Sven Steckmann
136287

Teamkonflikte und Teamkonfliktpotentiale in der agilen Softwareentwicklung

Master-Thesis
zur Erlangung des akademischen Grades
Master of Business Administration
im
Studiengang General Management
Department Wirtschaft und Management
Deutsche Universität für Weiterbildung

Betreuer: Thomas Fitzel
Datum der Einreichung: 16. Oktober 2016

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Leistungen selbst erbracht habe. Ich erkläre weiter, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte wurden gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch genaue Quellenangaben gekennzeichnet. Die wissenschaftliche Arbeit ist noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht worden. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt.

Schwanstetten, den 16. Oktober 2016

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Forschungsfragen 3
1.2 Methodik der Arbeit 4

2 Klassische Team-Modelle **7**

2.1 Gruppe 7
2.2 Team 9
2.3 Phänomene in Teams und Gruppen 10
2.4 Konflikte in Gruppen und Teams 15
2.5 Zeitbasierte Modelle der Teamentwicklung 19
2.6 Rollen-basierte Modelle der Teamentwicklung 26
2.6.1 Belbins Team-Roles Self-Perception Inventory 29
2.6.2 Spencer und Pruss 30
2.7 Zusammenfassung 31

3 Selbstorganisation **37**

3.1 Definition und Abgrenzung 38
3.2 Selbstorganisation beschreiben 38
3.2.1 Voraussetzungen nach Gloger und Rösner 39
3.2.2 Voraussetzungen nach Eoyang 40
3.3 Zusammenfassung 41

4 Agile Methoden **45**

4.1 Kennzeichen agiler Methoden 46
4.1.1 Werte 47
4.1.2 Merkmale der Arbeitsweise von agilen Teams 49
4.1.3 Rollen 51
4.2 Agile Frameworks 53
4.3 Scrum 53
4.3.1 Artefakte 56
4.3.2 Rollen im Scrum Team 57
4.3.3 Ereignisse 59

4.4 Leadership in agilen Teams 63

5 Verbindung der klassischen Modelle und der agilen Arbeitsweise 67
5.1 Phänomene in Gruppen 68
5.2 Konflikte 70
5.3 Phasen der Teamentwicklung 72
5.4 Rollen-basierte Modelle der Teamentwicklung 73
5.4.1 Spencer und Pruss 74
5.4.2 Belbins Team-Roles-Self-Perception Inventory 77
5.4.3 Zusammenfassung und Bedeutung für die Forschungsfrage 80

6 Best-Practices zum Einsatz agiler Methoden im Unternehmensumfeld 83
6.1 Experteninterview 83
6.1.1 Auswahl der Gesprächspartner 84
6.1.2 Interviewleitfaden 86
6.1.3 Durchführung des Interviews 86
6.1.4 Hinweise zur Typographie für Zitate aus den Interviews für diese Arbeit 87
6.2 Auswertung der Experteninterviews 87
6.2.1 Konflikte und negative Entwicklungen im agilen Team 88
Rahmenbedingungen für Agilität 89
Raum und Zeit 90
Autonomie, Feedback und Übernahme von Verantwortung 91
Persönliche Einstellungen und Merkmale 97
Schwierige Persönlichkeiten 98
Wissen über Methoden der agilen Entwicklung 100
Zusammensetzung des Teams 101
Teamspirit 102
Kommunikation 104
Rollen im Scrum 105
Notwendige Veränderungen 106
Disziplin im Scrum-Alltag 107
Zusammenfassung und Bedeutung für die Forschungsfrage 108
6.2.2 Erkennen von Konfliktherden und Problemen in agilen Teams 111

Bei Konflikten kommen die Leute oftmals auf den Coach / Scrum-Master zu. 111
Aktives Nachfragen 112
Nutze die Artefakte von Scrum und Meetings mit Interaktion im Team zur Diagnose 113
Anti-Pattern / Bad-Smell 113
Persönliche Gefühle 114
Veränderung im Verhalten 115
Wirtschaftliche Lage / Zahlen 117
Zusammenfassung und Bedeutung für die Forschungsfrage 117
6.2.3 Lösungswege und Vorbeugemaßnahmen im Alltag bei der Arbeit mit agilen Teams 118
6.2.4 Weitere identifizierte Themen 126
Einführung, Gründe, Effizienz, Leistung 126
Softwareprojekte sind nicht gleich — Scrum-Implementierungen auch nicht 127
Scrum im kulturellen Kontext 128

7 Zusammenfassung **131**

8 Ausblick **137**

9 Literatur **139**

A Das agile Manifest **149**

B Interviewleitfaden **151**

C Transkriptionen der Interviews **159**
C.1 Interviewtranskription vom 1. Interview 160
C.2 Interviewtranskription vom 2. Interview 191
C.3 Interviewtranskription vom 3. Interview 224
C.4 Interviewtranskription vom 4. Interview 246

D Abkürzungsverzeichnis **269**

E Glossar **271**

1 Kapitel 1

Einleitung

Die Informatik und deren praxisnahe Anwendung in Unternehmen haben für die Entwicklung von Software eine andersartige Herangehensweise als in den klassischen technischen Disziplinen entwickelt. Grundlegend hierfür war die Erkenntnis, oder besser gesagt die Unkenntnis der Methoden für eine optimale Softwareentwicklung. Dabei sind sich selbst die Wissenschaftler nicht einig, wie Softwareentwicklung zu verstehen ist: Ist es eher eine kreative Design- oder eher eine Ingenieurstätigkeit (Schaeffer 1996)? Durch die hohe Abstraktion von Software und die schlechte Begreifbarkeit verhält sich Software in vielerlei Hinsicht ähnlich wie Wissen in Unternehmen.

Während in anderen Disziplinen Prozesse eine vordergründige Rolle spielen, geht es im Softwarehandwerk in eine andere Richtung. Es gibt hier auch komplexe und schwerfällige Entwicklungsmodelle, wie das V-Modell, welche sich in der Praxis oftmals als zu schwerfällig herausgestellt haben. Gerade in der heutigen Zeit ist es wichtig, eng am Markt und am Kunden zu sein. Dadurch wird die Praxis stark geprägt von laufenden Änderungen der Anforderungen, was besser durch eine iterative und einem leichtgewichtigen Softwareentwicklungsprozess umgesetzt werden kann. Die agile Softwareentwicklung geht hierbei einen Weg, mit wenig bürokratischen Aufwand und Regeln auszukommen (vgl. Beck et al. 2001). Vermehrt werden diese Vorgehensweisen auch auf andere Disziplinen in der Entwicklung ja sogar auf Abteilungsebene, Projektebene (Pröpper 2012) oder Organisationsebene (HolacracyOne, LLC. 2015) eingesetzt. Die Arbeit wird sich im weiteren Verlauf vor allem für den Einsatz agiler Methoden auf Ebene der Teams interessieren.

Agile Methoden sind jedoch kein Garant für eine reibungslose Teamarbeit. Sie geben oftmals nur einen Rahmen vor, der von erfahrenen Managern als eine Art „Best Practice“ modelliert wurde. Agile Methoden können damit bei der Adaption helfen. Das Management muss hierfür die notwendigen Rahmenbedingungen herstellen, das Team und deren Zusammenarbeit verstehen und ein Gefühl für diese Art der Arbeit und die Kultur (vgl. Maximini 2015) entwickeln. Dies fällt nicht jedem leicht, da Agilität auch oft ein Verzicht von Eingriffen bedeutet, was jedoch nicht bedeutet, dass das Team nicht beobachtet werden sollte. Dabei muss sich ein Manager eher als Coach verstehen (vgl. Adkins 2010). Hierfür ist es unerlässlich, dass problematische Zustände rechtzeitig erkannt und beseitigt werden. Es gibt bereits erste Forschungsansätze, welche versuchen die Kommunikation in agilen Teams zu untersuchen oder auch sogenannte „must haves“ zu ermitteln, welche unbedingt vorhanden sein sollten. In Lindvall et al. 2002 werden z.B. „Success Factors“ and „Warning Signs“ angesprochen jedoch nicht näher ausgeführt. Diese zu ermitteln ist von hohem Interesse, denn entweder ist das System „agiles Team“ in der Lage diese zu erkennen oder es muss von extern erkannt werden. Gerade durch die Betriebsblindheit oder anderer psychologischer Effekte wie Groupthink (vgl. Gessler 2012; Janis 1982; Janis 1972) kann es passieren, dass das Team solche Faktoren nicht erkennt. In Boehm und Turner 2003a und Boehm und Turner 2003b wird dabei ein wichtiger Faktor angesprochen: Die Balance zwischen Agilität und Disziplin muss stimmen und individuell an die Rahmenbedingungen angepasst werden. Dieser Faktor ist z.B. in einem Medizintechnikunternehmen von hoher Bedeutung, da hier regulatorisch „must haves“ an den Entwicklungsprozess selbst gestellt werden. Es ist jedoch oftmals wichtig, auch abseits von solchen scharfen Randbedingungen weitere Spielregeln oder auch Ziele für das Team zu definieren, um eine langfristige hohe Performance des Teams zu gewährleisten. Aus meiner eigenen Erfahrung weiß ich, dass z.B. das Wissensmanagement von Entwicklern gerne zu Gunsten neuer technischer Herausforderungen vernachlässigt wird. Damit dies nicht zu einer Gefahr für den dauerhaften Geschäftserfolg wird, müssen solche Themen beobachtet werden.

Auch ein agiles Team ist letztendlich ein Team, das von seiner Konstituierung bis zu seiner Auflösung verschiedene Phasen durchläuft, was z.B. im Modell von Tuckman und Jensen (Tuckman und Jensen 1977) beschrieben wird. Die Frage, ob ein Framework wie Scrum (Schwaber und Beedle 2001) dazu beitragen kann, dass ein Team schneller produktiver wird und sich letztendlich auch in einer Phase mit hoher Leistung halten kann stellt sich da-

bei auch bei der Einführung von Scrum für die Entscheider. Dies wird in einer Gegenüberstellung der Modelle und der im Framework auffindbaren Elemente untersuchen (siehe auch Kapitel 2.5). Ebenso ist es auffällig, dass Scrum mit wohldefinierten Rollen arbeitet. Dies legt nahe, sich die verbreiteten Rollenmodelle für Teams anzusehen und einen Vergleich mit den auffindbaren Elementen anzustellen. Dies werden im Kapitel 2.6 weiter untersuchen.

Die Modelle stoßen letztendlich immer an Grenzen. Dies zeigt die Praxis in einer Art und Weise, so dass von ihnen gelernt werden kann. Daher wird im Anschluss eine empirische Erhebung durchführen, die aufzeigt, wie praxisnahe und erfahrene Coaches oder Teamleiter von außen ein agile Team führen und diagnostizieren. Es sollen dabei mithilfe von Experten wichtige Faktoren für die Diagnose ermittelt werden. Es ist davon auszugehen, das diese Situation bei den Experten im Alltag bereit vermehrt aufgetreten ist und sie daher schon für dieses Thema sensibilisiert sind.

1.1 Forschungsfragen

Die Einleitung macht deutlich, dass das Gebiet sehr vielfältig und interessant ist. Die Arbeit kann sich in ihrem Umfang nur einem Teil der Praxis nähern. In der Praxis geht dem Lösen von Problemen immer das Erkennen des Problems voraus. Daher wird sich der diagnostische Teil mit der Forschungsfrage widmen. Gerade aus der Teamdiagnose und -entwicklung sind vielfältige Modelle bekannt, welche ebenso auf agile Teams zutreffen.

Folgende Forschungsfrage wird im Rahmen dieser Arbeit bearbeitet:

- Wie können relevante negative Entwicklungen in agilen Scrum-Teams praxisgerecht beobachtet werden?

Folgende unterstützende Fragen werden in der Arbeit weiterhin eine wichtige Rolle spielen:

- Welche negativen Entwicklungen aus den klassischen Modellen der Teamentwicklung lassen sich auf agile Scrum-Teams übertragen?

- Welche inhärenten Merkmale sind in agilen Methoden zu finden, die einer negative Entwicklungen vorbeugen?
- Welche typischen negative Entwicklungen lassen sich in der Praxis beobachten?

1.2 Methodik der Arbeit

Die zahlreichen Publikationen in Form von Büchern aus der Praxis gibt uns schon einen Hinweis, dass es gerade im Scum-Umfeld viele Praktiker gibt, welche in ihrer täglichen Praxis mit vielschichtigen Problemen konfrontiert werden. Einige von ihnen, wie beispielsweise Mike Cohn, haben wertvolle Bücher aus ihren Praxiserfahrungen für Coaches und Interessenten verfasst (Cohn 2010b). Wiederum andere nutzen ihr Wissen aus der Praxis für ihre tägliche Arbeit als Scrum-Master oder Coach für agile Teams. Es liegt nahe, die Wissensquelle „Praktiker“ für die Arbeit zu nutzen.

Leider ist die Wissensquelle Mensch wenig strukturiert und durch das noch wenig vorhandene Vorwissen wird eine explorative Methode benötigt, die in der Lage ist, dieses Wissen sichtbar zu machen.

Die Arbeit wird daher die Technik des Leitfrageninterviews aus der qualitiativen empririschen Sozialforschung zur Anwendung bringen (vgl. Kühl, Strodtholz und Taffertshofer 2009, S. 32 ff.). Die Technik wird auch als (leitfragengestütztes) Experteninterview bezeichnet. Die interviewten Experten[1] zeichnen sich durch einen besonderen Wissensstand aus. Die gewählten Experten haben schon vielfältig im Bereich der agilen Entwicklung, bei der Zusammenstellung neuer Teams, im Alltag als Scrum-Master[2] oder durch vielfältige Coaching-Einsätze im agilen Umfeld ihr Praxiswissen erfolgreich angewandt. Damit weisen diese Personen einen deutlichen Wissensvorsprung aus der Praxis auf, der für diese Arbeit genutzt wird.

Die Leitfragen für das spätere Interview sind „das Resultat theoretischer-wissenschaftlicher Vorüberlegungen, mit denen die Forscher die Feldphase vorbereiten“ (Kühl, Strodtholz und Taffertshofer 2009, S. 38). Innerhalb

[1] Für den Experten gibt es verschiedene wissenschaftliche Definitionen (s.a. Kassl 2000, S. 29-38; Kühl, Strodtholz und Taffertshofer 2009, S. 33 f.). Im Rahmen der Arbeit wird die Definition genutzt, das es sich um eine Person handelt, die ein hohes Wissen im Forschungsfeld besitzt.

[2] Der Scrum-Master ist eine Rolle im agilen Vorgehensmodell Scrum, welche sich vor allem um die soziale Komponente im Team kümmert. Die Aufgabe der Rollen wird in Kapitel 4.3.2 beschrieben.

dieser Vorüberlegungen wird sich die Arbeit mit den klassischen Team-Modellen in Kapitel 2 auseinandersetzen . Da agile Teams, wie der Name schon aussagt, ebenfalls Teams sind, können wir bei einer Gegenüberstellung mit den agilen Methoden aus Kapitel 4 bereits interessante Zusammenhänge oder Unterschiede feststellen, die wiederum neue Fragen an die Experten aufwerfen. Weiterhin wurden in den agilen Methoden Mechanismen zur Vermeidung von Problemen vorgesehen. Diese wurde ebenfalls mit dem Experten hinterfragt, ob sie immer zum Erfolg führen und in welchen Konstellationen dies problematisch werden kann.

Mit den entwickelten Leitfragen wurde das Experteninterview durchgeführt (s.a. Kapitel 6). Anknüpfend an die Vorgehensweise für die Arbeit ist die Methodik des Leitfrageninterviews in Kapitel 6.1 dargestellt.

Kapitel 2

2 Klassische Team-Modelle

In der heutigen Arbeitswelt spielt das Team eine wichtige Rolle. Wir begegnen ihm bereits in Stellenanzeigen, wo Teamfähigkeit gefordert wird, in unserer täglichen Arbeit in Form von Projekt- oder Führungsteams, ja sogar im Vereinssport in unserer Freizeit. Für uns als soziale Wesen ist es selbstverständlich, dass wir in unserem beruflichen Alltag mit anderen Menschen zusammenarbeiten. Die Interaktion und wie wir als Mitglieder in Teams oder Gruppen zusammenarbeiten ist eine wichtiger Forschungsschwerpunkt in der Organisationspsychologie. Es gibt eine Vielzahl von Modellen, die jeweils aus Beobachtungen der Praxis hervorgegangen sind und einen Teilaspekt beschreiben, kein Modell kann dabei die komplexen Situationen in der Gänze erfassen. Sie helfen jedoch aufgrund ihrer Strukturiertheit die komplexe Situation zu ordnen und begreifbarer zu gestalten. Im Sprachgebrauch wird Team und Gruppe oftmals äquivalent behandelt, obwohl es deutliche Anzeichen gibt, dass es hier Unterschiede gibt. Eine Auseinandersetzung mit den Begrifflichkeiten „Team“ vs. „Gruppe“ und der genutzten Definition ist daher unerlässlich, bevor das Thema mit den klassischen Team-Modellen vertieft wird.

2.1 Gruppe

Für die Gruppe gibt es eine Vielzahl von Definitionen in der Literatur (siehe Tabelle 1 Kauffeld 2001, S. 13). Sehr treffend wird dies jedoch wie folgt beschrieben: „Die Gruppe besteht aus zwei oder mehr Personen, die miteinander interagieren und insofern interdependent sind, als ihre Bedürfnisse

und Ziele eine gegenseitige Beeinflussung bewirken." (Aronson, Wilson und Akert 2014, S. 311). Wichtig für die Ausbildung einer Gruppe ist vor allem, das die Personen interagieren können: „Gruppe ist das, was ZWISCHEN den Mitgliedern geschieht." (Antons 2000, S. 296). Damit „…werden in der Regel rein koagierende Arbeitsgruppen, in denen eine Koordination der Einzelleistungen der Mitarbeiter nicht erforderlich ist, nicht als Gruppe bezeichnet." (Kauffeld 2001, S. 13), was eine wichtige Abgrenzung beschreibt. Wird die Gruppe als Ganzes betrachtet, so bildet sie sich aufgrund von Gemeinsamkeiten der Individuen; so haben Gruppen z.B. ein gemeinsames Ziel oder eine gemeinsame Einstellung.

Gruppen gibt es in vielfältiger Ausprägung, denken wir z.B. an Volksgruppen. Wir interessieren uns im Folgenden nur für Gruppen in der Organisationsstruktur, dabei sind für uns in dieser Arbeit weiterhin nur die formellen[1] Gruppen interessant, da diese Art von Gruppen auch die Arbeitsgruppen bilden, welche später untersucht werden. Sie sind geprägt durch ein gemeinsamen Ziel und dieses ist in Hinblick auf die Aufgabenerfüllung geplant (vgl. Steinmann und Schreyögg 2005; Aronson, Wilson und Akert 2014; Kauffeld 2001).

Die Literatur hat verschiedene Definitionen für eine Gruppe hervorgebracht. Dabei zeigen sich vielfach Gemeinsamkeiten, die als Voraussetzungen zur Bildung einer Gruppe genannt werden (vgl. Antons 2000, S. 296 ff.; Steinmann und Schreyögg 2005, S. 595; Arrow und McGrath 1995; Eberhardt 2013, S. 8; Nerdinger, Blickle und Schaper 2014, S. 104; Winterhoff-Spurk 2002, S. 59):

- Drei oder mehrere Personen, deren Gesamtanzahl gering genug ist, dass die Gruppenmitglieder miteinander direkt interagieren können. Oft wird eine Maximalanzahl der Gruppenmitglieder von 12 bis 20 genannt. Darüber hinaus wird es schwierig für die Mitglieder den direkten Kontakt zu pflegen und es bilden sich Untergruppen.
- Das Auftreten von Interaktion zwischen den Gruppenteilnehmern und der daraus resultierten Bindung der Mitglieder untereinander.
- Durch eine längerfristige Gruppenzugehörigkeit entsteht ein Wir-Gefühl in der Gruppe und damit eine Bindung und Identifikation.

[1] Die *formelle Gruppe* wird in Hinblick auf ihre Aufgabenerfüllung geplant und in der organisatorischen Gliederung erfasst. Sie wird formal von der Organisation gebildet und aufgelöst. Die Leitung der Gruppe hat eine bestimmte Aufgabe innerhalb der Organisation (vgl. Steinmann und Schreyögg 2005, S. 596; Gabler 2015).

Dabei entwickeln sich gemeinsame Normen und Werte, welche ihr Denken und Handeln beeinflusst.

- Ein gemeinsames Ziel, Aufgabe oder Projekt sind vorhanden.

In einer Gruppe bilden sich Rollen aus, welche die einzelnen Mitglieder einnehmen.

2.2 Team

Das Wort Team ist aus der heutigen Arbeitswelt nicht mehr wegzudenken. Was zeichnet das Team im Vergleich zu einer Gruppe von Menschen besonders aus? Es gibt Stimmen in der Literatur, welche einen Unterschied für marginal halten oder Aussagen, dass aufgrund der Ähnlichkeit eine „…Unterscheidung zwischen Gruppe und Team kaum möglich ist." (Nerdinger, Blickle und Schaper 2014, S. 104). Andere sprechen davon, dass „Die Begriffe Gruppe und Team (…) von einigen Autoren synonym verwendet (werden), von anderen nicht." (Kauffeld 2001, S. 14). Allerdings verbinden wir mit Team etwas höherwertiges als eine Gruppe. Untersuchungen von Softwareteams haben den Vorteil von einem Team gegenüber einer Gruppe deutlich gezeigt: Bei einer Untersuchung von 190 Projekten in Huber und Lindenhahn 2010 schließen Teams Projekte zu 40% ab, während der Durchschnitt nur bei 25% lag. Eine oftmals genutzte Definition (auch für den Naturwissenschaftler in mir) ist die Formel:

$$\text{Team} = \text{Gruppe} + \text{Leistung}$$

Für ihr Buch *Mehr Erfolg im Team* haben Francis und Young Führungskräfte befragt, was ein Team besonders auszeichnet und sie fassten dies wörtlich zu folgenden Aussagen zusammen (Francis und Young 1996, S. 17):

- Erfolgreiche Teams vollbringen außerordentliche Leistungen auch unter schwierigen Bedingungen.
- Die Mitglieder fühlen sich für Arbeit des Teams verantwortlich und sie erörtern offen alle Probleme, die ihnen im Weg stehen.

Gerade diese Eigenschaften machen in vielen Situationen ein Team gegenüber einer Gruppe für ein Unternehmen wertvoller. Besonders die höhere Leistung eines Teams im Vergleich zur Gruppe spielt eine entscheidende

Rolle für deren Einordnung, wie etwa auch durch Winterhoff-Spurk 2002, S. 102 ff. festgestellt wird. Das Team hat im Vergleich zu einer Gruppe einen „Teamschwur“, der sich durch eine besonders enge Beziehung untereinander auszeichnet (vgl. DeMarco und Lister 2014, S. 135). Dem Team werden vorteilhafte Eigenschaften zugeschrieben, welche ihm zu dieser Leistung verhelfen (vgl. Francis und Young 1996, S. 19; Eberhardt 2013, S. 8; Niermeyer 2012, S. 15-16; Krüger 2013, S. 8):

- Es herrscht eine starke Kohäsion im Team.
- In der Regel hat das Team 2 bis ca. 9 Mitgliedern.
- Ein herrscht ein angenehmes, harmonisches Arbeitsklima.
- Es wird das Ziel des Teams verfolgt, Einzelinteressen treten zurück es gibt eine hohe Aufgabenorientierung.
- Das Team hat eine enge, offene Kommunikation, Vertrauen und gutes Konfliktmanagement.
- Das Team motiviert sich von innen heraus.
- Ein sehr starkes „Wir“-Gefühl.
- Das Team hat eine Struktur in Form von einem gemeinsamen Rollenverständnis (s.a. Kapitel 2.6).
- Das Team erbringt Leistungen die mehr als die Summe der Einzelleistungen sind und nutzt Synergieeffekte zwischen den Mitgliedern.

2.3 Phänomene in Teams und Gruppen

Sobald sich Menschen zu Gruppen zusammenschließen, ändern sie ihr Verhalten. Dies ist ein gewünschter Effekt, jedoch gibt es auch einige Handlungsmuster, welche problematisch sein können. Für die Organisation ist es wichtig, diesen problematischen Entwicklungen vorzubeugen oder zumindest zu erkennen und ins Bewusstsein zu rücken.

Zur Ausübung ihrer Aufgaben muss eine Gruppe immer wieder Entscheidungen treffen. Aus dem beruflichen Alltag ist bekannt, dass Entscheidungsfindung in einer Gruppe im Vergleich zum Individuum anders abläuft. Die Gruppe wird für ihre Entscheidung Kommunikation nutzen, um Informationen für die Entscheidung zusammenzutragen. Bei diesem Vorgang kann

es passieren, das sich bereits eingenommene Positionen verstärken, oder dass vorher belanglose Positionen zum Tragen kommen. Ein Beispiel für einen solchen Vorgang ist der Besuch in einem Restaurant. Nehmen wir an, das Essen ist nicht perfekt, aber durchaus gut und bei jedem in der Gruppe ist am Gericht nur eine Kleinigkeit auszusetzen. Durch die Kommunikation dieser Kleinigkeiten kann in der Gruppe sehr schnell die Stimmung soweit kippen, dass die Gruppe das gesamte Essen als extrem negativ empfindet und unter Umständen die Entscheidung fällt, das Restaurant nicht mehr aufzusuchen. Die Gruppe wird polarisiert und die Gruppe wird in ihrer Einstellung extremer. Zusammenfassen lässt sich damit: In Gruppen gibt es eine Tendenz, das Entscheidungen extremer ausfallen, als sie die einzelnen Individuen getroffen hätten. (vgl. Gerrig und Zimbardo 2015, S. 660 f.; Aronson, Wilson und Akert 2014, S. 330-332). In der Fachsprache wird hierfür der Begriff **Risikoschub (engl. risky shift)** genutzt. Die meiste Literatur spricht davon, dass der Risikoschub hin zu risikoreicheren Entscheidungen passiert. Gerrig und Zimbardo 2015, S. 660 weißt jedoch darauf hin, dass dies von der Gruppentendenz abhängig ist, so dass der Risikoschub in beide Richtungen auftreten kann; hin zu einem höheren Risiko aber auch zu einer Risikovermeidung. Die Risikovermeidung ist insofern ebenso problematisch, da z.B. Innovationen immer ein gewisses Risikos beinhalten und die Gruppe das Unternehmen lähmen kann.

In der Literatur gibt es zahlreiche Erklärungsthesen (vgl. Steinmann und Schreyögg 2005, S. 621 f.):

Verantwortung der Gruppe Das Risiko wird durch die ganze Gruppe getragen.

Mehr Informationen Die Mitglieder der Gruppe haben durch die Kommunikation im Vorfeld der Entscheidung mehr Informationen zur Verfügung, was die Unsicherheit absolut oder scheinbar reduzieren kann.

Führung Führungspersönlichkeiten in Gruppen haben einen hohen Einfluss auf die Gruppe und sind risikofreudiger; ihre Argumente werden daher eher von den anderen angenommen.

Risiko als sozialer Wert Risiko gilt als ein Teil des Geschäfts; um nicht als Bremse wahrgenommen zu werden, wird gerne mehr Risiko eingegangen.

Eine andere Erklärung, welche für hohes wie niedriges Risiko gleichermaßen gelten soll, ist, dass sich durch die vermehrte Kommunikation in der

Gruppe die Gruppentendenz letztendlich verstärken wird. Es ist damit abhängig, wie die Gruppe vom Grunde auf eingestellt ist, zu welcher Richtung sie letztendlich tendiert (vgl. Gerrig und Zimbardo 2015, S. 660).

Eng verwand mit dem Risikoschub ist das Phänomen **Gruppendenken (engl. groupthink)**. Der Begriff des Groupthink wurde durch Janis in Janis 1972; Janis 1982 geprägt. Ein sehr bekanntes Beispiel (welches auch durch Janis ausführlich analysiert wurde), das dem Gruppendenken zugeordnet wird ist Entscheidung zur Invasion Kubas über die Schweinebucht im Jahre 1961. Der Invasion gingen zahlreiche Treffen von Kennedy und seinem Kabinett voraus; es war bekannt, wie Kennedy darüber dachte und die Mitglieder hielten zum Teil Informationen zurück und schwiegen wie Arthur Schlesinger. Er schätzte seine Meinung als gering ein und fürchtete die Kritik der Gruppe. Er war doch nur Universitätsprofessor und die anderen von hohen Regierungsinstitutionen (vgl. Aronson, Wilson und Akert 2014, S. 327).

Janis fand in seinen Studien Bedingungen, welche Gruppendenken ermöglicht. Durch spätere Forschung wurden diese ursprünglichen Bedingungen noch erweitert. Es wird wahrscheinlicher auftreten, wenn folgenden Bedingungen anzutreffen sind (vgl. Gessler 2012, S. 242 f.; Aronson, Wilson und Akert 2014, S. 326-329):

Gruppenkohäsion Eine Gruppe besitzt eine hohe Kohäsion sowie Wertschätzung und Attraktivität. Andere Menschen möchten der Gruppe gerne angehören.

Isolation Die Gruppe ist von konträren Standpunkten isoliert. Andere Sichtweisen und Standpunkte dringen nicht zur Gruppe vor.

Direktiver Leiter Ein richtungsweisender Leiter beherrscht die Diskussion. Er gibt seine Meinung und Wünsche bekannt.

Hoher Stress Die Mitglieder haben aufgrund einer bedrohlichen Situation hohen Stress.

Schlechter Entscheidungsfindungsprozess Der Prozess zur Entscheidungsfindung ist wenig standardisiert; gerade das Einbinden alternativer Sichtweisen wird nicht systematisch verfolgt.

Gruppendenken kann nicht nur anhand der oftmals fehlerhaften Entscheidungen erkannt werden, es treten ebenfalls typische Merkmale auf, die einen Hinweis geben können (vgl. Aronson, Wilson und Akert 2014, S. 326-329; Nerdinger, Blickle und Schaper 2014, S. 109-111; Steinmann und Schreyögg 2005, S. 623 f.):

1. Selbstüberschätzung der Gruppe

   Illusion der Unverwundbarkeit Die Gruppe rechnet nicht mit einer Fehleinschätzung oder -entscheidung ihrerseits und fühlt sich unbesiegbar.

   Glaube an die moralische Richtigkeit der Gruppe Die Gruppe fühlt sich moralisch im Recht.

2. Geistige Einengung

   Kollektive Rationalisierung Die Motive des eigenen Handelns werden durch rationale, aber vorgeschobene Gründe überdeckt. Es werden damit legitime Gründe für ein ansonsten nicht legitimierbares Handeln geschaffen.

   Stereotypisierung von Außenstehenden Andere Gruppen, meist Gegenspieler, werden vorverurteilt, womit eine objektive Einschätzung nicht mehr möglich ist.

3. Uniformitätsdruck

   Selbstzensur Die Mitglieder halten Informationen oder ihre abweichende Meinung zurück. Die Motive sind verschieden, oftmals ist es Loyalität zum Meinungsführer, der Gruppenfrieden oder der Zeitdruck.

   Gruppendruck (Gruppenzensur) Die Äußerung widersprüchlicher Meinungen wird durch den Druck der Gruppe unterdrückt. Dies passiert z.B durch Infragestellen der Gruppenzugehörigkeit.

   Gedankenwächter (mind guards) Mitglieder der Gruppe halten bewusst Informationen zurück, die ihrem Ziel kontraproduktiv sein könnten. Besonders der Gruppenleiter wird von diesen Informationen ausgeschlossen.

   Illusion der Einmütigkeit Es wird der Eindruck erweckt, das die Gruppe einer Meinung ist. Mitglieder mit abweichender Meinung werden nicht befragt.

Um Gruppendenken bei Entscheidungen zu vermeiden, ist vor allem die Führung der Gruppe gefragt. Die Maßnahmen setzen bei den Voraussetzungen für das Gruppendenken an und vermeiden diese (vgl. Aronson, Wilson und Akert 2014, S. 328 f.; Wiendieck 2004):

- Der Gruppenführer soll keine direkte Rolle in der Gruppe einnehmen sondern unparteiisch bleiben.
- Fremdmeinungen von außerhalb der Gruppe sollten regelmäßig eingeholt werden. Diese Meinungsträger sind nicht der Gruppenkohäsion verpflichtet und daher frei in der Wiedergabe ihrer Meinungen. Die Gruppe erhält hierdurch neue Anreize für die Diskussion.
- Die Gruppe sollte in Untergruppe aufgeteilt werden, die sich zunächst separat beratschlagen und erst später die Vorschläge gemeinsam diskutieren. Damit wird eine voreilige Festlegung auf eine Lösung verhindert.
- Abstimmungen oder Meinungsäußerungen in der Gruppe können anonym durchgeführt werden. Dies verhindert, dass sich Mitglieder selbst zensieren und sich in ihrer Entscheidung oder Äußerung freier fühlen.
- In die Diskussion der Gruppe sollten gezielt gegenteilige Meinungen einfließen. Dies entflammt die Diskussion und kann ein Umdenken bewirken. Eine Methode hierfür ist Advocatus diaboli (s.a. Steinmann und Schreyögg 2005, S. 625) oder der dialektische Diskurs.

Oftmals verbinden wir mit Team- oder Gruppenarbeit ein Erhöhung der Leistung; in der Praxis kann auch das Gegenteil eintreten. Wie es sich entwickelt ist stark von der Aufgabe und den Randbedingungen abhängig. Die Soziologie hat hierfür die Begriffe der **social facilitation** und des **social loafing** geprägt.

Eine Gruppe versetzt den menschlichen Körper in einen erregten Zustand und Experimente (Schmitt et al. 1986) haben gezeigt, dass in diesem Zustand besonders gut dominante, also gut eingeübte Verhaltensmuster abgerufen werden können. Was in Anwesenheit von anderen jedoch schwer fällt, sind komplexe Dinge wie z.B. Lernen. Zusammenfassen lässt sich **social facilitation** zu: Wenn die Einzelleistung messbar ist, werden bei der Anwesenheit anderer einfache Dinge schneller erledigt, während komplexe länger benötigen (vgl. Aronson, Wilson und Akert 2014, S. 317 ff.).

Gerade bei Aufgaben mit nicht identifizierbarer Einzelleistung, kann sich **social loafing** (Latané, Williams und Harkins 1979) in Gruppen entwickeln. Hierbei lässt ein Mitglied in seiner Aufgabe nach und wird unter dem Deckmantel der Gruppe eine geringere Leistung erbringen. Für den Einzelnen in der Gruppe sinkt durch die Gruppensituation die Bewertungsangst und

er wird sich weniger anstrengen, denn er kann nicht erkannt oder bewertet werden. Dies wird dann besonders kritisch, wenn der Eindruck des Trittbrettfahrens aufgrund mangelnder Motivation einzelner für die Gruppe entsteht. Dann kann dies in einem Teufelskreis enden, bei dem wiederum die Leistungsträger in der Gruppe ihre Leistung ebenfalls reduzieren. (vgl. Aronson, Wilson und Akert 2014, S. 321 ff.; Wiendieck 2004).

2.4 Konflikte in Gruppen und Teams

Konflikte sind so alt wie der Mensch selbst. Der Mensch als soziales Wesen hat Bedürfnisse und Interessen. In der Interaktion mit anderen Mitmenschen treten zwangsläufig Interessenskonflikte auf. Die Lösung dieser Konflikte bietet Chancen und Risiken zugleich. Wie sich ein Konflikt letztendlich entwickelt hängt stark von den Beteiligten, dem Verständnis eines Konfliktes und der letztendlichen Lösung ab.

Die Felder der Forschung, welche sich der Erforschung und Beschreibung von Konflikten widmen, sind vielfältig. Traditionell beschäftigte sich vor allem die Psychologie und Soziologie mit Konflikten. Gerade im beruflichen Alltag sind Konflikte ein integraler Bestandteil, so dass sich die Arbeits- und Organsiationspsychologie für Konflikte interessiert. Im Rahmen dieser Arbeit wird der Fokus ausschließlich auf den beruflichen Alltag gelegt (vgl. Gius 2015, S. 28 ff.). Für die Organisation und Teams bieten Konflikte immer die Möglichkeit, Schwächen zu identifizieren und positive Änderungen zu bewirken.

Definition und Einordnung von Konflikten in Teams

Konflikte lassen sich nach dem Ort des Auftretens unterschieden:

- Als **innerer** oder **intrapersonal** wird ein Konflikt bezeichnet, der sich auf eine Person beschränkt. Der Konflikt spielt sich in den Gedanken der Person ab und hat seine Ursache in unüberschaubaren oder widersprüchlichen Gedanken, so dass sich die Person selbst nicht für eine bestimmte Handlung entscheiden kann.
- Der **äußere** oder **interpersonale** Konflikt hat immer zwei Beteiligte wie z.B. Personen, Gruppen, Staaten.

Der äußere Konflikt tritt vor allem in Teams in Erscheinung, da hier mehrere Personen miteinander interagieren. Sie können auch begleitet von inneren Konflikten der Personen selbst sein (vgl. Gessler 2012, S. 1013). Für die Teamarbeit sind beide Konfliktarten wichtig, jedoch wird sich diese Arbeit nur mit dem äußeren Konflikt beschäftigen. Er ist es, der Probleme in der Zusammenarbeit hervorruft; der innere Konflikt wirkt sich auf das Team nur indirekt durch einen nicht entscheidungsfähigen Mitarbeiter aus.

Der äußere Konflikt (auch sozialer Konflikt genannt) lässt sich wie folgt beschreiben: „Soziale Konflikte sind Spannungssituationen, in denen zwei oder mehrere Parteien, die voneinander abhängig sind, mit Nachdruck versuchen, scheinbar oder tatsächlich unvereinbare Handlungspläne zu verwirklichen und sich dabei der Gegnerschaft bewusst sind." (Rüttinger 1980, S. 22)

Diese Definition kann wie in Winterhoff-Spurk 2002, S. 111 f. in unterschiedliche Elemente unterteilt werden (die betroffenen Definitionen sind *kursiv* gesetzt):

- Eine *Spannungssituation* ist bei den Beteiligten durch körperliche oder emotionale Merkmale wie Stress und Aggression zu erkennen.
- *Zwei oder mehr Parteien* sind ein typisches Zeichen für einen interpersonalen Konflikt. Für die Teams spielen folgende zwei Fälle eine wichtige Rolle:
  - Intragruppenkonflikte, also die Konflikte innerhalb der Gruppe bzw. dem Team selbst
  - Intergruppenkonflikte zwischen verschiedenen Gruppen im Unternehmen
- Da die Konfliktparteien *voneinander abhängig* sind, hat die Handlung der einen Partei unmittelbar Einfluss auf die andere. Bei der Lösung kann es verschiedene Möglichkeiten geben. Es können beide Parteien gestärkt durch eine „win-win"-Situation hervorgehen. Wenn eine oder beide Parteien unterliegen, wird von einer „win-loose" oder „loose-loose"-Situation gesprochen (s.a. Konfliktlösungsstile Sturm, Opterbeck und Gurt 2011, S. 88).
- Handlungspläne spielen bei einem Konflikt eine entscheidende Rolle, denn er tritt dann auf, wenn *scheinbar oder tatsächlich unvereinbare Handlungspläne* existieren. Es geht damit um konkretes Verhalten und nicht um Meinungen oder Gefühle.

- *Bewusste Gegnerschaft* zeigt an, dass sich beide Parteien des Konfliktes bewusst sind und ihn annehmen, womit ein offener Konflikt vorliegt.

Dabei ist darauf zu achten, das eine Meinungsverschiedenheit nicht gleich ein Konflikt ist, sondern zu einem Konflikt werden kann. Stöwe und Keromosemito 2013 fassen zusammen, dass eine Meinungsverschiedenheit dann zu einem Konflikt wird, wenn (a) kein Argument überzeugt, (b) es nicht gleichgültig ist und (c) Emotionalität hinzu kommt.

Klassifizierung von Konflikten

Merkmal	Differenzierung
Parteien	Einzelpersonen, Gruppen, Organisationen
Machtverteilung	Gleichgestellte oder Ungleiche
Gegenstand	Ressourcen, Macht, sonstige Handlungsziele
Ursache	Informationsdefizite, Zielkonflikte
Phase	Beurteilung, Bewertung, Verteilung
Modus	konstruktiv, destruktiv

Tabelle 2.1: Klassifikation von Konflikten nach Merkmalen aus Winterhoff-Spurk 2002, S. 113 und Rüttinger 1980

Konflikte lassen sich nach unterschiedlichen Merkmalen unterscheiden. In den weiteren Untersuchungen ermöglicht die Klassifizierung eine Aussage zu treffen, in welchen Konstellationen Konflikte vermehrt auftreten können. Die Kategorisierung ermöglichen es weiterhin, den Konflikt einzuordnen, woran sich meist die Lösung orientiert. So verschiedenartig wie die Konflikte selbst sind, so vielfältig ist deren Möglichkeiten zur Kategorisierung. Tabelle 2.1 stellt eine Möglichkeit dar, wie Konflikte hinsichtlich unterschiedlicher Merkmale klassifiziert werden können. Witt-Bartsch und Becker 2010, S. 170 ff. teilen Konflikte in die Kategorien Beziehungs–, Verteilungs–, Vorgehens–, Werte– und Ziel–Konflikten ein. Eine andere Einteilung nach den Ursachen von Konflikten ist in Tabelle 2.2 zu finden. Diese Einteilung ist vielversprechend, da mit ihrer Hilfe von den Ursachen bzw. Bereichen auf die Wahrscheinlichkeit von Konflikten geschlossen werden kann. Dies wird besonders dann wichtig, wenn später die agilen Modelle aus Kapitel 4 und die Phasenmodelle der Teams in Kapitel 2.5 auf ihre Konfliktträchtigkeit hin analysiert werden.

Ursachengruppe	Ursache
Motivations- und Zielkonflikte	■ unterschiedliche Projekt- oder Bereichsziele ■ unterschiedliche Motivation, die Ziele erreichen zu wollen
Positionskonflikte	■ Hierarchie, Macht ■ Status ■ Einfluss
Organisatorische Konflikte	■ unklare Abläufe oder Aufträge ■ unklare Kompetenzen und Verantwortlichkeiten ■ unbefriedigender Informationsfluss
Rollenkonflikte	■ formelle und informelle Rollen ■ Rollen in verschiedenen Situationen
Beziehungs- bzw. interpersonale Konflikte	■ Normen und Werte ■ Sympathie/Antipathie ■ Chemie
Intrapersonale Konflikte	■ Neid ■ Unsicherheit ■ Angst ■ Über- oder Unterforderung

Tabelle 2.2: Typische Konfliktursachen wie sie nach Kuster et al. 2008, S. 251 auftreten können.

Eine für die Diagnose wichtige Einordnung ist die Unterscheidung zwischen dem **heißen** und **kalten Konflikt**. Wichtiges Unterscheidungsmerkmal zwischen diesen Konfliktypen ist die Art und Hefitgkeit der Interaktion. Während bei einem kalten Konflikt kaum Interaktion zwischen den Parteien vorhanden ist und der Konflikt das Team im Wesentlichen lähmt, ist die Interaktion bei einem heißen Konflikt spürbar. Jeder, der die Parteien beobachtet, kann bei einem heißen Konflikt feststellen, dass etwas in der Luft liegt. Das Knistern kann förmlich wahrgenommen werden. Dies macht es bei einem heißen Konflikt einfacher ihn zu erkennen und durch die Interaktion kann er auch schneller gelöst werden. Bei einem kalten Konflikt hingegen müssen die Rahmenbedingungen geändert werden und die Parteien müssen in Kontakt gebracht werden. (vgl. Kuster et al. 2008, S. 257 f.; Frank 2010, S. 169).

