Barke, Dissertationsprojekt „Diversity/Vielfalt in Scrumteams“
- Interesse an Ihrer persönlichen Perspektive und Erfahrung als Scrumteammitglied
- Ungewöhnliche Situation: Ich werde Ihnen ein paar offene Fragen stellen und Sie dürfen einfach ganz ausführlich erzählen. Sie dürfen selbst entscheiden was und wieviel. In der Regel dauert so ein Interview etwa ein Stunde.
- Offene Fragen = Methodik der Grounded Thoery. Später evtl. detaillierter.
- Ich höre nur zu und werde nichts kommentieren. Das entspricht also weniger einem natürlichen Dialog / Gespräch. Es geht vor allem um Ihre Perspektive.
- Wenn Sie einverstanden sind, würde ich das Interview gerne aufnehmen und abtippen. Selbstverständlich wird alles anonymisiert und streng vertraulich behandelt. Das Interview wird für meine Dissertation und Forschungsarbeit verwendet. Alles ist dann so verfremdet, dass keine Rückschlüsse auf Ihre Person mehr möglich sind.
- Hinterher Datenschutzerklärung geben

### Icebraker:

- Was motiviert Sie in Scrumteams zu mitzuarbeiten?

### Ablauf:

- Erzählen Sie mit bitte wie eine Iteration abläuft und wie das Team dabei vorgeht.

### Ihr Team:

- Wie setzt sich Ihr Team zusammen?
- Erzählen Sie mir etwas über jedes Teammitglied.
- Welche Kompetenzen haben die einzelnen Teammitglieder?
- Welche Kompetenzen würden Sie sich mehr wünschen?

### Zusammenarbeit im Team:

- Wie sieht die Rollen und Aufgabenverteilung in Ihrem Team aus?
  - Wie organisiert sich das Team?
  - Wie gestaltet das Team den Prozess? (von Aufgabe bis ?)
  - An welchen Punkten nimmt es Einfluss und wie?
  - Welche Personen außerhalb des Teams sind noch wichtig?
  - Wie arbeiten die einzelnen Teammitglieder miteinander?
  - Wie ist die Beziehung der einzelnen Teammitglieder zueinander?
- Woran merken Sie, dass es im Team gut läuft?
- Woran merken Sie, dass es im Team nicht gut läuft?
  - Wie geht das Team damit um?

### Ihre Person:

- Was machen Sie besonders gerne?
- Was machen Sie besonders ungern?
- Welche Kompetenzen hätten Sie gern noch?
- Was haben Sie von anderen Teammitgliedern lernen können?
- Was haben andere Teammitglieder von Ihnen gelernt?

Was gibt es noch Interessantes zu berichten?

Vielen Dank!  
Erklärung Datenschutz  
Personalbogen

Abbildung A.1: Interviewleitfaden

**Fragen an Agile Coach:**

Zur Teamzusammenarbeit

- Kannst du mir bitte von einem Team erzählen, dass annähernd ideal zusammenarbeitet?
  - Wie kommt das Team zu gemeinsamen Entscheidungen?
  - Wie geht das Team mit Veränderungen um?
  - Wie geht das Team mit unterschiedlichen Kompetenz/Entwicklungsstufen einzelne Personen um?
  - Wie wird Crossfunktionalität im Team erreicht?
  - Wie löst das Team Konflikte?
  
- Kannst Du mir bitte von einem Team erzählen bei dem die Entwicklung von User Stories und die Abarbeitung annähernd ideal klappt?
  - Wie sieht eine ideale User Story aus?
  - Wie entstehen hier User Stories?
  - Wie werden die User Stories abgearbeitet?
    - Wer nimmt/bekommt was? Warum? Nach welchem Prinzip? Vorlieben?
  
- Erzähl mir bitte wie die Idealbesetzung der einzelnen Scrumrollen aussieht.
  - Scrummaster, PO, Developer
  - Wie funktioniert die Kommunikation zwischen PO und Dev Team am besten? Was sollte der PO mitbringen?  
Praxisbeispiele: Sind die in der Praxis bereits ideale Besetzungen begegnet und kannst du mir kurz von erzählen?
  
- Wie können Teammitglieder voneinander lernen? Praxisbeispiel? Methoden?

Vielen Dank!  
Erklärung Datenschutz  
Personalbogen

Abbildung A.2: Zusätzliche Fragen für agile Expert\_innen

Datenschutzerklärung zum Dissertationsprojekt:  
"Diversity in Scrumteams"

Die Auswertung des Interviews dient der Doktorarbeit und der wissenschaftlichen Arbeit von Helena Barke.

Alle Daten, die über die Veröffentlichung der Doktorarbeit, wissenschaftlichen Publikationen und Ähnliches an die Öffentlichkeit gelangen, sind so stark anonymisiert und aus dem Zusammenhang gerissen, dass die Identität der Befragten nicht mehr nachvollziehbar ist. Alle Informationen werden streng vertraulich behandelt.

Ich bin damit einverstanden, dass meine Aussagen derart anonymisiert als Material für wissenschaftliche Zwecke genutzt werden. Unter der Bedingung bin ich bereit, an dem Interview teilzunehmen und damit einverstanden, dass es aufgenommen, abgetippt, anonymisiert und ausgewertet wird.

-----  
Ort/Datum

-----  
Unterschrift

Abbildung A.3: Datenschutzerklärung

Fragebogen zur Forschungsarbeit „Diversity in Scrumteams“  
von Helena Barke

Name/Code: \_\_\_\_\_

**Zur Person:**

Geburtsjahr: \_\_\_\_\_

Geburtsort: \_\_\_\_\_

Familienstand: \_\_\_\_\_

Anzahl Kinder: \_\_\_\_\_

**Studium:**

Studiengang: \_\_\_\_\_

Studiensort: \_\_\_\_\_

Studienabschluss: \_\_\_\_\_

Jahr des Abschlusses: \_\_\_\_\_

**Beruf:**

Berufseinstiegsjahr: \_\_\_\_\_

Beim aktuellen Unternehmen seit: \_\_\_\_\_

Abteilung/Team: \_\_\_\_\_

Berufsbezeichnung: \_\_\_\_\_

Arbeitszeitmodell: \_\_\_\_\_

Abbildung A.4: Datenblatt Interviewte