Je nach den Rahmenbedingungen im Unternehmen kann es sein, dass kalte Konflikte begünstigt werden. Gerade wenn Konflikte im Unternehmen als störend und wider der Werte oder den Regeln des Unternehmens empfunden werden, dann ist die Wahrscheinlichkeit groß, dass sich kalte Konflikte bilden und akzeptiert werden (vgl. Müller 2014, S. 27).

Kuster et al. gibt einige Beispiele, welche Einflüsse Konflikte im Projektalltag begünstigen können (vgl. Kuster et al. 2008, S. 250):

- Sachverhalte werden nicht rechtzeitig geklärt.
- Es fehlt an Transparenz und Offenheit.
- Weil Schnittstellen zu wenig miteinander kommunizieren.
- Unterschiedliche Positionen werden nicht gemeinsam ausdiskutiert.

2.5 Zeitbasierte Modelle der Teamentwicklung

Ein Team ist nicht von Beginn an ein Team, sondern es entwickelt sich. Während dieser Entwicklung durchläuft es verschiedene Stadien, die jeweils typische Charakteristika in Hinblick auf die Arbeitsweise, dem Konfliktpotential, der Kommunikation und letztendlich der Effizienz haben. Um komplexe Vorgänge bei der Bildung und Entwicklung von Teams beschreiben zu können, haben sich Modelle entwickelt, welche den Prozess in verschiedene Phasen einteilen. Damit wird es möglich zu beschreiben, in welcher Phase sich ein Team befindet und wie diese Phase bestmöglich unterstützt werden kann.

Ein weithin akzeptiertes Modell der Teamentwicklung ist das Modell nach Tuckman und Jensen 1977. Es ist ein sehr frühes Modell, welches ursprünglich vier Phasen einer Teamentwicklung darstellte und dann auf fünf erweitert wurde. Kauffeld weist in Kauffeld 2001, S. 30 ff. darauf hin, das dieses Modell weithin akzeptiert ist und es von anderen Forschern aufgenommen und in abgewandelter Form wiedergegeben wird. Es ist damit ein sehr etabliertes und in der Praxis durch Studien vielfach überprüftes Modell, dass sich zur Darstellung sehr gut eignet. Ausgehend von diesem Modell haben sich weitere Modelle entwickelt, welche im Grunde eine verbesserte Übersicht bilden und Begriffe plastischer benennen. Ein hier sehr prominentes Modell ist das von Francis und Young 1996, welches auch gerne in einer Teamentwicklungsuhr (vlg. Abbildung 1 Kauffeld 2001, S. 31) dargestellt wird.

Aufgrund der großen Gemeinsamkeiten und der weithin großen Akzeptanz auf das Basismodell von Tuckman und Jensen wird sich diese Arbeit hierauf beschränken.

Phasen der Teamentwicklung nach Tuckman und Jensen

Das Modell beschreibt die Entwicklung eines Teams in Phasen. Das Modell nimmt an, dass die Phasen nacheinander durchlaufen werden müssen. Jede Phase hat dabei seine Eigenheiten in der Art der Interaktion von Gruppenmitgliedern untereinander. Die Phasen ermöglichen zu erwartenden Probleme durch die gerade statt findende Entwicklung im Team einzuordnen und dem Team eine gezielte Hilfestellung bei der Lösung von Probleme zu geben. Tuckman und Jensen teilen den Entwicklungsprozess in 5 Phasen ein: Forming, Storming, Norming, Performing und Adjourning. Letztere Phase war im ersten Modell nicht enthalten, wurde jedoch in Tuckman und Jensen 1977 eingeführt. Sie beschäftigt sich mit der Auflösung des Teams nach getaner Arbeit und ist daher für diese Arbeit nicht von belang und wird hier nicht wiedergegeben. Neben der ursprünglichen Beschreibung in Tuckman und Jensen 1977 greifen verschiedene Autoren die Teamentwicklung auf, die zum Teil eine spezifische Blickrichtung auf die Entwicklungsphasen haben (vgl. Gloger und Rösner 2014, S. 45 ff.; Gessler 2012, S. 238 ff.; Francis und Young 1996, S. 21; Steinmann und Schreyögg 2005, S. 631; Nerdinger, Blickle und Schaper 2014, S. 105). Für einen groben Überblick über die Kernelemente der Phasen ist die Team–Entwicklungs–Uhr in Abbildung 2.1 empfohlen.

Forming

In Organisationen werden Gruppen meist zusammengestellt, ohne dass die Mitglieder hierauf einen Einfluss haben. Sobald die neuen Gruppenmitglieder die Information erhalten, dass sie zur Gruppe gehören, ist dies für sie eine Situation, die sie nicht komplett abschätzen können und sie werden daher verunsichert sein. Die Mitglieder werden versuchen sich in der Gruppe zurecht zu finden, was im Wesentlichen in 3 Schritten erfolgt: Die Aufgabe selbst erfassen, sich selbst einordnen und letztendlich sich selbst individuell ausrichten. Dabei werden sich die Mitglieder über ihre gewünschte informelle Rolle im Team klar.

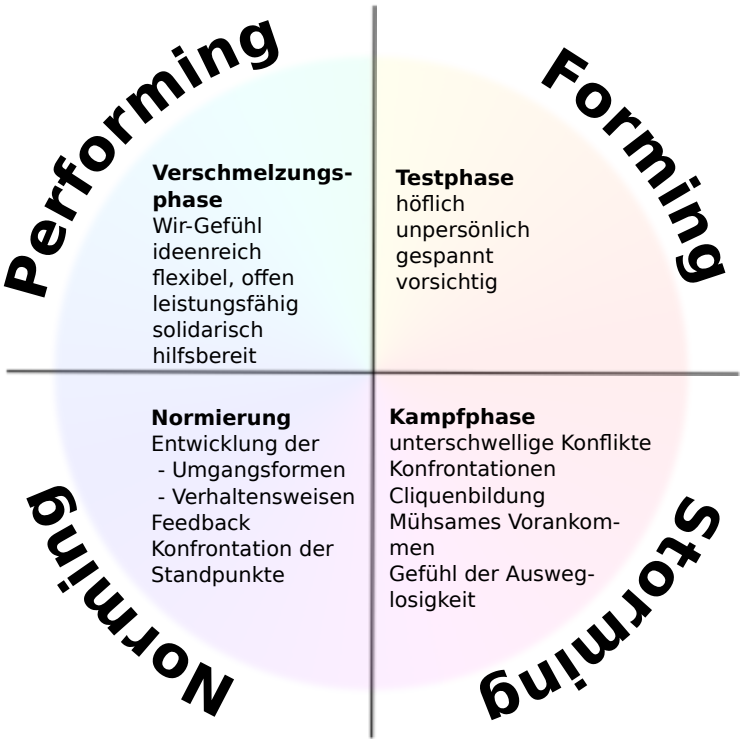


Abbildung 2.1: Teamentwicklungsuhr, zeigt beispielhaft die Phasen zur Teamentwicklung. Dabei beginnt das Team um kurz nach 12 und durchläuft im Uhrzeigersinn die Phasen. Die wichtigsten Eigenheiten der Phasen sind in der Uhr mit aufgeführt. Abbildung angelehnt an Francis und Young 1996 und Abbildung 1 Kauffeld 2001, S. 31.

Kommunikation und Interaktion Die Teammitglieder verhalten sich individuell und unabhängig. Sie beobachten sich gegenseitig, um von den anderen Mitgliedern zu erfahren, wie sie kommunizieren, welche Einstellungen, Werte und letztendlich Arbeitshaltung sie haben. Dabei kann das Teammitglied selbst sehr passiv oder aktiv sein. Jedes Mitglied erkundet dabei wie es seinen Platz und seine Rolle im Team findet. Es entstehen erste lockere Beziehungen der Mitglieder untereinander.

Anforderung an die Führung Die Führung muss für die Zusammenkunft und eine Anregung der Kommunikation im Team sorgen. Sie wird daher Meetings einberufen und dafür Sorgen, dass sich das Team oft trifft, um erste Übereinkünfte zu treffen. Die Führung muss jedoch darauf achten, sich nicht in den Vordergrund zu drängen und wichtige Entscheidungen im Team zu belassen. Dies fördert die Selbstorganisation im Team, was gerade für agile Teams von Bedeutung ist. Die Führung gibt dabei bestenfalls die grobe Richtung vor und greift nur ein, wenn es notwendig ist.

Storming

Das Team gewinnt Vertrauen zueinander, wodurch sich die einzelnen Personen öffnen und ihre Persönlichkeit deutlicher zum Tragen kommt. Das Team beginnt sich mit Werte und Normen sowie die Rollen und Aufgabenverteilung in der Gruppe auseinanderzusetzen. Dabei legen sie fest, wie sie in Zukunft vorgehen wollen, wo oftmals Ideen und Vorstellungen miteinander konkurrieren. Die Rolle der Führung wird genau beobachtet und möglicherweise infrage gestellt. Die Gruppe entscheidet über das Wie der Zusammenarbeit, indem sie folgende Fragen stellt:

1. Wer übt Kontrollfunktionen aus?
2. Wie werden Kontrollfunktionen ausgeübt?
3. Was geschieht bei Verstößen gegen Gruppenregeln?

In dieser Phase festigt sich auch die innere Einstellung des einzelnen Mitgliedes zur Gruppe und der Aufgabe. Im schlimmsten Fall kommt es zu einer innerlichen Kündigung und das Mitglied ist dann für das Team nicht mehr brauchbar. Die Phase ist manchmal auch geprägt durch das Entlassen einzelner Mitglieder aus der Gruppe, wenn bemerkt wird, dass diese Mitglieder nicht zur Gruppe passen.

Kommunikation und Interaktion Die Phase ist gekennzeichnet durch einen starken Beziehungsaufbau der Mitglieder untereinander. Typisch für diesen Abschnitt sind offene wie auch verdeckte Konflikte der Gruppenmitglieder untereinander. Bei den Konflikten handelt es sich vornehmlich um Positionskonflikte (siehe Tabelle 2.2 auf Seite 18). Oft ist die Konfliktursache auch in der Verteilung der informellen Rollen zu finden.

Im Team bilden sich zum Teil Untergruppen und Koalitionen, welche z.B. eine Meinung vertreten oder sich besonders nah fühlen. Gerade in dieser Phase ist es nicht untypisch, dass die Führung in Frage gestellt wird, da sich die Rollen im Team festigen. Auch werden in dieser Phase Grenzen gegenüber der Führung und Gruppenmitgliedern ausgetestet.

Anforderung an die Führung Die Führung ist in dieser Phase besonders gefragt, Fingerspitzengefühl zu beweisen. Sie sollte sich nach Möglichkeit im Hintergrund halten, damit das Team die Konflikte unter sich löst und damit von der Führungskraft keine Partei bevorzugt behandelt wird. Die Führung

agiert in dieser Phase überwiegend als Coach und Beobachter. Meist regeln sich die Konflikte im Team von selbst, es ist jedoch gut, wenn in dieser Phase eine gute Konfliktkultur im Team vorherrscht; diese gilt es zu fördern. Wichtig für die Akzeptanz als Führungsperson in dieser Phase ist der Aufbau von Vertrauen zum Team und die Zusammenarbeit der eventuell vorhandenen informellen Führer im Team. Es kann auch notwendig werden, sich von einzelnen Teammitglieder zu trennen oder sie zu tauschen.

Norming

Die Rollenverteilung ist geklärt und das Team nimmt die Arbeit auf. Es wird sich vermehrt auf seine Aufgabe konzentrieren. Das Team entwickelt gemeinsame Ziele und wird sich mit seiner eigenen Leistung kritisch auseinandersetzen. Es wird sich bemüht, die Arbeit effektiver zu gestalten und es entsteht Routine. Die Phase kann zeitintensiv sein, wie folgende Feststellung widerspiegelt: „Es lässt sich nicht umgehen, dass die Organisationsphase Zeit in Anspruch nimmt. (…) sie müssen Lösungsstrategien einüben, um dauerhafte „Spielregeln“ zu begründen …“ (Francis und Young 1996, S. 23).

Kommunikation und Interaktion In dieser Phase lernt das Team mit Problemen umzugehen und sie zu lösen. Nur wenn dies gelingt, kann es zu einer hohen Leistung kommen. Weiterhin wächst das gegenseitige Verständnis und die Akzeptanz untereinander, es entsteht eine zunehmende Identifikation mit dem Team. Dazu trägt auch bei, dass sich eine positive Feedbackkultur etabliert und die Kommunikation offener wird. Die Interaktion und Kommunikation dreht sich um die Sache selbst. Das Team wirkt deutlich harmonischer und gefestigter. Das Team stellt sich Regeln und Vereinbarungen auf, die es auch selbst überwacht.

Anforderung an die Führung Die Führung kann sich in der Phase aus dem Geschehen weiter zurücknehmen. Das Team wird die neu gewonnene Freiheit im Rahmen seiner Kompetenzen nutzen, um autonomer zu agieren. Die Führung kann jetzt Aufgaben an das Team delegieren und durch einen partizipativen Stil das Team in seiner Aufgabe unterstützen. Allerdings ist es auch wichtig, dass sich die Führung an selbst gesetzte Regeln hält und diese auch gegenüber anderen Hierarchiestufen durchsetzen kann.

Performing

Das Team arbeitet als eine Einheit zusammen. Für das Team ist das wichtigste Gut, die Aufgabe zu erfüllen, danach richtet es seine eigene Struktur und seine Rollen aus. Das Team arbeitet reibungslos zusammen und hat eine klare Vorstellung von seinem Aufgabenbereich. Diesen nimmt es voll und ganz an und benötigt in seinem Kompetenzrahmen weder Überwachung noch übermäßige Hilfe.

Kommunikation und Interaktion Es hat sich ein Team gebildet, das eng zusammenarbeitet und das einen starken Zusammenhalt pflegt. Es ist ein „Wir"-Gefühl im Team entstanden, das nach außen hin repräsentiert und verteidigt wird. Die Mitglieder ziehen an einem Strang und erreichten damit eine hohe Leistung. Die Kommunikation ist offen, wertschätzend und kaum von Konflikten geprägt. Die Normen und Standards im Team sind gefestigt und das Team reflektiert seine Arbeitsweise um sich stetig zu verbessern. Das Team unterstützt sich gegenseitig, es ist ein Geben und Nehmen.

Anforderung an die Führung Die Führung lässt das Team die Entscheidungen im Rahmen seines Aufgabengebietes selbständig treffen und wird daher noch ein wenig mehr das Team loslassen. Sie stellt jedoch sicher, dass der Leistungsstandard gehalten wird und sorgt für die Rahmenbedingungen. Die Führung wird dadurch mehr zum Dienstleister für das Team, gerade wenn es auch darum geht, den Kontakt zur Umwelt herzustellen oder Entscheidungen die nicht in der Kompetenz des Teams liegen, zu treffen. Typischerweise wird die Führung Arbeiten an das Team delegieren und eine unterstützende Rolle einnehmen.

Eigenschaften des Modells

Auch wenn das Modell in der Literatur sowie im Bereich der Fortbildung gerne verwendet wird, hat es in der Praxis einige Nachteile, die nicht vergessen werden dürfen. Gloger und Rösner 2014, S. 45 weisen darauf hin, dass die reale Welt sich selten in Schubladen abbilden lässt. Mit Schubladen sind hier die einzelnen Phasen und die in diesen Phasen typischen Probleme gemeint. Er warnt vor Stereotypisierung und einer damit verbundenen Vorverurteilung. Dies ist potentiell dann gefährlich, wenn Situationen falsch eingeschätzt wer-

den und z.B. Konflikte einen anderen Grund haben, als sie auf die Phasen der Teambildung zurückzuführen wären.

Das Modell sieht in seinen Phasen einen linearen und strukturierten Entwicklungsablauf vor. In der Praxis ist dies jedoch nur selten anzutreffen. Die Team-Uhr aus Abbildung 2.1 deutet dies schon durch ihren farblichen Verlauf an, die Phasen werden schleichend durchlaufen und im Team treten dann Phänomene aus mehreren, oftmals benachbarten Phasen auf (vgl. Francis und Young 1996). Danach ist es auch so, dass die einzelnen Phasen in Gruppen immer wieder durchlaufen werden können. Dies kann viele Gründe haben, oftmals anzutreffen ist:

- Mitglieder kommen oder verlassen das Team, was dazu führt, dass sich die Teammitglieder wieder formieren müssen. Dies kann auch dazu führen, dass die Führung, Rollen, Regeln, … wieder angezweifelt werden.
- Die Führung wird angezweifelt, wenn besondere Ereignisse eintreten. Dann entbrennt unter Umständen ein erneuter Rollenkampf im Team.
- Sich ändernden Rahmenbedingungen können es erforderlich machen, das sich Normen und Regeln verändern.
- …

Untersuchungen haben weiterhin gezeigt, dass die Phänomene, welche in den Gruppenphasen auftreten, nicht auf diese fixiert sind. Vielmehr können Phänomene (wie z.B. Unsicherheit, Angst, Rollenkämpfe, …) in beliebiger Reihenfolge und in beliebigen Phasen auftreten, womit gerade der Rückschluss von Phänomenen auf die Gruppenphase nicht möglich ist (vgl. Ardelt-Gattinger und Gattinger 1998; Kauffeld 2001, S. 30 f.).

Gerade bei langjährig zusammenarbeitenden Teams kann das Phasenmodell auch trügerisch sein. Solche Teams befinden sich in der Performing-Phase, jedoch können in solchen Teams dann Symptome einer Teamkrise auftreten. Solche Symptome sind beispielsweise unzureichende Kommunikation, ineffektive Besprechungen oder verminderte Leistung. Diese Effekte sind jedoch nicht unbedingt auf einen Rückschritt in den Teamphasen zurückzuführen, sondern haben Ursachen wie z.B. mangelnde Motivation oder dass sich das Team nicht mehr bewusst ist, das es die Zusammenarbeit pflegen muss (vgl. Kauffeld 2001, S. 31; Comelli, Rosenstieland und Nerdinger 2014). Für ein Team ist es demnach immer wichtig, an sich selbst zu arbeiten und darauf zu

achten, dass es auch nach längerer Zusammenarbeit sich mit sich selbst beschäftigt. Hierdurch schärft es seinen eigenen Blick auf Konflikte im Team und kann erkennen, wenn Korrekturmaßnahmen für negative Entwicklungen einzuleiten sind. Der Impuls hierzu muss nicht immer von innen kommen, es können auch äußere Anstöße genutzt werden.

2.6 Rollen-basierte Modelle der Teamentwicklung

Unsere heutige Arbeitswelt ist geprägt durch die Arbeitsteilung. Dadurch wird es möglich, komplexe Probleme, welche unterschiedlichste Arten von Wissen wie auch Erfahrungen oder persönliche Eigenschaften benötigen, zu meistern. In Teams ist auch eine Form der Arbeitsteilung vorzufinden; es bilden sich unterschiedliche Rollen heraus, die von den Rollenträgern besetzt werden. „Eine Rolle kann definiert werden als ein Bündel von **Verhaltenserwartungen**, die von anderen an einen Positions- oder Statusinhaber herangetragen werden." (Steinmann und Schreyögg 2005, S. 612). Durch diese Verhaltenserwartung wird es möglich, für die Teammitglieder Orientierung zu schaffen, da sie wissen, was die Umwelt von ihnen erwartet und im Umkehrschluss auch die anderen Teammitglieder wissen, was sie vom Rollenträger erwarten können (vgl. Aronson, Wilson und Akert 2014, S. 312, Gerrig und Zimbardo 2015, S. 651).

Die Zuordnung zu den einzelnen Teammitgliedern muss nicht immer eindeutig sein, es können daher sowohl Mehrfachbelegungen als auch Personalunion vorkommen. Ähnlich wie bei den Gruppen lässt sich bei den Rollen zwischen formellen und informellen Rollen unterscheiden. Formelle Rollen sind deutlich definiert und offiziell anerkannt (z.B. Abteilungsleiter). Dem gegenüber stehen die informellen Rollen, die sich neben den formellen Rollen ausbilden und mehr oder minder deutlich sichtbar werden. Die formellen und informellen Rollen müssen nicht deckungsgleich sein (vgl. Antons 2000, S. 197). Formelle Rollen innerhalb der Organisation sind unmittelbar sichtbar und können damit klar umschrieben werden. Bei den informellen Rollen ist dies nicht ohne weiteres möglich. Die Organisationspsychologie versucht daher geeignete Modell zu entwickeln, Rollen zu beschreiben und daher die Möglichkeit zu schaffen, sie zu analysieren. Diese Rollen sind in einem Team oftmals als informelle Rollen besetzt, da sie nicht explizit for-

mal festgelegt werden. Es können jedoch formelle Rollen definiert werden, in denen die Rollen des Teams gefordert sind und diese unterstützen.

Die Definition der Rollen in einem Rollenmodell ist nicht immer eindeutig, da es hier wesentlich auf die Art der Fragestellung ankommt. Dabei lassen sich die Rollen in den Modellen mit charakteristischen Eigenschaften oder Aufgaben beschreiben. Anhand der Modelle wird es möglich, das Team zu analysieren (Teamdiagnose) und die Rollenverteilung zu beurteilen und gegebenenfalls Entwicklungen einzuleiten (vgl. Kauffeld 2001, S. 78-91).

In der Literatur gibt es verschiedene Modelle, die einen unterschiedlichen Ausgangspunkt der Forschung und damit der Ausrichtung des Modells haben. Von hoher Bedeutung ist das Modell Belbins Team-Roles Self-Perception Inventory (BTRSPI) (Belbin 1981), welches die Team–Rollen Forschung wiederbelebt und für Forscher interessant gemacht hat. In der Folge entwickelten sich unter anderem das Modell des Team Management System (TMS) von Margereson und McCann aus Margerison 2002; Tscheuschner und Wagner 2008; Tscheuschner 2012 sowie das nach Spencer und Pruss (Spencer und Pruss 1995). Diese Modelle sind weit verbreitet und werden gerne zur Teamentwicklung und für das Coaching von Teams genutzt.

Für diese Arbeit wird das BTRSPI weiter verfolgt, da es sich hierbei um ein weit verbreitetes Modell nach einem psychologisch basierten Ansatz handelt. Psychologisch basiert bedeutet, dass es eine Rolle mit psychologischen Merkmalen der Rollenträger verknüpft. Das Modell selbst ist gut erforscht. Über den praktischen Einsatz gibt es Literatur und Erfahrung.

Dem gegenüber steht das Modell nach Spencer und Pruss, das davon ausgeht, dass bestimmte Funktionen im Team besetzt sein müssen. Diese Funktionen werden dabei aus dem Arbeitsalltag gebildet. Im Vergleich zum BTRSPI wird nicht versucht, die Persönlichkeit der Teammitglieder zu nutzen, um Rollen zu bilden, sondern es werden die notwendigen Funktionen in den Vordergrund gestellt. Dies ist ein wichtiger Aspekt in der modernen Arbeitswelt, denn Spencer und Pruss haben beobachtet, dass sich der Mensch an solche Funktionen unabhängig von seiner Persönlichkeit entsprechend der Erfordernisse anpasst (vgl. Spencer und Pruss 1995, S. 56 ff.). Da dies ein wichtiger Blickwinkel ist, wird dieses Modell in den Untersuchungen genutzt, um besonders die Funktionen im Team zu untersuchen.

Rolle	Gruppenbeitrag	Stärke / Schwäche
	Handlungsorientierte Rollen (Action)	
Macher (SH)	Er drängt die anderen zum Handeln, hat den Mut, Hindernisse zu überwinden	dynamisch, pragmatisch, stressresistent; aber: ungeduldig, neigt zur Provokation
Umsetzer (IMP)	Er setzt Pläne in die Tat um, verfügt über Organisationstalent und praktischen Verstand	diszipliniert, pflichtbewusst, effektiv; aber: häufig auch unflexibel, eigensinnig
Perfektionist (CF)	Der Qualitätskontrolleur kümmert sich um die Details und vermeidet Fehler	sorgfältig, gewissenhaft, pünktlich; aber: zaghaft, kontrollsüchtig, delegiert ungern
	Kommunikationsorientierte Rollen (Social)	
Koordinator (CO)	Er ist der ideale Teamleiter, fördert Entscheidungen, verstärkt gute Ideen	ruhig, selbstsicher, kontrolliert; aber: nur durchschnittliche Fähigkeiten
Teamarbeiter (TW)	Als Helfer im Hintergrund verbessert er die Kommunikation und baut Reibungsverluste ab	sensibel, sanft, kooperativ, diplomatisch; aber: selten entscheidungsstark
Weichensteller (RI)	Der Netzwerker bringt nicht nur neue Ideen, sondern richtet die Gruppe nach Bedürfnissen externer Schnittstellen aus	Enthusiastisch, neugierig, kommunikativ; aber: verliert schnell Interesse und ist zu optimistisch
	Wissensbasierte Rollen (Thinking)	
Erfinder (PL)	Er ist der Spinner der Truppe, bringt frische Ideen, denkt quer und provokant	unorthodox, individualistisch; aber: oft abgehoben, ignoriert formale Vorgaben
Beobachter (ME)	Der Skeptiker untersucht die Vorschläge vor allem auf ihre Machbarkeit und verliert nie die Bodenhaftung	zäh, nüchtern, klug, strategisch; aber: wenig inspirierend und motivierend, bremst andere schon mal aus
Spezialist (SP)	Er ist der Tüftler, steuert das nötige wie stets aktuelle Fachwissen bei	selbstbezogen, engagiert; aber: verliert sich oft in technischen Details

Tabelle 2.3: Teamrollen nach Belbin 1993 übernommen aus Mai 2009, Abschnitt vom 17. Mai und ergänzt (vgl. Belbin Associates 2011)

2.6.1 Belbins Team-Roles Self-Perception Inventory

In den 1970er Jahre führte Belbin empirische Studien durch, als Ergebnis beschrieb sie in Belbin 1981 8 Teamrollen. Das Modell ist vor allem anhand psychologischer Profile der Teammitglieder ausgerichtet (s.a. Stärken und Schwächen der Rolle in Tablle 2.3). Nach erweiterten Studien wurden diese Rollen in Belbin 1993 um den Spezialisten erweitert. Die 9 Rollen (siehe Tabelle 2.3) wurden durch Belbin in 3 Gruppen eingeteilt: Action, Social und Thinking. Auch wenn das Modell nicht frei von Kritik ist (vgl. Kauffeld 2001, S. 87 f.), genießt es eine hohe Akzeptanz und wird daher vielfach für die Teamdiagnose oder zur Bildung eines ausgewogenen Teams genutzt (vgl. Kauffeld 2001, S. 82 ff.; Dick und West 2013, S. 26 ff.; Gessler 2012, S. 250 ff.).

Anhand des Modells wurden vielfach Studien durchgeführt, um das Modell zu bestätigen oder um weitere Erkenntnisse zu erlangen. Eine interessante Studie ist Chong 2007, welche beabsichtigt herauszufinden, ob alle Rollen für ein effektives Team notwendig sind. Die Studie kam zum Ergebnis, dass besonders die Rollen des Erfinders (PL), Koordinators (CO) und Teamarbeiters (TW) einen entscheidenden Einfluss auf die Performance des Teams haben. Diesen Rollen werden später betrachtet und untersucht, wo diese in agilen Teams zu finden sind.

Die Rollen sind oftmals nur als informelle Rollen im Team wiederzufinden, denn die tatsächlichen Rollen werden seltenst so benannt. Erinnern wir uns an Tabelle 2.2 (S. 18), so ist dort die Konfliktursache „Rollenkonflikte“ aufgeführt. Eine typische Ursache ist eine unterschiedliche Zuordnung von formeller und informeller Rolle im Team. Formell vergeben ist meist die Rolle des Teamleiters, der im BTRSPI in der Rolle des Koordinators (CO) erscheint. Diese Rolle ist besonders heikel, auch wenn wir uns zurück an die phasenbasierten Modelle (s.a. Kapitel 2.5) erinnern, bei der die Teamleitung eine wichtige Rolle im Fortschreiten der Phasen spielt.

In einem realen Team wird es aufgrund der Mitgliederanzahl immer eine Personalunion der Rollen geben. Durch die oftmals notwendige Mehrfachzuordnung von Rollen ergibt sich eine weitere Konfliktursache. Ein Mitglied wird demnach in verschiedenen Situationen durchaus in unterschiedliche Rollen schlüpfen, je nachdem welche gefordert ist und wo auch seine Präferenz liegt. Dies kann für die anderen Mitglieder im Team verwirrend oder aber auch problematisch sein, wenn sie sich in ihrer eigenen Rolle be-

droht fühlen. Kann es sein, dass sich z.B. je nach Situation die Rolle des Koordinators in unterschiedlichen Teammitgliedern wiederfinden? Im Verlauf dieser Arbeit wird dies wieder aufgriffen und anhand der agilen Methoden noch einmal untersuchen werden. Bei Konflikten im Team, gerade wenn es Rollenkonflikte sind, werden wir froh sein, wenn sich im Team ein guter Teamarbeiter (TW) findet. Dieser kann so manchen Konflikt erkennen und entschärfen, indem er seine sich positiv einbringt.

2.6.2 Spencer und Pruss

Teamrolle	Beschreibung der Teamrolle
Visionär	Überblickt den Teamauftrag in der Gesamtheit; schwebt über den Wolken und greift nach den Sternen.
Pragmatiker	Gegengewicht zum Visionär; erinnert an praktische Punkte und Schwierigkeiten; zeigt dem Team aber auch, wie man das Unmögliche möglich macht.
Entdecker	Holt Informationen, Materialien und Unterstützung von außen ein; Botschafter des Teams
Herausforderer	Stellt Bestehendes in Frage; pedantisch
Uparteiischer	Außenseiter; Berater
Friedensstifter	Beseitigt Ungerechtigkeiten; legt Meinungsverschiedenheiten bei; sieht Konfliktbereiche voraus und wendet sie ab
Arbeitstier	Macht die Arbeit
Trainer	Stärkt die Moral; Sorgt für neuen Antrieb; erinnert alle an die Aufgabe; spornt an
Bibliothekar	Dokumentiert; zeichnet Aktivitäten auf; Informationsquelle
Vertrauensperson	Beichtvater; jemand bei dem man sich ausweinen kann

Tabelle 2.4: Teamrollen nach Spencer und Pruss 1995 aus (Tabelle 13 Kauffeld 2001, S. 83)

Der erstmals in (Spencer und Pruss 1995) vorgestellte, nach Funktionen orientierte Ansatz, teilte das Team in 10 Rollen (siehe Tabelle 2.4) ein, welche als „Ergebnis von langer Beobachtung und Erfahrung“ (Spencer und Pruss 1995, S. 74) ermittelt wurden. Der funktionale Ansatz wurde vor allem deswegen gewählt, da in der heutigen Arbeitswelt die Arbeit aus Funktionen

besteht, die bestimmte Personen erfüllen. Der Ansatz bietet die Möglichkeit anhand der Funktionen das agile Team zu untersuchen und Schlüsse zu ziehen. Das Modell nimmt weiterhin an, dass immer alle Rollen besetzt sind, nur wie gut sie besetzt sind, lässt es offen.

Spannend im Modell ist vor allem die Beziehung zwischen Visionär und Pragmatiker, da diese konträr zueinander sind. Dies beinhaltet einen Herd für Konflikte und eine gewisse Spannung, ermöglicht es aber auch, verschiedene Seiten der Medaille zu betrachten. Gerade dies kann sehr wertvoll sein, wenn es darum geht, negative Gruppenphänomene wie Groupthink zu verhindern. Auffällig bei diesem Modell ist, dass es keine Rolle gibt, die eine typische Führungsrolle widerspiegelt. Vielmehr besteht die Gefahr, dass die Führungskraft nicht vergebene Rollen übernimmt. Dies hat eine Konzentration von Rollen in einer Person zur Folge, die problematisch werden kann (vgl. Spencer und Pruss 1995, S. 91 f.).

2.7 Zusammenfassung

Das Bilden von und das Arbeiten in Gruppen und Teams ist geprägt von komplexer sozialer Interaktion und Zusammenhängen. Diese können einzeln nicht verstanden werden, was auch in der engen Verzahnung der Modelle ersichtlich wird. Sie bauen aufeinander auf, referenzieren verschiedene Aspekte aus anderen Modellen und sind damit aufeinander angewiesen. Sie beleuchten immer nur einen kleinen Aspekt von einem Team in seiner aktuellen Zusammensetzung, der Veränderung über die Zeit oder spezieller Interaktionen wie Konflikten.

Bei den Rollenmodellen fällt auf, dass der verwandte Ansatz zu einem deutlich unterschiedlichen Ergebnis kommt. Sie haben eine bestimmte Blickrichtung, welche dem Forscher als vorteilhaft erschienen. Dies muss in der täglichen Arbeit nicht zwangsläufig sein, vielmehr sind sie ein Anhaltspunkt, damit ein Team strukturiert werden kann. Keinesfalls sind sie als absolut zu sehen. Wir werden später noch sehen, dass agile Teams eine weiteres Rollenmodell postulieren und es werden dort Gemeinsamkeiten sichtbar werden.

Konflikte aus dem Alltag

Der Zusammenhang zwischen den Modellen und Konflikten ist für den Alltag besonders wichtig, da es Potential für positive Entwicklungen oder gar die Gefahr des totalen Stillstand gibt. Neben den typischen Zielkonflikten (z.B. zwischen Zeit, Qualität und Kosten) stehen in Teams besonders die folgenden Konflikte im Mittelpunkt und sind für die agilen Teams von besonderem Interesse:

- Rollenkonflikte treten besonders während der Entwicklungsphasen in der Storming-Phase auf. Jedoch treten sie auch immer wieder erneut auf, wenn sich das Team ändert oder selbst wenn sich unterschiedliche Randbedingungen ergeben. Typisch sind Konflikte um die formellen und informellen Rollen, gerade wenn es um die Führungsposition im Team geht. Selbst ein akzeptierter Leader kann in bestimmten Situationen in seiner Rolle angezweifelt werden. Dies kann z.B. passieren, wenn er auf einer fachlichen Lösungsmöglichkeit beharrt, die keine breite Zustimmung im Team hat.

- Erwartungen an Rollen können ebenso zu Konflikten führen, denn die Teammitglieder haben bestimmte Erwartungen an ihre eigene Rolle ebenso wie an die Rolle der anderen. Werden die unterschiedlichen Rollen-Modelle betrachten, so kann festgestellt werden, dass diese so vielfältig wie die Menschen selbst sind. Ebenso wird jeder sein eigenes Rollenmodell und die damit verbundenen Aufgaben und Pflichten in seiner Vorstellung haben. Diese gilt es abzugleichen, ansonsten kann es vorkommen, dass Erwartungen nicht erfüllt werden und die Zusammenarbeit ins Stocken gerät.

- Zielkonflikte treten vor allem dann auf, wenn Teammitglieder unterschiedliche Vorstellungen oder Motivationen haben, bestimmte Ziele zu erreichen. Diese Konflikte können auch bedingt durch die unterschiedlichen formellen Rollen sein, womit Teammitglieder bestimmte Interessen vertreten. Das Team kann jedoch nur effizient arbeiten, wenn es sich auf ein Ziel einigt und dabei die einzelnen Ziele der formellen Rollen berücksichtigt. Es muss ein Überziel geben, an dem alle arbeiten. Damit kann die eingesetzte Energie für das gemeinsame Team-Ziel genutzt werden.

- Organisatorische Konflikte treten gerne auf, wenn sich Teams mit ihrer Arbeitsweise und den damit verbundenen Kompetenzen und Ver-

antwortlichkeiten intern, wie extern nicht genügend auseinandersetzen. Dann ist unklar, wer dem Team z.B. Arbeit geben darf, was es wie erledigt oder welche Informationen es bekommt. Dies führt zu vielen Reibungsverlusten und Konflikten, wenn dies nicht rechtzeitig geklärt wird. Dabei muss dieser Konflikt nicht nur das Team selbst betreffen, sondern es entsteht oftmals ein Konflikt mit anderen Parteien mit denen das Team zusammenarbeiten muss.

Besonders kritisch sind die verdeckten Konflikte. Das Team wirkt nach außen hin intakt, kann aber in seiner täglichen Arbeit tief gespalten sein. Gerade nicht Ausgesprochenes kann auf Dauer belasten, daher ist es immer sinnvoll, solche Konflikte aufzudecken oder ganz zu vermeiden. Vielfach lassen sich verdeckte Konflikte finden, wenn sich die Teammitglieder einander vertrauen und eine offen, regelmäßige Kommunikations- und Feedbackkultur pflegen. Dann werden Probleme frühzeitig angesprochen, so das sie sich nicht manifestieren können. Das Team wächst dann an ihnen.

Erkenntnisse für die Forschungsfragen

Von den gruppendynamischen Prozessen werden sich agile Teams von den normalen Teams nicht wirklich unterscheiden. Diese Annahme kann getrost getroffen werden, denn ein agiles Team ist zunächst einfach ein Team, welches eine bestimmte Ordnung und Vorgehensweise mit speziellen Normen pflegt (welche das sind wird im nächsten Kapitel noch geklärt). Die Modelle der klassischen Teams können also auf agile Teams übertragen werden. Die dabei aufkommende Frage ist: Welche Teile der klassischen Modelle sind für agile Teams besonders relevant und wo haben agile Teams durch ihre Struktur und Vorgehensweise Vorteile? Diese Fragen werden später noch geklärt (siehe Kapitel 5); aus der klassischen Sicht kann bereits die Aussage getroffen werden, welche negativen Entwicklungen es aus dieser grundlegenden Sicht gibt.

Negative Entwicklungen

In der Forschungsfrage muss zunächst geklärt werden, worin negative Entwicklungen bestehen können, da dies ein zentraler Kern ist. Aus dem vorangegangenen Kapitel konnten bereits wichtige Erkenntnisse gewonnen werden. Mit jedem untersuchten Feld, seien es Teams und Gruppen allgemein,

Phasen, Konflikte oder Rollen haben sich die Problemfelder gezeigt, die bei der Arbeit in Teams immer vorhanden sind. Das gesamte Feld der negativen Entwicklungen ist dabei sehr groß, doch welche Definition von negativ wird in dieser Arbeit genutzt? Im Kern ist für das Unternehmen ein Punkt besonders wichtig: Unternehmen sind auf eine hohe Leistung des Teams und eine gute Integration in das Unternehmen angewiesen, um ihre Ziele nachhaltig zu erreichen.

Bei den meisten Unternehmen ist das Ziel eine Gewinnerziehlungsabsicht (wie es das Steuerrecht definiert) und wenn Adam Smith Glauben geschenkt werden darf, streben Unternehmen nach Gewinnmaximierung. Heute ist dies bei Organisationen nicht immer der Fall, so gibt es auch Non Profit Organisation, die keinen Gewinn erwirtschaften wollen. Gleichwohl haben auch sie ein Ziel, das es zu erreichen gilt. Das Kernthema ist demnach das zu erreichende Ziel; hier lässt sich überlegen, wie das Ziel am besten erreicht werden kann und es spielt immer Effizienz eine wichtige Rolle, denn je höher diese ist, umso besser und einfacher lässt sich ein Ziel erreichen. Zurück zu den negativen Entwicklungen in Teams, die erkannt werden sollen. Es lässt sich sagen, dass auf das Team im organisatorischen Kontext alles negativen Einfluss nimmt, was die Effizienz und damit den Output des Teams schmälert. Mit dieser Definition kann überlegt werden, wann der Output sinken wird.

Aus den vorangegangenen Überlegungen lässt sich damit ableiten, dass ein eingespieltes Team in der Performing-Phase ohne effizienzmindernde Konflikte als Idealzustand anzusehen ist. Weiterhin ist darauf zu achten, dass negative Effekte im Team nicht zum Tragen kommen. Damit lassen sich folgende negativen Entwicklungen ableiten:

- Das Team ist kein Team, sondern eine Gruppe und bringt daher eine geringere Leistung. Dabei können zwei Fälle unterschieden werden:
  - Das Team befindet sich in der Entwicklung und hat diese noch nicht abgeschlossen. Hier kann nicht von einer negativen Entwicklung gesprochen werden, sondern es muss sich erst zum Positiven entwickeln.
  - Es fehlen die Voraussetzung zur Bildung eines Teams auf Ebene der Organisation.
- Das Team hat Probleme mit negativen Gruppeneffekten, wie Risikoschub, Gruppendenken oder soziales Faulenzen.

- Im Team selbst oder mit dem Team bestehen Konflikte, die ungelöst sind.
- Das Team fällt aus der Performing-Phase in eine frühe Entwicklungsphase zurück.

Beobachtung weiterer negativen Entwicklungen

Die oben aufgeführten negativen Effekte sind gut beschrieben und werden auch während der Teamdiagnose gerne genutzt. Jedoch stellt sich die Frage, ob dies ausreicht, um ein Team zu beschreiben. So kann es sein, dass in einem Team keine Konflikte anzutreffen sind, das Team sich nach Tuckman und Jensen 1977 in der Perfomingphase befindet und die Leistung trotzdem nicht stimmt. Gerade in Entwicklungsteams gibt es viele Randbedingungen, die notwendig sind, damit sie ihre Arbeit erfolgreich umsetzen können. Die dort notwendigen Strukturen sind weit vielfältiger als die der klassischen Teammodelle.

Als Führungskraft im täglichen Umgang mit dem Team wird häufig auf das reine Beobachten zurückgegriffen. Der Grund ist so banal, wie einfach, denn Führungskräfte beobachten das Team immerwährend, wenn sie mit ihm arbeiten. Die direkte Beobachtung von Leistung ist gerade in Teams, bei denen Kreativität und Entwicklung eine entschiedene Rolle spielt, schwierig. Gerade in Entwicklungsteams ist es daher nicht ohne weiteres möglich, die Leistung in Form des Outputs aus dem Team direkt zu betrachten. Neben der reinen Teamleistung spielt noch die Aufgabe und deren Komplexität einen entscheidenden Rolle. Eine Führungskraft wird daher selten durch ihren Blick von außen die Möglichkeit haben, die Leistung des Teams objektiv einzuschätzen. Es müssen andere, beobachtbare Kriterien gefunden werden als die reine Leistung.

Kapitel 3

3 Selbstorganisation

Unterhalte ich mich mit klassischen (Projekt-)Managern über agile Methoden, so beobachte ich Faszination und Skepsis zugleich. Besonders das Loslassen des Teams und ihm zu vertrauen fällt manchem gebeutelten Manager in einer Sandwichposition sichtlich schwer im Kopf. Den Verlust von Einfluss und Kontrolle gegenüber dem Team stellen viele in Frage: Wie kann ein solches Team dann geführt werden und wie funktioniert dieses Team? Gerne antworte ich darauf mit einer Gegenfrage: Wie kann ein Manager, welcher das Team nur durch ein Schlüsselloch beobachtet, sicher sein, dass er die richtigen Schlussfolgerungen und Entscheidungen trifft? Gerade die letzte Frage zeigt, gerade Manager sind meist nur punktuell am Geschehen mit dem Team beteiligt und können viele Probleme oder Lösungswege nicht direkt oder nur mit Zeitaufwand nachzuvollziehen.

Gerade in Zeiten komplexer werdender Technik und eine oftmals wiederkehrenden Anpassung von Strukturen und Arbeitsweisen in Organisationen überfordern das Management im klassischen Sinne zunehmend. Um eine Anpassung durchzuführen muss zunächst die Notwendigkeit gesehen werden und dann müssen weiterhin die Anpassungen geplant und umgesetzt werden. „Complexity theory suggests that self-organisation is the natural "default"beahvior.“ (Coleman 1999, S. 33) weist uns darauf hin, dass die Selbstorganisation in jedem Teil unserer Natur, wie auch Gesellschaft eine treibende Kraft ist. Dies wurde durch die Begründer der agilen Methoden frühzeitig festgestellt und eine der Maßnahmen ist, dass agile Teams selbstorganisiert arbeiten. Dazu verhalf nicht zuletzt die auch im agilen Manifest festgehaltene Beobachtung „Die besten Architekturen, Anforderungen und Designs kommen durch selbst organisierte Teams zustande.“ (Beck et al. 2001).

Sich selbst organisierende Teams unterliegen nach wie vor der Steuerung des Managements und sind in die Organisation eingebettet. Durch die Selbstorganisation werden jedoch die Grenzen der Systeme entlang der Kompetenzen der Teammitglieder und Manager deutlicher abgesteckt. Die Teams erhalten einen höheren Freiheitsgrad bei der Gestaltung ihrer Arbeit (vgl. Cohn 2010b, S. 251 ff.). Dies entlastet das Management, verlagert Entscheidungen zu den Kompetenzträgern und als positiver Effekt sind solche Teams deutlich effektiver und motivierter.

3.1 Definition und Abgrenzung

In der englischsprachigen Literatur finden sich sehr ähnliche Begriffe, die jedoch unterschiedliche Bedeutungen haben. So spricht der englische Orginaltext des agilen Manifests von „self–organizing“. Diese Art der Selbstorganisation bezieht sich auf die tägliche Arbeit. Derby fasst in ihrem Blog die wesentlichen Punkte zusammen: „Aus der Perspektive sozialer Systeme betrachtet, bedeutet Selbstorganisation, dass das Team neue Herangehensweisen entwickelt und sich so an neue Herausforderungen in seiner Umgebung anpasst. Selbstorganisierende agile Teams besitzen eine beschränkte Autorität, um eigene Zusagen auszusprechen, sich zu organisieren und ihre Arbeit aufzuteilen. Sie entwickeln ihre eigene Strategie um Ziele zu erreichen und sie fällen ihre eigenen Entscheidungen mit (wieder eingeschränkten) wirtschaftlichen und organisatorischen Auswirkungen.“ (Derby 2011, übersetzt aus dem englischen)

Damit ist Selbstorganisation im agilen Kontext eine begrenzter Raum, in dem das Team sich selbst organisieren kann. Dieser Raum ist so gewählt, dass es seinen Erfordernissen entspricht und dem Team selbst Sicherheit über seine Randbedingungen gibt. Das Team macht sich diesen Raum zu eigen. Die Anbindung zur Organisation findet über Schnittstellen statt, die vom Team mitgestaltet werden können.

3.2 Selbstorganisation beschreiben

Selbstorganisation ist zunächst nur schwer beschreibbar und die Ergebnisse sind nicht direkt vorhersagbar. Soziale Interaktionen zwischen Menschen sind ein ebenso komplexes wie faszinierendes Muster, jedoch ist dies auch

nur schwer zu beschrieben. In unserem täglichen Leben sind wir vielfach schon mit Selbstorganisation in Berührung gekommen. Denken wir an einen Verein oder an das Team in der Arbeit, so werden dort immer Prozesse der Selbstorganisation ablaufen. Wir kennen Selbstorganisation aus unserer täglichen Erfahrung und sie ist tief in uns verwurzelt.

In einer immer komplexer werdenden Welt der Organisationen wurde lange Zeit versucht, die Probleme in immer kleinere Teile zu teilen und diese Teilbereiche zu beschreiben. Selbstorganisation ist ein komplexes Thema, das nur dann erfasst werden kann, wenn eben diese Zerlegung nicht statt findet. Diesen Ansatz verfolgt die systemische Herangehensweise, oftmals auch als Systemtheorie bezeichnet. Sie wählt dabei den Ansatz, das System als Ganzes zu erfassen und die Interaktion in und mit diesem System zu untersuchen. Die Systemtheorie ist dabei nicht auf einen Bereich fixiert sondern wird vielfältig z.B. in der Technik, Biologie, Psychologie, Soziologie, … (die Liste lässt sich beliebig fortsetzen) genutzt. Im Grunde ist sie weniger eine Theorie im klassischen Sinne, sondern eine Art von Denkansatz: „Vernetzes statt kausales, dynamisches statt statisches Denken ist heute die Grundlage so gut wie jeder Diskussion über Management und Führung.' (Gloger und Rösner 2014, S. 34).

3.2.1 Voraussetzungen nach Gloger und Rösner

Gloger und Rösner beschreibt fünf Voraussetzungen, die notwendig sind, um Selbstorganisation stattfinden zu lassen (vgl. Gloger und Rösner 2014, S.37):

1. Eine für das System existenzielle Mission / Aufgabe
2. Einen die Systembalance gefährdeten äußeren / inneren Anpassungsdruck
3. Ein dem System eigene und bewusste / unbewusste Anpassungsbereitschaft
4. Das Geschehenlassen und Installieren systemeigener Strukturen und Regulationsprozesse
5. Die Integration der Anpassung in die neu entwickelte Systemrealität

Im Wesentlichen beschreiben diese Aussagen zwei Komponenten: Der Punkt 1 beschreibt die Aufgabe, welche aus der Definition von Teams bereits

bekannt ist: Das gemeinsame Ziel des Teams. Die Punkte 2 bis 5 beschreiben wesentliche Punkte, welche für Veränderungen im Unternehmen wichtig sind. Veränderungen werden nach den klassischen Theorien meist mit einem Auftauen (vgl. Lauer 2014, S. 68) bzw. mit einer Dringlichkeitsvermittlung (Kotter und Rathgeber 2011) begonnen. Dieser Anpassungsdruck wird in den klassischen Theorien vom Management, in der agilen Welt vor allem vom Markt (oder seinem Stellvertreter) und dem Team selbst ausgeübt. Das ganze System muss Änderungen selbst zulassen, geschieht dies nicht, wird auf lange Sicht die Wettbewerbsfähigkeit des Unternehmens leiden. Schlussendlich schließt der Punkt 5 die Veränderung ab, indem sie in das System aufgenommen wird. In der agilen Literatur ist das klassische Freezing nicht zu finden, denn ein agiles System bedeutet, es ist beständig im Wandel. Dies ist eine wesentliche Eigenschaft des agilen Systems.

3.2.2 Voraussetzungen nach Eoyang

Eoyang beschreibt in ihrer Arbeit (Eoyang 2001) das CDE Modell, Darin werden die Voraussetzungen zur Selbstorganisation zu **C**ontainer, Significant **D**ifferences (Unterschiede) und Transforming **E**xchange (Austauschvorgänge) zusammengefasst. Das Modell wird hier vorgestellt, da es sich gut eignet, um Probleme zu strukturieren und anhand der 3 Felder zu identifizieren (vgl. Cohn 2010a, S. 253 ff.).

Der CONTAINER ist die Begrenzung, welche das System von seiner Umwelt absteckt. Im klassischen Fall von Teams ist der Container das Team oder eine Arbeits- und Projektgruppe im Unternehmen. Die Selbstorganisation findet nur innerhalb dieses Containers statt, so dass die Grenzen des Containers besonders zu beachten sind. Wird versucht, Selbstorganisation über die Containergrenzen hinaus auszudehnen, ist dies meist nicht von Erfolg gekrönt, da die zu ändernden Dinge nicht in der Macht der Teammitglieder stehen.

In einem realen Team wird es UNTERSCHIEDE der einzelnen Teammitglieder untereinander geben. Unterschiedliches Fachwissen, Erfahrungen oder Einflüsse formen einen individuellen Mitarbeiter. Allerdings kann ein Einzelner die Aufgabe nicht in der notwendigen Zeit erfüllen, so dass er auf andere angewiesen ist, die in einem anderen Bereich wichtiges Wissen haben. Unterschiede sind daher wichtig, um einer Vielzahl von unterschiedlichen Anforderungen gerecht zu werden und dennoch zum Ergebnis zu kommen.

Um die Unterschiede zu nutzen oder auszugleichen, treten die Mitglieder innerhalb des Containers in einen AUSTAUSCHVORGANG ein. Durch Kommunikation kann Wissen ausgetauscht werden, jedoch können auch Güter, Geld oder z.B. Motivation ausgetauscht werden. Der Inhalt des Austausches ist sehr vielfältig und ist nicht vorgegeben. Es kann sich sowohl um etwas materielles, wie auch etwas immaterielles handeln.

3.3 Zusammenfassung

Die Selbstorganisation eines Teams läuft immer einher mit seiner Teamentwicklung. In den Phasen der Teamentwicklung können wir die Selbstorganisation wiederfinden, denn sie findet in einem gewissen Umfang in jedem Team statt. Im Modell nach Tuckman und Jensen aus Kapitel 2.5 ist gerade die Phase des Normings stark von Selbstorganisation geprägt. Wird Selbstorganisation als einen kontinuierlichen Prozess des Neu- und Umgestaltens begriffen, so ist dies auch in der Performingphase als Teil der täglichen Arbeit von agilen Teams zu finden.

Selbstorganisation in Unternehmen ist nicht selbstverständlich, denn es gibt viele Barrieren, die bewusst, wie unbewusst in die Selbstorganisation eingreifen. Gerade für Mitarbeiter, welche aus einem Umfeld mit klassischen Management stammen, können zusätzliche Barrieren entstehen. Balkema und Molleman 1999, S. 136 ff. sprechen hier von unterschiedlichen Barrieren. Im Management verändern sich die Aufgaben eines Managers zu einem Coach oder Facilitator. Der Mitarbeiter kann eine andere Einstellung zur Arbeit haben, die eine Verantwortungsübernahme ablehnt oder bei der die neue Organisationsform der eigenen Identität des Mitarbeiters zuwider ist. Bei den Kompetenzen werden zusätzlich noch managementskills und soziale Kompetenz gefragt. Zuletzt muss auch die Reichweite der Selbstorganisation an die zu erfüllende Aufgabe angepasst sein.

Erkenntnisse für unsere Forschungsfragen

Selbstorganisation ist ein inhärentes Merkmal agiler Methoden. Das Fehlen oder nur unzureichende Funktionieren der Selbstorganisation sind Probleme für agile Teams und sie werden dann kaum den an sie gestellten Erwartungen gerecht. Daher ist es wichtig, dies zu beobachten und gegebenenfalls korrigierend einzugreifen. Meist ist es jedoch nicht einfach ersichtlich, ob es

wirklich an der Selbstorganisation im Team mangelt oder die Rahmenbedingungen für Selbstorganisation nicht stimmen.

Gerade bei den Voraussetzungen nach Gloger und Rösner sind meist die Kultur im Unternehmen oder die Rahmenbedingungen nicht ideal, weswegen es ein Hemmnis gibt. Welches dies ist, kann jedoch nicht pauschal beantwortet werden, sondern muss individuell an der Situation beobachtet werden. Es kann auch an inneren Voraussetzungen fehlen, hier ist besonders der fehlende innere Anpassungsdruck sowie die Anpassungsbereitschaft zu nennen. Ein immer wieder beobachtetes Problem ist eine länger Zusammenarbeit von Teams. Es schleift sich eine Art von Alltag und Bequemlichkeit ein. Das Team wird sich dann auf alten Erfolgen ausruhen, der Anpassungsdruck von innen heraus sinkt und die Anpassungsbereitschaft wird dann abnehmen. Dabei ist es schwierig zu entscheiden, ob alles gut läuft oder ob Anpassungen notwendig wären. Anzeichen für ein Abnehmen des Anpassungsdrucks sind sicher weniger effektiv durchgeführte Retrospektiven, die zudem als überflüssig empfunden werden.

Bei den Voraussetzungen nach Eoyang fällt auf, dass der dynamische Vorgange im System der Austauschvorgang ist. Die Container sowie die Unterschiede sind relativ statisch und ändern sich über die Zeit nur wenig. Sie sind vielmehr das Setting für die Selbstorganisation, während die Austauschvorgänge das Leben mit den Containern und Unterschieden sind. Die Austauschvorgänge lassen sich beobachten und analysieren, da sie zumeist eine Interaktion der Teammitglieder darstellt. Ein Indiz für ein Problem können gestörte Austauschvorgänge sein oder sich über die Zeit verändernde Austauschvorgänge. Wird das lang zusammenarbeitende Team aufgegriffen, so werden dort Austauschvorgänge vermindert werden, da das Team sich so eingerichtet hat, dass es möglichst wenige benötigt. Dies kann förderlich, wie auch hinderlich sein, vor allem wenn es um Entscheidungen geht.

Auch die Zusammensetzung des Teams ist von entscheidender Bedeutung, da dies einen wesentlichen Einfluss auf die Unterschiede in diesem Modell ausmacht. Sind die Unterschiede zu gering, so werden wenige Austauschvorgänge statt finden und das Team verfügt zudem über vielfache Redundanz. Es wird dann zwar in der Lage sein, spezielle Aufgaben besonders gut zu bearbeiteten, jedoch außergewöhnliche werden zu Problem. Sind die Differenzen zu groß, können Konflikte im Team zunehmen und die Mitglieder sprechen keine gemeinsame Sprache mehr. Ebenso kann es dann vermehrt zu zwischenmenschlichen Problemen führen, wenn sich z.B. ein alleiniger

Leader herausbildet und dieser dann das Team komplett anführt. Dies ist weniger wünschenswert, da die Hypothese „The degree of shared leadership in a team is positively related to the team performance." (Carson, Tesluk und Marrone 2007, S. 1224) durch die Studie starke Unterstützung erhielt.

Der Begriff des Containers gibt zunächst keinen brauchbaren Hinweis, wie daran negative Entwicklungen erkannt werden können. Jedoch ist zu sagen, das es gerne die Container sind, die ein Team in seiner Selbstorganisation einschränkt und manche positive Entwicklung unmöglich macht. Dies heraus zu finden ist für das Management von besonderem Interesse, denn eine Veränderung des Containers kann durchaus die Lösung sein. Ein klassisches Beispiel für solch einen ungünstig gewählten Container ist ein agiles Team mit einer eigenständigen Testabteilung. Zwar ist es zu bevorzugen, wenn Test unabhängig von der Entwicklung ablaufen, jedoch finden diese dann nicht entwicklungsbegleitend statt und werden immer nach Abschluss der Arbeit erstellt, oder wenn die Zeit nicht mehr reicht, einfach weggelassen. Dies baut mit der Zeit eine nicht mehr einzuholende technische Schuld auf. Eine Veränderung des Containers, so dass einige Mitglieder aus der Testabteilung direkt im agilen Entwicklungsteam tätig sind, könnte dieses Problem wirkungsvoll lösen. Die Frage ist, wie lässt sich der falsche Container erkennen? Eine Möglichkeit ist sicher ein Beobachten des Teams und der Retrospektiven, um daraus zu erkennen, wenn das Team Probleme erkennt, die es nicht selbständig lösen kann. Eine gezielte Frage an das Team kann hier auch helfen: Kann das Problem XY gelöst werden, wenn Strukturen umgebaut werden?

Mitarbeiter die neu zum Team hinzustoßen und die die Form der Selbstorganisation nicht kennen, können das Team negativ beeinflussen, wenn sie wenig flexibel und anpassungsfähig sind. Selbstorganisation verlangt eine erhöhte Verantwortungsübernahme seitens des Mitarbeiters, die auch genommen werden muss. Gerade wenn sich Mitarbeiter machlos gegenüber Veränderungen fühlen oder notwendige Veränderungen nicht anstoßen können, werden sie vieles hinnehmen und sich auf einen gefrusteten Standpunkt zurückziehen und Verantwortung nicht mehr wahrnehmen. Sie haben das Gefühl, dass die Verantwortung von ihnen nicht getragen werden kann, da eine Veränderung nicht in ihrem Machtbereich liegt. Für eine sinnstiftende Arbeit (solche sucht der Mensch in seiner Arbeit) benötigt es Freiräume, die gestaltet werden können. „Menschen wollen durch ihre Arbeit sich selbst und ihre Umwelt verändern …" (Sprenger 2009, S. 230) und wenn dies nicht zugelassen wird, werden sie resignieren und Demotivation wird entstehen.

Kapitel 4

4 Agile Methoden

Die Ursprünge der agilen Methoden reichen schon in das vorherige Jahrhundert zurück. Die erste Systematik und damit auch der Startschuss zu einer weiteren Verbreitung wurde durch Kent Beck mit seinem Buch *Extreme programming explained: embrace change* (Beck 1999) gelegt. Durch diese Veröffentlichung wurde der grundlegende Gedanke und das „Warum“ für die agilen Methoden aus Managementsicht beschrieben.

Dies brachte eine Bewegung auf, die schlussendlich das agile Manifest[1] (Beck et al. 2001) hervorbrachte. Das Manifest beschreibt dabei nur, welchen Werten in der agilen Entwicklung Vorzug gewährt werden soll. Es beschreibt eine Art von Kultur, die ein Unternehmen für Teams oder das gesamte Unternehmen übernehmen kann. Stellvertretend für agile Frameworks wurde dies für das Scrum-Framework in Maximini 2015 durchgeführt. Die Frage nach der Kultur ist wichtig, denn nur wenn die Vorbedingungen im Unternehmen erfüllt sind, lässt sich agile Softwareentwicklung etablieren.

Das Wort Agilität kommt ursprünglich aus dem lateinischen von „agilis“ für beweglich und kennzeichnet damit eine Eigenschaft der Konzepte hinter der agilen Softwareentwicklung. Das Zitat „An agile process is both light and sufficient. The lightness is a means of staying maneuverable.“ (Cockburn 2006) vereinigt dabei das Prinzip des effizienten Ressourcennutzung durch Aufwandsminimierung wie auch der dadurch erzielten Möglichkeit zur schnellen Reaktion auf Ereignisse wie neue Technologien oder geänderte Kundenwünsche.

[1] Nachdruck des Textes in deutscher Übersetzung im Anhang A

Aus diesem Grundgedanken heraus haben sich unterschiedliche Vorgehensweisen, Frameworks oder Teileaspekte der Softwareentwicklung entwickelt. Einige der wichtigsten sind unter anderen:

- Agiles Testen (Baumgartner et al. 2013)
- Agile Unified Process (Edeki 2013)
- Crystal (Cockburn 2004)
- Extreme Programming (XP) (Beck 1999; Larman 2001)
- Feature Driven Development
- Behavior Driven Deleopment
- Test Driven Development
- Kanban
- Scrum (Rising und Janoff 2000)
- Scrumban
- …

4.1 Kennzeichen agiler Methoden

Die agile Welt ist für viele Manager und Angestellte fremdartig. Die agilen Methoden stellen vielfach erprobte Muster und Werteordnungen in Frage. Gerade für Menschen, die nur in der klassischen Welt gearbeitet haben, ist es nur schwer nachvollziehbar, wie Entwicklung mit hoher Qualität aber mit weniger Regeln und z.B. Dokumentation funktionieren soll. Einen guten Überblick über die wesentlichen Unterschiede für Teilaspekte gibt uns Nerur, Mahapatra und Mangalaraj 2005, S. 75 welche in Tabelle 4.1 dargestellt sind.

Da das Feld agile Methoden mittlerweile sehr unübersichtlich ist und ständigem Wandel unterliegt, wird sich auf wichtige Gemeinsamkeiten in der folgenden Beschreibung beschränkt. Es ist jedoch wichtig zu verstehen, warum und wieso diese Gemeinsamkeiten besonders wichtig für das Funktionieren der Methoden sind.

	Traditionelle, plangetriebene Methoden	Agile Methoden
Fundamentale Annahmen	Systeme sind zur Gänze spezifizierbar, vorhersagbar und können durch akribische und extensive Planung erzeugt werden.	Hochqualitative, adaptive Software kann nur von kleinen Teams entwickelt werden, die das Prinzip der ständig Design-Verbesserung und des ständigen Testens anwenden, basierend auf schnellem Feedback und schneller Veränderung.
Kontrolle	Prozesszentriert	Menschenzentriert
Managementstil	Kommando und Kontrolle	Führung und Zusammenarbeit
Wissensmanagen	Explizit	Durch Weitergabe in der Zusammenarbeit
Rollenzuteilung	Individuelles Spezialistentum	Selbstorganisierende, cross-funktionale Teams, in denen jeder unterschiedliche Aufgaben übernehmen kann
Kommunikation	Formell, Programmieren als einsamer geistiger Prozess	Informell, Programmieren als kooperativer, kommunikativer Prozess
Rolle des Kunden	Wichtig	Entscheidend
Entwicklungsmo	Lebenszyklusmodelle (Wasserfall, Spirale o.Ä.)	Modelle evolutionären Lieferns
Gewünschte Organisati-onsstruktur	Mechanistisch (bürokratisch mit einem hohen Formalisierungsgrad)	Organisch (flexibel und partizipativ, fördernde, kooperative soziale Interaktion)

Tabelle 4.1: Traditionelle- versus Agile Softwareentwicklung aus Tabelle 1.1 Gloger und Häusling 2012, S. 9 (aus dem englischen Original Tabelle 1 Nerur, Mahapatra und Mangalaraj 2005, S. 75)

4.1.1 Werte

Agile Methoden zeichnen sich durch ein grundlegendes Werteverständnis aus (siehe Kapitel 4 und Anhang A). Für eine erfolgreiche Umsetzung müssen diese im Team und optimaler weise auch in der Organisation verankert sein. Sie bilden daher eine zentrale Richtlinie für das Verhalten der Teammitglieder untereinander und liefern die Einstellung zu ihre Arbeit.

Die Werte, welche im Agilen Manifest niedergeschrieben wurden sind aus einer Erfahrung von Softwareentwicklern von erfolgreichen Unternehmen erwachsen. Sie sind in erster Linie auf eine agile, sich schnell ändernde Umgebung zugeschnitten. Dies ist in der heutigen von schnellen Wandeln unterworfenen Zeit eine wertvolle Eigenschaft. Werden die vier Wertepaarungen betrachtet, so ist zu erkennen, dass es besonders der Mensch in Form der Entwickler und Kunden, sowie die Kommunikation und eine positive, zielgerichtete Produktentwicklung im Vordergrund steht.

Agile Vorgehensweisen sind geprägt durch Kommunikation (und Feedback), Simplicity und Verantwortung (vgl. Turk, France und Rumpe 2005, S. 3):

Kommunikation und Feedback Ein wichtiger Aspekt ist die Kommunikation unter den Entwicklern wie auch zu den Kunden (oder seinen Vertretern). Die Kommunikation erfüllt dabei folgende Aufgaben:

- Wissenstransfer unter den Teammitglieder
- Transfer der Kundenwünsche in das Team und schnelles Feedback für das entstandene Produkt.
- Transparenz im Team und nach außen

In agilen Methoden wird eine direkte „face-to-face“-Kommunikation wertvoller als formale Kommunikation angesehen. Besonders die Möglichkeit Rückfragen zu stellen und neben der Sprache noch andere Kommunikationswege bei der direkten Kommunikation wählen zu können kann Missverständnisse schneller ausräumen und unterschiedliche „Fach-Sprachen“ verbinden. Die Softwareentwicklung ist eine überwiegend interdisziplinäres Feld, da die entwickelten Produkte für Kunden aus verschiedenen Branchen mit einem jeweils eigenen Sprachgebrauch und Vorgehensweisen bestimmt sind.

Das Feedback von außen ermöglicht es dem Team, sich und das Produkt zu verbessern. Das Team in einem agilen Umfeld ist jedoch nicht nur auf Feedback von außen angewiesen, sondern reflektiert die eigene Arbeitsweise durch Retrospektiven.

Transparenz und Offenheit sind weitere Aspekte der agilen Methoden. Sie werden sowohl innerhalb des Teams als wichtig erachtet, als auch bei der Kommunikation nach außen hin. Die Kommunikation nach außen hin ist vielfältig und umfasst z.B das Reporting an das Management und die Kommunikation zum Kunden.

Simplicity (Einfachheit) Ein wesentlicher Grundsatz agiler Methoden ist, dass nur lohnenswerte Dinge durchgeführt werden. Anforderungen für Softwareprojekte unterliegen oftmals schon Änderung während der Entwicklung. Die Anforderungen oder Techniken verändern sich schnell. Es ist empfehlenswert, nur umzusetzen, was gerade notwendig ist, denn es kann sonst sein, dass Umsetzung veraltet und nutzlos wird. XP postuliert daher, dass nur die aktuellen Anforderungen erfüllt werden und nicht mehr. Dies schließt eine Arbeitsweise, die langfristig angelegt ist, nicht aus, denn das Projekt bleibt einfacher zu ändern, wenn es keine überflüssigen Element enthält. Hier ist der Ansatz von „Lean" zu erkennen, was Toyota mit dem Kanban-Prinzip in der Automobilfertigung umgesetzt hat (Ōno 1988): Nur das was benötigt wird, wird aktuell vorgehalten bzw. produziert, Verschwendung (Muda) ist zu minimieren.

Die agilen Methoden selbst sind auch von Einfachheit geprägt. Die meisten sind auf wenigen Seiten beschreibbar und verfolgen den Ansatz sich nur auf das Wesentliche zu konzentrieren.

Verantwortung Durch agile Methoden wird Verantwortung an Stellen gebracht, die sie fachlich tragen können. So sind Entwickler für technischen Details verantwortlich, während sich das Management um Businessthemen kümmert (Pixton, Gibson und Nickolaisen 2014, vgl.).

Die Abgabe von Verantwortung in ein selbst organisiertes Team in Zusammenhang mit den agilen Grundsätzen verspricht eine Zunahme von Vertrauen, wie eine Untersuchung von Entwicklungsgruppen hervorbrachte: „Participants (…) reported that trust existed among their members. We found that agile methods increased trust by increasing transparency, accountability, communication, (…) knowledge sharing and feedback." (McHugh, Conboy und Lang 2012, S. 73).

Es ist jedoch auch wichtig, dass diese Verantwortung an das Team abgegeben wird, sonst wird das Team niemals selbständig arbeiten können und seine volle Leistung entfalten.

4.1.2 Merkmale der Arbeitsweise von agilen Teams

Den Werten folgend haben sich spezifische Arbeitsweisen in agilen Teams entwickelt. Einige von ihnen haben eine Auswirkung auf die Leistungsfä-

higkeit und die Art zu Arbeiten und sind damit von besonderem Interesse. Diese sollen im Folgenden näher betrachtet werden.

Das Prinzip hinter dem Agilen Manifest greift auf ein selbstorganisiertes Team (s.a. Punkt 11) zurück. Mit der Selbstorganisation ist hier gemeint, dass das Team seine Umsetzung der Aufgaben und Entscheidungen für seinen Bereich selbst treffen muss (vgl. Steinmann und Schreyögg 2005, S. 593), nicht jedoch die Zusammensetzung und Initiierung des Teams. Dies gibt die Verantwortung für die fachliche Aufgabenlösung und das direkte Umfeld in das Team. Um die Selbstorganisation zu unterstützen, haben sich verschiedene Best-Practice-Ansätze etabliert und werden im Rahmen der agilen Methoden genutzt:

Inkrementelle Produktentwicklung wird eingesetzt, um die Fertigstellung des Produktes zu steuern und frühzeitig eine lauffähige Software zu erhalten. Für den Kunden entsteht der Vorteil, dass die Software sehr früh nutzbar ist und Feedback, ob dies dem Kundenwunsch entspricht, möglich ist. Die Time-To-Market wird dadurch verkürzt. Oftmals wird hier auch von iterativen Vorgehen gesprochen, da sich die Entwicklungszyklen laufend wie in einem Regelkreis wiederholen (siehe den Kreislauf im Scrum in Abbildung 4.1 auf Seite 55).

Pull-Prinzip zur Aufgabensteuerung entstand durch die Kanban-Entwicklung bei Toyota für die Produktion. Das Pull-Prinzip in der Entwicklung hat den gleichen Grundgedanken: Aufgaben werden nicht an einzelne Ressourcen zugewiesen sondern sie werden vom Team oder den Personen nachgefragt (gepullt).

Der klassische Push-Workflow wird auch als spekulative Aufgabenzuweisung angesehen, indem davon ausgegangen wird, dass das Team die Ressourcen hat, die Aufgabe auszuführen. Durch die direkte Zuteilung an Aufgaben wird die Auslastung der Mitarbeiter maximiert. Dies hat Grenzen, denn es muss immer jemand die Aufgaben managen, zuweisen und vor allem die Abhängigkeiten zwischen ihnen bis ins letzte Detail überblicken. Verzögerungen oder Störungen wirken sich massiv auf die Planung und das Gesamtsystem aus. Gerade in der Entwicklung von Software sind solche Störungen mannigfaltig vorhanden: Defekte (Bugs), nicht absehbare technische Probleme, fehlendes Wissen, ungenügende Architektur der Software, ...

Dem gegenüber steht das Pull-Prinzip, bei dem das System sich selbst aussteuert und die Verantwortung in das Team gibt. Das Pull-System verlangt mehr Verantwortung und Initiative vom Team und basiert auf den Fluss von Aufgaben. Es ist vergleichbar mit einer Autofahrt im Nebel: Wenn kein Nebel ist, können wir weit geradeaus blicken, jedoch kann die weite Landschaft davon ablenken, dass wir demnächst ohnehin abbiegen müssen und es war unwichtig, bis zum Horizont blicken zu können. Im Nebel konzentrieren wir uns dagegen auf das Hier und Jetzt, die Fahrbahn und die nahestehenden Schilder. Dies ermöglicht es uns, dass wir mit weniger Informationen und Planungsaufwand (sowie unnötiger Arbeit, da der gerade Weg nicht mehr relevant ist) unser Ziel erreichen und wir keine unnötigen Arbeiten erledigen. (vgl. M. Poppendieck und T. Poppendieck 2006, S. 112 ff.; Reddy 2015, S. 198)

Retrospektive ist eine Methode, die eine Selbstreflexion im Team ermöglicht. Diese wird in verschiedenen Frameworks (z.B. bei Scrum) in regelmäßigen Zyklen genutzt, sie kann auch eingesetzt werden, wenn ein besonderes Ereignis oder Erfordernis sichtbar wird.

Rollenverteilung ermöglicht es, sich im Team zurechtzufinden. Mit dem Aufkommen der Agilen Methoden wurden typische formale Rollen beschrieben. Diese sind in so gut wie allen agilen Teams vorhanden und sind dort zum Teil explizit genannt (s.a. Kapitel 4.1.3).

4.1.3 Rollen

Agile Frameworks definieren formale Rollen, welche es ermöglichen, einen Aufgabenbereich und die Anforderungen an die Rolle für alle sichtbar zu machen. Die Rollen werden dabei im Team oder ihm nahestehenden Personen besetzt. Durch die Ausfüllung der Rollen durch Teammitglieder werden Entscheidungen zu Personen mit den notwendigen Kompetenzen verlagert: „In the Agile world, decisions are purposefully pushed down to the lowest possible level to the folks that have the most information and most to gain or lose from the decision.“ (Moreira 2010, S. 55). Die Rollenverteilung hilft auch bestimmte Verhaltensmuster im Team aufrecht zu halten oder den Dialog zu fördern und Verantwortlichkeiten klar zu regeln. Eine Übersicht über die häufig angetroffenen Rollen ist in Moreira 2010, S. 55-58 zu finden.

Coach

Der Coach spielt vor allem bei der Einführung von agilen Methoden oder neu gegründeten Teams eine zentrale Rolle. Er leistet Hilfestellung bei Problemen des Alltags und gibt seine eigene Erfahrung an die Teams weiter. Ein agiler Coach betreute meist mehrere Teams gleichzeitig.

Der Coach ist oft auch in Personalunion mit dem agilen Projektmanager wie z.B. bei Scrum in der Rolle des Scrum-Masters zu finden.

Product Owner

Der Product Owner stellt die Verbindung des Entwicklungsteams zum Produkt her. Er ist verantwortlich, dass die zu bearbeitenden Aufgaben ausreichend beschrieben und auch nach Prioritäten (z.B. nach Business Value) geordnet sind, so dass diese vom Team bearbeitet werden können. Er agiert damit als „Stimme des Kunden“ und wird oft auch als (Produkt-)Manager beschrieben.Die anfallenden Arbeiten muss er nicht selbst erledigen, sondern kann sie auch delegieren, er bleibt aber immer in der Verantwortung, dass die Arbeiten ausgeführt werden (vgl. Schwaber und Sutherland 2013).

Kunde

Eine wesentliche Rolle in der agilen Entwicklung spielt der Kunde. Dies ist schon im agilen Manifest zu erkennen, in dem die Zusammenarbeit mit dem Kunden als besonders wichtig herausgestellt wird (vgl. Beck et al. 2001). Inwieweit der Kunde direkt an der Entwicklung beteiligt ist, kommt oftmals auf die Situation und das Verhältnis des Kunden zum Produkt an. Der Kunde ist oft auch in Form eines Repräsentanten (z.B. durch den Product Owner) durch eine Rolle beteiligt.

Agiles Team

Das Team ist zentral für die Erfüllung der gestellten Aufgaben. Das Team besteht zumeist aus erfahrenen Entwicklern mit Kenntnissen aus den Disziplinen der Softwareentwicklung. Das Team beinhaltet normalerweise auch die anderen Rollen, in der agilen Methodik wird das Team jedoch meist als „die Entwickler“ verstanden. Daher ist der Begriff Team doppelt belegt: Wenn das „Agile Team“ als Rolle aufgefasst wird, sind die Entwickler gemeint, ansonsten das komplette Team.

Im Scrum-Framework ist diese Rolle als Scrum-Master bekannt. Eine der wichtigsten Aufgaben der Rolle ist es, als Facilitator zu agieren. Er schafft dabei eine Atmosphäre, die offen für Transparenz, Kommunikation und Veränderung ist. Dazu begleitet er das Team durch Retrospektiven, die Planungsmeetings und beim den Daily Standup Meetings. Weiterhin ist er Ansprechpartner und Manager für das Backlog und kommuniziert mit anderen Teams oder dem Management.

4.2 Agile Frameworks

Wie im Abschnitt 4 dargestellt wurde, gibt es eine Vielzahl von agilen Methoden. Im Rahmen dieser Arbeit können nicht alle Methoden betrachtet werden, weswegen sich diese Arbeit mit der Methode Scrum näher auseinandersetzen wird. Diese Methode wird vielfach eingesetzt, dass sie einen guten Rahmen vorgibt. Dies erweist sich besonders bei der Einführung der Methode als praktisch, um die mangelnde Erfahrung in der agilen Entwicklung auszugleichen.

4.3 Scrum

Auch wenn Scrum erst in diesem Jahrtausend weite Verbreitung fand, reichen die Wurzeln weiter zurück. Die Anfänge wurden durch Takeuchi und Nonaka mit der Veröffentlichung Takeuchi und Nonaka 1986 gelegt. Die Autoren beschreiben im Artikel einen neuen Ansatz, um die Entwicklung flexibler und schneller zu machen. Sie bezeichnen ihren als „holistic or rugby„ (Takeuchi und Nonaka 1986, S. 141) und prägten durch den Rugby-Vergleich letztendlich den Begriff Scrum[2]. In der darauf folgenden Zeit haben Jeff Sutherland und Ken Schwaber das Scrum-Framework entwickelt. Sie experimentierten und reflektierten über ihre Erfahrungen (vgl. Sutherland 2004) was letztendlich in die Veröffentlichung Schwaber 1997 mündete. Nach einiger Reifezeit wurde das erste Buch Schwaber und Beedle 2001 über Scrum veröffentlicht. Er erfolgten weitere Veröffentlichungen (Schwa-

[2]Scrum ist im Rugby eine Standardsituation, bei der alle Spieler auf einem Haufen stehen und die Köpfe zusammen stecken.

ber 2004) sowie die Ausweitung der Idee auf Unternehmensebene Schwaber 2008 (vgl. ScrumGuides.org 2015).

Um Scrum zu unterstützen und zu fördern, wurde um 2002 die Scrum Alliance (ScrumAlliance 2015) gegründet. Die Scrum Alliance „…encourages and supports the widespread adoption and effective practice of Scrum." (ScrumAlliance 2015) indem sie z.B. das „Certified Scrum-Master" Programm aufgelegt hat. Dies ist eine Zertifizierung, die als Scrum-Master durchlaufen werden sollte. Weiterhin gibt es noch die Scrum.inc von Jeff Sutherland sowie die Scrum.org von Ken Schwaber, welche den von ihm aufgebrachten Enterprise-Gedanken weiterträgt.

Die Grundlagen für Scrum in der Anwendung werden im erstmals 2010 veröffentlichten „*Der Scrum Guide*" (Schwaber und Sutherland 2013) zusammen gefasst. Im Folgenden wird der Guide genutzt, um Scrum darzustellen. Hiernach zerfällt Srcum in 3 wesentliche Elemente: Rollen im Scrum Team, Ereignisse und Artefakte. Diese Elemente sind für jeden sichtbar und machen die nach außen hin sichtbaren Zeichen des Frameworks aus. Sie bedingen sich jedoch gegenseitig und sind daher nur in ihrem Zusammenhang nützlich und zu verstehen. Für ein besseres Verständnis, vor allem wie die Ereignisse zusammen wirken, ist der typische Zyklus (ein Sprint) in der Abbildung 4.1 dargestellt. Dem Leser wir empfohlen, diese Abbildung als Referenz zu nutzen, wo die Artefakte entstehen und in welcher zeitlichen Abfolge die Ereignisse stattfinden.

Scrum baut essentiell auf dem Gedanken des „inspect and adapt"-Prinzips auf. Das Prinzip ist eine Art von Regelkreis, bei der ein Ergebnis oder ein Zustand inspiziert (inspect) und im Anschluss daran eine neue Vorgehensweise bzw. Strategie erarbeitet wird, wie es verbessert werden kann (adapt). Scrum wendet dieses Prinzip auf der Ebene des Produktes ebenso an, wie beim beteiligten Team und den Methoden. Damit dies regelmäßig durchgeführt werden kann, unterteilt Scrum die Arbeit in Einzelschritte und durchläuft diese immer wieder. Das Scrum als Framework wurde eine kontinuierliche Entwicklung konzipiert und wird in den meisten Unternehmen über einen langen Zeitraum fortgeführt.

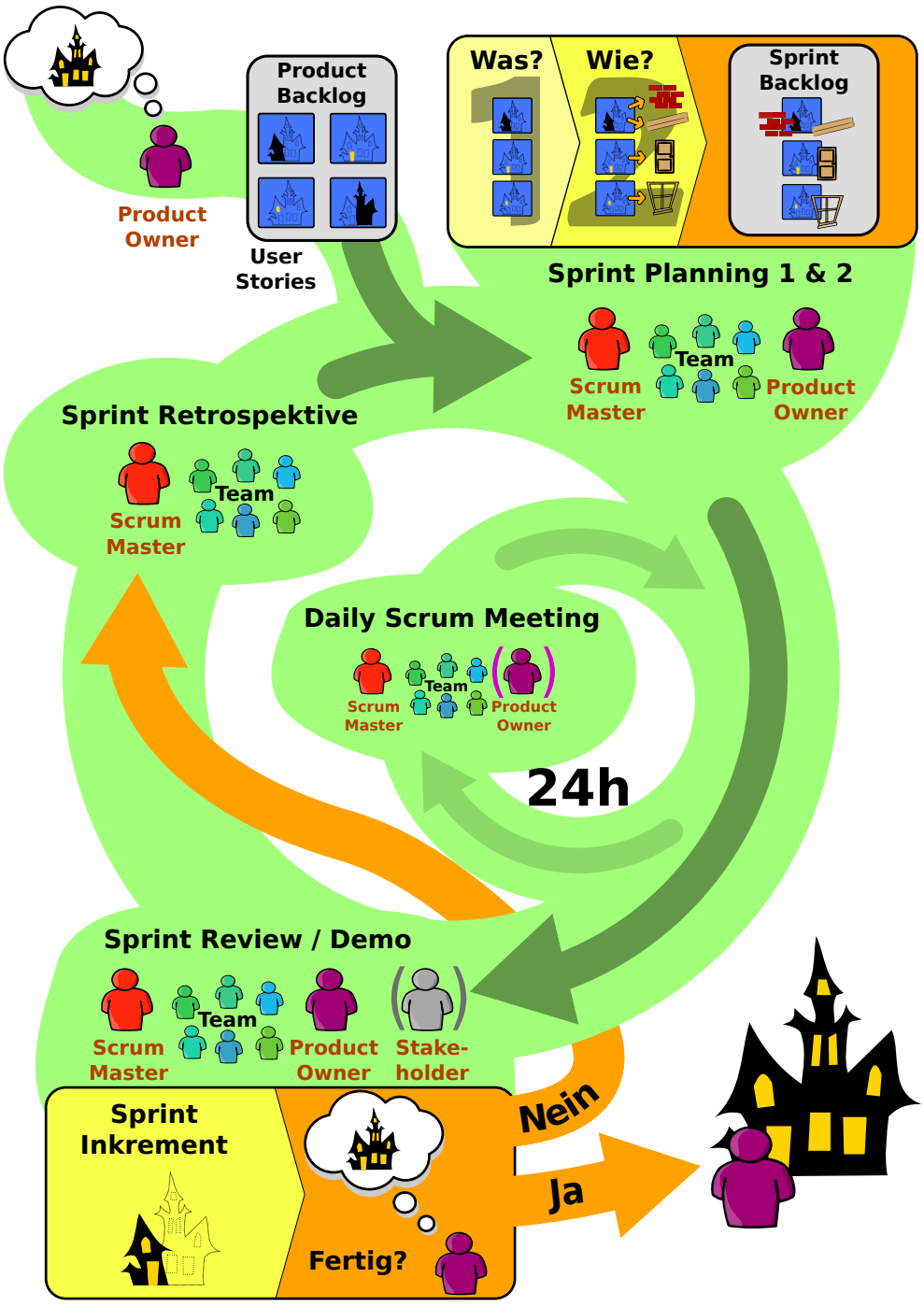


Abbildung 4.1: Der dynamische Ablauf im Scrum-Framework anhand eines Sprints mit den beteiligten Rollen.
Der Ablauf beginnt mit der Vorstellung eines Produktes des Product Owners. Diese Vorstellungen werden in sog. User Stories umgesetzt und im Product Backlog eingetragen. Ab hier beginnt die iterative Vorgehensweise. Zunächst finden die Sprint-Planungsmeetings statt, bei dem die User-Stories in einzelne Task umgewandelt werden. Weiterhin wird die Menge der bearbeiteten User-Stories für einen Sprint festgelegt. Während eines Sprints spricht sich das Team im Daily Scrum Meeting ab. Nach dem Spring folgt ein Review, bei dem der Fortschritt in Form des Sprint-Inkrements vom Product Owner bewertet wird. So lange das Produkt nicht fertig ist, erfolgt eine Sprint-Retrospektive, bei dem das Team aus dem vergangenen Sprint lernt.

4.3.1 Artefakte

Ein Scrum-Artefakt steht stellvertretend für Arbeit oder Wert. Die Artefakte haben das Ziel, wesentliche Informationen festzuhalten, Transparenz zu fördern und ein gemeinsames Verständnis zu schaffen.

Product-Backlog Im Product-Backlog werden die Informationen für das Produkt festgehalten. Es besteht aus einer priorisierten Liste mit Items (z.B. User Stories), die zum Produkt beitragen können. Im Wesentlichen finden wird dort Items der folgenden Typen: Feature, Funktionalität, Verbesserung und Fehler. Ein Item selbst hat eine Art von Reifegrad, der abhängig vom Verständnis des Items ist. Je besser ein Item verstanden wird, umso höher ist der Reifegrad bis das Item letztendlich die „Definition of Ready“ erfüllt und damit bereit für eine Aufnahme in das Sprint-Backlog und damit eine Implementierung ist. In der „Definition of Ready“ wird festgehalten, was das Entwicklungsteam an Informationen benötigt, um das Item bearbeiten zu können.

Das Product-Backlog ist dynamisch und wird sowohl vom Entwicklungsteam als auch vom Product-Owner gepflegt. Die Pflege ist dabei aufgeteilt oder kann gemeinsam erfolgen. Bei diesem Prozess des „Refinements“ (gerne auch „Grooming“ genannt) werden Details und Schätzungen zu den Items hinzugefügt oder die Priorität durch den Product-Owner verändert.

Im Zusammenhang mit den Items taucht oftmals auch der Begriff der „User-Stories“ auf. Dies ist eine Art, wie die Items beschrieben werden können. Mit ihnen lassen sich Systeme aufgrund ihres gewünschten Verhaltens aus Nutzersicht beschreiben (s.a. Cohn 2004; Wirdemann 2011).

Sprint-Backlog Dieses Backlog ist eine Ausschnitt aus dem Product-Backlog, das durch das Sprint Planning erstellt wird. Es beinhaltet die geplanten Items für genau einen Sprint. Es stellt damit den Plan auf, wie das Produkt-Inkrement erstellt werden kann. Die Items in diesem Backlog sind meist noch feiner ausgearbeitet, so dass diese sofort umgesetzt werden können. Während des Sprints gibt das Sprint-Backlog den aktuellen Entwicklungsstand wieder und dient damit auch als „Fortschrittsanzeige“ für den Sprint.

Während des Sprints wird das Sprint-Backlog vom Umfang nicht erweitert oder verringert. Jedoch ist Entwicklung nicht immer optimal und es kann

notwendig werden. Dies wird nur durch das Entwicklungsteam durchgeführt.

Produkt-Inkrement In einem Sprint wird das Product-Inkrement erstellt. Am Ende eines Sprints muss das erstellte Inkrement in einem verwendbaren (potentiell auslieferbaren) Zustand sein. Dies wird auch als „Done“ bezeichnet.

4.3.2 Rollen im Scrum Team

Das Scrum Team setzt sich aus mehreren Mitgliedern zusammen, wobei explizit die Rollen Product Owner, Entwicklungsteam und Scrum-Master unterschieden werden (vgl. auch Kapitel 4.1.3). Das Team selbst arbeitet selbstorganisiert. Das Team trifft Entscheidungen durch seine Kompetenzen selbständig und ist damit unabhängig. In der Praxis ist dies nicht immer gegeben, denn das Team muss sich vielfach auch Entscheidungsgrundlagen schaffen und Informationen für seine Entscheidungen von externen Spezialisten beschaffen. „Das Team-Modell in Scrum wurde konzipiert, um Flexibilität, Kreativität und Produktivität zu optimieren. Scrum Teams liefern Produkte iterativ und inkrementell und maximieren somit die Gelegenheiten für Feedback.“ (Schwaber und Sutherland 2013, S. 4).

Entwicklungsteam Das Team besteht aus professionell arbeitenden Mitgliedern, welche das „Produkt-Inkrement“ (die Verbesserung am Produkt) erstellen. Das Team muss unabhängig arbeiten können und dem Team wird von der Organisation vertraut, dass es dies auch leistet. „Entwicklungsteams sind von der Organisation so strukturiert und befähigt, dass sie ihre eigene Arbeit selbst organisieren und managen.“ (Schwaber und Sutherland 2013, S. 6).

Eigenschaften des Entwicklungsteams (vgl. Schwaber und Sutherland 2013, S. 6):

- Das Team organisiert sich selbst und bestimmt, wie es das Product-Backlog umsetzen wird. Dabei gibt es keine Einmischung von Außen.
- Das Team verfügt über alle notwendigen Mittel und Fähigkeiten um sein Vorhaben umzusetzen.

- Es gibt keine Spezialbehandlungen im Team, wie zusätzliche Titel oder Unterteilungen in Aufgabenbereiche. Dies schließt nicht aus, dass Mitglieder des Teams spezielle Fähigkeiten oder Fachgebiete haben können. Das Team bleibt immer als Ganzes für sein Handeln verantwortlich.
- Das Team besteht mindestens aus 3 Mitgliedern, mehr als 9 Mitglieder wird nicht als sinnvoll erachtet, da sonst die Interaktion zwischen den Teammitgliedern sinkt.

Product Owner „Der Product Owner ist für die Wertmaximierung des Produkts sowie der Arbeit des Entwicklungsteams verantwortlich." (Schwaber und Sutherland 2013, S. 5). Er pflegt als alleinige Person das Product-Backlog und ist dafür verantwortlich. Das Entwicklungsteam arbeitet nur aufgrund seiner Entscheidungen über das Product-Backlog. Für den Erfolg ist es wichtig, das die gesamte Organisation seine Entscheidungen respektiert.

Aufgaben des Product-Owners ist …

- … Backlog-Einträge klar zu formulieren. Das Entwicklungsteam muss die formulierten Einträge verstehen können.
- … die Zuordnungen von Prioritäten, indem das Product-Backlog anhand der Priorität sortiert ist. Die Prioritäten sind so festzulegen, das der höchste Business-Value, also der Wert eines Items für den Kunden, die höchste Priorität hat.
- … die Optimierung des Wertes der Arbeit des Entwicklungsteams.
- … sicherzustellen, dass die Transparenz des Teams in Form des Product-Backlogs gewahrt wird. Damit wird von außen jederzeit sichtbar, woran das Scrum-Team als nächstes arbeitet.

Scrum-Master Der Scrum-Master hat Verantwortlichkeiten im Team und darüber hinaus. Im Team agiert er als Servant Leader (Greenleaf 1977) und unterstützt das Team insbesondere bei der Durchführung von Scrum, indem er die Werte vermittelt und dafür sorgt, dass die Praktiken und Regeln von jedem gelebt werden. Nach außen hin vermittelt er ein Verständnis von Scrum, indem er Außenstehenden ein Verständnis für Prozesse im Team vermittelt und nur für das Team förderliche Interaktionen unterstützt.

Der Scrum Guide (Schwaber und Sutherland 2013) zählt ein Vielfältiges Spektrum an Aufgaben für den Scrum-Master auf. Diese sind dort hinsichtlich der unterschiedlichen Stakeholder gruppiert und lassen sich im Wesentlichen auf folgende Aufgaben zusammenfassen:

- Scrum Ereignisse werden von ihm nach Bedarf und auf Anfrage in ihrer Durchführung unterstützt.
- Er hat eine vielfältige Coaching-Aufgabe, bei der er insbesondere folgende Felder abdeckt:
  - Verwaltung und Pflege des Product-Backlogs schulen
  - richtiges Verständnis von Agilität vermitteln
  - Selbstorganisation und Teamarbeit im Entwicklungsteam fördern
- Unterstützung des Entwicklungsteams bei der Beseitigung von Hindernissen.
- Initiierung von Veränderungen, um die Produktivität zu steigern.
- Austausch und Zusammenarbeit mit anderen Scrum-Mastern im Unternehmen.

Der Scrum-Master ist in besonderem Maße bei der Einführung von Scrum im Unternehmen gefragt. Die Einführung von Scrum geht oftmals einher mit starken Widerständen gegen den Wandel, da hierarchische Führungsstrukturen aufgebrochen, Zuständigkeiten verändert und neue Werte eingeführt werden. Weiterhin muss das Unternehmen sich zu den agilen Grundsätzen bekennen, was sich als schwierig herausstellen kann. Hier gilt es für den Scrum-Master die Voraussetzungen für funktionierenden Scrum darzulegen und schädliche Einflüsse sofort zu unterbinden. Dieses Thema führt für die Arbeit zu weit, für den geneigten Leser empfehle ich das Buch Cohn 2010b, das sich mit der Einführung von Scrum und ihrer Widerstände beschäftigt.

4.3.3 Ereignisse

Von Ereignissen in Scrum wird immer gesprochen, wenn das Team zusammen arbeitet. Sie geben dem Team eine Anleitung, was notwendig ist, um das Ziel zu erreichen. Sie werden eingeübt und durch die regelmäßige Wiederholung erhält das Team Routine in diesen Tätigkeiten, es wird eine Art

von Ritual. Dies macht es einfach, sich als Teammitglied daran zu orientieren. Die Ereignisse selbst sind so ausgelegt, dass sie möglichst kurz sind und dabei den maximalen Nutzen für einen bestimmten Zweck entfalten. Eine wichtige Eigenschaft aller Ereignisse ist, dass sie Time-Boxed sind und damit eine festgelegte maximale Dauer haben.

Sprint Ein wichtiges Ereignis ist der Sprint (in Abbildung 4.1 der große Zyklus). Er stellt eine Time-Box mit einer maximalen Dauer von einem Monat dar (üblich sind 2 bis 4 Wochen). Die Dauer eines Sprints ist normalerweise konstant und wird auf die Situation angepasst. Im Sprint selbst sind alle Tätigkeiten eingebettet, welche zu einem potentiell auslieferbaren Sprint-Inkrement (ein weiterentwickeltes Produkt) führen.

Ein Sprint umfasst die folgenden Ereignisse:

- Sprint Planning (zu Beginn des Sprints)
- Daily Scrum (jeden Tag)
- Entwicklungsarbeit (jeden Tag)
- Sprint Review / Demo (am Ende des Sprints)
- Sprint Retrospektive (Abschluss des Sprints)

Ein Sprint kann als ein kleines Projekt mit einem gut überschaubaren Zeithorizont aufgefasst werden. Durch den kurzen Zeithorizont ist das Risiko überschaubar, denn die Unsicherheit ist nur gering. Nach Abschluss des Sprints ist ein Produkt verfügbar, mit dem überprüft werden kann, ob eine Entwicklung in die gewünschte Richtung statt findet. Ebenso kann die eigene Vorgehensweise zügig reflektiert werden und für das nächste „Projekt“ in Form des Sprints alsbald genutzt werden.

Sprint Planning Bei der Planung für den nächsten Sprint wird durch das gesamte Scrum-Team ein gemeinsames Verständnis und eine Vorgehensweise für den nächsten Sprint entwickelt. Bei der Planung werden Teile des Product-Backlog während eines Meetings in ein Sprint-Backlog überführt, welches im Anschluss für diesen Sprint als Vorlage zur Durchführung dient. Beim Meeting werden die folgenden zwei Teilaspekte unabhängig hintereinander betrachtet:

1. **Was** soll im nächsten Sprint als Product-Inkrement erstellt werden?
2. **Wie** wird die Arbeit erledigt?

Sprint Planning I (Was?) Im ersten Teil des Sprint Planning wird festgelegt, was im kommenden Sprint erstellt werden soll. Das Product-Backlog dient dem Team dabei als Arbeitsgrundlage, denn hier hat der Product-Owner den Wert der einzelnen Items hinterlegt. Das Entwicklungsteam wählt aus dem Product-Backlog die Items aus, die in diesem Sprint umgesetzt werden sollen, indem es folgende Kriterien berücksichtigt:

- Die eigene Kapazität in diesem Sprint. Damit übernimmt es die Ressourcenverwaltung für den anstehenden Zeitraum und berücksichtigt die Verfügbarkeit.
- Die bisherige Leistung wird genutzt, um die zukünftige Leistung vorherzusagen.
- Die Wertigkeit der Items.
- Synergieeffekte oder Abhängigkeiten der Items untereinander, die dem Product-Owner nicht bekannt sind.

Nur das Entwicklungsteam wählt die Items für den nächsten Sprint aus. In der Praxis wird es dabei auf seine eigene Erfahrung zurückgreifen und die Items hinsichtlich ihrer Komplexität oder des Aufwandes schätzen. Das schätzen in agilen Projekten unterscheidet sich von klassischen Methoden bei denen versucht wird der Aufwand zu schätzen. Dies ist in der Software nur schwierig möglich, denn um Aufwand schätzen zu können, muss auf der Aktivitätsebene geschätzt werden. Oftmals wird durch die vorhandene Komplexität wichtige Aktivitäten vergessen oder erst im Nachhinein sichtbar. Die Einträge im Product-Backlog sind auf Feature-Ebene beschrieben. Meist ist es gar nicht notwendig eine exakte Schätzung zu haben, weswegen auf Verfahren zurückgegriffen wird, die als Grundlage versuchen die Komplexität zu schätzen (vgl. Cohn 2005). Im Folgenden ist das Schätzen von geringerer Interesse, da es sich um ein komplexes Feld handelt und für den Rahmen dieser Arbeit nicht von Belang ist.

Nachdem das Entwicklungs-Team den Umfang für den Sprint festgelegt hat, wird ein Sprint-Ziel festgelegt. Es bildet einen Leitfaden, was in diesem Sprint erstellt werden soll. In der Praxis sind in den User-Stories oftmals Gemeinsamkeiten zu finden, die z.B. eine gemeinsame Feature-Gruppe umfassen. Es gibt dem Team ein gemeinsames Verständnis, was das fertige Product-Inkrement enthalten soll.

Sprint Planning II (Wie?) In diesem Teil wird die konkrete Umsetzung der Items geplant. Dabei unterteilt das Team die Items in einzelne Aufgabenschritte und entwickelt ein tiefes Verständnis, wie die Umsetzung durchgeführt werden soll. Dies beinhaltet schon die Abklärung von technischen Details oder Rückfragen an Spezialisten, falls dies notwendig ist.

Am Ende dieses Meetings hat das Entwicklungsteam selbständig einen Umsetzungsplan erstellt, nach dem es für diesen Sprint arbeiten wird. Damit ist jeder im Team in der Lage, jedem Außenstehenden zu erklären, wie es das Sprint-Ziel erreichen will.

Daily Scrum Der Daily Scrum ist ein wertvolles Element zur Kommunikation im Scrum. Er hat das Ziel, Wissen auszutauschen, das Team zu synchronisieren und Hindernisse schnell zu erkennen und zu beheben. Das Meeting selbst wird täglich zur gleichen Uhrzeit am selben Platz, Time-Boxed für maximal 15 Minuten durch das Entwicklungsteam durchgeführt. Jedes Teammitglied beantwortet dabei folgende Leitfragen:

- Was habe ich gestern erreicht?
- Woran werde ich heute arbeiten?
- Welche Impediments (Hindernisse) gibt es momentan bei meiner Arbeit?

Sprint Review / Demo Am Ende des Sprints wird das Sprint Review abgehalten und hat den Zweck, die Ergebnisse in Form des Product-Inkrements zu bewerten. Wie die Ausgestaltung des Meetings ist, hängt stark vom Produkt und der Entwicklung selbst ab. Je nach Erfordernis sind Stakeholder und / oder der Product-Owner anwesend und bewerten oder testen die neuen Features anhand der Sprint-Backlog-Einträge. Oftmals hat das Meeting einen Vorführcharakter, weswegen es gerne Demo genannt wird. Das Meeting ist jedoch mehr. Durch den „inspect and adapt"-Gedanken liefert es wertvollen Input für die kommenden Sprint Plannings. Das Meeting bietet die Chance, die aktuelle Situation hinsichtlich des Produktes zu eruieren und daraus neue Erkenntnisse zu gewinnen. Diese Erkenntnisse fließen in das Product-Backlog ein und werden im kommenden Sprint behandelt.

Hieran ist ein wichtiges Zeichen agiler Methoden zu erkennen, denn durch diesen Einfluss ist es möglich, sehr zeitnah auf geänderte Anforderungen, neue Ideen oder geänderte Marktsituationen zu reagieren. Ebenso wird

der Zeitraum bis zur Begutachtung des Produkts wesentlich verkürzt und mit Abschluss des Sprints liegt immer ein potentiell auslieferbares Produkt vor.

Sprint Retrospektive Die Retrospektive stellt für das Team die Möglichkeit dar, sich Gedanken über seine Arbeitsweise und seine Umgebung zu machen. Im Meeting wird mit unterschiedlichen Methoden (s.a. Derby, Larsen und Schwaber 2006) gearbeitet, um die Arbeitsweise, Prozesse, Werkzeuge, und das Zwischenmenschliche zu hinterfragen. Dabei identifiziert das Team Best-Praktices und Verbesserungspotential. Es erarbeitet selbständig einen Plan, wie die Verbesserungen im nächsten Sprint umgesetzt werden können.

Gerade die Umsetzung gestaltet sich manchmal schwierig, vor allem wenn es um externe Abhängigkeiten geht. Hier ist der Scrum-Master in seiner Vermittlerrolle gefragt, um dies im Unternehmen weiter voranzubringen.

Das Team wird auch schon während des Sprints Verbesserungspotential erkennen und in seiner Arbeit berücksichtigen, jedoch zeigt es sich in der Praxis das ein formelles Meeting noch deutlich fokussierter mit speziellen Methoden an der Verbesserung arbeiten kann.

4.4 Leadership in agilen Teams

Das Scrum-Framework ist sehr gut definiert, macht jedoch kaum eine Aussage über die Führung von einem Scrum-Team. Die Führung in einem agilen Team ist aufgespalten und es wird sich eine geteilte Führung entwickeln. Werden die Rollen im Scrum betrachtet, so ist ersichtlich, dass für unterschiedliche Bereiche und Aufgaben unterschiedliche Rollen zuständig sind. Diese Rollen beinhalten im Wesentlich folgende Führungsaufgaben:

- Die Führung im Sinne der umzusetzenden Kundenwünsche durch den Product-Owner.
- Die Führung im Sinne des Scrum-Frameworks, also die Einhaltung der Vorgehensweisen, zur Vermeidung von Konflikten und andere vor allem soziale Teamfaktoren durch den Scrum-Master.
- Die technische Führung durch die Mitglieder selbst. Hier werden sich immer wieder unterschiedliche Leader für bestimmte Aufgaben erge-

ben, die vor allem davon abhängen, wie viel Erfahrung ein Mitarbeiter in einem bestimmten, gerade geforderten Bereich hat.

Die geteilte Führung ist vor allem dann von Vorteil, wenn die jeweiligen Mitarbeiter ihr Potential hierdurch bessere einbringen können. Dies ist sehr vorteilhaft, so hat die Studie Carson, Tesluk und Marrone 2007 gezeigt, dass geteilte Leadership eine positive Relation mit der Teamperformance zeigt. Weiterhin entlastet es die einzelnen Leader in ihrer Arbeit, so dass sich diese auf ihren Teilbereich konzentrieren können.

Die Führungsstile, welche von den einzelnen Leadern in ihren jeweiligen Situationen[3] genutzt werden, sind dabei relativ vielfältig. Jedoch ist zu bedenken, dass der Führungsstil in eine agile Kultur passen muss. Die Kultur ist geprägt von Offenheit und dem Team-Gedanken, weswegen sich dies auch im Führungsstil widerspiegeln muss. In der Literatur wird von einem Team-Coach oder Agile-Coach gesprochen (vgl. Adkins 2010; Stellman und Greene 2014), welcher vor allem durch den Scrum-Master zum Einsatz kommt. Ebenso ist dies sehr wichtig für den Product-Owner, welcher vermittelt, wie der Kunde denkt und was er benötigt. Die Interaktion der Teammitglieder untereinander ist von einer oftmals fachlichen Führung, die eine große Portion Wissensvermittlung beinhaltet, geprägt.

Der Transformationale Führungsstil (Burns 1978) ist gerade in Zusammenhang mit den agilen Methoden zu nennen. Bei agilen Teams spielt die Kultur eine wichtige Rolle, die neben den Werten ein Miteinander einfordert. Die beste Effizienz, gerade auch in Hinblick auf die Selbstorganisation des Teams, bring demnach ein Transformationaler Führungsstil, der es ermöglicht, an Beispielen das Team zu entwickeln. Durch das gute Beispiel und den alltäglichen Gebrauch werden die Vorgehensweisen und Grundgedanken an die Teammitglieder weitergegeben und dort in ihrer Team-Kultur verankert. Gerade bei der Einführung von agilen Methoden ist es wichtig, die Kultur der Agilität im Team zu verankern, was vor allem durch einen Leader funktioniert, der überzeugend und authentisch diese Kultur vertritt. Dies ist ein Merkmal des transformativen Führungsstil: „Transformative Führer handeln aus tiefer Überzeugung, aus dem Glauben an bestimmte Werte und Ideen." (Steinmann und Schreyögg 2005, S. 659). Weiterhin handelt es sich

[3] Die „Situationale Führungstheorie" nach Hersey und Blanchart (Steinmann und Schreyögg 2005, S. 672 f.) ist hier nicht allgemein passend, da die Dimensionen des Modells (Personen- und Aufgabenorientierung) die Situationen bei den Rollen ungenügend beschreiben. Wohl aber kann ein Leader für die Führung eines Mitarbeiters auf das Modell zurück greifen und so z.B. einen jungen Mitarbeiter fachlich entlang des Modells anleiten.

bei der Entwicklung um eine hochgradig kreative Tätigkeit, so dass die Führungskraft bei den Teammitglieder die Kreativität anregen müssen und als Mentor und Coach zur Verfügung stehen sollen (vgl. Bass 1985).

Kapitel 5

5 Verbindung der klassischen Modelle und der agilen Arbeitsweise

Die in Kapitel 2 vorgestellten Modelle sind allesamt gut bekannt und im Alltag der Teamentwicklung erprobt. Die Einführung von agilen Methoden bedeutet nicht, dass alles bis jetzt Dagewesene keine Gültigkeit mehr besitzt. Vielmehr kommt eine neue Komponente hinzu, welches es zu beachten gibt. Agile Teams sind in ihrer Struktur, Aufgabenbearbeitung und wie sie Probleme lösen deutlich freier und müssen daher den ihnen gegebenen Freiraum ausfüllen. Dieser neue Freiraum bedeutet aber auch eine erhöhte Verantwortung und unter Umständen andersartige Probleme, als dies von klassisch geführten Teams bekannt ist.

Das folgende Kapitel wird sich mit der Frage auseinandersetzen, ob agile Methoden negative Entwicklungen vorbeugen können und wenn dies so ist, wie sie dies tun (s.a. Forschungsfragen aus Kapitel 1.1. Ebenso wird analysiert werden, welche negativen Entwicklungen aus den klassischen Modellen der Teamentwicklung eine hohe Relevanz für agile Teams haben und welche davon nur eine untergeordnete Rolle spielen (weil sie z.B. durch agile Methodiken abgemildert werden können). Daher wird hier der Ansatz verfolgt, die auffälligen negativen Entwicklungen aus den vorangegangenen Kapiteln zu nutzen und zu überprüfen, ob agile Methoden eine Vorsichtsmaßnahme haben, die dieser Entwicklung entgegen wirkt.

5.1 Phänomene in Gruppen

Die Phänomene Risikoschub, Gruppendenken und das Soziale Faulenzen sind negative Erscheinungen in Gruppen, welche genauer betrachtet werden.

Risikoschub

Agile Teams sind im Bereich der Entwicklung anzutreffen. Technische Entwicklung birgt aber immer gewisse unternehmerische Risiken, dass z.B. eine bestimmte Technologie nicht wie geplant eingesetzt werden kann oder sich eine Entwicklung verzögert. Risikoschub betrifft hauptsächlich die Entscheidungen von Teams. Das Team ist bereit ein größeres Risiko bei einer Entscheidung einzugehen. Zunächst muss gefragt werden, welche Entscheidungen hier gemeint sind, denn in einem agilen Team werden laufend Entscheidungen getroffen, welche z.B. in Bereiche wie Vorgehensweise, Organisation und Implementierungsstrategie unterteilt werden können. Ein wesentlicher Vorteil für das Team ist es, dass die meisten Entscheidungen das Team oder seine Arbeit selbst betreffen. Dies hat den Vorteil, dass die Teammitglieder eine hohe Kompetenz haben, diese Entscheidungen zu treffen.

Das iterative Vorgehen, sowohl bei der eigentlichen Arbeit in Form von Erstellung des Produktes als auch die Änderung der Vorgehensweise und damit der Organisation im Team wird laufend überprüft. Am Ende eines jeden Sprints wird durch die Sprint-Demo das Produkt (und damit die getroffenen Entscheidungen) einem Review unterzogen. Damit beschränkt sich das Risiko auf einen relativ kurzen Zeitraum. Die Vorgehensweise reflektiert das Team immer wieder, indem es regelmäßig Sprint-Retrospektiven durchführt. Viele Teams sind auch gerne bereit zu Experimenten, um seine Zusammenarbeit oder Organisation in vielen evolutionären Schritten zu verbessern. Was sich bewährt, bleibt bestehen, anderes wird revidiert. Hierdurch bekommt das Team regelmäßig Zeit, sich über seine Entscheidungen bewusst zu werden und diese gegebenenfalls auch wieder zu korrigieren.

Es lässt sich sagen, dass ein Scrum-Team im Vergleich zu einem herkömmlichen Team es gewohnt ist, Entscheidungen in Frage zu stellen. Dies macht es ihm einfacher möglich, entweder Risikoschub von vornherein zu vermeiden, oder besonders risikoreiche Entscheidungen wieder zu revidieren. Besonders letztere Eigenschaft ist zu erwähnen. Dennoch wäre zu hinterfragen, ob dies

immer funktioniert und ob dies auch von der Erfahrung der Teammitglieder oder auch der Kultur im Team abhängig ist.

Groupthink

Auf Seite 12 wurden neben Groupthink allgemein auch die begünstigenden Bedingungen sowie Hinweise zur Vermeidung aufgeführt. Vorbeugende Maßnahmen besteht im Wesentlichen aus den Komponenten Leadership und Vorgehensweise bei Entscheidungen in der Gruppe.

Leadership ist speziell in Scrum-Teams verteilt auf mehrere Gruppenmitglieder (s.a. Seite 63), so dass es keinen alleinigen Gruppenführer gibt. Dies mildert die Gefahr eines starken Gruppenführers nur bedingt, da es je nach Situation einen Leader gibt, der vor allem die fachliche Seite sieht. Die Gruppe ist es allerdings gewöhnt, dass es nicht eine einzige Meinung gibt. Hier können agile Methoden nur bedingt weiter helfen.

Vorgehensweisen sind daher umso wichtiger, um eine einseitige Entscheidung zu vermeiden. Im Scrum-Framework sind hierfür keine Elemente vorgesehen, die den vorbeugenden Maßnahmen ähneln. Es ist daher darauf zu achten, wie die Meinungsbildung in der Gruppe funktionieren wird. Dies ist eine Aufgabe des Scrum-Masters, da gerade er als Coach agiert und daher nicht komplett in das Team integriert ist. Es kann sein, dass dies nicht ausreicht. Daher sollte die Gruppe stets in Hinblick auf die Symptome von Groupthink beobachtet werden.

Soziales Faulenzen

Ein wichtiger Vorteil der agilen Methoden und insbesondere von Scrum ist es, die Arbeitsweise des Teams transparent zu machen. Dies macht es möglich, Probleme sichtbar zu machen und dann in einem weiteren Schritt daran zu arbeiten. Genau diese Transparenz hilft dem Team, zu beurteilen, welches Mitglied welchen Beitrag zum Gelingen beisteuert. Die Transparenz wird insbesondere durch den Daily-Scrum sowie die spätere Retrospektive am Sprint-Ende sichergestellt. Der Fortschritt von einzelnen Aufgaben wird durch das Team selbst am Sprint-Backlog sowie der engen Zusammenarbeit fortgeschrieben und aufgezeichnet. Damit ist jeder immer im Bilde, was der

andere gerade beisteuert und die Tendenz zum sozialen Faulenzen wird erheblich verringert. Daher sind Scrum-Teams wenig anfällig für diesen Effekt und wenn er auftritt, dann wird er durch die anderen Mitglieder zuverlässig erkannt.

Zusammenfassung und Bedeutung für die Forschungsfrage

Scrum als agile Methode besitzt Vorkehrungen, um Gruppeneffekte zu verringern. Insbesondere zwei Eigenschaften erweisen sich dabei als positiv:

- **Iteratives Vorgehen** hilft vor allem, um Fehler zu korrigieren und schnellstmöglich eine Rückmeldung zu bekommen. Damit lassen sich Probleme frühzeitig erkennen und beseitigen.
- **Transparenz** ist besonders dann von Vorteil, um Probleme zu erkennen. Ein solches Problem kann ein überlasteter Mitarbeiter sein, aber auch jemand, der als Trittbrettfahrer das Team abbremst.

Soziales Faulenzen lässt sich gut vermeiden, beim Risikoschub sind Mechanismen zur Vermeidung vorhanden. Anders sieht es beim Groupthink aus. Probleme in Entscheidungsprozessen kann das Scrum-Framework nicht vermeiden. Hierzu müssen die traditionellen Vorgehensweisen genutzt werden. Da es im Team selbst keinen ausgesprochenen Leader gibt, muss dies in die Verantwortung des Scrum-Masters oder einem externen Beobachter gelegt werden. Das Team sollte jedoch für diesen Effekt sensibilisiert werden. Da der Scrum-Master an den Meetings des Teams teilnimmt, kann er dies beobachten und gegebenenfalls Maßnahmen einleiten.

5.2 Konflikte

In agilen Teams kommt es wie in jedem anderen Team ebenfalls zu Konflikten. Diese Konflikte können innerhalb des Teams stattfinden oder aber an seinen Schnittstellen zum restlichen Unternehmen. Für uns wesentlich sind hier die Konflikte im Team selbst.

Die auftretenden Konflikte sind von unterschiedlicher Natur und es ist nur schwer vorhersehbar, welche Art von Konflikten in einer bestimmten Konstellation auftreten werden. Vielfach löst das Team die Konflikte auch ohne externe Hilfe, da es eine offene Kommunikationskultur pflegt und daher

entstehende Konflikte in einem frühen Stadium ansprechen und lösen kann. Ebenso sind die täglichen Meetings und die Sprint-Retrospektive eine gute Möglichkeit, Konflikte anzusprechen und in einem dafür vorgesehenen Rahmen zu bearbeiten. Das Team wird dabei immer versuchen, Konflikte selbst zu lösen. Manchmal ist es jedoch wichtig, dass Konflikte angesprochen werden, dies unterliegt der Obliegenheit des Scrum-Masters. Er spielt eine wichtige Rolle, wenn es um das Aufdecken von Konflikten geht. Ebenso wird er versuchen, diese möglichst unabhängig zu lösen. Konfliktlösung ist ein integraler Bestandteil der Aufgaben des Scrum-Masters (vgl. Kompetenzprofil in Gloger und Häusling 2012, S. 51 ff.). Ein wichtiger Konfliktherd ist auch die Zusammenarbeit zwischen Team und Product–Owner. Gerade in zeitlich angespannten Situationen können hier problematische Konflikte entstehen, da oftmals die Interessen von Entwickler und dem Product-Owner nicht identisch sind.

Zusammenfassung und Bedeutung für die Forschungsfrage

Teams, welche nach dem Scrum–Framework arbeiten, haben im Vergleich zu klassisch arbeitenden Teams den Vorteil, dass Konfliktmanagement Teil der Aufgaben eines Scrum–Master ist. Speziell bei der Ausbildung zum Scrum-Master spielt das Thema Konflikte eine wichtige Rolle. Der Scrum-Master muss Konflikte (auch außerhalb des Teams) klären und eine Lösung herbeiführen, wenn sie langfristig ein Impediment darstellen (also einen negativen Einfluss auf die Performance des Teams haben). Weiterhin hilft die offene Kommunikationskultur, dass Konflikte gar nicht erst entstehen oder sie frühzeitig unter den Konfliktparteien geklärt werden.

Konflikte und ihre Bewältigung wird aber immer ein Teil des Arbeitsalltages bleiben. Hier stellt sich vor allem die Frage, welcher Art diese Konflikte sind. Hier hat jeder, abhängig von den Personen im Team und der Umgebung im Unternehmen andere Erfahrungen. Daher ist es eine interessante Fragestellung, welche Konflikte in welchem Zusammenhang schon beobachtet wurden.

5.3 Phasen der Teamentwicklung

Die Phasen der Teamentwicklung sind für agile Teams ebenso relevant, wie für klassische Teams. Wenn die Definition von Scrum betrachtet wird, sind viele Dinge sehr detailliert geregelt, jedoch ist eine Implementierung von Scrum im Unternehmen immer von den äußeren Rahmenbedingungen beeinflusst. Es wird damit keine zwei identisch arbeitenden Teams geben. Scrum kann damit für ein Team nur ein grobes Gerüst bilden, das aus einem Best-Practice-Ansatz besteht.

Es stellt sich die Frage: **Kann das Scrum-Framework bei der Teamentwicklung behilflich sein?**

Zwar stellt Scrum ein gutes Framework zur Verfügung, in der Praxis kommt es jedoch immer wieder zu Problemen. Gerade die Verteilung der Rollen bietet manchmal eine nur trügerische Sicherheit, wie folgendes Zitat zeigt: „Die eindeutige Zuordnung der Verantwortlichkeiten in Scrum … lässt sich in der Praxis nicht immer so einfach einhalten. … Wem die Verantwortlichkeit nicht klar ist, der ist oft versucht, einer der anderen Scrum-Rollen die Verantwortung zuzuschreiben." (Wolf 2015, S. 38). Gerade die Rollenverteilung ist ein typisches Beispiel, welches in jedem Team geklärt werden muss. Klassischer weise wird in der Storming-Phase die Führungsrolle in Frage gestellt. In agilen Teams wird diese Führungsrolle von verschiedenen Mitgliedern ausgeübt, weswegen diese Rollen zunächst definiert werden müssen. Im angeführten Beispiel war ein großes Problem, dass die Verantwortung von keinem getragen werden wollte und diese hin und hergeschoben wurde. Eine der wohl schwierigsten Aufgaben für das Team in der *Storming-Phase*, ist zu verstehen, dass es als Team verantwortlich ist (vgl. Cohn 2010a, S. 231 ff.). Interessant an diesem Beispiel ist, dass das Problem gelöst werden konnte, indem sich das Team regelmäßig mit sich selbst beschäftigt hat. Es hat damit selbständig eine Teamentwicklung durchgeführt. Dies ist ein integraler Bestandteil des Scrum-Frameworks, so dass solche Probleme strukturiert und in einem entsprechenden zeitlichen Plan mit beinhaltet sind.

Es darf jedoch nicht vergessen werden, dass es sich bei Softwareentwicklung um einen höchst individuellen Prozess handelt. Das Entwicklungsteam trifft je nach Situation sehr individuelle Entscheidungen und wird auch die Lösung von Problemen unterschiedlich angehen. So ist z.B. in einem regulierten Umfeld wie der Medizintechnik ein direkter Test durch Nutzer nur schwer möglich, denn es muss immer vorab eine Zulassung erlangt werden. Bei der

momentan populären App-Entwicklung für Smartphones ist dies ohne weiteres möglich. Das Team wird sich daher gerade in der Phase des *Normings* eine sehr individuelle Vorgehensweise aneignen. Das Scrum-Framework hilft hier nur bedingt, jedoch schafft es dem Team eine Möglichkeit in einem kontinuierlichen Verbesserungsprozess diese Entscheidungen zu treffen und umzusetzen. Das Team hat durch die regelmäßige Retrospektive den Vorteil, Missstände zu erkennen und entsprechende Maßnahmen einzuleiten. Solch ein Missstand kann auch ein Zurückfallen in eine vorherige Phase sein. Dies verhilft dem Team auch Probleme in der *Performing-Phase* zu erkennen und eine Lösung zu erarbeiten.

Zusammenfassung und Bedeutung für die Forschungsfrage

Scrum bietet durch die regelmäßige Retrospektive die Möglichkeit, sich einen Einblick in die aktuelle Team-Situation zu verschaffen. Es erstellt damit quasi eine Art von Teamdiagnose mit unterschiedlichen Mitteln (für einen Überblick über mögliche Methoden siehe auch Derby, Larsen und Schwaber 2006). Es hat damit die Möglichkeit, die Phasen der Teamentwicklung bewusst zu erleben und das Team hat damit auch die Möglichkeit einen Rückfall aus der Performance-Phase zu erkennen und geeignete Gegenmaßnahmen einzuleiten.

Es stellt sich natürlich die Frage, ob dies immer gelingt oder ob es Situationen gibt, wo dieser Sicherungsmechanismus nicht mehr greift. Zwar gibt es immer noch den Scrum-Master, welcher dann in solchen Situationen eingreifen kann, jedoch kann es auch sein, dass eine disziplinarisch vorgesetzte Führungskraft das Team darauf hinweisen und Maßnahmen einleiten muss.

5.4 Rollen-basierte Modelle der Teamentwicklung

In der vorherigen Sektion wurde es schon angesprochen: Rollen sind auch im Scrum ein wichtiger Faktor, damit sich Teammitglieder orientieren können und wissen, wer welche Verantwortung zu tragen hat. Dass dies selbst bei gut definierten Rollen nicht immer eindeutig ist, hat das Beispiel leider

auch klar gemacht. Für ein funktionierendes Team sind mehr Rollen notwendig, als die von Scrum definierten. Scrum definiert seine Rollen aus einer an Funktionen orientierten Sicht, die stark auf die eigentliche Aufgabe der Entwicklung fixiert ist. Viele notwendige Rollen im Team sind nicht beachtet und es soll nun geklärt werden, welche Rollen durch Scrum schon besetzt sind und welche Rollen einem besonderen Augenmerk unterliegen sollten.

5.4.1 Spencer und Pruss

Spencer und Pruss definieren wichtige Teamrollen auf Basis eines ebenfalls auf Funktionen basierten Ansatzes. Die Funktionen sind im Vergleich zu Scrum mehr auf das Innere im Team und dessen Zusammenarbeit fokussiert. Sie sind allgemeiner gefasst und nicht an die spezifische Situation eines Teams angepasst. Werden die Teamrollen in Tabelle 2.4 betrachtet, so lassen sich in den Rollen von Scrum und deren Rollenbeschreibung viele Gemeinsamkeiten finden. Es zeigt sich auch, dass eine Rolle doppelt besetzt bzw. auf mehrere Personen aufgeteilt sein kann (siehe auch Tabelle 5.1).

Der Product-Owner kann ganz klar als **Visionär** bezeichnet werden (s.a. Gloger und Häusling 2012, S. 14). Er hat eine Vision, wie das Produkt gestaltet werden soll und überblickt als einziger in der Gänze die Anforderungen an das Produkt.

Als Gegengewicht zum Product-Owner steht der Scrum-Master mit dem Entwicklungsteam und ist damit der **Pragmatiker**. Er verhilft dem Team dazu, die Vorstellungen des Product-Owner umzusetzen. Dabei agiert der Scrum-Master nicht direkt, sondern viel mehr als Katalysator. Natürlich wird nicht das ganze Entwicklungsteam aus Pragmatikern bestehen, vielmehr werden sich dort einzelne Personen herauskristallisieren, die Erfahrung haben und damit einen Weg aufzeigen können.

Der Scrum-Master fördert durch seine „Unterstützung des Entwicklungsteams bei der Schaffung hochwertiger Produkte ... (und) Beseitigen von Hindernissen, die das Entwicklungsteam aufhalten“ (Schwaber und Sutherland 2013, S. 7) die Entwicklung und hat besonders die Aufgabe das Team zu unterstützen. Dies macht ihn zum **Entdecker** für das Team. Weiterhin ist er der Botschafter für das Team: „Der Scrum Master hilft denjenigen, die kein Teil des Scrum Teams sind, zu verstehen, welche ihrer Interaktionen mit dem

Team sich hilfreich auswirken und welche nicht." (Schwaber und Sutherland 2013, S. 6).

Der **Herausforderer** ist in so gut wie allen Rollen zu finden. Die eigene Arbeitsweise, das Produkt und die Umgebung infrage zu stellen, ist ein Kernelement des Scrums und ein wichtiger Teil der „inspect and adapt"-Philosophie. Daher ist eine direkte Zuordnung nicht wirklich sinnvoll.

Der Scrum-Master soll sich idealerweise nicht direkt in die Belange des Teams einmischen, sondern als Berater und Coach agieren. Dies kann er nur wenn er als **Unparteiischer** auftritt und einen neutralen Blick wahrt. In vielen Bereichen kann er diese Rolle erfüllen, ganz kann er sie jedoch nicht ausfüllen, denn er ist gleichzeitig im Dienste des Teams, was ihn mit dem Team verbindet.

Auch wenn dies durch das Rollenverständnis des Scrum-Masters nicht abgedeckt ist, so tritt er in vielen Teams durch seine Stellung als **Friedensstifter** auf. Sein Ziel ist es, die Zusammenarbeit zu fördern (vgl. Cohn 2010a, S. 147 f.), was auch beinhaltet, Konflikte aufzulösen oder vorzubeugen. Dies ist jedoch auch für jeden anderen die Pflicht, dies zu tun.

Das **Arbeitstier** ist ganz klar der Entwickler. Er hat mit dem Entwicklerteam die Hauptaufgabe des Teams zu erfüllen und die Erstellung der Software umzusetzen.

Ein Teilaufgabe des Scrum-Masters ist die Rolle des **Trainers**. Er gibt dem Team Halt und begleitet das Team darin, sich selbst zu finden und ihre Aufgaben wahrzunehmen. Er hat auch eine „Überwachende" Rolle, indem er dafür sorgt, dass notwendige Kommunikationsmeetings wahrgenommen werden und deren Wert geschätzt wird. Aus der fachlichen Sicht wird auch ein erfahrener Senior Entwickler neue Teammitglieder einarbeiten und für sie ebenfalls als **Trainer** auftreten. Der Trainer wird damit immer durch ein Teammitglied wahrgenommen werden, dass über eine entsprechende Expertise und Erfahrung verfügt. Gleichzeitig macht es auch klar, dass Teammitglieder mit der entsprechenden Expertise ein wichtiger Bestandteil des Teams sind.

Der **Bibliothekar** dokumentiert und zeichnet Aktivitäten auf und dient als Informationsquelle. Dies ist im Scrum an unterschiedlichen Stellen zu finden. Ein Teil der Dokumentation ist die tägliche Arbeit am Quellcode, der zugleich als Informationsquelle dient. Weiterhin entwickelt das Team selbst über seinen Projektverlauf die Backlogs weiter und macht sie damit zu einer

wertvollen Wissensquelle. Die Rolle ist überwiegend im Entwicklerteam sowie beim Product-Owner zu suchen.

Der Scrum-Master kann für einige im Team die **Vertrauensperson** werden, dies ist jedoch nicht immer so. Die Rolle selbst ist nur schwer zu lokalisieren, weswegen es keine allgemeingültige Zuordnung gibt. Jedoch hat es enorme Vorteile, wenn der Scrum-Master das Vertrauen des Teams genießt. Wie sich später in den Interviews (s.a. Kapitel 6.2.2) noch zeigen wird, ist das für viele Scrum-Master die Basis ihrer Arbeit, das sie diese Rolle besetzen und mit den daraus gewonnenen Informationen dem Team dienen um Konflikte und Probleme zu lösen.

Teamrolle	**Scrum-Rolle**	**Bemerkung**
Visionär	Product-Owner	in Bezug auf das Produkt
Pragmatiker	Entwickler Scrum-Master	
Entdecker	Scrum-Master	
Herausforderer	alle	inspect & adapt
Unparteiischer	Scrum-Master	teilweise
Friedensstifter	Scrum-Master	nicht alleinig
Arbeitstier	Entwickler	
Trainer	Scurm-Master erfahrenes Teammitglied	je nach Trainingsgegenstand sehr unterschiedlich
Bibliothekar	Entwickler Product-Owner	abhängig von der Art der Dokumentation
Vertrauensperson	Scrum-Master	muss nicht, bringt viele Vorteile

Tabelle 5.1: Teamrollen nach Spencer und Pruss in Verbindung mit den Scrum-Rollen. Es zeigt sich, das es einige wichtige Rollen gibt, die durch Scrum-Rollen bedacht sind, wiederum andere sind sehr vom Thema und der Situation abhängig.

Fazit Es lassen sich viele Rollen wie in Tabelle 5.1 lokalisieren, jedoch sind einige auch nicht direkt von Scrum-Rollen besetzt. Diese Rollen müssen im Team im Auge behalten werden. Augenfällig ist die klare Rollentrennung zwischen Visionär und Pragmatiker, welche in unterschiedlichen Scrum-Rollen zu finden sind. Viele Rollen aus Spencer und Pruss können auf auf die Scrum-Rolle des Scrum-Masters zumindest teilweise zugeordnet werden. Dies kann der Scrum-Master daher nicht alleinig ausführen und es muss ich

Gedanken machen werden, was diese starke Personalunion für Auswirkungen haben kann. Gerade dieser starke Fokus kann zu einer Überlastung des Scrum-Masters sowie dem Product-Owner führen, da dies Schlüsselrollen sind die nur durch eine einzelne Person besetzt werden.

5.4.2 Belbins Team-Roles-Self-Perception Inventory

Eine andere Sicht auf die Scrum-Rollen und insbesondere die Verteilung der Rollen im Scrum-Team ergibt sich, wenn die Rollen nach Belbin aus Abschnitt 2.6.1 herangezogen werden. Belbin unterteilt die Rollen dabei in Gruppen, die einen interessanten Rückschluss auf die notwendigen Kompetenzen der Teammitglieder ermöglichen. In Tabelle 5.2 sind wahrscheinliche Rollenverteilungen dargestellt. Dabei ist zu beachten, dass diese Rollenverteilungen nicht in jedem Team vorzufinden sind, sondern aus rein theoretischen Überlegungen entstanden sind.

Rolle nach Belbin	**Scrum-Rolle**
Handlungsorientierte Rollen (Action)	
Macher (SH)	Product-Owner Entwickler
Umsetzer (IMP)	Entwickler
Perfektionist (CF)	Entwickler
Kommunikationsorientierte Rollen (Social)	
Koordinator (CO)	Scrum-Master Entwickler
Teamarbeiter (TW)	Scrum-Master Product-Owner Entwickler
Weichensteller (RI)	Scrum-Master
Wissensbasierte Rollen (Thinking)	
Erfinder (PL)	Entwicklerteam Product-Owner
Beobachter (ME)	Entwickler
Spezialist (SP)	Entwickler

Tabelle 5.2: Teamrollen nach Belbin in Verbindung mit den Scrum-Rollen. Die Rollenverteilung ist dabei nicht absolut und eindeutig zu sehen, vielmehr handelt es sich um eine Tendenz, welche Funktion durch die Entsprechende Scrum-Rolle wahrscheinlich wahrgenommen wird.

Zunächst werden die **wissensbasierten Rollen** betrachtet. Diese sind vor allem bei der Umsetzung während der Entwicklung sehr wichtig (zur Erinnerung: Softwareentwicklung ist Wissensarbeit). Diese Rollen sind überwiegend bei den Softwareentwicklern zu besetzen, denn sie haben das notwendige Fach- und Spezialwissen bei der Umsetzung und ihre Hauptaufgabe ist es, das Wissen bzw. die Informationen des Product-Owners in einen ausführbaren Code umzusetzen. Nicht zu vergessen ist auch der Product-Owner als eine starke Wissensquelle. Nur im Zusammenspiel zwischen ihm und den Entwicklern wird es den Entwicklern möglich, die Vorstellungen der Kunden in das Produkt umzusetzen. In gewisser Weise hat der Product-Owner die Funktion des Spezialisten für den Kunden. Meist ist der Product-Owner auch ein Erfinder, da er die Kundenvorstellung schon in Ideen und Lösungsvorschläge umsetzen kann, die er mit den Entwicklern weiter verfeinert (s.a. Sprint-Planung). Der Scrum-Master ist in diesem Bereich als Rolle selbst nur wenig gefragt.

Der Scrum-Master ist im sozialen Bereich anzutreffen. Hier ist er umso mehr gefragt. Beim Betrachten der **kommunikationsorientierten Rollen** wird schnell klar, wo das Hauptbetätigungsfeld des Scrum-Masters liegt. Er ist ein wichtiger Bestandteil als Koordinator, zum Teil als Teamarbeiter und vor allem, wenn es darum geht Impediments auszuräumen, als Weichensteller. Wichtig ist jedoch das Verständnis der Rolle Koordinator als nicht alleinig durch den Scrum-Master wahrgenommene Rolle. Im Abschnitt 4.4 wurde bereits das Leadership-Modell von Scrum näher beleuchtet, welches für die Rolle des Koordinators ebenso zutrifft. Es ist davon abhängig, welche Aufgabenbereich betrachtet wird.

Die **handlungsorientierten Rollen** werden überwiegend durch die „Arbeiter“ ausgeführt. Sie sind für die Durchführung der eigentlichen Arbeit zuständig: Aufnahme der Kundenbedürfnisse, Design des Produktes, Implementierung und abschließender Test. Hier ist vor allem der Product-Owner in Zusammenarbeit mit den Entwicklern gefragt. Der Scrum-Master spielt darin so gut wie keine Rolle. Er begleitet zwar die Entwicklung und sorgt für einen reibungslosen Ablauf, jedoch ist er nicht direkt in die eigentliche Arbeit involviert.

Wichtige Rollen nach Chong Chong merkte an, dass besonders die Rollen Koordinator, Teamarbeiter und Erfinder von zentraler Bedeutung für das funktionieren eines Teams sind (Chong 2007).

Der Scrum-Master ist für den Koordinator sowie für die Rolle des Teamarbeiters eine wichtiger Kandidat und wird viele Aufgaben dieser Rollen ausfüllen. Als Koordinator wird er das Team vor allem anleiten, Entscheidungen zu treffen und das Team hierbei bestärken. Er wird das Team jedoch weniger kontrollieren, wie dies von einem Koordinator normalerweise gefordert ist. Diese Tätigkeit wird er in das Team durch strukturierte Reviews zurück geben und damit dafür sorgen, dass sich das Team selbst überwacht und gegebenenfalls Maßnahmen ergreifen kann. Er wird natürlich bei einer für ihn erkennbaren Fehlentwicklung dies zur Sprache bringen. Die Beschreibung der Rolle des Teamarbeiters liest sich, als wäre dies eine Stellenausschreibung für einen Scrum-Master. Er hat die wichtige Aufgabe, als Helfer im Hintergrund für das Team zu agieren und es zu unterstützen, gerade wenn es um die Kommunikation mit anderen Abteilungen geht. Dabei versucht er Impediments auszuräumen und damit dem Team zu helfen. Dabei fungiert er beispielsweise auch als Botschafter für agile Arbeitsweisen.

Der Erfinder ist im Scrum-Team nicht eindeutig zu identifizieren. Klassisch finden sich Erfinder sowohl auf seitens der Produktdenker, wie auch unter den Entwicklern. Dabei ist dies sehr vom Aufgabenfeld abhängig. Entwickler werden dabei mehr den technischen Fokus oder die Umsetzung im Blick haben, während es auf Seiten des Produktes der Product-Owner sein wird. Dort werden neue Vorgehensweisen und Möglichkeiten für die Lösung eines Kundenproblems erfunden. Oft ist es aber auch ein hochgradig komplexes Zusammenspiel des Teams, so dass aufgrund neuer Sichtweisen etwas besonderes entsteht. Es ist also nur schwer möglich, diese Rolle in einem Framework, wie Scrum abzubilden.

Fazit Durch die Aufteilung der Rollen in Gruppen ergibt sich ein interessantes Bild:

1. Die handlungsorientierten Rollen werden überwiegen von den Entwicklern sowie dem Product-Owner eingenommen. Sie sind dafür verantwortlich, das Produkt zu erschaffen und zu designen. Für die Durchführung ihrer Arbeit ist es weiterhin notwendig, dass sie die wissensbasierten Rollen ebenso besetzen. Nur durch das Wissen der Rollenträger ist es möglich, das Produkt zu erschaffen. Sie führen die Arbeit im eigentlichen Sinn aus.

2. Die kommunikationsorientierten Rollen werden durch alle Teammitglieder gleichermaßen besetzt. Dabei sticht heraus, dass der Scrum-

Master sein Aufgabenfeld in diesem Bereich hat. Er ist das soziale Bindeglied im Team und sorgt für positive Rahmenbedingungen, die zum Funktionieren des Teams beitragen.

Gerade der Fokus auf die wichtigen Rollen zeigt dabei, dass der Scrum-Master durchaus eine zentrale Rolle ist, die für das Team sehr wichtig ist. Der Scrum-Master ist zwar nur ein Dienstleister seines Teams, hat dort aber eine wichtige Aufgabe, indem er das Team dort unterstützt, wo es notwendig ist. Gleichzeitig ist er auch davor geschützt, dass sein Team zu viel von ihm erwartet. Der Scrum-Master wird nur solche Dinge tun, in denen er besonders gut ist; dies beugt einer Überlastung und eines Abladens von Verantwortung (wie es gerne in klassischen Teams auf die Manager getan wird) auf den Scrum-Master vor. Dieser Rolle gilt daher ein besonderes Augenmerk, denn dort ist ein hochgradig sozialer Mitarbeiter vonnöten, der das notwendige Feingefühl hat.

Zu beobachten ist vor allem die Rolle des Erfinders, da diese nicht explizit an ein Teammitglied zugewiesen werden kann. Sie muss sich finden. Wird diese nicht besetzt, so kann es für das Team schwierig werden, Höchstleistungen zu erbringen.

5.4.3 Zusammenfassung und Bedeutung für die Forschungsfrage

Die Scrum-Rollen können zum Teil auf die Rollen der vorgestellten Modelle abgebildet werden. Scrum ist jedoch nicht dafür geeignet, eine vollständige Abbildung durchzuführen. Vielmehr kümmert es sich partiell um einige Rollen, die es als wichtig erachtet. Dabei besetzt es aus seiner Sicht kritische Rollen durch die Scrum-Rollen und gibt diesen Rollen einen funktionsorientierten Ansatz mit. Dabei wurde von den Entwicklern von Scrum aus der Praxis heraus wichtige Rollen benannt, was besonders beim Vergleich mit den Rollen von Belbin deutlich wurde. Gerade die soziale Komponente des Scrum-Masters ist dort wichtig. Gerade auch die Studien von Chong und der Besetzung von Scrum zeigen, dass hier viele Vorkehrungen getroffen wurden, um zum einen negative Entwicklungen vorzubeugen und zum anderen eine hohe Performance des Teams durch eine klare Struktur zu erreichen. Weiterhin wird durch Scrum möglich, Rollen explizit zu machen. Somit haben Teammitglieder an diese Rollen die selben Erwartungen und die Rollenträger wissen was von ihnen in dieser Rolle gefordert wird. Dies

beugt Rollenkonflikte frühzeitig vor und gibt dem Team zusätzliche Orientierung. Die Gegenüberstellung hat allerdings auch gezeigt, das viele mögliche Rollen nicht durch eine einzelne Person abgedeckt werden kann. Dies kann zu Problemen durch Überlastung der Personen führen.

Kapitel 6

6 Best-Practices zum Einsatz agiler Methoden im Unternehmensumfeld

Wie können agile Framewoks aus Sicht des Unternehmens bzw. der Führungskräfte (im Sinne der Verantwortlichen für den Erfolg des Teams / Projektes) im Unternehmen eingesetzt werden und welche Methoden werden in der Praxis genutzt, um die theoretisch auftretenden Lücken zu füllen und eine effektive Teamarbeit sicherzustellen.

6.1 Experteninterview

In Kapitel 1.2 wurde bereits die grundlegende Methodik der Arbeit vorgestellt. Anknüpfend an die dort vorgestellte Vorgehensweise müssen wir uns mit dem Experteninterview auseinandersetzen. Das Experteninterview als Methode ist nicht standardisiert oder ausdrücklich so benannt worden. Unter den Forschern hat sich eine einheitliche Vorstellung von der Methode etabliert: „Im Grunde scheint (…) Konsens darüber zu herrschen, dass das Experteninterview ein Leitfadeninterview zu sein habe (…)." (Liebold und Trinczek 2009, S. 32) definiert das Experteninterview damit als ein (halbstandardisiertes) Interview entlang von Leitfragen. Das Thema und die Richtung des Interviews wird durch die Leitfragen vorgegeben. Die in der Situation des Interviews gestellten Fragen sind offen und können im Laufe des Gesprächs angepasst werden. Es ist weiterhin möglich, Erkenntnisse aus vorherigen Antworten zu hinterfragen und zu vertiefen. Damit ermöglicht die

Methode auch noch unbekannte Informationen zu gewinnen, deren Existenz vor dem Interview unbekannt war. (Hildebrandt et al. 2015, S. 241-255; Liebold und Trinczek 2009; Blöbaum, Nölleke und Scheu 2015)

Grundsätzlicher Ablauf Das Leitfrageninterview als Methode benötigt folgende Schritte:

1. **Vorbereitung**
    a) Erarbeiten der theoretisch-wissenschaftlichen Vorüberlegungen für die Leitfragen
    b) Auswahl (und Kontaktaufnahme) der Gesprächspartner
    c) Erstellung der Leitfragen und dessen Aufteilung in sinnvolle Abschnitte
2. **Durchführung** der Interviews mit anschließender Transkription
3. **Auswertung** der Interviews

Rahmenbedingungen Die Interviews wurden persönlich mit den Gesprächspartnern durchgeführt. Vorab wurde die Zeit auf 1 Stunde für das reine Interview-Gespräch fixiert, jedoch wurden die Personen gebeten, sich 1,5 Stunden Zeit frei zu halten. Dies ermöglichte es, in einer angenehmen Atmosphäre ohne Zeitdruck arbeiten zu können. In einem Vorgespräch wurden die Personen über die Ziele aufgeklärt. Dies ermöglichte es, dem Gesprächspartner, sofern sie mich noch nicht kannten, einen kurzen Eindruck von mir als Person zu geben. Damit konnte Vertrauen für das folgende Gespräch aufgebaut werden.

6.1.1 Auswahl der Gesprächspartner

Es stellt sich die Frage, was für uns ein Experte ist. Eingangs im Kapitel 1.2 von der Methodik der Arbeit wurde die Frage schon teilweise beantwortet, es ist eine Person mit hohem Wissen im Forschungsfeld. Die Frage stellt sich natürlich, wo sind solche Person zu finden oder gehört sie sogar einer Personengruppe an. Kühl, Strodtholz und Taffertshofer 2009, S. 33 schreibt dazu sehr treffend „(…) aufgrund der zunehmenden Ausdifferenzierung (…) spezialisierter Rollen entsteht, was sich auf der Wissensebene als ein System differenzierten Fachwissens darstellt.“. Dies beschreibt im Wesentlichen die

Merkmale der modernen Arbeitsteilung und einen Trend zu einer deutlichen Spezialisierung der Mitarbeiter bzw. Rollenträger. Für uns ist daher das Kriterium zur Auswahl des Experten, dass bei ihm ein spezielles Wissen vorhanden ist, an dem Interesse besteht.

Blicken wir zurück auf das vorangegangene Kapitel, so ist Wissen, welches in erster Linie ein Coach oder eine Führungskraft besitzt, interessant. In der agilen Welt ist die Führungskraft als Rolle im Team nicht explizit vorhanden (es handelt sich um ein selbstorganisiertes Team), der Coach hingegen schon. Im Scrum-Umfeld gibt es einen sehr wichtigen Coach, den Scrum-Master. Personen, welche diese Rolle ausgefüllt haben, sind daher von besonderem Interesse für uns. Weiterhin gibt es zahlreiche agile Coaches, welche bei der Einführung und Durchführung von agilen Methoden Unternehmen und Mitarbeiter sowie die Teams beraten. Diese verfügen im Allgemeinen einen unternehmensübergreifenden, breiten Horizont und werden besonders in kritischen Situationen eingesetzt. Zurück zu den Führungskräften, in einer typischen Organisation werden die agilen Teams dennoch eine vorgesetzte Führungskraft haben und wenn es der Geschäftsführer ist. Damit ist die relevante Personengruppe abgesteckt, indem sie über die folgenden Rollen im Unternehmen definiert wird:

- Scrum-Master
- Coach für agile Teams im Allgemeinen
- Führungskraft, welchen in einem Unternehmen agile Teams unterstellt sind.

Während des Interviews wurden 4 Personen befragt, die mindestens eine der vorherigen Kriterien erfüllten. Die Personen haben ihr Wissen zum Großteil in verschiedenen Unternehmen gewonnen, was eine gute Durchmischung der Unternehmenskultur bringt. Da jedoch für agile Vorgehensweisen ein gewisses Mind-Set vorhanden sein muss, sind die Firmen, in welchen agile Methoden erfolgreich eingesetzt werden, von der Kultur ähnlich, vor allem wenn dies mit allen Unternehmen verglichen wird. Aufgrund der zum Teil erheblichen Bedenken der Interviewpartner wurden die Inhalte, wie auch die Namen der Befragten anonymisiert.

Im Folgenden eine kurze Auflistung der Interviewpartner. Weitere Informationen zur Fachexpertise sind im Anhang C zu finden.

1. Interview Die Person arbeitet überwiegend als Scum-Entwickler und hat vorher eine lange Zeit auch als Scrum-Master Erfahrung gesammelt.

2. Interview Die Person ist als Consultant im agilen Umfeld tätig. Dabei unterstützt die Person Unternehmen bei der agilen Transition, wie auch der konkreten Ausgestaltung des Scrum-Prozesses im Unternehmen. Dabei werden die Unternehmen über einen längeren Zeitraum begleitet. Weiterhin hat die Person auch bereits als Scrum-Master in diversen Teams für kurze Zeiträume gearbeitet.

3. Interview Die Person ist aktuell als Scrum-Master für Teams tätig.

4. Interview Die Person ist Teamleiter für mehrere Teams bei einem Softwaredienstleiter. Dabei wird auch die Rolle des Scrum-Masters für aktuell zwei Teams übernommen.

6.1.2 Interviewleitfaden

Der Vorbereitung der Leitfragen ist für das spätere Interview mit den Experten ein wichtiger Schritt. Bei der Zusammenstellung der Leitfragen muss sowohl das Ziel wie auch die Eigenart eines Leitfrageninterviews mit berücksichtigt werden. Die Vorbereitungen aus den vorangegangenen Kapiteln fließen dabei ebenso ein, wie die Randbedingungen: Im Interview selbst steht nur eine begrenzte Zeit zur Verfügung. Gläser und Laudel geben den Hinweis, das pro Stunde zwischen 8 und 15 Fragen beantwortet werden können (vgl. Gläser und Laudel 2010, S. 144). Der Leitfragen enthält zu Beginn eine Einleitung, in welcher der Befragte über das Ziel, seine Rolle und die Randbedingungen für die Veröffentlichung informiert wird. Der Leitfaden wurde unter Zuhilfenahme von Gläser und Laudel 2010, S. 142 ff. erstellt. Der letztendlich genutzte Leitfragen ist in Anhang B abgedruckt.

6.1.3 Durchführung des Interviews

Das Interview selbst wurde durch eine Tonaufnahme begleitet, so dass dies im Nachgang ein Transkript angefertigt werden konnte. Die Durchführung der Interviews dauerte zwischen 90 und 120 Minuten. Dabei wurde vor allem zu Beginn den Interviewpartnern der Hintergrund der Forschung erläutert, wie der Ablauf des Interviews ist und wie das Interview verwertet wird. Da die Interviews Details zu Abläufen und Erfahrungen von Personen und Unternehmen beinhalten, wurden die Interviews anonymisiert erhoben. Die Transkriptionen sind im Anhang C abgedruckt.

6.1.4 Hinweise zur Typographie für Zitate aus den Interviews für diese Arbeit

Die Transkriptionen aus den Interviews sind im Anhang C zu finden. Für die Auswertungen der Interviews und die korrekte Zitierung der Quellen aus den Interviews wird für eine verbesserte Les- und Auffindbarkeit folgendes Schema zur Identifikation verwendet:

- Die erste Stelle besteht aus einem „T“ für Transkription.
- Die zweite Stelle ist die Nummer des Interviewpartners (1, 2, …).
- Die dritte Stelle kennzeichnet die Referenz innerhalb des Gesprächs. Diese werden fortlaufend nummeriert.

Zum Beispiel, das erste Zitat aus dem 2. Interview wird wie folgt in der Arbeit dargestellt: „Entnommenes Zitat“ (T2.1 auf S. 192). Zum leichteren Auffinden der Zitate wird zusätzlich noch die Seitenzahl genannt. An der zitierten Stelle ist wiederum die Identifikation hochgestellt abgedruckt, wie in diesem Beispiel: Zitierte Stelle$^{T2.1}$. Das Datum des Interviews kann ebenfalls dem Anhang entnommen werden.

6.2 Auswertung der Experteninterviews

Bei der ersten Durchsicht der Interviews wurde bereits eine grobe Einteilung in negative Entwicklungen / Konflikte, Diagnostik und mögliche Lösungen vorgenommen. Zusätzlich wurden weitere interessante Stellen herausgearbeitet, welche das Thema agile Teams tangieren, jedoch nicht direkt mit den Forschungsfragen in einem direkten Zusammenhang stehen. Dabei zeigte sich, dass die einzelnen Bereiche nicht immer klar abzugrenzen sind und von den Beteiligten oftmals zwischen negativen Entwicklungen, Diagnose und Vorsorgemaßnahmen gesprungen wurde. Für manche negative Entwicklung wurden im Vorfeld bereits präventiver Maßnahmen getroffen, so dass die ein oder andere negative Entwicklung gar nicht erst auftritt.

Strukturieren des Wissens aus den Interviews Nach der ersten groben Durchsicht wurden die Themenbereiche unterhalb der groben Einteilung identifiziert. Zunächst sind die Themenblöcke „negative Entwicklungen“ und „Diagnose“ herausgearbeitet worden. Diese wurden zunächst unabhängig betrachtet; es stellte sich schnell heraus, das diese beiden Themen nur

schwierig in einem losgelösten Kontext betrachtet werden können. Vielfach kann eine negative Entwicklung direkt beobachtet werden, meist ist dann auch die Lösung nicht weit entfernt. Es macht daher mehr Sinn, die einzelnen Felder nicht komplett getrennt zu betrachten, sondern sie als Themenblöcke zu nutzen und anhand dessen die weitere Arbeit zu gliedern. Zuerst werden die negativen Entwicklungen und Voraussetzungen für Scrum vorgestellt (siehe Kapitel 6.2.1), wenn dort jedoch spezifische Diagnose und Lösungsmöglichkeiten vorhanden waren, wurden diese dort eingepflegt. Dies erleichtert die Lesbarkeit deutlich, da ansonsten die Themenblöcke über mehrere Kapitel verstreut worden wären. Es gibt dann weiterhin noch einige allgemeinen Diagnoseinstrumente (siehe Kapitel 6.2.2) bzw. Voraussetzungen welche für die Diagnose in Zusammenhang mit Teams unabdingbar sind. Ebenso gibt es später allgemeine Beobachtungen und Lösungsmöglichkeiten (siehe Kapitel 6.2.3).

Erfahrungen und wie explizit liegt das Wissen vor Bei der Auswertung zeigten sich einige Auffälligkeiten, die vor allem mit der Erfahrung des Befragten zusammenhängt. Es zeigten sich sowohl Unterschiede bei der Art der Vorgehensweise in der täglichen Arbeit wie auch wie explizit das Wissen schon in den Äußerungen des Gegenübers vorliegt. So zeigte sich, dass gerade Scrum-Master in ihrer täglichen Arbeit überwiegend instinktiv handeln, dieses Handeln aber auch reflektiert wird. Je mehr sich eine Person mit dem Themengebiet beschäftigt hat und darüber auch Vorträge gehalten hat oder sein Wissen verschriftlichen musste, desto einfacher fällt es dieser Person das Wissen in eine explizite Form überzuführen und strukturiert vorzugehen. Dies zeigte sich besonders im Unterschied des Interviews zwischen den Scrum-Mastern und dem Consultant.

6.2.1 Konflikte und negative Entwicklungen im agilen Team

Grundsätzlich sind alle Themenfelder aus der normalen Teamentwicklung ebenso für agile Teams relevant und wurden durch die Experten bestätigt. Daneben gibt es jedoch einige essentielle Voraussetzungen, damit ein agiles Team zu einer unabhängigen Arbeitsweise und einer hohen Performanz kommen kann. Eine naheliegende Einteilung ist in Kontext-Faktoren, also von außen auf das Team treffende Faktoren, und Teamfaktoren. Diese Ein-

teilung kann auch für einen ersten Hinweis gesehen werden, wer zur Lösung beizutragen hat:

Teamfaktoren können immer nur durch das Team selbst gelöst werden. Hierzu muss das Team sich und seiner Art und Weise des Arbeitens verändern. Es kann jedoch der Fall sein, dass das Team unüberwindbare Hindernisse in seiner Zusammenarbeit findet, diese bedürfen dann unter Umständen Entscheidungen von außen (z.B. Entfernung oder Hinzufügen eines Teammitglieds).

Kontext-Faktoren werden meist zwar vom Team identifiziert, dann aber durch den Scrum-Master oder einen Beauftragten weiter verfolgt, so dass eine Lösung gefunden werden kann. Meist sind sie auch die Voraussetzung, damit ein agiles Arbeiten überhaupt möglich ist.

Leider sind nicht alle Dinge exakt dem einen oder anderen zuzuschreiben. Beide Faktoren können sich gegenseitig beeinflussen und es können Themenfelder auch auf beide zutreffen. Im Ganzen ist es jedoch eine sinnvolle Einteilung, die wir weiter nutzen möchten. Dabei werden zunächst die Kontext-Faktoren verwendet und diese gehen dann fließend in die Teamfaktoren über.

Rahmenbedingungen für Agilität

Die Organisation muss die Mittel und vor allem den Rahmen schaffen, dass agile Methoden überhaupt gelebt werden können. Im Grunde handelt es sich hierbei um einen sehr großen Block; eine der Hauptprobleme ist es, dass in der Übergangszeit eine Veränderung statt findet. Je nachdem, wie das Mind-Set des Managements in der Organisation ist, wird diese schwierig, oder gar unmöglich. Wenn sich ein Unternehmen zu einer Aussage, wie „Wir machen Scrum, aber es darf sich nichts ändern!“ (T1.11 auf S. 167) hinreisen lässt, dann wird das nichts. Gerade die Veränderungen von Zuständigkeiten, Rollen und damit auch der für viele Manager gefühlte Kontrollverlust kann zum Widerstand bei der Transition oder auch später werden. Dieses Bedürfnis nach Kontrolle eines klassischen Managers (vgl.T2.21 auf S. 205) kann von vorn herein vorhanden sein oder immer wieder während des Prozesses auftreten, vor allem wenn Probleme sichtbar werden oder durch das Management vermutet werden.

Raum und Zeit

Ein zentraler Wert in agilen Vorgehensweisen ist die Kommunikation (s.a. 4.1.1). Eine enge Zusammenarbeit des Teams fördert die Kommunikation. Dabei ist diese Zusammenarbeit sowohl räumlich als auch zeitlich zu verstehen. Zwar ist es gerade heute in Mode, Heimarbeit und flexible Arbeitszeiten zu haben, jedoch darf darunter die Kommunikation und die soziale Interaktion nicht leiden (vgl. T2.43 auf S. 209).

Colocation wird gerade im agilen Kontext gefordert, um die Zusammenarbeit zu fördern. Nur wenn genug Teammitglieder verfügbar sind, können auch Praktiken wie Pair-Programming oder Reviews sinnvoll und zeitnah durchgeführt werden (vgl. T4.4 auf S. 249 und T2.42 auf S. 209). Die Auflösung einer ungünstigen Situation kann einem Team durchaus einen messbaren Schub bei der Performanz bringen, wie in T4.5 auf S. 250 angemerkt wurde. Klassischerweise wird unter Colocation nur der räumliche Aspekt verstanden, eine ebenso große Auswirkung hat auch der zeitliche Aspekt (vlg. T4.8 auf S. 251). So unterschiedlich wie die Persönlichkeiten sind, so unterschiedliche Präferenzen haben sie für die Lage ihrer Arbeitszeit. Wenn die Hälfte des Teams um 7 Uhr beginnt, die andere jedoch erst um 11 Uhr, so haben sie bei einer 40-Stunden Woche nur eine gemeinsame Zeit von 4 Stunden, die eine Zusammenarbeit deutlich erschweren können. Es treten dann vermehrt Impediments in Form von nicht Anwesenheit von Review-Partnern oder Wissensträgern auf (vgl. T3.43 auf S. 239). Es ist damit wichtig, dass Teammitglieder fürs Team verfügbar sind, wenn sie benötigt werden.

In einem typischen auf Projekte zentrierten Unternehmen werden Personen oftmals in vielen Projekten gleichzeitig verplant. Hiervon ist in einem agilen Umfeld dringend abzuraten, da hierdurch der Arbeitsfluss, die Planbarkeit und Effizienz verloren geht: „(…) die Zeit fehlt, wenn er einen halben Tag irgendwas anderes machen soll (…) Kontextwechsel und der Flow geht damit verloren (…)“ (T4.22 auf S. 260). Nicht zu verachten ist bei einer solchen Vorgehensweise auch die Belastung, die als Unternehmen zu vermeiden sind: „(…) dieses ständige Springen zwischen mehreren Projekten führt zu zusätzlicher kognitiver Last und das führt halt dazu, dass man irgendwann aus- bzw. überlastet ist und Überlast, da kann ich dann wieder die ganzen Stresstheorien anwenden, d.h. das führt dann wieder zu Burnout, Motivationsverluste, usw. (…)“ (T2.9 auf S. 198)

Aus diesem Grund ist es sicher sinnvoll, zu definieren, welchen Anteil an Arbeit aufzuwenden ist, damit eine sinnvolle Zusammenarbeit möglich ist:

„Jemanden in 2 Scrum-Teams gleichzeitig drinnen zu haben geht nicht, das ist kein Scrum-Team mehr. Jemand, der unter 80% für ein Projekt verfügbar ist, ist ein Teilzeitteammitglied, jemand der unter 60% verfügbar ist, ist kein Teammitglied mehr. D.h. ich kann keine Leute in 2 bis 3 Scrum-Teams rein setzen.“ (T2.8 auf S. 198). Dies scheint ein wichtiger Punkt zu sein, damit ein Scrum-Team wirklich funktionieren kann.

Negative Entwicklungen können sich besonders bei Änderungen im Arbeitsablauf oder durch die Organisation selbst ergeben. Wenn plötzlich Mitarbeiter in mehreren Teams gleichzeitig eingesetzt werden sollen oder ein zusätzlicher Dienstleister genutzt wird, der jedoch seien Hauptsitz an einem entfernten Ort hat. Die Situationen können sehr unterschiedlich sein, jedoch sollte dies immer im Hinterkopf gehalten werden, dass, je besser das Kriterium Colocation erfüllt werden kann, umso besser kann die Kommunikation laufen und umso weniger Probleme sind zu erwarten.

Autonomie, Feedback und Übernahme von Verantwortung

Durch einen Interviewpartner wurde in T2.10 auf S. 199 das Job-Characteristics-Model aus Hackman und Oldham 1976 vorgestellt. Dieses Modell ist insoweit sehr interessant, da es zwei Kernaspekte der agilen Arbeitsweise enthält: **Autonomie** und **Feedback**. Zwar sind die anderen Aspekte auch von Bedeutung, hierzu gibt es im Agilen zunächst einmal keinen direkten Bezug, außer dass sie in den Werten indirekt verwurzelt sind. Feedback und Autonomie ist jedoch ein zentraler Bestandteil des agilen Konzepts, denn es ist ganz wichtig: „(…) Leute müssen der Arbeit eine Bedeutung zumessen, sie müssen sich für die **Ergebnisse verantwortlich fühlen** und sie müssen auch die **Ergebnisse tatsächlich sehen und mitkriegen** (…)“ (T2.11 auf S. 199, Herv. d. Verf.).

Feedback In agilen Methoden wird auf vielfältige Weise Feedback (siehe Tabelle 6.1) erhoben und hat dort vor allem folgende Aufgaben:

Motivation erlangt das Team unter anderem aus dem Feedback. Es unterstreicht, dass die vom Team geleistete Arbeit einen wichtigen Zweck erfüllt und das diese gebraucht wird. Das motiviert das Team auch, seine Ziele zu erreichen, auch wenn die Erreichung manchmal schwierig oder mit Entbehrungen verbunden sein kann (vgl. T4.14 auf S. 255).

Verbesserungen kann das Team nur anstoßen, wenn es sich dem Problem bewusst ist. Hierfür ist Feedback notwendig, wobei das Team selbst Feedback generiert oder Feedback von außen für diesen Zweck nutzen kann. „Scrum ist immer ein evolutionärer Prozess (…)“ (T4.16 auf S. 257) beschreibt dies sehr gut. Die Bewertung, ob eine Vorgehensweise positiv ist, wird vor allem durch das Feedback gesteuert und dann angepasst oder eingestellt.

Scrum-Artefakt	**Feedback-Art**	**Regelmäßigkeit**
Stand-Up	Feedback über Arbeitsweise / Produkt /aktuelle Arbeit	täglich
Sprint-Retrospektive	Feedback über Arbeitsweise	Sprintende
Sprint-Review (Demo)	fachliches Feedback vom Kunden	Sprintende

Tabelle 6.1: Scrum generiert regelmäßig verschiedene Arten von Feedback.

Scrum bietet hier vor allem den Vorteil, das es inhärente Merkmale besitzt, um schnelles und wiederkehrendes Feedback zu ermöglichen: „Scrum gibt Mittel für schnelles Feedback vor, auf verschiedenen Ebenen.“ (T1.27 auf S. 176). Schnelles Feedback ist vor allem durch die Regelmäßigkeit der Feedbackschleifen begründet (vgl. T1.32 auf S. 178 und T1.33 auf S. 178), wobei im Scrum zwei wesentliche Arten von Feedback generiert werden, das fachliche Feedback und das Feedback über die Arbeitsweise (siehe Tabelle 6.1).

Es zeigte sich, dass es Probleme mit dem Feedback geben kann, wobei die Befragten dies auf zwei verschiedene Blöcke aufteilten:

1. Das Feedback wird durch das Team nicht oder nur ungenügend genutzt T1.28 auf S. 177.
2. Das Feedback ist nicht vorhanden oder nicht ausreichend.

Während Punkt 1 nur von einem Einzelnen direkt angesprochen wurde, ist Punkt 2 in der Praxis relevanter. Dies zeigt sich in den Interviews, wo über die Qualität des Feedbacks deutlich mehr gesprochen wurde, als die Auswertung des Feedbacks. Es war für die Interviewpartner selbstverständlich, dass ein gegebenes Feedback immer genutzt und verarbeitet wird. Punkt 1 ist wohl vor allem in demotivierten Teams anzutreffen oder in Teams, die Probleme nicht sehen möchten, wie in T2.64 auf S. 219 beschrieben wurde.

Feedback ist oft an ein zu erreichendes Ziel geknüpft. So steckt sich das Team beim Scrum ein Sprint-Ziel, das es erreichen möchte und gibt darüber ein

Commitment ab. Dafür muss das Ziel für das Team erreichbar und vor allem lohnenswert sein. Ersteres hat es beim Scrum in der Hand, bei letzterem kann es sein, dass äußere Randbedingung dem widersprechen, wenn z.B. nach dem Sprintende das Produkt doch nicht ausgerollt wird und das vorher schon bekannt war (vgl. T3.28 auf S. 234). Hier bekommt das Team eine Art von Feedback, das das Team nutzen wird und sich für das nächste Ziel unter Umständen weniger anstrengen wird. Das ist dann ein Scrum-Prozess, bei dem das Inkrement nicht sofort dem Kunden zur Verfügung gestellt wird T3.27 auf S. 234. Ein typisches Beispiel hierfür ist die Medizintechnik. Diese Sparte ist streng reguliert, dort darf eine fertige Software erst nach weiteren und oftmals arbeitsaufwändigen Qualifizierungen dem Kunden zur Verfügung gestellt werden. Damit verliert das Ziel, ein wirklich lieferbares Produkt zu erstellen, an Anreiz. Der Kunde spiegelt nichts zurück, außer dass es jetzt nicht sofort eingesetzt werden kann. Dies muss in besonderem Maße beobachtet werden, damit dieser negative Einfluss nicht zu groß wird.

Auch eine Art von Feedback ist: „(…) Product-Owner, Kunde (…) aufhört, das zu überprüfen und das Interesse an den einzelnen Inkrementen einzustellen, dann wird auch das Team aus meiner Sicht in den Sprints oder in den Iterationen keinen Aufwand treiben, sich da zu kontrollieren oder das Ziel zu erreichen.“ (T4.13 auf S. 254), „Worst-Case: Kein Feedback oder das die Arbeit nicht gebraucht wird.“ (T2.15 auf S. 201). Das ist natürlich ein problematisches Feedback, denn hier wird dem Team signalisiert, dass die aktuelle Arbeit gar nicht so wichtig ist. Nicht selten wird dann der Fokus aufzugeben und nach folgender Prämisse weitergearbeitet: „(…) die Leute machen halt im Zweifelsfall eher das, was ihnen Spaß macht, was sie gut finden, was sie richtig finden, aber nicht unbedingt das, was der Kunde, der halt sozusagen das Produkt am Ende haben möchte, das Ergebnis haben möchte als Wichtigstes braucht (…) .“ (T2.45 auf S. 210). Im Sinne des Unternehmens ist das nicht, da wichtigen Dinge wie „(…) Kundenfokus, (…) Ergebnisfokus, (…) Qualitätsfokus aus den Augen verloren werden (…)“ (T2.44 auf S. 210). Gerade bei Scrum ist es daher wichtig, dass immer wieder Feedback erzeugt oder gegeben wird. Nur so kann ein Inspect & Adapt erfolgen. Das Feedback ist ein wichtiger Steuerungsmechanismus für das Team selbst. Ebenso ist die Qualität des Feedbacks ausschlaggebend: "(…) Feedback **muss vorhanden sein**, wichtig ist ein **schnelles** und **präzises inhaltliches** Feedback vom Kunden.“ (T2.14 auf S. 201, Herv. d. Verf.). Wenn einer dieser Punkte unterbleibt, so sinkt die Qualität und es müssen die Ursachen ergründet werden.

Autonomie Die von agilen Vorgehensweisen geprägte Veränderung und Agilität wird erreicht, indem es dem Team ermöglicht wird, sich in einem weiten Umfang selbst zu organisieren (s.a. Kapitel 3). Dies bedeutet allerdings auch, dass hierfür dem Team ein großes Maß an Autonomie eingeräumt werden muss. In der Praxis gibt es dort einige Fallstricke, die durch die erfahrenden Interviewer genannt wurden.

Die befragten Personen haben folgende Beispiele für notwendige oder fehlende Autonomie genannt:

- Prozesse der Arbeit müssen sich ändern lassen, um Anpassungen zu erlauben (vgl. T2.12 auf S. 200).
- Hierarchien können ein Problem sein, da diese im klassischen Verständnis ein anderer Zuständigkeitsbereich ist. Wenn sich jedoch in einem anderen Feld ein Problem befindet oder dort eine Lösung ansetzen soll, kommt es vermehrt zu Konflikten. (vgl. T2.13 auf S. 200 und T3.9 auf S. 226). Gerade wenn das Team agil ist, aber der Rest der Organisation nicht, kann es schwierig werden und ein Team demotivieren (vgl. T3.10 auf S. 227). Das Team stellt unter Umständen sogar die agile Vorgehensweise infrage und resigniert: „Warum machen wir eigentlich Agilität, wenn der Rest um uns herum auch nicht agil ist?“ (T3.8 auf S. 226).
- Das Team erstellt eine Planung, diese sollte im Normalfall so eine Iteration lang stehen bleiben. Wird diese häufig aufgrund von externen Einflüssen neu erstellt und angepasst, gefährdet das die Autonomie im Team (vgl. T4.23 auf S. 260).
- Rollen aus dem agilen Kontext müssen auch eine organisatorische Ermächtigung haben. Sie müssen soweit legitimiert sein, das sie ihre Aufgaben auch erfüllen können (vgl. T1.12 auf S. 168 und T1.29 auf S. 177).
- Externe Probleme belasten das Team, woran das Team selbst nichts ändern kann. Im Beispiel wurde dem Team ein Product-Owner zugeteilt, der noch nicht die notwendige Reife hatte und zunächst viele für das Team nicht umsetzbare Stories hervorbrachte. Dann kann es passieren, dass „(…) ein Team schnell konsterniert wirkt, wenn man merkt, dass von außen Probleme hereinkommen, an denen man relativ wenig tun kann.“ (T3.20 auf S. 231).

Die Autonomie kann allerdings auch durch eine falsch verstandene Rolle des Scrum-Masters verloren gehen (vgl. T4.11 auf S. 253). Gerade bei einer

Transition zu einer agilen Vorgehensweise kann es passieren, das der Scrum-Master seine Aufgabe weiterhin mehr als Projektmanager versteht und damit versucht das Team zu managen, anstatt sich in die neue Rolle des Coachs einzufinden. Auch eine oftmals schwierige Situation ist, wenn der Scrum-Master gleichzeitig der Linienvorgesetzte ist. Je nach Persönlichkeit des Linienvorgesetzten und dem Vertrauen ihm gegenüber, kann dies eine gute Lösung sein, oder zu Problemen im Team führen (Cohn 2010a, S. 437). Das verlangt allerdings Zurückhaltung, wie schon der erfahrene Scrum-Master, der auch gleichzeitig Linienvorgesetzter ist, sagt: „(…) du darfst dich natürlich auch nicht so verhalten, dass du der Chef von der Truppe bist." (T4.12 auf S. 253)

Auch wenn der Vorgesetzte außerhalb des Teams ist, passieren immer wieder negative Einmischungen in das Team, was besonders die Selbstorganisation und die Autonomie des Teams infrage stellt. Gerade dieses in das Team „hineinregieren" (vgl. T2.19 auf S. 204) kann schwere Auswirkungen auf das Verhältnis zwischen Führungsperson und Team haben. So kann innerhalb einer Sekunde das vorher mühsam aufgebaute Vertrauensverhältnis nachhaltig gestört werden, vor allem, wenn die Meinung des fachlichen Teams von dem des Managements abweicht (vgl. T2.20 auf S. 205). Das Management muss sich im Umgang mit agilen Teams eine andere Art der Einflussnahme aneignen, die vielmehr durch Geben von Feedback gekennzeichnet sein muss (vgl. T2.24 auf S. 205).

Der agile Manager versteht sein Team als einen Regelkreis, der genau weiß, wie er mit Problemen umgeht. Der Regelkreis weiß im Allgemeinen mehr über sein Fachgebiet, seine Probleme und die Lösungen als es das Management je könnte. Von daher gilt es für das Management, eine Struktur aufzubauen, so dass diese Wissenquellen genutzt werden und es nur im äußersten Notfall dort einzugreifen. Es arbeitet nach dem Paradigma: „Als agiles Management baue ich einen Regelkreis, ich sorge dafür, das ich ein System baue, das sich selbst verbessert und dann: „Get out of the way!" Dann lass ich dieses System in Ruhe." (T2.22 auf S. 205). Im Anschluss sollte ein agiler Manager das Ganze nur noch gelegentlich beobachten und Vertrauen in das Team entwickeln: „Ich schaue nur, ob diese Regelkreise funktionieren und optimiere ggf. die Regelkreise, aber ich rede nicht in die tägliche Arbeit, in die tägliche Veränderung rein" (T2.23 auf S. 205).

Das fällt allerdings nicht jedem Manager leicht, denn noch sind die alten Denkmuster in ihrer Arbeit verankert: „Das klassisches Management ist un-

heimlich kontrollbedürftig (…) .“ (T2.21 auf S. 205). Es benötigt eine große Portion Selbstvertrauen und Vertrauen in die Mitarbeiter, sich zu ändern und Mitarbeitern das Vertrauen entgegen zu bringen, Arbeiten zu verrichten, die vorher in seinem Aufgabenbereich lagen.

Natürlich ist eine Art von Kontrolle trotzdem notwendig: Das Team muss seine eigenen Vorgaben regelmäßig kontrollieren. Gerade aus Retrospektiven kommen Änderungsvorschläge und es werden Umsetzungen geplant. Diese müssen dann aber auch in den täglichen Arbeitsablauf integriert werden. Daher ist es ratsam, dass sich das Team selbst kontrolliert und die Wirksamkeit der Maßnahmen in der nächsten Retrospektive oder dem Stand-Up selbst prüft (vgl. T4.21 auf S. 259). Es muss verantwortungsvoll mit der gegebenen Autonomie umgehen.

Autonomie und Alignment Ein Reibungspunkt ist oftmals, dass kein gemeinsames Verständnis vorliegt, was wichtig ist und was am Ende herauskommen soll. Dies wird im klassischen Management von einigen wenigen Managern, wie Projektmanager oder Produktmanager vorgegeben. Im agilen ist es jedoch wichtig, da das Team auch strategische Entscheidungen treffen muss, dies zu verstehen und auch abwägen zu können. Es ist damit wichtig, dass „Autonomie und Alignment“ (T2.16 auf S. 202) hergestellt wird. Das Team muss sein Handeln an den Zielen des Unternehmens ausrichten können, dazu bedarf es deutlich mehr Wissen über diese. Dies wurde auch in (vgl. T3.23 auf S. 232) festgestellt, dass wenn keine Vision, kein Fokus vorhanden ist, dann führt das mitunter zu Unzufriedenheit im Team selbst. Dies kann besonders dann für das Team kritisch werden, wenn die gemeinsamen Normen und Standards nicht vorhanden sind. Dann hat jedes Teammitglied bzw. auch Abteilungen oder Personen, die mit dem Team eng zusammen arbeiten, eine andere Vorstellung. Dies führt dann zu vermehrten Konflikten im Team oder bei der Zusammenarbeit mit externen Teams (vgl. T3.24 auf S. 232).

Ziele des Teams versus persönliche Ziele Im klassischen „Management by Objectives“ ist es üblich, die Mitarbeiter über wohl definierte Einzelziele zu führen. Dies ist ein sehr gefährliches Mittel, wenn mit agilen Teams gearbeitet werden soll. Meist ist die Gefahr relativ gering, dass es unterschiedliche Ziele zwischen dem Team und den persönlichen gibt, jedoch für den Fall, dass es passiert, kann es das Team tief erschüttern. Im Zweifelsfall, vor

allem wenn das persönliche Gehalt an diesen Zielen hängt, wird sich die Person dazu entscheiden zunächst seine eigenen Ziele und nicht die Ziele des Teams zu verfolgen (vgl. T2.18 auf S. 204). Das kann sowohl das Vertrauen der Teammitglieder untereinander erschüttern, wie auch den Erfolg des Teams maßgeblich gefährden. Gerade wenn wir dabei an das Thema „Autonomie und Alignment" denken, fällt auf, dass das Team kein gleiches Alignment mehr hat, sondern das Alignment differiert. Meist zwar nur in einzelnen Punkten, aber es differiert in diesen Punkten und es kann damit zu Mikropolitik kommen. Diese Mikropolitik ist undurchsichtig und intransparent, es kommt weiterhin zu Misstrauen und dies ist einem agilen Team nicht zuträglich.

Verantwortungsübernahme Mit der gegebenen Autonomie erwartet das Management auch ein Übernahme der Verantwortung für einen Teilbereich des Unternehmens: Das agile Team! Dies kann jedoch nicht vom Management angeordnet werden, denn ob jemand die Verantwortung übernimmt ist letztendlich eine sehr persönliche Entscheidung einer Einzelperson und letztendlich dann auch des ganzen Teams. Nur durch die Übernahme der Verantwortung kann ein Team wirklich agil werden (vgl. T2.26 auf S. 206). Damit übernimmt das Team die Verantwortung über den Prozess (vgl. T3.35 auf S. 237) und macht es zu seinem eigenen, da es diesen gestaltet. Es übernimmt weiterhin die Verantwortung für seine eigenen Ziele, plant diese, denn für „Ziele, für die du nicht selbst verantwortlich bist, die sind für dich auch nicht attraktiv." (T4.24 auf S. 260). Es ist allerdings auch wichtig, dass Verantwortungsübernahme gelernt wird, denn das können viele von Natur aus nicht (vgl. T2.33 auf S. 207). Für eine Sache verantwortlich sein kann jedoch nur jemand, der auch genug Einfluss und Gestaltungsspielraum hat, so dass er wichtige Entscheidungen treffen und umsetzen kann und letztendlich dann auch sagt: „Ja ich bin bereit, ich übernehme die Verantwortung dafür." (T2.27 auf S. 206). Für die Übernahme der Verantwortung ist es jedoch auch notwendig, dass Fehler möglich sind und dass das Unternehmen eine für das Team akzeptable Fehlerkultur etabliert hat (vgl. T2.28 auf S. 206).

Persönliche Einstellungen und Merkmale

Jeder Mensch ist einzigartig, so hat er individuelle Bedürfnisse, eine individuelle Einstellung zu Arbeit und bestimmten Themen, sprich er hat eine Persönlichkeit. Diese Persönlichkeit kann es aber auch schwierig machen,

mit ihm in einem Team zu arbeiten oder sie bestimmt, wie gut ein Team zusammen arbeitet. So können sich Persönlichkeiten ergänzen, oder es kann sich zwischen zwei Menschen eine unüberwindbare Barriere aufbauen (vgl. T1.20 auf S. 171). Nicht jede Facette der Persönlichkeit hat eine Auswirkung auf die Leistung und Harmonie des Teams, jedoch gibt es gerade im Agilen einige wichtige und erwähnenswerte Dimensionen, die von den Interviewpartnern angesprochen wurden (vgl. T1.16 auf S. 170 und T2.29 auf S. 206).

Durchsetzungsstarke Persönlichkeiten In der Welt gibt es Persönlichkeiten, die sehr durchsetzungsstark sind und dann auch ihre Meinung besonders stark einbringen (vgl. T1.49 auf S. 188). Dies ist nicht unbedingt schlecht, vor allem wenn es darum geht, bestimmte Prinzipien zu vertreten und einer fehlenden Disziplin im Team vorzubeugen. Solche Personen lassen dann keine Wiederworte gelten und halten damit die Disziplin hoch. Ein Problem kann dies allerdings werden, wenn es darum geht, einen Weg zu finden (vgl. Situation in T1.47 auf S. 187) wie etwas konkret gestaltet werden soll und dann zwei Dickköpfe aufeinander prallen. Dann kann es sein, dass diese Reibereien zu verhärteten Fronten führen, vor allem, wenn wie im obigen Beispiel verschiedene Hintergründe der Personen und verschiedene Prinzipien gegeneinander kollidieren [1]. Gerade bei Softwarearchitektur kann dies ein Hemmnis für das Team werden, da hier Kompromisse immer zu einer unklaren und undeutlichen Architektur führen.

Schwierige Persönlichkeiten

Nicht immer lassen sich Probleme an einem bestimmten Persönlichkeitsmerkmal fest machen. Es gibt Personen, die im Umgang zum Teil wechselhaft, ja sogar launisch sind und die an einem schlechten Tag einen erheblichen negativen Einfluss auf das Team haben können. An einem schlechten Tag kann es passieren, dass ein zunächst unüberwindbarer Konflikt entsteht. In T3.12 auf S. 228 wird eine solche Situation beschrieben, wo eine wichtige Person aus dem Team gegen das Team bzw. den Zweck eine Scrum-Meetings gearbeitet hat. Dies lag im Anschluss wie ein „schwarzer Block“ (T3.14 auf S. 228), der das Team über einen längeren Zeitraum in seiner Kommunikation und Arbeit hemmte. Dies ist dann nur durch weitere Interventionen

[1] Anm.: Dem Verfasser ist die Situation aus T1.47 auf S. 187 persönlich bekannt, von daher ist dies die weitere subjektive Einschätzung des Verfassers.

wieder zu beheben. Hier gilt es, als Scrum-Master möglichst schnell auf solche Situationen zu reagieren und zur Not das Meeting auszusetzen, bevor sich die Situation weiter verschlimmert. Zu retten ist in dieser Situation ohnehin nichts mehr, ein produktives Arbeiten ist in einem gespannten Umfeld nicht möglich. Es ist sinnvoller vorher die Hintergründe zu klären und dann wieder in die gemeinsame Teamarbeit einzusteigen. Hier ist durch den Scrum-Master viel Erfahrung und Fingerspitzengefühl notwendig, damit er nur eingreift, wenn unbedingt notwendig, aber die Situation zuverlässig erkennt. Dabei hilft es ihm sehr, wenn er die Personen in seinem Team gut einschätzen kann.

Persönliche Einstellung zum Thema Verantwortungsübernahme Die gewonnene Freiheit bei der Arbeit in einem agilen Team gefällt nicht jedem Entwickler gleich gut. Immerhin heißt es, dass in einem agilen Team nicht der Chef bestimmt, was zu tun ist, sondern das Team selbst. Es übernimmt die informelle Führung für seinen Teilbereich (vgl. T2.34 auf S. 207). Vielfach haben sich (besonders ältere) Entwickler daran gewöhnt, dass es bequem und angenehm ist, wenn jemand bestimmt und ihnen vorgibt, was sie zu machen haben: „Die fühlen sich viel wohler, wenn ihnen jemand sagt, was sie zu tun haben und sie sich dann darauf zurückziehen können: „Ich hab ja nur das gemacht, was mir gesagt wurde."" (T2.30 auf S. 206). In der IT ist oftmals das Gegenteil der Fall und es ist eher folgende Situation vorzufinden: „Es gibt die Menschen im Team, die total begeistert sind an dem agilen Vorgehen und das auch für sich mit entdecken und lieben und auch das total für sich entdeckt haben, das verstehen und auch Freude dran haben (…) ." (T3.11 auf S. 227). Die Einstellung der Personen zum Thema Verantwortungsübernahme und Arbeiten im agilen Team sind meist zwischen diesen beiden Extremen zu finden (vgl. T2.31 auf S. 207). Bis zu einem gewissen Grad lassen sich negative Einstellungen durch das Team ausgleichen, jedoch ist darauf zu achten, dass es ein Team eher lähmt, wenn die Persönlichkeiten überhand nehmen, die sich mit den agilen Mind-Set schwer tun (vgl. T2.32 auf S. 207). Hierauf ist besonders zu achten, wenn neue Leute in das Team aufgenommen werden oder bei einer Neuzusammenstellung des Teams (s.a. Kapitel 6.2.1: „Zusammensetzung des Teams").

Collective Code Ownership Für viele aus der klassischen Entwicklungswelt, aber auch für die Genies unter den Entwicklern ist es oftmals schwierig, „ihren" Code abzugeben, so dass ein anderer „ihr Baby" auch bearbeiten darf.

Im Agilen wird groß geschrieben, dass jeder eine bestimmte Stelle bearbeiten kann, weswegen hier das Prinzip der Collective Code Ownerships genutzt wird (vgl. T1.15 auf S. 170 und (T4.9 auf S. 252). Diese Art von Teilen ist nicht für jeden einfach, denn dies erfordert, dass sich jemand immer wieder auch an neue Situationen anpasst (wenn sich z.B. der Code ändert) und auch seine Konzepte überarbeitet werden. Hier kann es durchaus zu Reibereien im Team kommen, die letztendlich auch dazu führen können, dass einzelne Inseln im Team entstehen, die sich dann doch wieder einen Teil aneignen und dort die Experten sind. Dies führt zwangsläufig zu einem Problem, da dies zum einen das sogenannte Refactorings erschwert und gleichzeitig einen Flaschenhals erzeugt.

Machtmotivierte Mitarbeiter Es gibt verschiedene Motive, warum ein Mensch Arbeit auf nimmt. Einige dieser Motive können jedoch auf das Team einen negativen Einfluss haben. Sehr stark machtorientierte Menschen, welche vor allem die Macht nutzen, um sich einen eigenen Vorteil zu verschaffen, können ein Team in seiner Zusammenarbeit stören (vgl. T2.46 auf S. 210). Für solche Personen ist das Team nur ein Mittel zum Zweck, um sich selbst einen Vorteil zu verschaffen. Sie werden dann nicht im Sinne des Teams handeln, sondern werden versucht sein, das Team so zu manipulieren, dass es ihren eigenen Zielen dient.

Persönlichkeitsstörungen Im Interview kam die Persönlichkeitsstörung Narzissmus zur Sprache (vgl. T2.47 auf S. 211). Bei dieser Persönlichkeitsstörung fehlt demjenigen eine aus sich kommendes Selbstwertgefühl, so dass dies von Außen kommen muss. Hier wird das Team zu einem Instrument, um dieser Person die Wertschätzung und ein positives Feedback zukommen zu lassen. Dies geht natürlich zu Lasten des Teams an sich, da der eigentlich Zweck des Teams verfehlt wird. Die Person wird die Teammitglieder für seinen Zweck manipulieren.

Wissen über Methoden der agilen Entwicklung

Agile Vorgehensweisen erfordern ein großes Maß an methodischem Wissen, Handwerkszeug, das beherrscht werden muss. Nur durch dieses Handwerkszeug ist es überhaupt möglich, eine immer wiederkehrende und reproduzierbare Software mit einem hohen Qualitätsstandard zu entwickeln. Dieses Handwerkszeug wird oftmals als gegeben vorausgesetzt, jedoch zeigt sich,

dass dies auch ein wichtiger Erfolgsfaktor sein kann (vgl. T1.13 auf S. 169). Gerade wenn das Team oder das Unternehmen aus einer klassichen Projektstruktur mit vielen Prozessen kommt, sind diese nicht immer bekannt und es muss erst erlernt werden, „(…) Wasserfallmodell, (…) diese Techniken eben nicht erfordert, weil man es halt durch Prozessmittel anders löst." (T1.14 auf S. 169). Es muss dann dazu gelernt werden. Dies sind jedoch nicht unbedingt universell einsetzbare Mittel, sondern durch die unterschiedlichen technischen oder organisatorischen Randbedingungen müssen die Methoden je nach Situation anders angewandt werden oder es kann auch sein, das ein Einsatz nicht möglich ist.

Ein Beispiel, was höchst unterschiedlich sein kann, ist das sogenannte Story-Schneiden, welches der Product-Owner durchführen muss. Da jedes Projekt andere Stories, andere technische Mittel enthält, müssen die Stories unterschiedlich geschnitten werden, was zunächst gelernt werden muss (vgl. T4.1 auf S. 248). Sicherlich, Erfahrung ist hier wichtig, aber auch dies ist ein Lernprozess, dem gerade zu Beginn eines Projekts oder nach dem Umstieg auf agile Entwicklung ein besondere Bedeutung beigemessen werden muss (vgl. T3.21 auf S. 231 und T4.2 auf S. 248).

Manche Werkzeuge der agilen Entwicklung haben ein hohes Potential, bei einer falschen Einstellung dazu, einen Konflikt heraufzubeschwören. In agilen Teams wird vermehrt auf die Technik des Code-Reviews gesetzt, um den Code weiter abzusichern. Das kann bei einem falschen Verständnis des Feedbacks als persönliche Kritik schnell in einem handfesten Konflikt im Team enden (vgl. T4.10 auf S. 252).

Zusammensetzung des Teams

Die Zusammensetzung eines agilen Teams kann den Erfolg des Teams beeinflussen. Dabei spielt es vor allem eine große Rolle, dass die agilen Werte des Teams überhaupt gelebt werden können und dass auch die Bedürfnisse der einzelnen Teammitglieder erfüllt werden können. Die Interviewpartner haben dazu einige Aussagen getroffen:

- „Ich brauche im Agilen einigermaßen homogene Teams." (T2.35 auf S. 207) war eine wichtige Aussage, die hier besonders im Bezug auf das Wissenslevel gemeint war. So wird es einen erfahrenen Entwickler demotivieren, wenn er nur mit Anfängern zusammen arbeitet und den

Anfängern wird er unter Umständen keine Chance gegeben, sich in das Team einzufügen.

- Das Vorhandensein von bestimmten Skills spielt aber ebenso eine Rolle (vgl. T4.6 auf S. 250). Angesprochen wurden hier vor allem Skills, die Requirementserhebung als auch den Test betreffen (s.a. Kapitel 6.2.1: „Wissen über Methoden der agilen Entwicklung").
- Persönliche Abneigungen im Team können die Zusammenarbeit erschweren (vgl. T1.20 auf S. 171).

Das sind jedoch keine Ausschlusskriterien, denn bei entsprechenden Persönlichkeiten der einzelnen Teammitglieder kann eine Zusammensetzung funktionieren, weil ein Senior-Entwickler z.B. sehr gerne Wissen weitergibt, bei einem anderen Entwickler, der seine Erfüllung im Schreiben von Quelltext findet, kann es dann aber schwierig werden, wenn er sich von den anderen isoliert, weil er keine Lust auf Diskussionen hat. Eine homogene Zusammensetzung ist zwar ein Erfolgsfaktor, jedoch kann es auch nicht oder trotzdem funktionieren. In der Regel sind meist die optimalen Personen nicht verfügbar (vgl. T4.33 auf S. 263) und es müssen andere Wege gefunden werden, wie z.B. das Team zu entwickeln.

Teamspirit

Agile Teams sind geprägt von einer sehr engen Zusammenarbeit, Vertrauen und gegenseitigem Respekt: „Agile Teams zeichnen sich vor allem dadurch aus, dass sie nicht nur ein: „Wir arbeiten irgendwie miteinander." sind, sondern meistens darüber hinaus die anderen Mitarbeiter als Menschen sehr stark respektieren schätzen (…)" (T2.36 auf S. 207). Eine enge Beziehung der Teammtiglieder untereinander schafft Vertrauen, so dass Teammitglieder auch unangenehme Dinge ansprechen (vgl. T3.45 auf S. 239). Nur mit diesen Grundwerten ist es möglich, eine offene Kommunikation und Feedbackkultur zu pflegen. Diese Eigenschaften haben eine Reihe von Vorteilen in der Arbeit von Teams, wie die Interviewpartner hervorhoben.

Konfliktlösung und -vorbeugung Es vereinfacht die Lösung von Konflikten, da es möglich ist, die emotionale Ebene in Konflikten fast vollständig auszuschalten und auf der fachlichen Ebene die Lösung zu suchen. Durch den sehr starken Zusammenhalt ist die emotionale Ebene soweit geklärt, dass eine heftige Diskussion nicht in einem Konflikt mündet. Es ist klar, das

ein gegenseitiger Respekt vorhanden ist, der durch die aktuelle Diskussion nicht infrage gestellt wird (vgl. T2.39 auf S. 208). Es wird für die Beteiligten möglich, sofort in eine starke fachliche Diskussion einzusteigen, ohne dass Diplomatie oder verletze Gefühle eine Rolle spielen. Es ist dann sogar möglich, die Arbeit des anderen in einer Art und Weise zu kritisieren, die sonst unmöglich wäre und als Affront gewertet werden würde.

Misserfolge Die enge Bindung der Teammitglieder untereinander führt auch dazu, dass Fehler von einzelnen oder Misserfolge einzelner oder des Teams einfacher und ohne negative Auswirkungen auf das Team weggesteckt werden können. Es ist für das Team einfacher, diesen Misserfolg einzustecken und gemeinsam zu beheben (vgl. T2.38 auf S. 208).

Feedback geben Nur in einem vertrauten Umfeld ist es möglich, Probleme offen anzusprechen und Feedback zu geben. Wenn Feedback gegen eine Person ausgenutzt wird, dann wird dies ihr letztes ehrliches Feedback gewesen sein. Daher ist eine vertrauensvolle Basis notwendig, damit Feedback offen gegeben werden kann (vgl. T2.60 auf S. 216).

Gleichgültiges Verhalten In schlimmeren Fällen, wenn der Teamspirit nicht mehr passt, kann es zu schädlichem Verhalten einzelner Mitglieder kommen. Dabei verhalten sie sich „(…) passiv aggressiv (…).“ (T2.40 auf S. 208), womit sie sich zwar nicht an einer aktiven Störung beteiligen, jedoch auch nichts dagegen unternehmen werden, wenn sie Probleme oder Missstände erkennen. Sie freut es dann schon fast, dabei zuzusehen, wie das Projekt oder das Team in Schieflage gerät. Dies kann in einem agilen Kontext sehr negative Auswirkungen haben, da hierdurch die Mechanismen zur Verbesserung außer Kraft gesetzt werden. Das Team arbeitet dann zwar noch, aber die gewünschte Agilität, die Anpassung an neue Umgebungen findet nicht mehr im vollen Umfang statt. Das gleiche Verhalten ist vorzufinden, wenn Teammitglieder gleichgültig gegenüber Problemen geworden sind, weil das Verhältnis zwischen Teammitgliedern oder zum Vorgesetzten gestört wurde: „Wenn das Verhältnis nicht mehr funktioniert, dann sage ich es nicht mehr, sondern dann ist es mir einfach wurscht (…) .“ (T2.41 auf S. 209).

Kommunikation

Bereits im Kapitel 6.2.1 „Raum und Zeit“ war das Thema Kommunikation der Teammitglieder ein zentrales Thema. Hier war der Aspekt vor allem von äußeren Einflüssen geprägt, jedoch ist dies nicht immer so. Grundsätzlich müssen zwei Arten von Kommunikation unterschieden werden:

- „fachliche Ebene“ (T1.4 auf S. 164)
- „persönliche Ebene“ (T1.5 auf S. 164)

Die fachliche Kommunikation dient hauptsächlich dem Wissensaustausch im Team. Diese wird durch Scrum aktiv in Form des Stand-Ups gefördert (vgl. T1.22 auf S. 171), jedoch findet auch sehr viel Kommunikation abseits davon statt. Hier kann es passieren, dass durch schlechte Kommunikation Wissenslücken entstehen und wichtige Informationen nicht weiter verbreitet werden (vgl. T3.7 auf S. 226). Manchmal tritt dieser Effekt auch auf, da davon aus gegangen wird, dass das Gegenüber die Information bereits besitzt und daher nicht mehr erwähnt werden muss. Die fachliche Kommunikation bekommen auch Teams hin, deren Mitglieder sich nicht so eng verbunden fühlen (vgl. T1.18 auf S. 170). Gerade bei der fachlichen Kommunikation ist es einfach, diese anzustoßen bzw. zu intensivieren: „(…) richtet Dinge wie regelmäßig Meetings ein, oder man stellt jemanden ab, dass Leute miteinander reden oder man bringt den Leuten bei: Schreibt bitte eine Mail, wenn irgendwas ist.“ (T1.17 auf S. 170). Ein solches regelmäßiges Meeting ist schon das Stand-Up, jedoch kann es Sinn machen, für spezielle fachliche Themen noch einen kleinen Arbeitskreis zu bilden, der sich regelmäßig trifft und dort Wissen austauscht.

Die Kommunikation auf der persönlichen Ebene wurde vom Interviewpartner auch als Metaebene bezeichnet (vgl. T1.19 auf S. 170). Hierbei geht es um Informationsaustausch abseits der fachlichen Information und sie dient dem sozialen Miteinander und dafür, dass das Team enger zusammen wächst. Störungen hier könne vor allem auftreten, wenn Personen nicht viel reden und Introvertiert sind oder wenn sie eine Abneigung zueinander haben (vgl. T1.20 auf S. 171). Gerade wenn eine persönliche Abneigung zueinander vorhanden ist, wird es schwierig, eng zusammenzuarbeiten und Methoden wie Pair-Programmings können dann nicht so effektiv sein (vgl. T1.21 auf S. 171).

Rollen im Scrum

Auch wenn in den Interviews viel über Rollen geredet wurde, so spielten sie nur eine untergeordnete Rolle bei der Nennung von negativen Entwicklungen oder der Diagnose. Zwar wird viel Augenmerk auf die Rolle des Scrum-Masters und des Product-Owners gelegt, abseits hiervon kamen andere Rollen nur wenig vor. Einer der Interviewpartner ließ sich über die Gestaltung und Aussage von Scrum über die Rollen aus und stellte fest: „(…) Struktur mit Rollen und Tätigkeiten und das geht natürlich weit über die Scrum-Rollen hinaus, diese sind ja nur sehr minimalistisch (…).“ (T1.24 auf S. 173). Damit wird klar, dass eine Scrum-Implementierung im Unternehmen deutlich mehr bedarf als das Ausfüllen der Rollen des Scrum-Masters, Product-Owner und der Entwickler. Vielmehr ist es wichtig, dass die notwendigen Rollen für den spezifischen Anwendungsfall erkannt und besetzt werden. Explizit genannt wurden Rollen wie Bedenkenträger (vgl. T4.31 auf S. 262) und Bewahrer (vgl. T4.32 auf S. 262), welche für ein Team wichtig sein können. Sie ermöglichen es dem Team Disziplin zu halten und Entscheidungen zu hinterfragen bzw. darüber zu diskutieren bevor Entscheidungen getroffen werden.

Im Sinne von Scrum erfolgt dies in einem evolutionären Prozess, so dass sich notwendige Rollen herausbilden und nicht eingeteilt werden (vgl. T4.30 auf S. 262). Letztendlich ist es nicht wichtig, wer eine bestimmte Rolle ausfüllt, sondern nur, dass sie vorhanden ist und gelebt wird. Es entsteht eine Organisation, die folgendes leistet: „Die Organisation definiert ja nicht die Personen, sie definiert ihre Zuordnung.“ (T1.25 auf S. 174).

Ungeklärte oder nicht wahrgenommene Rollen Unklare Rollen (vgl. T1.7 auf S. 166) können zu vermehrten Reibungsverlusten führen. In der Praxis kommen solche Probleme dann vor, wenn sich Zuständigkeiten oder Verantwortungen infolge von organisatorischen Umstrukturierungen ändern. Ebenso kann dies der Fall sein, wenn sich Wissensträger ändern, weil z.B. ein neues Gebiet bearbeitet wird und dann die alten Wissensträgerrollen nicht mehr stimmen. Besonders prägnant wird dies, wenn essentielle Rollen im Scrum nicht klar sind oder sich durch eine Änderung der alte und neue, inoffizielle Rollenträger sich das nicht bewusst machten. Auch haben Entwickler vielfach mehrere Ansprechpartner in einem komplexen Unternehmen und dann kann es vor kommen, dass wie im Beispiel, „nicht so richtig klar war, wer da jetzt das Sagen hat“ (T1.6 auf S. 166) oder ein Rollenträger sich seiner

Verantwortung nicht bewusst ist: Eine Produktmanagerin sich selbst nicht als Product-Owner bezeichnet, weil kein Wissen über die Existenz dieser Rolle vorhanden ist (vgl. T1.9 auf S. 167). Wenn solch ein Zustand bemerkt wird, ist es notwendig das Rollenverständnis zu schulen, oder aber sich klar zu machen, wer welche Rolle besetzt (vgl. T1.8 auf S. 167).

Notwendige Veränderungen

Agile Methoden werden eingeführt, enger am Markt sein zu können und sich an eine verändernde Umwelt schneller anpassen zu können. Erreicht wird dies, indem Teams, welche dicht am Markt sind, relevante Entscheidungen selbst treffen können. Jedoch kann dieser Mechanismus auch versagen oder nicht mehr intensiv gelebt werden.

Veränderte Umwelt Es gibt Fälle, wo sich das Marktumfeld geändert hat und bei denen das Versprechen zu schnellen Änderung bei agilen Methoden nicht automatisch eingelöst wurde, wie folgende Textpassage anregt: „Nach einer gewissen Zeit nimmt das aber ab und ein lang zusammen arbeitendes Team, haben sich halt gewisse Methodiken, Interaktionen, die haben sich einfach eingeschliffen, die sind selbstverständlich, die werden auch gar nicht mehr diskutiert, nicht mehr hinterfragt, (…) Was aber dadurch passieren kann, ist das man disfunktionale Verhaltensweisen hat, manche die sogar in der Vergangenheit sinnvoll waren, jetzt aber in einem geänderten Umfeld nicht mehr sinnvoll sind, (…) und das gar nicht mehr erkennen.“ (T2.63 auf S. 218).

Was ist in einem solchen Team passiert, dass es die Eigenschaft verloren hat, sich an ein neues Umfeld anzupassen?

Routine in der Arbeit ermöglicht es, kognitive Last zu senken; Normen und Standards im Team helfen den Teammitgliedern bei der Orientierung im Team. Sie sind der Schlüssel zu einem Team, das in der Performing-Phase angekommen ist. Routine und Standards werden schnell zu einer Selbstverständlichkeit und ein Hinterfragen findet immer weniger statt, je öfters diese erfolgreich waren. Zu Problemen kann dies führen, wenn sich die Umwelt so verändert, dass diese Routinen und Standards nicht mehr sinnvoll sind und eigentlich geändert werden müssen. Dann liegt es am Team, dies zu erkennen und dem entgegenzusteuern.

Veränderungsresistenz Ein typische Herausforderung bei Veränderung ist die Überwindung von Resistenz gegenüber Änderungen. Dies kann so weit gehen, dass Probleme ignoriert werden: „(…) Leute haben massive Probleme aber sie wollen das gar nicht sehen." (T2.64 auf S. 219); sie verschließen sprichwörtlich die Augen vor den Problemen, da ihnen z.B. das aktuelle Umfeld vertraut ist und sie Ängste vor den anstehenden Veränderungen haben. Dann kann es passieren, dass Probleme einfach ignoriert oder sogar als gegeben hingenommen werden.

Agile Vorgehensweisen sind keine Garantie Agiles Vorgehen ist nicht per se erfolgreicher beim Anstoßen von Veränderungen. Im Scrum gibt es zwar viele eingebaute Schutzmechanismen, wie das Konzept von Inspect & Adapt, die potentielle Quellen für die Initiierung von Veränderungen sind, jedoch ist es keineswegs ein Garant für Erfolg. Hier empfiehlt es sich von Zeit zu Zeit unabhängige Personen einzubeziehen, die eine Art Audit durchführen und dem Team neuen Input, neues Feedback (s.a. Kapitel 6.2.1: „Feedback") geben. Dabei hilft der unvoreingenommene Blick von Außen genauso wie es auch die andere Formulierung von Problemen tun kann.

Disziplin im Scrum-Alltag

Ein typisches Problem von Teams ist die Disziplin im Alltag. Gerade Teams, die schon längere Zeit erfolgreich zusammenarbeiten (vgl. T3.3 auf S. 226) bekommen das Gefühl, dass sich manches als Ballast anfühlt oder dass es ja gut läuft und dass ein oder andere nicht mehr so ernst gesehen werden muss. Es schleifen sich dann kleine Unachtsamkeiten ein, die immer größer werden und im schlimmsten Fall dazu führen, dass das Team nicht mehr seinen Anforderungen gerecht werden kann (vgl. T3.2 auf S. 226 und T3.30 auf S. 235).

In manch einem Team lässt die Disziplin nach (vgl. T3.2 auf S. 226 und T3.30 auf S. 235), so dass z.B. der Daily Scrum bzw. das Stand-Up, wie es oft genannt wird, weniger intensiv gelebt und gerne vernachlässigt wird (vgl. T3.5 auf S. 226). Die Ursachen dafür sind nicht immer eindeutig, so wurde genannt, dass es sich ein Problem von Teams handelt, die länger zusammenarbeiten (wie beim Interviewten mit fünf Jahren (vgl. T3.3 auf S. 226)) oder dass der Kommunikationskanal nach Meinung der Teammitglieder nicht benötigt wird, da sie auch so genug kommunizieren können (vgl.)T3.6 auf S. 226). Auch Situationen, in denen auf das Team externer Druck ausgeübt

wird und das Team damit in Stresssituationen kommt, führen schnell zu einer Verschiebung der Priorität und die scheinbar unwichtigen Elemente werden über Board geworfen (vgl. T3.26 auf S. 233). Dies wird auch durch folgende Aussage untermauert: „(…) auch dass man gar nicht die Zeit hat, sich um Organisationsthemen zu kümmern, weil man viel zu sehr damit beschäftigt ist, Bugs zu fixen oder so was in der Richtung (…)“ (T1.30 auf S. 177).

„Nee, Retrospektive haben wir mal am Anfang gemacht und das finden manche ein bisschen lächerlich.“ (T4.19 auf S. 259), solche oder ähnliche Aussagen zeugen von einem weiteren Problem: Das Team sieht keinen Sinn in diesem Element und stellt es daher ein. Dies kann passieren, wenn das Meeting z.B. nicht den gestellten Erwartungen gerecht wird. Im Beispiel der Retrospektive kann dies z.B. sein, dass sich doch nichts ändert oder dass sich das Team dabei nicht wohl fühlt.

Einige typische Anzeichen für fehlende Disziplin bei der Anwendung von Scrum sind:

- Das Stand-Up wird nicht mehr als so wichtig erachtet, das Team geht lieber der fachlichen Arbeit nach.
- Die Retrospektive wird nicht mehr intensiv gelebt oder fällt ganz aus (vgl. T4.20 auf S. 259).

Als Scrum-Master ist es wichtig, dass diese Situationen erkannt werden, um Gegenmaßnahmen einzuleiten. Bewährt hat sich hier, dem Team neue Impulse zu geben (vgl. T3.4 auf S. 226) und dem Team noch einmal die Wichtigkeit der Artefakte aufzuzeigen.

Zusammenfassung und Bedeutung für die Forschungsfrage

Aus den Interviews mit den Experten sind viele bekannte und neue Erkenntnisse entstanden, die sich mit den bisherigen Modellen nicht erkennen ließen. Einige davon sind sehr spezifisch für agile Teams und spiegeln wichtige Voraussetzungen wider, die benötigt werden, um agile Methoden in Unternehmen zu etablieren. Die Praxis zeigt, das sich im Laufe der Zeit Voraussetzungen und Rahmenbedingungen ändern können. So kann ein neues Management mit einem anderen Führungsstil die Arbeit mit agilen Teams wieder zurückwerfen oder verbessern.

Unsere Forschungsfrage stellt sich die Frage zur Identifikation von negativen Entwicklungen. Hier konnten wir die zwei Gruppen herausarbeiten:

Der Kontext-Faktoren und die Teamfaktoren. Diese Unterteilung ist jedoch nicht neu, sondern wird bereits in der bestehenden Literatur verwendet.

In der Literatur sind meist sehr viele negative Entwicklungen zu finden, der Vorteil aus der praktischen Befragung ist jedoch, dass sich dabei eine Relevanz abzeichnet. Da unsere Experten allesamt aus der Praxis sprechen, haben sie durch ihre Aussagen, die für sie im Alltag relevanten Entwicklungen skizziert. Eine kurze Zusammenfassung der Themenfelder ergibt:

Raum und Zeit Das Team sollte möglichst viel gemeinsam Zeit an einem gemeinsamen Arbeitsort verbringen. Dies fördert die Kommunikation und beugt Problemen vor. Wenn sich hier Veränderungen ergeben hat dies meist einen negativen Einfluss auf das Team.

Feedback Es ist wichtig, gutes Feedback zu bekommen und zu kreieren. Wenn dies ausbleibt, wird sich dies negativ auf das Inspect & Adapt Konzept auswirken und das Team wird anfälliger, Veränderungsnotwendigkeiten zu erkennen. Der Prozess des Sich-Selbst-Verbesserns kommt ins Stocken, was sich wiederum negativ auf die gewünscht Anpassungsfähigkeit auswirkt.

Autonomie Das Team muss eigenständig arbeiten können, Entscheidungen fällen und deren Umsetzung ermächtigt sein. Wenn diese ihm entzogen wird, auch wenn dies nur vom Team so wahrgenommen wird, wird das Team die Verantwortung für seinen Arbeitsbereich abgeben und die Agilität wird darunter leiden.

Alignment Um sinnvolle Entscheidungen treffen zu können, muss das Team wissen, in welche Richtung es arbeiten muss. Wird dies nicht ausreichend im Team verankert, so werden Entscheidungen gefällt, die zwar aus Sicht des Teams die richtigen sind, aber die z.B. nicht zur strategischen Ausrichtung des Unternehmens passen, was sich wiederum negativ auf das Gesamtunternehmen auswirkt.

Persönlichkeiten Der Faktor Mensch ist bei Teamarbeit immer wichtig und wurde durch die Interviewpartner auch bestätigt. Dabei ist dies ein großes Gebiet und umfasst z.B. die persönliche Einstellung zur agilen Arbeitsweise, persönliche Lebenslagen und Bedürfnisse, Motivation für die Arbeit und problematische Persönlichkeiten. Durch die Interviewten wurde dargestellt, dass es sich hier immer wieder um Konfliktpotentiale handelt, die sich dann negativ auf das Team als Ganzes auswirken.

Skills und Wissen Durch die sehr selbständige Arbeitsweise von agilen Teams, müssen diese über spezielles Wissen und Skills verfügen. Wenn diese Skills im Team nicht gelernt und angewendet werden können, so ist niemand da, der dies erkennt und einbringen kann. Dies wiederum macht es dem Team schwierig State-of-the-Art zu arbeiten. Dies kann sich dann negativ auf die Qualität wie auch den agilen Prozess auswirken.

Teamzusammensetzung Wie ein Team aufgebaut ist, ist besonders wichtig. Agile Teams arbeiten eng zusammen, benötigen eine offene und kreative Atmosphäre sowie eine gute Portion Skills. Wenn dies bei der Zusammensetzung nicht berücksichtigt wird, hemmt dies die Zusammenarbeit, Feedback kann erschwert werden.

Teamspirit Eine positive Stimmung im Team kann manchen Konflikt gar nicht erst aufkommen lassen. Es kann vor allem dafür sorgen, das fachliche Gräben einfach und schnell ausdiskutiert werden und eine Lösung gefunden wird.

Kommunikation sie ist die Basis für die Teamarbeit und in einem Hochleistungsteam unverzichtbar. Es muss dafür gesorgt werden, dass diese immer fachlich präzise und menschlich nah ist.

Rollen Sie sind zu definieren oder sie müssen den Beteiligten klar sein. Es muss ersichtlich sein, wer für was im Team zuständig ist und es muss dabei gesichert sein, dass nicht eine Rolle den „schwarzen Peter“ hat, nur weil die z.B. immer dagegen ist. Konträre Standpunkte aus den Rollen heraus sind ebenso wichtig wie Harmonie bei Entscheidungen. Negative Entwicklungen ergeben sich hier vor allem bei Umgestaltungen, Neuzusammensetzungen oder Veränderungen im Team oder der Arbeitsweise.

Veränderungen Sie sind das Wichtigste am agilen Entwickeln. Durch sie wird es möglich, dem Marktgeschehen in der erforderlichen Geschwindigkeit zu folgen. Eine negative Entwicklung hier kann für das Unternehmen existenzgefährdend sein. Es muss sichergestellt werden, dass das Team am Zahn der Zeit bleibt und die Bewegungen am Markt aufnimmt und zum Vorteil des Unternehmens ausbaut.

Disziplin Es ist wichtig, Festlegungen einzuhalten, beschlossenes Umzusetzen und die wichtigen Elemente von agiler Entwicklung mit Leben zu

füllen. Hier sind in der Praxis immer wieder negative Entwicklungen zu verzeichnen, da es scheinbar bequemer ist.

Es gibt viele negative Entwicklungen, die sich anhand der Themenblöcke ergeben haben. Die Manifestation der eigentlichen negativen Entwicklung kann dabei sehr unterschiedlich sein. Ob so eine negative Entwicklung letztendlich dem Unternehmen schadet, ist nicht sicher. Es ist allerdings gesichert, das die Wahrscheinlichkeit für einen Misserfolg steigt. Viele dieser Entwicklungen lassen sich anhand ihrer Auswirkungen direkt beobachten, so kann die fehlende Disziplin im Alltag direkt beobachtet werden. Damit lassen sie sich in der Praxis, wenn man sich diesem bewusst ist, einfach beobachten. Es ist dabei ein tiefer Einblick in das Team erforderlich, die Führungskraft muss sich mit dem Team und seiner Arbeit intensiv beschäftigen.

6.2.2 Erkennen von Konfliktherden und Problemen in agilen Teams

Im vorangegangenen Kapitel wurden Faktoren für erfolgreiche agile Teams und negative Entwicklungen vorgestellt. Zum Teil können diese Themen direkt beobachtet werden, zum Teil sind jedoch weitere Mittel für eine Diagnose notwendig. Eine Vielzahl der Mittel sind ebenso einfach, wie logisch und werden einige sicher überraschen. Meist bedarf es nicht einer komplizierten Diagnose, wie es bei Teams üblich ist, sondern die Mittel sind viel banaler und einfacher. Eine wichtige Grundeigenschaft des Scrum-Masters oder einer Führungskraft ist dessen Beobachtungsgabe und seiner Beziehung zum Team. Gerade eine gute Beziehung, die vor allem durch Vertrauen geprägt ist, kann wesentlich wertvoller sein, als so mancher ausgeklügelter, psychologischer Fragebogen oder externe Coach. Diagnose heißt auch immer, eine Führungskraft muss sich mit seinem Team beschäftigen. Wie ich aus leidvoller Erfahrung aus dem Alltag weiß: Ein Faktor, der gerne vergessen wird.

Bei Konflikten kommen die Leute oftmals auf den Coach / Scrum-Master zu.

Den Einen oder Anderen wird dieses Mittel überraschen, dabei ist es die einfachste Methode, die es überhaupt gibt, frei nach dem agilen Gedanken,

der Mensch steht im Mittelpunkt: Höre den Teammitgliedern zu. Alle erfahrenen Scrum-Master haben dies meist als besonders wichtigen Punkt genannt (vgl. auch T4.29 auf S. 261, T3.44 auf S. 239), stellvertretend sei hier einer zitiert: „Der häufigste Fall ist es, die Leute sagen es einem." (T2.56 auf S. 215).

Bei einem besonderen Verhältnis zum Team, werden auch die Meetings im Team genutzt: „Hat mit der Teamleitung zu tun und mit dem Vertrauen zueinander (…) haben zum Stand-Up ganz offen und ehrlich gesagt oder sind auf einen zu gekommen." (T3.46 auf S. 239). Damit Teammitglieder diesen Weg wählen, ist ein hohes Maß an Vertrauen zum Team oder dem Scrum-Master bzw. der Führungskraft vonnöten. Die Leute benötigen Vertrauen, dass ihre Äußerung nicht gegen sie verwendet wird und dass sie auch vertraulich behandelt wird, wenn dies notwendig ist.

Dies benötigt eine gute persönliche Beziehung zwischen beiden Parteien, wie der interviewte Coach durch seine Aussage untermauert: „(…) darauf basieren alle meine erfolgreichen Projekte, nicht alle meine Projekte sind erfolgreich, aber die, die erfolgreich sind, basieren immer darauf, dass ich zum Kunden eine sehr gute persönliche Beziehung aufgebaut habe." (T2.57 auf S. 215) Wiederum andere sprechen von einer Feedbackkultur, die etabliert werden sollte, um dies zu fördern: „(…) und wenn man eine Kultur etabliert hat, in der man sich auch offen Feedback gibt, wird das auch normalerweise angesprochen. Ich geh davon aus, das normalerweise Leute zu mir kommen und sagen, hier das und das passt mir nicht." (T4.29 auf S. 261).

Aktives Nachfragen

Wenn Leute nicht von sich aus auf jemanden zu kommen, kann es ganz hilfreich sein, wenn diese befragt werden. Dies kann in einem geplanten, förmlichen Rahmen statt finden, wie hier: „Ich führe normalerweise, weil es meistens so ist, das ich Teamleiter bin, individuelle Gespräche eh mit den Leuten (…)" (T4.28 auf S. 261), oder es findet in einem nicht förmlichen Rahmen statt: Es reicht auch oft, sich am Kaffeeautomaten oder in der Raucherecke aufzuhalten und dort Smalltalk zu führen. Ebenso kann es hilfreich sein, direkt zu fragen, wie es läuft. Dabei ist wichtig, dies nicht zu tun, weil es gelernt wurde, sondern aus eigenem Interesse. Nur dann wirkt dies auch authentisch und die Leute haben das Gefühl, verstanden zu werden. Hierfür ist ebenso wie beim direkten auf die Person zukommen aus dem vorherigen

Punkt ein hohes Maß an Vertrauen notwendig (vgl. T2.59 auf S. 216 und T2.61 auf S. 217).

Nutze die Artefakte von Scrum und Meetings mit Interaktion im Team zur Diagnose

Relativ einfach und unkompliziert lassen sich die Interaktion der Teammitglieder untereinander ansehen und beobachten. Hierfür gibt es vielfältige Möglichkeiten, denn als agiler Coach, Scrum-Master ist ein häufiges Treffen mit dem Team ohnehin obligatorisch. Im Vergleich zu einem normalen Projektmanager, Produkt-Manager oder Vorgesetzten ist die Arbeit mit dem Team besonders wichtig und es gibt viele gemeinsame Schnittpunkte in der Arbeit; im Scrum sind dies die Artefakte, wie Stand-Ups, Retrospektiven, Planungs-Meetings und die tägliche Arbeit (vgl. T4.34 auf S. 264 und T4.35 auf S. 264).

Anti-Pattern / Bad-Smell

Der Begriff „Anti-Pattern“ stammt ursprünglich aus der Softwarearchitektur. Dort gibt es die Bezeichnung „Design Pattern“ (deutsch: Entwurfsmuster), die eine Art von Schablone bezeichnet, wie bestimmte Typen von Problemen gelöst werden können. Schnell hat sich dort für schlechte Muster der Begriff „Anti-Pattern“ durchgesetzt. Nicht weit weg ist hier der Begriff „Bad-Smell“, der ebenso genutzt wird und eine Stelle im Quelltext bezeichnet, die einer Überarbeitung bedarf, da sie nicht gut aussieht und Probleme macht. Solche Muster lassen sich auch in der täglichen Arbeit mit Scrum finden und definieren (vgl. T1.2 auf S. 162, T1.40 auf S. 182 und T4.18 auf S. 259). Dinge, auf die man besonders achten sollte, sind zuhauf im Internet zu finden. So fand Google beim Schreiben dieser Arbeit am 10. Oktober 2016 für die Suchbegriffe „bad-smell scrum“ ca. 31700 Treffer und für „anti-pattern scrum“ ca. 84600 Treffer. In den Interviews wurden im Wesentlichen folgende Pattern angesprochen:

- Product-Owner-Proxy: Der eigentliche Product-Owner möchte die Aufgabe nicht übernehmen oder kann sie nicht übernehmen und ein anderer füllt diese Rolle aus, indem er sein Stellvertreter wird (vgl. T1.1 auf S. 162). Ein Stellvertreter vertritt nur jemanden, es geht Wissen verloren, das direkte Feedback fehlt.

- Langes Standup: Es gibt Teams, die ein Stand-Up falsch verstehen und ausführlichst und detailliert erzählen, was wo gemacht wurde. Solch ein Standup kann dann gerne statt den üblichen 10 Minuten bis zu einer Stunde am Tag andauern (vgl. T1.41 auf S. 182).
- Retrospektive weglassen: Die Retrospektive wird nicht durchgeführt (vgl. T4.18 auf S. 259). Eine Reflexion über die Arbeit findet nicht statt.

Beim Lesen der Pattern denken sich viele: Das kann mir nicht passieren! Aber manchmal ertappt man sich dann dabei zu sagen: Ja, das trifft auf uns zu, vielleicht ein bisschen. Die Pattern zu lesen und sich mit ihnen zu beschäftigen, schärft den Blick für eventuelle Probleme. Dabei wird auch gleich das eigene Team in diesem Sektor analysiert.

Persönliche Gefühle

Gefühle begleiten uns auch in der Arbeitswelt und helfen uns den Weg durchs Leben zu finden. Gefühle sind eine Art unbewusste Warnglocke, die uns helfen kann, für das Bewusstsein verdeckte Mängel zu identifizieren. Dabei können wir unsere eigenen Gefühle genauso benutzen, wie auch die der anderen Teammitglieder, wenn wir sie lesen können.

Der Softwareingenieur hat einen Ausdruck, im Englischen „Pain“, also Schmerz, den er nutzt, wenn ihm eine Arbeit besonders schwer fällt oder er vor Problemen steht, die eigentlich aus seiner Sicht nicht sein müssten, aber vorhanden sind. Er wird immer versucht sein, diesen „Pain“ abzustellen, indem er die Ursache beseitigt. Dieses: „Wenn es weh tut.“ (T1.35 auf S. 179) oder „Wenn man sich ärgert.“ (T1.34 auf S. 179) wird sehr häufig angetroffen, wenn Infrastruktur-Probleme vor liegen (vgl. T1.36 auf S. 179). Gerade bei der heutigen Komplexität und in einer agilen Vorgehensweise ist der Softwareentwickler auf mehr als einen einfachen Rechner angewiesen. Er bedient sich Methoden, die oftmals auf einen hohen Automatisierungsgrad von Aufgaben zurückgreifen, sofern sie vorhanden sind und reibungslos funktionieren. Es können aber auch andere Teile der Arbeit betroffen sein, wie die Qualität des Quellcodes oder die Zusammenarbeit mit anderen Teams. Dies alles kann ein Hemmnis für agile Entwicklung sein und zu Konflikten führen, da dann z.B. viel manuelle und fehlerträchtige Arbeit notwendig ist, was ein Team ausbremsen kann. Gute Teams arbeiten an diesen Problemen und versuchen diese Dinge zu beheben. Es kann aber auch ein Anzeichen sein, dass ein Team hierauf aufmerksam gemacht werden muss.

Weiterhin versuchen Führungskräfte, wie auch Scrum-Master die Gefühle anderer Leute zu lesen und zu deuten (vgl. T3.40 auf S. 238). Dabei achten sie auf zwischenmenschliche Beziehungen ebenso, wie auf die Einzelpersonen. In diesen Gefühlen können persönliche Probleme ebenso verborgen sein, wie eine allgemeine schlechte Stimmung im Team. Es ist daher gut zu wissen, woran das Team ist, und was diese Gefühle auslöst. Dann kann entsprechend gehandelt werden, denn starke Gefühle führen oftmals zu Emotionen im Miteinander und bergen damit die Gefahr einen Konfliktherd aufzutun.

Veränderung im Verhalten

Sehr eng mit dem Lesen der Gefühle ist auch das Beobachten von Verhaltensweisen der Teammitglieder verbunden. Vielfach ändern Menschen ihr Verhalten, wenn sie Probleme haben, sich Einstellung zur Arbeit ändert oder Demotivation breit macht. Dabei ist zu unterscheiden zwischen der Verhaltensänderung einer Einzelperson und den zwischenmenschlichen Beziehungen (vgl. T3.39 auf S. 238). Meist beginnt die Verhaltensänderung bei einer Einzelperson und beeinflusst dann auch die zwischenmenschliche Beziehung zu den anderen Teammitgliedern.

Warnsignale für geändertes Verhalten Offensichtliche Zeichen können nach den Interviewpartner sein:

- Das soziale Verhalten und die Umgangsformen oder das äußere Erscheinungsbild ändert sich: „Wenn ich merke, oh da ist gerade jemand unzufrieden, der grüßt nicht mehr richtig oder brummelt zu viel oder hat ziemlich viele Augenringe (…) .“ (T2.54 auf S. 215).
- Die Person wird unzuverlässig (vgl. T4.7 auf S. 251) und nimmt die Arbeit im Team weniger an: „Man kann Probleme bei bestimmten Personen, also jetzt nicht aufs ganze Team bezogen, bei bestimmten Personen sehen, wenn sie z.B. häufiger zu spät kommen zum Stand-Up oder wenn sie bei bestimmten Sachen nicht da sind oder einfach von sich aus gefühlt kein Commitment zeigen, das sieht man.“ (T3.42 auf S. 238)
- Die Person ziehen sich aus dem sozialen Kontext zurück (vgl. T4.26 auf S. 261) und beteiligen sich nicht mehr an sozialen Team-Ritualen (z.B. Pizza Essen, gemeinsames Kaffee trinken): „(…) wir haben bestimmte Rituale, die wir einfach machen und wenn sich jemand da raus

ziehen würde, ist das ganz offensichtlich, das er sich da nicht wohl fühlt (…)“ (T4.25 auf S. 261).

- Das Verhalten in der Kommunikation ändert sich, so dass sich jemand weniger oder anders als vorher äußert (vgl. T4.27 auf S. 261).
- Antriebslosigkeit im Team: „Wenn die Leute irgendwie unmotiviert sind, ein Meeting ist rum, sie könnten jetzt zur Arbeit gehen und sitzen immer noch da und "Naja, müssten wir eigentlich wieder was schaffen (Atmet tief ein und wieder aus) ... müsstmer erst mal ein Eis essen gehen.““ T2.58 auf S. 216

Dies alles können Warnsignale sein, die entweder eine temporäre Tiefphase des Mitarbeiters repräsentiert oder aber einen wichtigen Hintergrund haben. Beides sollte durch einen Scrum-Master bzw. die Führungskraft weiter erörtert werden, so dass eine mögliche negative Auswirkung auf das Team weniger wahrscheinlich wird.

Sichtbarkeit nach Außen Konflikte sind meist auch außerhalb des Teams deutlich sichtbar, vor allem wenn sie so stark sind, dass sie die Zusammenarbeit gefährden oder behindern. So beschrieb der Interviewer das ganz deutlich mit: „Das hat man gesehen.“ (T1.50 auf S. 188) als er über eine Situation im Team spricht, bei der zwei Mitarbeiter regelmäßig ihre Meinungsverschiedenheiten ausgetragen haben. Nach außen hin sichtbar wird das Verhalten meist dann, wenn zwischenmenschliche Beziehungen oder mehrere Teammitglieder involviert sind.

Während sich die Teammitglieder und das Umfeld auch schnell an die Situation anpassen und darauf einstellen, kann ein unbeteiligter Dritter Probleme oftmals deutlich besser sehen. Gerade die Ursachenforschung für ein Problem fällt meist schwer: „Das kann man von außen häufigerweise besser sehen.“ (T2.65 auf S. 219). Die Mitarbeiter im Unternehmen sind zu diesem Zeitpunkt schon betriebsblind und sehen die beobachtet Situation für normal an.

Interessanterweise kam von einem Interviewpartner auch die Aussage, dass die Lösung des Problems oder das Wissen darüber schon im Unternehmen vorhanden ist: „Eigentlich weiß das Unternehmen als Ganzes wo das Problem ist. Aber Teile, die Teile des Wissens haben, sind nicht in der Lage diese Aussage so zu transportieren, dass es vom Rest des Unternehmens gehört wird.“ (T2.66 auf S. 219). In diesem Fall fehlt nur eine Art von Katalysator,

um dieses Wissen zusammenzusetzen und dem Unternehmen noch einmal, aber anders darzulegen. Dies ist eine wichtige Aufgabe eines externen Beraters.

Wirtschaftliche Lage / Zahlen

Eine oftmals in der Psychologie vergessener Faktor ist, dass in einem modernen Unternehmen Zahlen über die Performanz von Projekten oder Abteilungen vorliegen (vgl. T4.15 auf S. 255 und T3.41 auf S. 238). Diese können ebenfalls eine Quelle für einen ersten Anhaltspunkt darstellen, wenn sich diese zum schlechteren entwickeln. Solche Zahlen beginnen bei Budgetierungen und Abschätzungen für bestimmte Aufgaben und die tatsächlich benötigte Zeit bzw. Geldmittel und können weiter über Fehlzeiten oder übermäßige Überstunden fort geführt werden. Dies alles können erste Anhaltspunkte sein. Unter Umständen kann es auch sinnvoll sein, gezielt Zahlen zu erheben und diese dann für eine Beobachtung des Teams zu nutzen. Nicht immer muss hierfür ein aufwendiges Controlling eingeführt werden, manchmal ist es sinnvoll auch andere Wege zu gehen, wie im Abschnitt 6.2.3: „Ungewöhnlicher Visualisierung und abstrakten Methoden“ dargestellt wird.

Zusammenfassung und Bedeutung für die Forschungsfrage

Für die Diagnose haben die Interviewpartner in ihrer Praxis vielfältig die Sinne geschärft. Sie haben meist den Vorteil, dass sie sehr eng mit dem Team zusammenarbeiten und daher nah am Team sind und ein hohes Vertrauen des Teams genießen. Dies erleichtert die Diagnose ungemein, da sich dann weder die Teammitglieder verstellen noch Information vorenthalten. Es haben sich folgende Mittel als praktikabel und wichtig für die Praxis herausgestellt:

Vertrauensverhältnis Das Verhältnis aller Coaches war sehr vertrauensvoll mit ihren Teams. Dies ermöglichte es ihnen, quasi darauf zu warten, bis die Teammitglieder ihnen von Problemen berichteten. Sie kommen bei entsprechendem Vertrauen in die Person und seiner Fähigkeiten auf den Scrum-Master, Führungsperson oder Coach aktiv zu und sprechen über Probleme im Team.

Nachfragen Vielfach hilft auch ein gezieltes Nachfragen. Dabei wird aktiv versucht Informationen über Probleme zu gewinnen. Dies setzt aber

ein gutes Vertrauensverhältnis voraus, ist dann aber ein mächtiges Mittel.

Treffen Die tägliche Arbeit erlaubt es, besonders die Interaktion im Team mitzubekommen. Hieraus lassen sich ebenso Schlüsse über die Stimmung im Team und aufkeimende Probleme und Konflikte ziehen.

Anti-Pattern Andere hatten auch schon das „Vergnügen“, mit Problemen in einem Team zu tun zu haben. Es gibt vielfältige Literatur und Internetseiten, die von solchen Situationen berichten. Meist entfährt einem beim Lesen der Darstellungen ein leichtes Schmunzeln, sie regen aber auch zum Nachdenken über das eigene Team und die eigene Situation an. Vergleiche helfen, ähnliche Situationen zu vermeiden oder sie zu erkennen.

Gefühle Die eigenen Gefühle und die Gefühle anderen sind ein wichtiger Faktor in der sozialen Interaktion. Ebenso sind sie ein wichtiger Faktor, wenn es darum geht, Probleme zu erkennen.

Verhaltensänderungen Meist ändern Menschen ihr Verhalten, wenn sie Probleme haben oder sich eine neue Einstellung ergeben hat. Dies lässt sich beobachten und dann gezielt zum Nachfragen nutzen.

Sichtbarkeit Viele Konfliktherde sind auch außerhalb des Teams sichtbar oder für einen Externen besser zu erkennen. Es lohnt sich Feedback von außen einzufordern oder der Gerüchteküche im Unternehmen zuzuhören. Auch ein externer Berater kann einen unabhängigen Blick und eine Einschätzung liefern.

In der Praxis haben hat sich besonders die Beschäftigung mit den Mitarbeitern als sehr lohnend für die Diagnose herausgestellt. Das Vertrauen des Teams zu genießen, ist dabei genauso wichtig, wie eine gute Beobachtungsgabe zu haben. So lassen sich viele Probleme bereits früh erkennen, bevor sie überhaupt in negative Entwicklungen umschlagen.

6.2.3 Lösungswege und Vorbeugemaßnahmen im Alltag bei der Arbeit mit agilen Teams

Auch wenn es nicht Kernthema dieser Arbeit ist, so stellt sich für einen Praktiker immer die Frage, wie kann ich Situationen auflösen, ihnen Vorbeugen

oder eine Linderung bewirken. Da dies ein essentieller Teil des Arbeitsalltages eines Scrum-Masters ist, möchte ich im Folgenden einige wichtige Aussagen zu diesem Thema aus den Interviews zusammenfassen.

Agile Konzepte vollständig leben

Wenn man die Vorteile agiler Entwicklung haben möchte, genügt es nicht, einige ausgewählte Konzepte oder Grundgedanken zu implementieren. Vielmehr muss Agilität in seiner Gänze zunächst im Team verankert werden und optimalerweise auch im Unternehmen. Die Gesprächspartner haben dies immer wieder indirekt angesprochen (vgl. T3.29 auf S. 235). Dabei spielt das gewählte Framework nur eine untergeordnete Rolle, so sind agile Teams auch erfolgreich, wenn sie z.B. Scrum abgewandelt haben wie in (vgl. T3.31 auf S. 236) oder andere Methodiken wie Kanban. Manchmal passt ein Framework für die Rahmenbedingungen besser, manchmal schlechter. Essentiell ist jedoch bei allen, den agilen Gedanken des schnellen Feedbacks, Inspect & Adapt sowie die Autonomie der Teams zu nutzen.

Team Verankerung / Zusammenhalt

Da es für agiles Vorgehen von extremer Wichtigkeit ist, dass sich ein Team bildet, das zusammen hält (s.a. Kapitel 6.2.1: „Teamspirit“) sollte dies einen wichtigen Fokus in der Arbeit mit agilen Teams ausmachen, dies zu fördern, wo immer es geht. Es ist wichtig, dass alle Teammitglieder eine feste Verankerung im Team haben (vgl. T1.42 auf S. 182). Der Zusammenhalt wird auch durch gemeinsame Arbeitsweisen und Ansichten sehr erhöht. Besonders wenn dies intensiv gelebt wird, hat dies positive Auswirkungen auf den Zusammenhalt im Team, aber auch auf die Qualität des Produktes, wie in T3.22 auf S. 232 beschrieben wird. Weiterhin bietet es bei der nie ausbleibenden Konfliktlösung Vorteile (s.a. Kapitel 6.2.3: „Konflikte lösen“).

Führung in agilen Teams

Auch wenn im Zusammenhang mit Führung in agilen Teams oft auf die Selbstorganisation verwiesen wird, so gibt es auch in diesen Teams immer wieder Führung. Dabei führt sich das Team situationsbedingt immer anders oder verteilt diese Rolle je nach Aufgabe immer wieder neu. Eine wichtige

Figur in der Führung agiler Teams bleibt jedoch der agile Coach, der Scrum-Master. Dieser darf sich jedoch nicht als klassische Führungskraft verstehen, sondern muss sich eher zurück halten, das Team im Vordergrund lassen und dem Team mit seinen Fähigkeiten und Netzwerken dienen. Dabei handelt er als Vorbild, als agiler Missionar, als Vertreter nach Außen, als Anschieber, je nachdem was das Team gerade braucht. Einige Zitate aus den Interviews geben einen Hinweis, wie sich die Rolle Verhalten sollte:

- „Und das ist jetzt wieder Servant-Leader, oder eine Führungsperson oder eine Führungskraft ist dann wesentlich effizienter und anerkannter, wenn sie natürlich auch das mitträgt und aktiv mit lebt und beispielhaft zeigt. " (T3.37 auf S. 238), es gilt also ein Vorbild zu sein, ein gutes Beispiel abgeben.
- „Take a little bit less of credit than you share and a little bit more of the blame than your share." (T2.48 auf S. 211). Dies ist nicht unbedingt neu seit es das Agilität gibt, sondern dem Sinne nach wurde ein sehr ähnliches Zitat durch den Geschäftsmann Arnold Henry Glasow (1905 - 1998) geprägt: „A good leader takes a little more than his share of the blame, a little less than his share of the credit." Der Gedanke hinter diesem Satz jedoch ist gerade im Agilen wichtig und weist darauf hin, dass es wichtig ist, sich als agiles Teammitglied oder Führungsperson immer ein Stück weit zurück zu nehmen für das Ganze.
- „(…) wenn man ein Problem identifiziert hat, geht es nicht darum, mit dem Hammer zu kommen, (…) sondern die Leute abzuholen (…) ." (T3.32 auf S. 236)
- „(…) es ging niemals darum zu sagen: „So, dann machen wir das jetzt so!", sondern es ging darum, Coach zu sein, in der Retrospektive das Team reden lassen und vielleicht auf bestimmte Dinge hinweisen: „Habt ihr schon mal das gesehen?", vielleicht indem man bestimmte Informationen speziell aufbereitet, so dass ein Problem sichtbar wird, (…) ." (T3.34 auf S. 237). Manchmal ist ein Scrum-Master auch der Meinung, dass es ein Problem gibt, das Team ist sich dem jedoch nicht bewusst. Im Sinne des Coaching-Gedankens kann der Scrum-Master in dieser Situation nicht eine Lösung vorschlagen, sondern optimalerweise sollte er das Team höchstens darauf hinweisen oder z.B. Informationen so aufbereiten, dass das Team deren Tragweite erfassen und Schlüsse daraus ziehen kann (vgl. T3.33 auf S. 236 und T3.50 auf S. 242).

- „(…) da versucht man dann als Scrum-Master neue Impulse einzubringen.“ (T3.4 auf S. 226), als Scrum-Master neue Impulse setzen, wenn z.B. die Disziplin im Team nachlässt. Dies erfolgt jedoch indirekt und ohne direkte Anweisungen.

Weiterhin ist es wichtig, dass ein Scrum-Master sowohl ein gutes Standing im Team, als auch im Unternehmen hat. So nutzt es niemanden als Scrum-Master eingesetzt zu werden, wenn das Team ihn ablehnt oder wenn er nicht über ein Netzwerk und Einfluss im Unternehmen besitzt, um Änderungen anzustoßen [2].

Vorbilder Übereinkommen haben die Interviews auch die Vorbildfunktion als wichtig und sehr mächtig befunden (vgl. T2.49 auf S. 213 und T3.36 auf S. 237). Die Vorbildfunktion ist in einem agilen Team sehr machtvoll und kann dazu genutzt werden, um dem Team eine Richtung vorzugeben oder schlechte Einflüssen bzw. Entwicklungen vorzubeugen. Da sich ein agiles Team zum Großteil selbst führt, dienen verschiedene Teammitglieder als spezielle Führungspersönlichkeiten, die je nach Situation wechseln (s.a. Kapitel 4.4). Dabei übernehmen immer wieder Schlüsselfiguren eine Vorreiterrolle, wenn es darum geht, neue Dinge auszuprobieren oder umzusetzen. Sie stehen dann quasi im Fokus und es wird beobachtet, ob derjenige an den Erfolg glaubt und das ebenso umsetzt. Besonders wenn die Person ein hohes Ansehen im Team besitzt, kann dies ein wichtiger Faktor sein, wie das Beispiel eines erfolgreichen Scrum-Masters zeigt, der durch sein eigenes Team gewählt wurde und damit ein hohes Standing (vgl. T3.1 auf S. 224) im Team hatte : „Dadurch, das ich dann auch als Entwickler dem Prozess vorbildlich gefolgt bin, habe ich gezeigt, dass es geht und habe auch gleichzeitig motiviert, dass die anderen sagen: „Ok, der macht das auch und dann zieh ich da auch mit.““ (T3.38 auf S. 238).

Mit Menschen befassen Die Arbeit als Scrum-Master oder Coach hat immer viel mit Menschen zu tun, daher ist es wichtig, sich mit den Menschen zu befassen und sie nicht als Ressourcen zu begreifen. „Was motiviert sie selbst zum arbeiten? Was ist ihnen besonders wichtig? Was soll ich auf gar keinen Fall tun?“ (vgl. T2.51 auf S. 214); das alles sind Fragen, die wichtig sind für einen Menschen und daher sollten diese Fragen auch wichtig für

[2] Dies ist mir aus vielen Gesprächen mit anderen agilen Entwicklern und Scrum-Mastern bekannt.

eine Führungskraft sein. Diese Fragen helfen der Führungskraft einzuschätzen, wieso derjenige etwas nach einem bestimmten Muster macht und wieso er vielleicht geraden nicht so motiviert in seiner Arbeit ist. Ein gutes Mittel, was dazu von einem Interviewpartner eingesetzt wurde, sind „1on1"-Treffen. Hierbei trifft man sich miteinander in einer entspannten Atmosphäre, losgelöst vom Arbeitskontext und spricht über diese Dinge (vgl. T2.52 auf S. 214). Die daraus gewonnenen Erkenntnisse werden dann für die weitere Arbeit als Coach genutzt. Gerade bei Verhaltensänderungen von Einzelpersonen ist es sinnvoll sich mit den Menschen intensiver zu beschäftigen. Es gilt dabei zu ergründen, was diese Änderung ausgelöst hat und wie eine Abhilfe geschaffen werden kann (vgl. T2.55 auf S. 215).

Mediation nutzen, wenn die Situation es erfordert Eine Vorbereitung auf jede erdenkliche Situation ist im vornherein nicht möglich. Es kommt immer wieder zu neuen Situationen, die einen Scrum-Master oder eine Führungskraft an die Grenze bringt, so dass sie nicht mehr weiter weiß und einen Rat benötigt.

So beschreibt ein Scrum-Master eine Situation, in der er selbst nicht mehr weiter wusste: „An einem Tag hat er wirklich einen schlechten Tag gehabt, einen wirklich schlechten Tag und da hat er die Retrospektive so dermaßen blockiert und zerschossen, dass das Team wirklich fertig war." (T3.12 auf S. 228) Dies ist nicht unüblich, denn manchmal entwickeln Situationen eine Eigendynamik und geraten aus dem Ruder. Hierbei jedoch den richtigen Punkt zu finden, an dem man z.B. besser abbricht und den Schaden begrenzt ist nicht einfach. Hierfür benötigt es an Erfahrung und selbst Leuten mit viel Erfahrung wird es immer wieder passieren, in eine schwierige Situation zu geraten. Manches Mal ist eine Situation so festgefahren oder der Scrum-Master oder die Führungskraft ist in den Konflikt involviert, dann ist es Zeit, sich nach anderen Mitteln und Wegen umzusehen. Ein probater Weg ist, sich einzugestehen, dass diese Situation Hilfe benötigt, Hilfe auch, da ein Weiterkommen unmöglich ist. Dies darf auch nicht als persönliches Versagen gewertet werden, sondern muss als natürlich verstanden werden, so dass dann z.B. ein externer Coach gerufen wird oder auf Mediation zurück gegriffen wird (vgl. T3.13 auf S. 228). Es ist wichtig zu wissen, dass dies nicht unüblich ist; in vielen Firmen oder auch unternehmensübergreifend in bestimmten Region gibt es für solche Situationen Runden, bei denen sich Scrum-Master treffen, um sich über Situationen auszutauschen.

Konflikte lösen

Viel wurde von den Interviewten über die Lösung von Konflikten gesagt. Wenn auch die Vermeidung von Konflikten immer dem Vorzug gegeben werden sollte und dafür gesorgt werden soll, dass Konflikte durch die Beteiligten selbst gelöst werden können, funktioniert dies nicht immer. Durch die Interviewpartner wurden sowohl Lösungsmöglichkeiten, Best-Practice aus ihrer Sicht als auch wichtige Voraussetzungen angesprochen.

Allgemein hilft die Selbstorganisation und die agilen Ansätze schon, dass die Mitarbeiter versuchen werden, Konflikte selbständig zu lösen. Oftmals ist es so, dass der Chef wenig bis gar keinen Einfluss auf die Lösung von Konflikte nehmen kann, er kann bestenfalls vermitteln oder die Ursache für einen Konflikt beseitigen. Der Konflikt selbst muss durch die beiden Parteien selbst gelöst werden (vgl. T1.48 auf S. 187).

Eine potentielle Lösung für einen Konflikt muss folgende Eigenschaften haben:

- Die Lösung muss vom gesamten Team getragen werden, wenn sie das Team betrifft (vgl. T3.19 auf S. 230).
- Die Umsetzung der Lösung muss wirksam sein (vgl. T3.18 auf S. 230).

Konfliktlösung benötigt Zeit und Raum Wenn ein Konflikt erst einmal ausgebrochen ist, dann ist vor allem eines für die Lösung notwendig: Raum und Zeit (vgl. T2.53 auf S. 214). Es ist so gut wie aussichtslos einen Konflikt unter Zeitdruck aufzulösen, da dies dazu führt, ihn nur oberflächlich zu behandeln. Es kostet Zeit, bis die Ursachen geklärt werden können, um Vertrauen wieder aufzubauen oder eine für beide akzeptable Lösung zu finden.

Manche Konflikte lassen sich nicht im Team selbst lösen, gerade wenn es sich um Konflikte handelt, bei denen die Persönlichkeit des Einzelnen sehr stark involviert ist. Dann kann es sinnvoll sein, mit der Einzelperson ein Gespräch zu führen, um zu ergründen, warum es diesen Konflikt gibt und welche Gründe vorliegen, dass es mit dem Team zusammen nicht funktioniert. Gerade auch wenn sich jemand mit „(…) seiner Kritik abgeholt fühlt (…)“ (T3.15 auf S. 229) kann dies sehr wichtig sein, um den Konflikt soweit zu deeskalieren, dass im Anschluss an den fachlichen Themen gearbeitet werden kann und die emotionale Ebene verlassen werden kann. Schlussendlich

kann dies dazu führen, den Konflikt zu bereinigen, indem das fachliche Thema gelöst wird, wie folgendes Zitat zeigt: „Da ist er plötzlich sehr aufgeblüht und hat sich geöffnet und einen sehr guten Vorschlag mit mir ausgearbeitet, der auch vom Team sehr akzeptiert wurde.“ (T3.16 auf S. 229).

Persönliche Wertschätzung und Vertrauen Bei der Lösung von Konflikten ist oftmals ein unbeteiligter Dritter als Vermittler notwendig. Die Rolle des unbeteiligten Dritten ist jedoch nicht einfach, denn zum einen darf dieser keine Partei ergreifen und muss das Vertrauen beider Parteien haben. Sie müssen ihm zutrauen, dass er den Konflikt lösen kann und dass er in ihrem Interesse handelt (vgl. T3.47 auf S. 240). Dabei ist es wichtig, gute persönliche Beziehungen zu haben, denn auch wie im Coaching sind diese von Vorteil für das Lösen eines Konfliktes (vgl. T2.57 auf S. 215).

Gerade wenn der Konflikt zwischen einer Einzelperson und dem Team besteht, kann es hilfreich sein, die Übermacht des Teams zu reduzieren und sich zunächst mit der Einzelperson zu beschäftigen, ihm eine Wertschätzung entgegen zu bringen und die Stimme hören: „Ich glaube die Menschen wollen einfach abgeholt werden. Wenn man ihnen dann den Respekt und ein Ohr schenkt, dann glaube ich das die meisten darauf eingehen und die meisten Menschen auch ein Interesse daran haben, ihre Meinung kund zu tun.“ (T3.17 auf S. 229). Dies kann der erste Weg sein, Deeskalierend auf einen Konflikt einzugehen und ihn auf die sachliche Ebene zu begrenzen.

Konflikte Vorbeugen durch frühzeitige Intervention Für einen Scrum-Master ist es wichtig, ein Gespür zu entwickelt, wann eine Diskussion im Team förderlich ist und ab wann eine Diskussion zum Problem wird. Dies ist eine nicht einfach Aufgabe, die gerade unerfahrene Scrum-Master vor eine schwere Aufgabe stellt, wie das Beispiel aus T3.12 auf S. 228 zeigt. Schnell kann die Stimmung von einer fachlichen Diskussion hinein in eine emotionale kippen. Eine gute Orientierung und ein Hilfsmittel kann dabei sein, die Idee der Time-Box hier aufzugreifen und die Situation nach einer gewissen Zeit abzubrechen und die Diskussion in einer kleineren Gruppe, „Task-Force“ (T3.18 auf S. 230) weiterzuführen. Dabei ist es im Anschluss allerdings wichtig, dass die Lösung vom gesamten Team getragen wird (vgl. T3.19 auf S. 230).

Mile-High-Retrospektive

Scrum ist nur ein Framework, das bewusst minimalistisch gehalten wurde, um auf möglichst viele Situationen anwendbar zu bleiben. In den Interviews kristallisierte sich jedoch heraus, dass viele Teams ein spezielles Element implementierten. Diese hatte zwar unterschiedliche Namen (Release-Retrospektive, Mile-High-Restrospektive, 10000-Meter-Retrospektive), aber im Wesentlichen beinhaltet es eine spezielle Art von Retrospektive, die einen längeren Zeitraum mit einem gewissen Abstand betrachtet. Damit wird es möglich, den Fokus weg von einem einzelnen Zyklus, den Sprint mit seinen spezifischen Problemen zu verlassen und Sprint-Übergreifend Probleme zu identifizieren. Das sind meist übergeordnete Probleme, die erst auffallen, wenn das eigene Verhalten über eine längere Zeit gesehen wird (vgl. T2.67 auf S. 221 und T3.48 auf S. 240). Aber auch übergreifende, organisatorische Themen können dort betrachtet werden. In T2.68 auf S. 222 wird hierfür z.B. das Managementverhalten und die Firmenkultur als ein Beispiel genannt.

Nicht alles kann das Team selbst lösen

Als letztes ist es wichtig zu wissen, das ein agiles Team nicht alle Probleme eigenständig lösen kann. Immer wieder stößt es an Grenzen seiner eigenen Autonomie. In Organisation ist dies klassischerweise erreicht, wenn es über den Horizont des Teams hinaus geht oder es die Struktur im Team selbst betrifft. Es ist z.B. für ein Team normalerweise nicht möglich zu entscheiden, dass es ein Teammitglied abgeben kann (vgl. T2.50 auf S. 213) oder Änderungen, die andere Abteilung (vgl. T3.9 auf S. 226) betrifft. Hierfür Lösungen zu schaffen, die einen Weg aus dem Dilemma ebnen ist wichtig und hilft vorzubeugen, das ein Team motiviert bleibt (vgl. T2.13 auf S. 200 und T3.8 auf S. 226). Dies sollte schon im Vorfeld geschehen und es sollte eine offene Atmosphäre und Verständniss im Unternehmen geschaffen werden, diesen Austausch zu fördern.

Ungewöhnlicher Visualisierung und abstrakten Methoden

Manchmal ist Ideenreichtum und Ausprobieren ein probates Mittel, neue Wege in der Arbeit mit Teams zu gehen. Manchmal genügt es schon, den gewohnten Trott zu verlassen und andere Formen des Feedbacks zu nutzen. So wurde von den Interviewpartnern auch zunächst befremdliche Vorgehensweise für die Aufarbeitung von Retrospektiven, wie die Fußballretrospektive

(vgl. T3.49 auf S. 241) genannt. Diese Methode wird schon vermehrt von anderen Trainern angewandt, sie beinhaltet eine Art von mentaler Arbeit, die es ermöglicht über Geschehnisse nachzudenken und sie andersartig einzuordnen und neue Erkenntnisse zu gewinnen. Ebenso kreativ wie einfach ist auch die Idee, Legosteine zur Visualisierung zu nutzen (vgl. T3.51 auf S. 244). Nicht immer bedarf es einem starken Controlling mit entsprechendem Personal, sondern oftmals kann schon mit vergleichsweise wenig Einsatz doch eine wichtige Erkenntnis gewonnen werden. Hier gilt es, gerade im Umfeld mit agilen Teams, kreativ zu sein und neues auszuprobieren, ganz im Sinne der agilen Entwicklung: Lean, also schlank.

6.2.4 Weitere identifizierte Themen

Während den Interviews kamen neben den Kernthemen weitere zur Sprache, die interessante Forschungsbereiche sind (wie z.B. „Scrum im kulturellen Kontext“) oder informativ sind und für weitere Ideen und Anregungen zum Verständnis sein können.

Einführung, Gründe, Effizienz, Leistung

Der Begriff Leistung im agilen Kontext entflammte eine längere Rede mit einem Interviewpartner (vgl. T2.6 auf S. 196). Der Begriff Leistung ist gerade im Umgang mit agilen Methoden und Teams mit Vorsicht zu genießen, da er meist mit einem höheren Output gleich gesetzt wird. Viele denken bei Output jedoch nur an Quantität, also dass mehr in der gleichen Zeit erstellt werden kann. Hier wird eine klare Aussage getroffen, dass dies nicht das primäre Ziel ist: „Die Qualität der Outputs sollte bei einem gut funktionierenden Team, agilen Team hoch sein und im Schnitt auch höher, als es bei anderen Teams ist, was aber nicht sozusagen hoch geschraubt wird, das kann mal ein Nebeneffekt sein, es gibt auch Fälle bei denen es sicher passiert ist, aber es ist nicht der Default-Fall, dass das Team plötzlich viel mehr leisten wird, plötzlich viel mehr produzieren, mehr Ergebnisse machen wird, es wird eher die richtigen und bessere Ergebnisse produzieren.“ (T2.6 auf S. 196)

Agile Methoden versprechen per se keine höhere Leistung im Sinne von mehr erstellte Software, sondern sie verfolgen unter anderem folgende Ziele (vgl. auch T2.6 auf S. 196):

- Responding to Change: Das Team und damit letztendlich das Unternehmen kann sich deutlich schneller an neue Situationen im Markt und oder Technolgieänderungen anpassen und davon profitieren (vgl. T2.5 auf S. 195).
- Teams produzieren mehr von dem, was der Kunde wirklich braucht (vgl. T2.3 auf S. 195).
- Crossfunktionalität: Ein Mitarbeiter kann die Arbeit eines anderen übernehmen, es entstehen weniger Situation, bei denen es auf ein einzelnes Teammitglied ankommt. Es wird das Risiko für das Unternehmen minimiert. (vgl. T2.7 auf S. 196)
- Die Qualität der Software ist besser, da das Team eine höhere Verantwortungsübernahme für diese Arbeit hat.
- Zufriedene Mitarbeiter und damit weniger Wechsel und Verlust von Wissen: „(…) wenn ich jemanden 10, 12, 15 Jahre im Unternehmen halten kann, hat er für mich im Normalfall einen ganz anderen Wert, als wenn alle 3 Jahre die Leute weg sind (…) .“ (T2.4 auf S. 195).

Diese Definition von Leistung wurde dann auch in den anderen Interviews vorgezogen, so dass der Begriff von Leistung nicht auf den Output eingeschränkt wurde, sondern die anderen Elemente vom Leistungsbegriff ebenso mit einbezogen wurden.

Softwareprojekte sind nicht gleich — Scrum-Implementierungen auch nicht

Jedes Projekt in der Softwareentwicklung ist andersartig und einzigartig. Es gibt (noch) keinen Prozess für eine Massenfertigung von Software, der immer wieder gelingt. Es gibt zwischen den Projekten Ähnlichkeiten, vieles aus anderen Projekten kann als Anregung und Best-Practice genutzt werden, manches, was jedoch in einem Projekt perfekt funktionierte kann ein anderes zum scheitern bringen. Das liegt oft an den wechselnden Rahmenbedingungen, aber ebenso an Teammitgliedern, dem Team selbst, der Aufgabenstellung, der verwendeten Technologie, … diese Liste ließe sich beliebig fortsetzen. Dies haben mir die Interviewpartner durch einige Zitate zurückgespiegelt:

- „Dann weiß der genau, was da alles zu tun ist, was man so braucht, um so ein Projekt erfolgreich durchzuführen, wenn er Glück hat; wenn

man Pech hat, dann weiß er das zwar für die Projekte, die er bisher gemacht hat, aber nicht für das tolle neuartige Projekt, das ganz anders ist und dann wundert man sich, warum es nicht funktioniert (…)“ (T1.26 auf S. 175)

- „Das ist halt in der Software, ist jedes Projekt anders. Man kann die Erkenntnisse von einem auf das nächste Projekt nur bedingt mitnehmen.“ (T4.17 auf S. 257)

So verschieden wie Softwareprojekte sind, so unterschiedlich müssen die tatsächlichen Implementierungen von Scrum in den Unternehmen sein. Sie haben zwar immer wieder die einzelnen Elemente, allerdings, was dort außen herum noch genutzt wird, wie die Schnittstellen zu anderen Abteilungen und Prozessen verteilt sind, wie die eigentliche Arbeit durchgeführt wird, daher fällt es manchem Unternehmen schwer, Scrum zum ersten Mal auf sich zu adaptieren. Scrum-Gurus antworten häufig auf gestellte Fragen: „Das schreibt Scrum nicht vor!“ (T1.43 auf S. 184), was leider dem neuen Scrum-Master nicht wirklich weiter hilft. Es kann dann schnell passieren, was folgendes Zitat deutlich macht: „(…) man in der Praxis meint: „Wir machen jetzt Scrum!“ Es aber gar nicht richtig umsetzt (…)“ (T1.44 auf S. 184) Manchmal lässt sich Scrum auch nicht wirklich sinnvoll einsetzen. Hier können dann andere Vorgehensweisen wie z.B. Kanban besser passen. Ein Beispiel, wann Scrum nicht viel Sinn macht: „…vielleicht gar nicht sinnvoll iterativ arbeiten kann (…) wir haben z.B. einen festen Scope und einen festen Zeitraum …“ (T1.45 auf S. 184).

Wann es Sinn macht und wann nicht, das kann nur über die Praxis herausgefunden werden. Gut, dass Scrum kein fixer Prozess ist und das Kernprinzip hat, sich immer weiter zu verbessern und zwar ergebnisoffen (vgl. T1.46 auf S. 185). Daher braucht es Zeit, um Erfahrungen zu sammeln, wie es im aktuellen Kontext funktionieren kann: „Scrum ist immer ein evolutionärer Prozess, d.h. es ist in dem Prozess-Framework eingebaut, das man gewisse Zeit braucht, bis man herausgefunden hat wie es funktioniert.“ (T4.17 auf S. 257).

Scrum im kulturellen Kontext

Die befragten Personen haben ihren Hintergrund zur Arbeit und Problemen in europäischen, fast ausschließlich deutschen Teams gesammelt. Wie durch einen Interviewpartner angemerkt wurde, kann die Kultur auch ein

wichtiger Faktor für den Erfolg eines Scrum-Teams sein: „(…) je nach Kultur passen verschiedene Aspekte von Agilität besser oder schlechter.“ (T2.1 auf S. 192). Es wurde insbesondere die These aufgestellt: „Also je autoritärer eine Kultur ist, desto weniger passen halt auch solche agilen Vorgehensweisen mit Delegationskonzepten sehr gut rein.“ (T2.2 auf S. 192). Dies ist durchaus vorstellbar, es ist auch denkbar, dass noch andere Themen betroffen sind. In einem anderen Kulturkreis kann es daher ganz andere negative Entwicklungen oder Möglichkeiten zur Diagnose geben oder manche können auch komplett weg fallen, da sie in der anderen Kultur als problematisch anzusehen sind. Der Kulturaspekt ist ein interessanter Ansatz für weitere Forschungen.

7

Kapitel 7

Zusammenfassung

Gute agile Teams sind in erster Linie eines: Sie sind ein sogenanntes High-Performance-Team. Sie entscheiden nicht selten über den Geschäftserfolg eines Produktes auf lange Sicht. Es stellt sich damit die Frage: Unterscheiden sie sich von anderen High-Perfomance-Teams in anderen Bereichen?

Ein Team ist zunächst einmal ein Team, egal ob es sich High-Perfomance-Team oder agiles Team nennt. Es gelten somit die gleichen Gesetzmäßigkeiten, die auch für klassische Teams gelten. Wir haben dies im Kapitel 2.7 bereits ausführlichst erläutert. Wenn wir unseren Experten aus den Interviews analysieren, so ist festzustellen, dass sie sehr ähnliche Techniken zur Teamentwicklung einsetzen und auf die gleichen Modelle wie die für klassische Teams zurückgreifen. Ein agiles Team durchläuft z.B. genauso die Phasen nach Tuckman und Jensen aus Kapitel 2.5 (vgl. T1.23 auf S. 171, T2.62 auf S. 218 und T3.25 auf S. 232). Warum sollte sich ein Mensch oder ein Team auch anders verhalten, nur weil es agiles Team genannt wird? Die soziologischen Gruppenprozesse sind immer die Gleichen, da immer Menschen in einem Team anzutreffen sind. Agile Teams kämpfen unter Umständen mit anderen Problemen und sind meist mit Menschen eines ähnlichen Typus besetzt (meist Akademiker mit einer positiven Einstellung zu Neuem und was für Entwickler typisch ist: Die meisten sind neugierige „Spielkinder“ und damit probieren sie gerne Neues aus.). Beschreibungen von negativen Effekten aus den Modellen über Teams und Gruppen sind im Interview kaum angesprochen worden, da die Teams der Interviewten damit wenig Probleme hatten. Bei den Rollen gab es in einem schwierigen Team durchaus Probleme, hier waren die Rollen nicht klar abgesteckt und die Verantwortung wechselte stark. Dies hat zu Konflikten und Hindernissen in der Arbeit geführt.

Auch die klassischen Rollen spielen in diesen Teams neben den normalen Scrum-Rollen eine große Rolle, wenn sie auch oft nicht so wahrgenommen und deswegen nicht erwähnt werden (s.a. Kapitel 6.2.1: „Rollen im Scrum“). Einen umfassenden Überblick zwischen der Verbindung von agilen Methoden und den klassischen Modellen ist in Kapitel 5 zu finden. Dort werden auch die beiden unterstützenden Forschungsfragen beantwortet:

- Welche negativen Entwicklungen aus den klassischen Modellen der Teamentwicklung lassen sich auf agile Scrum-Teams übertragen?
- Welche inhärenten Merkmale sind in agilen Methoden zu finden, die einer negative Entwicklungen vorbeugen?

Auf einen Punkt gebracht: Für agile Teams gelten die klassischen Modelle der Teamentwicklung ebenso. Die Methoden, wie Teamentwicklung und Diagnose durchgeführt werden können, sind allerdings erweitert. So verfügt das agile Team im Scrum-Prozess von sich aus über Mechanismen zur Reflexion: Daily-Standup, Sprint-Retrospektive, Sprint-Review und was viele Teams zusätzlich installieren, die Mile-High-Retrospektive. Dies ermöglicht es ihnen, die negativen Entwicklungen früher als in anderen Teams zu erkennen und ihnen entgegenzuwirken. Untermauern lässt sich dies auch durch die Interviews, wobei für eine abschließende Bewertung dieser neuen These eine breitere Basis an befragten Experten aus der IT oder die Beobachtung von deutlich mehr Teams im Vergleich zu konventionellen Teams notwendig wäre.

Die Ergebnisse der Interviews, welche die eigentliche Forschungsfrage beantworten sollen, decken sich weitgehend mit eigenen Erfahrungen aus meiner täglichen Arbeit. Gerade die negativen Entwicklungen sind in der Praxis in der einen oder anderen Form immer wieder zu sehen. Die Forschungsfrage

- Welche typischen negative Entwicklungen lassen sich in der Praxis beobachten?

wird hauptsächlich durch die Experten aus den Interviews beantwortet, denn nur sie verfügen über die Sicht aus der Praxis. Die Zusammenfassung der wichtigsten und damit in der Praxis beobachtbaren negativen Entwicklungen sind in Kapitel 6.2.1 auf Seite 108 zu finden. Durch die Nennung der negativen Entwicklungen durch die Experten ist davon auszugehen, dass sie in ihrer Arbeit eine hohe Relevanz haben. Viele der Themenbereiche decken sich auch mit den vorher getätigten theoretischen Überlegungen an-

hand der Literatur und der Modelle. So war z.B. das Stichwort Alignment schon bei den Voraussetzungen nach Glogger für die Selbstorganisation (siehe Kapitel 3.2.1) zu finden. Aber auch negativen Entwicklungen aus den Bereichen Kommunikation, Teamzusammensetzung und Teamspirit zeigen große Übereinstimmungen mit den Untersuchungen aus dem Kapitel 2.2 und den dort genannten Eigenschaften für Teams im Vergleich zu einer Gruppe. Ebenso zeigen sich deutliche Übereinstimmungen mit den schlechten Einflüssen von Kuster et al. im Kapitel 2.4 mit den im Alltag wahrgenommenen Konflikten. Die Elemente von Scrum zielen dabei darauf ab, diese Einflüsse auszuschalten und damit weniger Konflikte aufkommen zu lassen. Viele der weiterhin genannten präventiven Maßnahmen aus Kapitel 6.2.3 haben ebenfalls zum Ziel, Konflikte zu vermeiden oder sie schnell und eigenständig durch das Team lösbar zu machen.

Letztendlich wurden die eigentliche Forschungsfrage beantwortet:

- Wie können relevante negative Entwicklungen in agilen Scrum-Teams praxisgerecht beobachtet werden?

Durch die Experten im Abschnitt „Konflikte und negative Entwicklungen im agilen Team“ in Kapitel 6.2.1 konnten die relevante, also die auch in der Praxis bei den Experten auftreten, negative Entwicklungen aufgezeigt werden. Felder, die sich als besonders wichtig für agile Teams herausgestellt haben sind unter anderem:

- Raum und Zeit (Colocation)
- Feedback
- Autonomie und Alignment
- Persönlichkeiten
- Skills und Wissen
- Zusammensetzung des Teams
- Teamspirit
- Kommunikation
- Rollen
- Veränderungen
- Disziplin.

Besonders in Veränderungen dieser Faktoren finden sich laut der Experten die praxisrelevanten negativen Entwicklungen, die von ihnen regelmäßig beobachtet werden. Diese lassen sich durch ihren Charakter oftmals schon direkt beobachten, weswegen eine solche Entwicklung meist auch gleich die

Diagnostik beinhaltet. Darüber hinaus haben sich die Experten eine Art von Best-Practice angewöhnt. Diese Erfahrungen werden im Abschnitt „Erkennen von Konfliktherden und Problemen in agilen Teams“ in Kapitel 6.2.2 mit dem Leser geteilt. Diese Umfassen folgende Themenfelder, die für eine Diagnose nützlich sein können:

- Vertrauensverhältnis zum Team
- aktives Nachfragen und Zuhören
- Meetings beobachten
- nach Anti-Pattern Ausschau halten
- eigene und fremde Gefühle beobachten
- Verhaltensänderungen im Team und von Einzelpersonen
- Sichtbarkeit nach außen

Diese Möglichkeiten zur Diagnose sind besonders wichtig, denn sie enthalten Möglichkeiten, wie neuartige negative Entwicklungen erkannt werden können, die bis jetzt noch nicht erfasst sind oder durch geänderte Situationen auftreten können.

Darüber hinaus ist es in der täglichen Arbeit mit Teams auch wichtig: Was kann ich tun, wenn ich diese Probleme erkannt habe und gibt es Vorbeugemaßnahmen hierzu? Natürlich kann dies nicht in einem Umfang beantwortet werden, wie die anderen Fragen; einige Anregungen haben die Experten dazu gehabt, die in Abschnitt „Lösungswege und Vorbeugemaßnahmen im Alltag bei der Arbeit mit agilen Teams“ in Kapitel 6.2.3 zusammengestellt sind. Hier wurden Themen angesprochen, die von Konfliktlösungen über Führung in agilen Team bis hin zu speziellen Methoden oder zusätzlichen Elementen wie die Mile-High-Retrospektive handeln. Sie runden damit den praktischen Einsatz ab und geben wichtige Hinweise.

Fazit

Agile Teams sind kein Allheilmittel für Probleme während der Softwareentwicklung, allerdings können sie vor allem durch ihren Ansatz der kontinuierlichen Verbesserung viele Vorteile für die Zukunft bringen. Sie sind vor allem eine langfristige Alternative zum klassischen Management. Gerade in Zeiten steigender Komplexität und kürzerer Innovationszyklen ist eine agile Reaktion notwendig. Sie sind jedoch nicht davor geschützt, das sich negative Entwicklungen im Team bilden und Konflikte entstehen. Eine richtig gelebte Agilität hat einige Sicherungsmaßnahmen entworfen, damit sich das

Team selbständig verbessern und auch selbst organisieren kann. Dies entlastet nicht nur das Management an sich, das daraufhin andere Aufgaben wahrnehmen kann, sondern bringt noch zusätzlich den Vorteil, dass es näher am Markt ist und für den Kunden potentiell bessere Produkte liefert. Darum gehen viele Unternehmen den Weg in die Agilität und bilden neue Manager aus, die nicht mehr nur managen, sondern sich als Coach, als Servant-Leader für die Mitarbeiter verstehen. Dabei müssen diese neuen Manager besonders im sozialen Bereich eine hohe Kompetenz haben, damit sie gerade bei der Diagnose von negativen Entwicklungen und der Vorbeugung von Konflikten ihren Vorteil ausspielen können. Autonomie, Übernahme von Verantwortung, Vertrauen und ein effizientes Feedback, was für die Arbeit von agilen Teams unerlässlich ist, müssen erlernt und durch Authentizität der Führung widergespiegelt werden.

8 Kapitel 8

Ausblick

Agile Ansätze in der Softwareentwicklung sind mittlerweile nicht mehr komplett neu, dennoch sind viele Aspekte noch vollkommen unerforscht oder nur aus der Praxis bekannt. Wie schon durch die interviewten Personen zu sehen war, die Grenze eines agilen Teams sollte nicht an der Teamgrenze sein. Auch ein Unternehmen wird zwangsläufig agiler werden müssen, wenn es auf agile Teams setzt. Dies fällt nicht immer leicht, heißt es nicht auch Hierarchien abbauen und ein sich veränderndes Managementverständnis. Einige Unternehmen sind diesen Weg schon gegangen und haben eine agile Transformation hinter sich gebracht, ja sogar schon den Posten des Geschäftsführers zur Wahl gestellt, wie das Beispiel des Unternehmens Haufe zeigt. Hier ist ein weites Feld für Experimente, aber auch für Feldstudien. Diese Art des Managements 3.0, als Mitarbeiter sich überall beteiligen zu können, Dinge einfach ändern zu können, ist ein großer Zukunftstrend, der viele Vorteile bringt. Das Wissen der Mitarbeiter wird effektiver genutzt, gleichzeitig bedarf es hierzu ein Umdenken im Management. Dies ist auch gleichzeitig einer der Herausforderungen bei dieser Entwicklung, denn Management, wie es heute üblich ist, steht einer agilen Transformation meist im Wege.

In unseren Gesprächen hat sich noch ein interessanter Punkt ergeben: Wie verhält sich Scrum im interkulturellen Kontext? Hier sind durchaus Unterschiede zu erwarten. Hier lohnt es sich zu forschen. Wie werden Kulturen, die sich mit agilen Grundwerten schwer tun, in diesem Bereich erfolgreich sein? Wo können dennoch solche Konzepte umgesetzt werden, so dass das Wissen aller im Unternehmen genutzt wird und nicht ein einzelner die Geschicke des Unternehmens lenkt, sondern viele?

Es ist und bleibt spannend, was sich in den Köpfen, in der Einstellung zum Arbeiten, in der Art Menschen zu führen ändern wird. Eines ist gewiss, die heutige Komplexität verlangt nach einer geteilten Verantwortung, nach einer breiten Wissensbasis. Ein Kopf alleine kann weder das Marktgeschehen mehr erfassen, noch jede Technologie begreifen. Der Erfolg liegt mittlerweile nicht mehr in der Hand eines Menschen, sondern in den Köpfen vieler. Nehmen wir die Verantwortung in die Hand, forschen an neuen Ansätzen und zeigen den Menschen, wie wir ein lohnendes Arbeitsumfeld für die zukünftigen Generationen schaffen können.

9 Kapitel 9

Literatur

Adkins, Lyssa (2010). *Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition*. Addison-Wesley Signature Series (Cohn). Pearson Education. ISBN: 9780321660350 (siehe S. 2, 64).

Antons, Klaus (2000). *Praxis der Gruppendynamik. Übungen und Techniken*. 9., durchgesehene und ergänzte Auflage. Göttingen, Bern, Toronto, Seattle: Hogrefe. ISBN: 9783801723545 (siehe S. 8, 26).

Ardelt-Gattinger, Elisabeth und Erich Gattinger (1998). "Gruppenarten und Gruppenhasen". In: *Gruppendynamik. Anspruch und Wirklichkeit der Arbeit in Gruppen*. Hrsg. von Elisabeth Ardelt-Gattinger, Hans Lechner und Walter Schlögl. Verlag für Angewandte Psychologie, Göttingen, S. 2–9. ISBN: 978-3801710934 (siehe S. 25).

Aronson, Elliot, Timothy Wilson und Robin Akert (2014). *Sozialpsychologie*. Pearson. ISBN: 978-3-86894-217-0 (siehe S. 8, 11–15, 26).

Arrow, Holly und Joseph E McGrath (1995). "Membership dynamics in groups at work: A theoretical framework". In: *Research in organizational behavior* 17, S. 373–373 (siehe S. 8).

Balkema, Ageeth und Eric Molleman (1999). "Barriers to the development of self-organizing teams". In: *Journal of Managerial Psychology* 14.2, S. 134–150 (siehe S. 41).

Bass, B.M. (1985). *Leadership and Performance Beyond Expectations*. Free Press. ISBN: 9780029018101 (siehe S. 65).

Baumgartner, Manfred et al. (2013). *Agile Testing. Der agile Weg zur Qualität*. Carl Hanser Verlag. ISBN: 978-3446431942 (siehe S. 46).

Beck, Kent (1999). *Extreme programming explained: embrace change*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. ISBN: 0-201-61641-6 (siehe S. 45, 46, 271).

Beck, Kent et al. (2001). *Manifesto for Agile Software Development*. URL: http://www.agilemanifesto.org/ (besucht am 09. 03. 2015) (siehe S. 1, 37, 45, 52, 149).

Belbin Associates (2011). *Team Role Summary Descriptions*. URL: http://www.belbin.com/content/page/49/BELBIN(uk)-2011-TeamRoleSummaryDescriptions.pdf (besucht am 18. 10. 2015) (siehe S. 28).

Belbin, R Meredith (1981). *Management Teams: Why They Succeed Or Fail*. Oxford: Butterworth-Heinemann (siehe S. 27, 29).

– (1993). *Team roles at work: A strategy for human resource management*. Oxford : Butterworth Heinemann (siehe S. 28, 29).

Blöbaum, Bernd, Daniel Nölleke und Andreas M. Scheu (2015). "Das Experteninterview in der Kommunikationswissenschaft". In: *Handbuch nicht standardisierte Methoden in der Kommunikationswissenschaft*. Hrsg. von Stefanie Averbeck-Lietz und Michael Meyen. Springer Fachmedien Wiesbaden. ISBN: 9783658016555 (siehe S. 84).

Boehm, Barry und Richard Turner (2003a). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley Professional (siehe S. 2).

– (2003b). "Observations on balancing discipline and agility". In: *Agile Development Conference, 2003. ADC 2003. Proceedings of the*. IEEE, S. 32–39 (siehe S. 2).

Burns, James MacGregor (1978). *Leadership*. Harper & Row, New York (siehe S. 64).

Carson, Jay B, Paul E Tesluk und Jennifer A Marrone (2007). "Shared leadership in teams: An investigation of antecedent conditions and performance". In: *Academy of management Journal* 50.5, S. 1217–1234 (siehe S. 43, 64).

Chong, Eric (2007). "Role balance and team development: A study of team role characteristics underlying high and low performing teams." In: *Journal of Behavioral & Applied Management* 8.3 (siehe S. 29, 78, 80).

Coad, Peter, Jeff de Luca und Eric Lefebvre (1999). *Java Modeling Color with Uml: Enterprise Components and Process with Cdrom*. Prentice Hall PTR (siehe S. 272).

Cockburn, Alistair (2004). *Crystal Clear. A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*. Addison-Wesley Professional. ISBN: 078-5342699470 (siehe S. 46).
– (2006). *Agile Software Development: The Cooperative Game*. 2. Aufl. Adison-Wesley (siehe S. 45).
Cohn, Mike (2004). *User Stories Applied*. Addison-Wesley Professional (siehe S. 56).
– (2005). *Agile estimating and planning*. Pearson Education (siehe S. 61).
– (2010a). *Agile Softwareentwicklung. Mit Scrum zum Erfolg!* Adison (siehe S. 40, 72, 75, 95).
– (2010b). *Succeeding with Agile. Software Development Using Scrum*. Addison Wesley. ISBN: 978-0321579362 (siehe S. 4, 38, 59).
Coleman Jr, Henry J (1999). "What enables self-organizing behavior in businesses". In: *Emergence* 1.1, S. 33–48 (siehe S. 37).
Comelli, Gerhard, Lutz Rosenstieland und Friedemann W. Nerdinger (2014). *Führung durch Motivation. Mitarbeiter für die Ziele des Unternehmens gewinnen*. Hrsg. von 5. Auflage. Vahlen. ISBN: 978-3800648399 (siehe S. 25).
DeMarco, Tom und Timothy Lister (2014). *Wien wartet auf dich! Produktive Projekte und Teams*. 3., überarbeitete und erweiterte Ausgabe. Carl Hanser Verlag (siehe S. 10).
Derby, Esther (2011). *Misconceptions about Self-Organizing Teams*. URL: http://www.estherderby.com/2011/07/misconceptions-about-self-organizing-teams-2.html (besucht am 20. 01. 2016) (siehe S. 38).
Derby, Esther, Diana Larsen und Ken Schwaber (2006). *Agile retrospectives. Making good teams great*. Pragmatic Bookshelf Raleigh, NC (siehe S. 63, 73, 272).
Dick, Rolf van und Michael A West (2013). *Teamwork, Teamdiagnose, Teamentwicklung*. Hogrefe Göttingen (siehe S. 29).
Eberhardt, Daniela, Hrsg. (2013). *Together is better? Die Magie der Teamarbeit entschlüsseln*. Springer Berlin Heidelberg. ISBN: 978-3-642-34436-7. DOI: 10.1007/978-3-642-34437-4 (siehe S. 8, 10).
Edeki, Charles (2013). "Agile Unified Process". In: *International Journal of Computer Science* 1.3 (siehe S. 46).
Eoyang, Glenda Holladay (2001). "Conditions for selforganizing in human systems". Diss. The Union Institute und University (siehe S. 40, 42).

Francis, Dave und Don Young (1996). *Mehr Erfolg im Team. Ein Trainingsprogramm mit 46 Übungen zur Verbesserung der Leistungsfähigkeit in Arbeitsgruppen*. Windmühle (siehe S. 9, 10, 19–21, 23, 25).

Frank, Thomas A. (2010). *Schnelleinstieg Psychologie im Unternehmen*. Haufe Praxisratgeber. Freiburg: Haufe-Lexware GmbH & Co. KG. ISBN: 978-3-448-10128-7 (siehe S. 18).

Gabler, Hrsg. (2015). *Gabler Wirtschaftslexikon - Definition Gruppe*. URL: `http://wirtschaftslexikon.gabler.de/Definition/gruppe.html?referenceKeywordName=formelle+Gruppe` (besucht am 15. 09. 2015) (siehe S. 8).

Gerrig, Richard J. und Philip G. Zimbardo (2015). *Psychologie*. Pearson. ISBN: 978-3-86894-213-2 (siehe S. 11, 12, 26).

Gessler, Michael, Hrsg. (2012). *Kompetenzbasiertes Projektmanagement (PM3). Handbuch für die Projektarbeit, Qualifizierung und Zertifizierung*. 5. Aufl. Deutsche Gesellschaft für Projektmanagement, Nürnberg. ISBN: 9783924841409 (siehe S. 2, 12, 16, 20, 29).

Gius, Evelyn (2015). *Erzählen über Konflikte: Ein Beitrag zur digitalen Narratologie*. Narratologia. De Gruyter. ISBN: 9783110422405 (siehe S. 15).

Gläser, Jochen und Grit Laudel (2010). *Experteninterviews und qualitative Inhaltsanalyse: als Instrumente rekonstruierender Untersuchungen*. 4. Auflage. VS Verlag für Sozialwissenschaften. ISBN: 3531172387 (siehe S. 86).

Gloger, Boris und André Häusling (2012). *Erfolgreich mit Scrum - Einflussfaktor Personalmanagement. Finden und Binden von Mitarbeitern in agilen Unternehmen*. Carl Hanser Verlag München. ISBN: 978-3-446-42515-6 (siehe S. 47, 71, 74).

Gloger, Boris und Dieter Rösner (2014). *Selbstorganisation braucht Führung: die einfachen Geheimnisse agilen Managements*. Carl Hanser Verlag GmbH Co KG (siehe S. 20, 24, 39, 42).

Greenleaf, Robert K. (1977). *Servant leadership. A journey into the nature of legitimate power and greatess*. New York: Paulist Press (siehe S. 58, 272).

Hackman, J Richard und Greg R Oldham (1976). "Motivation through the design of work: Test of a theory". In: *Organizational behavior and human performance* 16.2, S. 250–279 (siehe S. 91, 199).

Hildebrandt, Achim et al. (2015). *Methodologie, Methoden, Forschungsdesign. Ein Lehrbuch für fortgeschrittene Studierende der Politikwissenschaft*. Springer Fachmedien Wiesbaden. ISBN: 978-3-531-18256-8. DOI: 10.1007/978-3-531-18993-2 (siehe S. 84).

HolacracyOne, LLC., Hrsg. (2015). *HolacracyOne, LLC.* URL: http://holacracy.org (besucht am 08. 08. 2015) (siehe S. 1).

Huber, Eberhard und Sven Lindenhahn (2010). "Teamwork: Warum Projektteams erfolgreicher sind als Projektgruppen". In: *OBJEKTspektrum* 2, S. 10–16 (siehe S. 9).

Janis, Irving Lester (1972). *Victims of Groupthink*. New York: Houghton Mifflin (siehe S. 2, 12).

– (1982). *Groupthink: Psychological studies of policy decisions and fiascoes*. Houghton Mifflin Boston (siehe S. 2, 12).

Kassl, Siegfried (2000). "Expertenbefragung und Experteninterview als empirische Forschungsmethoden am Beispiel von Diplomarbeiten". Diplomarbeit (siehe S. 4).

Kauffeld, Simone (2001). *Teamdiagnose*. Verlag für Angewandte Psychologie. ISBN: 9783801714123 (siehe S. 7–9, 19, 21, 25, 27, 29, 30).

Kotter, John und Holger Rathgeber (2011). *Das Pinguin-Prinzip. Wie Veränderung zum Erfolg führt*. Droemer HC. ISBN: 978-3426275726 (siehe S. 40).

Krüger, Wolfgang (2013). *Teams führen*. Haufe Verlag (siehe S. 10).

Kühl, Stefan, Petra Strodtholz und Andreas Taffertshofer, Hrsg. (2009). *Handbuch Methoden der Organisationsforschung. Quantitative und Qualitative Methoden*. VS Verlag für Sozialwissenschaften. ISBN: 978-3-531-15827-3. DOI: 10.1007/978-3-531-91570-8 (siehe S. 4, 84).

Kuster, Jürg et al. (2008). *Handbuch Projektmanagement*. Springer Berlin Heidelberg. ISBN: 978-3-540-76431-1. DOI: 10.1007/978-3-540-76432-8 (siehe S. 18, 19, 133).

Larman, Craig (2001). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall PTR, Upper Saddle River, NJ (siehe S. 46).

Latané, Bibb, Kipling Williams und Stephen Harkins (1979). "Many Hands Make Light the Work:The Causes and Consequences of Social Loafing". In: *Journal of Personality and Social Psychology* 37.6, S. 822–832 (siehe S. 14).

Lauer, Thomas (2014). *Change Management. Grundlagen und Erfolgsfaktoren*. Springer Berlin Heidelberg. ISBN: 978-3-662-43736-0. DOI: 10.1007/978-3-662-43737-7 (siehe S. 40).

Liebold, Renate und Rainer Trinczek (2009). In: *Handbuch Methoden der Organisationsforschung. Quantitative und Qualitative Methoden*. Hrsg. von Stefan Kühl, Petra Strodtholz und Andreas Taffertshofer. VS Verlag für

Sozialwissenschaften. Kap. Experteninterview, S. 32–56. ISBN: 978-3-531-15827-3. DOI: 10.1007/978-3-531-91570-8 (siehe S. 83, 84).

Lindvall, Mikael et al. (2002). "Empirical Findings in Agile Methods". English. In: *Extreme Programming and Agile Methods — XP/Agile Universe 2002*. Hrsg. von Don Wells und Laurie Williams. Bd. 2418. Lecture Notes in Computer Science. Springer Berlin Heidelberg, S. 197–207. ISBN: 978-3-540-44024-6. DOI: 10.1007/3-540-45672-4_19 (siehe S. 2).

Mai, Jochen (2009). *Die Karriere-Bibel: Definitiv alles, was Sie für Ihren beruflichen Erfolg wissen müssen*. Deutscher Taschenbuch Verlag. ISBN: 9783423400732 (siehe S. 28).

Margerison, Charles J. (2002). *Team Leadership: A Guide to Success with Team Management Systems*. Cengage Learning Emea. ISBN: 9781861528636 (siehe S. 27, 273).

Maximini, Dominik (2015). *The Scrum Culture. Introducing Agile Methods in Organizations*. Management for Professionals. Springer International Publishing. ISBN: 978-3-319-11826-0. DOI: 10.1007/978-3-319-11827-7 (siehe S. 2, 45).

McHugh, Orla, Kieran Conboy und Michael Lang (2012). "Agile practices: The impact on trust in software project teams". In: *Software, IEEE* 29.3, S. 71–76 (siehe S. 49).

Moreira, Mario E. (2010). *Adapting Configuration Management for Agile Teams. Balancing Sustainability and Speed*. Wiley. ISBN: 9780470746639 (siehe S. 51).

Müller, E.W. (2014). *Konfliktmanagement: So lösen Sie Konflikte und verbessern das Betriebs- und Arbeitsklima!* Handbuch deutsches Arbeitsschutz-Recht. ecomed Verlagsgesellschaft in Hüthig Jehle Rehm. ISBN: 9783609619286 (siehe S. 19).

Nebulon Pty. Ltd (2015). URL: http://www.nebulon.com (besucht am 17. 09. 2015) (siehe S. 272).

Nerdinger, Friedemann W., Gerhard Blickle und Niclas Schaper (2014). *Arbeits- und Organisationspsychologie*. Springer-Lehrbuch. Springer Berlin Heidelberg. ISBN: 978-3-642-41129-8. DOI: 10.1007/978-3-642-41130-4 (siehe S. 8, 9, 12, 20).

Nerur, Sridhar, RadhaKanta Mahapatra und George Mangalaraj (2005). "Challenges of migrating to agile methodologies". In: *Communications of the ACM* 48.5, S. 72–78 (siehe S. 46, 47).

Niermeyer, Rainer (2012). *Teams führen*. Freiburg: Haufe-Lexware GmbH & Co. KG, S. 221–. URL: https://www.wiso-net.de/document/HAUF__9783648024560221 (siehe S. 10).

Ōno, Taiichi (1988). *Toyota Production System: Beyond Large-Scale Production*. Taylor & Francis. ISBN: 9780915299140 (siehe S. 49).

Pixton, Pollyanna, Paul Gibson und Niel Nickolaisen (2014). *The Agile Culture: Leading through Trust and Ownership*. Adison-Wesley (siehe S. 49).

Poppendieck, Mary und Tom Poppendieck (2006). *Implementing Lean Software Development. From Concept to Cash*. Addison-Wesley. ISBN: 978-0321437389 (siehe S. 51).

Pröpper, Nils (2012). *Agile Techniken für klassisches Projektmanagement: Qualifizierung zum PMI-ACP*. EAN 9783826692222. MITP Verlags GmbH (siehe S. 1).

Reddy, Ajay (2015). *The Scrumban [R]Evolution. Getting the Most Out of Agile, Scrum, and Lean Kanban*. Addison-Wesley. ISBN: 978-0134086217 (siehe S. 51).

Rising, Linda und Norman S. Janoff (2000). "The Scrum Software Development Process for Small Teams". In: *IEEE Software* 17.4, S. 26–32. ISSN: 0740-7459. DOI: 10.1109/52.854065 (siehe S. 46).

Rüttinger, Bruno (1980). *Konflikt und Konfliktlösen*. Bratt-Institut für Neues Lernen. ISBN: 9783885980056. URL: https://books.google.de/books?id=PeV1mwEACAAJ (siehe S. 16, 17).

Schaeffer, Ingo Schulz (1996). "Software Entwicklung zwischen Ingenieur und Designwissenschaft Überzeugungskraft und nützliche Widersprüchlichkeit von Software-Engineering und Software-Gestaltung". In: *Technikleitbilder auf dem Prüfstand. Leitbild-Assessment aus Sicht der Informatik- und Computergeschicht*. Sigma. Berlin: Hans Dieter Hellige, S. 115–140 (siehe S. 1).

Schmitt, Bernd H. et al. (1986). "Mere presence and social facilitation: One more time". In: *Journal of Experimental Social Psychology* 22 (siehe S. 14).

Schwaber, Ken (1997). "Scrum development process". In: *Business Object Design and Implementation*. Springer, S. 117–134 (siehe S. 53).

– (2004). *Agile project management with Scrum*. Microsoft Press (siehe S. 53).

– (2008). *Scrum im Unternehmen*. Microsoft Press (siehe S. 54).

Schwaber, Ken und Mike Beedle (2001). *Agile Software Development with Scrum*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR. ISBN: 0130676349 (siehe S. 2, 53).

Schwaber, Ken und Jeff Sutherland (2013). *Der Scrum Guide*. URL: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf (besucht am 26. 08. 2015) (siehe S. 52, 54, 57–59, 74, 75).

ScrumAlliance (2015). *ScrumAlliance Homepage*. URL: https://www.scrumalliance.org (besucht am 25. 09. 2015) (siehe S. 54).

ScrumGuides.org (2015). *The History of Scrum*. URL: http://www.scrumguides.org/history.html (besucht am 26. 08. 2015) (siehe S. 54).

Sendjaya, Sen (2015). *Personal and Organizational Excellence Through Servant Leadership*. Springer (siehe S. 273).

Spencer, J. und A. Pruss (1995). *Top-Teams: der Königsweg zu mehr Flexibilität, Effizienz und Erfolg im Betrieb*. Droemer, Knaur. ISBN: 9783426830208 (siehe S. 27, 30, 31).

Sprenger, Reinhard K (2009). *Mythos Motivation. Wege aus einer Sackgasse*. Campus Verlag. ISBN: 978-3-593-39000-0 (siehe S. 43).

Steinmann, Horst und Georg Schreyögg (2005). *Management. Grundlagen der Unternehmensführung*. Gabler. ISBN: 978-3409633123 (siehe S. 8, 11, 12, 14, 20, 26, 50, 64).

Stellman, Andrew und Jennifer Greene (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly Media, Inc. (siehe S. 64).

Stöwe, Christian und Lara Keromosemito (2013). *Führen ohne Hierarchie - Laterale Führung. Wie Sie ohne Vorgesetztenfunktion Teams motivieren, kritische Gespräche führen, Konflikte lösen*. Springer Fachmedien Wiesbaden. ISBN: 978-3-8349-3975-3. DOI: 10.1007/978-3-8349-3976-0 (siehe S. 17).

Sturm, Alexandra, Ilga Opterbeck und Jochen Gurt (2011). *Organisationspsychologie*. VS Verlag für Sozialwissenschaften. ISBN: 978-3-531-16725-1. DOI: 10.1007/978-3-531-93074-9 (siehe S. 16).

Sutherland, Jeff (2004). *Agile Development: Lessons Learned from the First Scrum*. online. URL: https://www.scrumalliance.org/resources/35 (besucht am 25. 09. 2015) (siehe S. 53).

Takeuchi, Hirotaka und Ikujiro Nonaka (1986). "The new new product development game". In: *Harvard business review* 64.1, S. 137–146 (siehe S. 53).

TMS-Zentrum (2015). URL: http://www.tms-zentrum.de/ (besucht am 23. 08. 2015) (siehe S. 273).

Tscheuschner Marc andWagner, Hartmut von (2012). *30 Minuten TMS - Team Management System*. ABAL-Verlag. ISBN: 978-3-86200-732-5 (siehe S. 27, 273).

Tscheuschner, Marc von und Hartmut Wagner (2008). *Das Team Management System. Der Weg zum Hochleistungsteam*. Offenbach: GABAL Verlag. ISBN: 978-3-89749-794-8 (siehe S. 27, 273).

Tuckman, Bruce W und Mary Ann C Jensen (1977). "Stages of small-group development revisited". In: *Group & Organization Management* 2.4, S. 419–427 (siehe S. 2, 19, 20, 35, 41, 131).

Turk, Dan, Robert France und Bernhard Rumpe (2005). "Assumptions underlying agile software development processes". In: *Journal of Database Managemen* 16.4, S. 62–87 (siehe S. 48).

Wiendieck, Gerd (2004). In: *Handwörterbuch Unternehmensführung und Organisation*. Hrsg. von Georg Schreyögg und Axel v. Werder. 4. Auflage. Schäffer-Poeschel Verlag Stuttgart. Kap. Gruppenverhalten und Gruppendenken, S. 388–398 (siehe S. 13, 15).

Winterhoff-Spurk, Peter (2002). *Organisationspsychologie. Eine Einführung*. Kohlhammer, Stuttgart. ISBN: 978-3-17-017107-7 (siehe S. 8, 10, 16, 17).

Wirdemann, Ralf (2011). *Scrum mit User Stories*. 2. Auflage. Hanser (siehe S. 56).

Witt-Bartsch, Angela und Thomas Becker, Hrsg. (2010). *Coaching im Unternehmen. Die individuellste und nachhaltigste Form der Personalentwicklung und Mitarbeiterführung*. Haufe Verlag (siehe S. 17).

Wolf, Henning, Hrsg. (2015). *Agile Projekte mit Scrum, XP und Kanban. Erfahrungsberichte aus der Praxis*. 2., aktualisierte Auflage. dpunkt.verlag GmbH (siehe S. 72).

A

Anhang A

Das agile Manifest

Das agile Manifest in deutscher Übersetzung (Beck et al. 2001) als vollständiges Zitat:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Individuen und Interaktionen mehr als Prozesse und Werkzeuge
Funktionierende Software mehr als umfassende Dokumentation
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
Reagieren auf Veränderung mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein."

Die Prinzipien hinter dem agilen Manifest

Der folgende Wortlaut ist ein vollständiges Zitat welches das agile Manifest (Beck et al. 2001) ergänzt:

„Wir folgen diesen Prinzipien:

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.

2. Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

“

B

Anhang B

Interviewleitfaden

Vorwort

Das Ziel der Untersuchung ist, problematische Entwicklungen in agilen Teams frühzeitig zu diagnostizieren. Mit diesem Wissen können Verantwortliche und Coaches frühzeitig gegensteuern. Das Ziel des Interviews ist es, Praxiswissen über Art und Erscheinungsform von negative Entwicklungen zu sammeln.

Folgende Dinge sind im Vorfeld des Interviews mitzuteilen und die Einverständnis des Interviewpartners einzuholen

- Besteht Einverständnis zur Tonaufzeichnung?
- Wie steht der Interviewer zur Veröffentlichung seiner Daten?

Weitere eingesetzte Mittel Das Interview wird begleitet von einer Aufzeichnung in Papierform, die für den Interviewpartner stets sichtbar ist. Diese wird genutzt, um eine anschauliche Diskussionsgrundlage zu haben, um eine Art von Landkarte zu erstellen, mit der es im weiteren Verlauf des Interviews möglich ist, diese Punkte wieder aufzugreifen oder nicht relevante Punkte auszusortieren. Dies ist besonders dann von Interesse, wenn es um die negativen Entwicklungen geht. Diese Landkarte soll auch genutzt werden, um eine Einschätzung von den erfahrenen Partnern zu bekommen, welche Punkte sie für besonders wichtig halten.

Einleitende Fragen

Wie lange befassen Sie sich schon mit agilen Teams?

- Fachexpertise erfassen

Welche agilen Aufgaben bzw. Rollen wurden schon übernommen?

- Fachexpertise erfassen
- Klären, welche Rollen schon eingenommen wurden

Negative Entwicklungen

Was kann sich negativ auf die Leistung eines agilen Teams auswirken?

- Erstellung der Landkarte beginnen
- Ist eine einleitende Frage um auf das Thema zu kommen und die Einstellung des Gegenüber kennen zu lernen.
- Bei der Aufzeichnung eine Priorisierung des Partners durchführen lassen.
- Vorsicht, hier sind auch externe Faktoren von belang, die uns in der Arbeit aber nicht interessieren.

Welche Probleme im Team haben Sie schon beobachtet?

- Welche negativen Entwicklungen können in einem agilen Team auftreten?
- Darauf achten, das er hier um innere Entwicklungen im Team geht.
- Ideen zum Ansprechen, wenn nicht viel Information kommt (Vorsicht nur als letztes Mittel einsetzen, da dies das Gegenüber beeinflusst):
  - Konflikte im Team
  - Wurden schon einmal Rollen infrage gestellt? Warum?
  - Trittbrettfahren (wenige machen die Arbeit, andere hängen sich dran)
  - Risikoschub
  - fehlendes Wissen / Erfahrung mit agilen Methoden
  - Selbstorganisation

Welche Situationen bzw. Konstellationen im Team kennen Sie, bei denen die Scrum Elemente zur Selbstorganisation versagten?

- CDE-Modell (Container, Differences, Exchange)
- Schleichende Veränderung oder Organisationsproblem?
- Probleme mit Rollen bzw. starken Persönlichkeiten im Team?

Was haben Sie in diesen Situationen in ihrer Rolle getan, um diese Probleme aufzulösen?

- Daten für die Zusammenfassung und den Ausblick sammeln.
- Die Ursache ergründen, was an der konkreten Situation das Problem war und wie sich solche Probleme beheben lassen.

Diagnostik

Wie bemerken Sie, das im Team etwas nicht stimmt?

- Möglichkeiten zur Diagnose finden
- Vorgehensweise des Befragten ergründen
- Worauf achtet der Befragte?
- Wie lange dauert es, bis problematische Entwicklungen bemerkt werden?
  - Wie schnell geht die Diagnose?
  - Ist diese Art der Diagnose geeignet?
- Ergründen, was dagegen getan wurde

(Für Coaches) Wie gehen Sie vor, wenn sie zu einem Coaching gerufen werden?

- Coaches sind normalerweise strukturiert, da sie sich schnell einen Überblick verschaffen müssen.
- Welche Instrumente setzen Sie ein?
- Kann das Team mit dem Scrum-Master einen Coach ersetzen bzw. wo sind die Grenzen von einem Scrum-Team?

Welche Situationen sind Ihnen bekannt, in denen das Team ein Problem nicht erkennt?

- Ergründen, welche Probleme dies sind
- Ergründen, was an diesen Situationen speziell ist: Rahmenbedingungen erfragen.
- Ergründen, was dagegen getan wurde
- Einsatz der Landkarte erwägen.

Wie unterstützt Sie das Scrum-Framework bei dieser Tätigkeit?

- Ergründen, in wie weit das Framework hilft.

- Diese Frage auch im Zusammenhang mit der nächsten sehen.
- Einsatz der Landkarte erwägen.

> *Wo gibt es aus ihrer Sicht Grenzen des Scrum-Frameworks bzw. dessen praktischer Ausübung hinsichtlich dem Erkennen von Problemen?*

Im Scrum fest verankert ist die Retrospektive nach jedem Sprint. Es ist interessant zu erfahren, ob dieses Mittel alleinig ausreicht, um einen Dauerhaften Teamerfolg zu sichern, oder ob der Einfluss von außen ebenso wichtig wie notwendig ist.

- Punkte suchen, auf die eine Diagnose besonderen Wert legen muss, da sie vom Team nicht erkannt werden können.
- Bei den Lücken nachfragen, was dagegen unternommen wurde.
- Lücken eignen sich zur weiteren Diagnose, denn dies sind die Stellen, auf die besonders zu achten sind.

Rollen bei Scrum

Rollen in Scrum sind explizit benannt, das hilft bei der Orientierung. Da die meisten diese Rollen schon ausgefüllt haben, ist es eine gute Möglichkeit, das Gegenüber in eine Situation zu versetzen und dies für die Exploration zu nutzten. Dei Folgenden Fragen sind optional, können jedoch dann helfen, wenn der Befragte wenig Feedback geben konnte. Sie eröffnen einen neuen Blickwinkel, der auf die Ich-Perspektive direkt abzielt und etwas konkreter sind.

> *Wann waren Sie als Coach / Scrum-Master besonders gefragt bzw. gefordert?*

Diese Frage versucht, konkrete Situationen oder Beispiele durch den Befragten zu bekommen.

- Erfahrung des Gegenübers erkunden und Beispiele dafür nutzen.
- Was ist in dieser Situation schlecht gelaufen?
- Was war die mögliche Ursache?
- Wie wurde das Problem erkannt?

> *Die Rolle des Scrum-Masters / Product-Owners ist eine wichtige Rolle, was sind die Probleme mit dieser Rolle ihrer Erfahrung nach?*

Diese Frage versucht die Vorstellung des Gegenübers anzuregen, was in den Rollen „abstrakt“ auftreten kann, ohne die eigene Person in den Fokus zu setzten. Dies erweitert die Suche so, das nicht nur konkrete Situationen interessant sind, sondern die Erfahrung, nach welchem der Interviewte in seiner Arbeit vor geht.

- Ergründen, welche Probleme in Zusammenhang mit dieser Rolle auftreten können.
- Ergründen, ob Überlastung ein Problem sein kann.

C

Anhang C

Transkriptionen der Interviews

C.1 Interviewtranskription vom 1. Interview

Datum 21.4.2016

Fachlicher Hintergrund Der Interviewpartner arbeitet im Wesentlichen als Teammitglied in Software-Entwicklungsteams mit, die Software für den stark regulierten, medizinischen Markt entwickeln. Dabei hat er schon unterschiedliche Teams kennengelernt und befasst sich schon seit ca. 8 Jahren mit dem Thema Agilität. Im Rahmen der Tätigkeit wurde auch die Schulung zum zertifizierten Scrum-Master abgelegt. Weiterhin gibt er Schulungen zu diversen Software-Themen.

Interview

Anmerkungen: Genannte Personen und Unternehmen bzw. Projekte wurden anonymisiert durch UnA, UnB und UnIv (Unternehmen des Befragten) für die Unternehmen. Personen wurden durch Person A, Person B, Person C, Person D, Person E, Person F, Person G und Person H ersetzt. Projekte wurden durch PRO ersetzt.

Interviewer: Hallo, als Erstes wollte ich von dir einmal wissen, wie lange du schon mit agilen Teams bzw. agiler Entwicklung / Umgebungen zu tun hattest und was du dort schon getan hattest, um deinen Hintergrund zu erfahren.

Befragter: Ich arbeite ja in unserem UnA-Projekt für UnA und mache Software und haben verschiedene Vorgehensmodelle schon einmal durchprobiert und sind ungefähr seit, ich bin nicht mehr ganz sicher, aber ganz grob seit 2008 oder 9 in die Agilität eingestiegen, haben also dann (*Anm.: jetzt ist 2016*) haben dann Scrum ausprobiert, uns eingelesen und das mal ausprobiert und haben das dann so nach und nach verbessert, was eigentlich so im Sinne von Agilität ist, Scrum-Schulung gegönnt und dann Sachen noch einmal anders, nämlich richtiger gemacht und des gab so verschiedene Phasen im Projekt, wo die Arbeit ganz unterschiedlich gelaufen ist, weil es mal eine Eigenentwicklung war, die man zugeliefert hat und abgegeben hat und später war es dann mehr so Zulieferung von einzelnen Paketen in den verschiedenen Phasen waren dann auch verschiedene Vorgehensmodelle unterschiedlich nützlich. Und da, wo wir sozusagen ein eigenes Produkt erstellt ist eigentlich Scrum sehr sehr gut geeignet, um die Arbeit zu planen und

zu priorisieren, außer wenn die Rahmenbedingungen so sind, dass alles fest vorgegeben ist (lachen). .

Interviewer: Das stimmt, ja.

Befragter: Das macht es dann etwas schwieriger, es ist dann aber immer noch nützlich, um einen Fortschritt zu beobachten. Das ist allerdings auch nicht die einzige agile Methode, sondern wir haben dann auch mal Kanban ausprobiert im Projekt, da warst du schon dabei, das hat sich in der Konstellation da nicht bewährt, wobei das sicherlich nicht an der Sache an sich liegt, sondern an der Umsetzung oder vielleicht war es auch nicht so ganz passend in dem Moment, da kommt auch wieder der Aspekt mit rein, da haben wir eben auch gemeinsam gelernt, dass es Situationen gibt, in denen so eine Technik ganz gut passt und wo sie eben nicht ganz gut passt und das man eigentlich für die verschiedenen Phasen in der Entwicklung vielleicht, dass es da Sinn macht zu identifizieren, in welcher Phase man jetzt so oder so arbeiten sollte am Besten und schaun wie man sich da unterstützt. Ja und was ich da ganz gelernt habe aus verschiedenen theoretischen Quellen und zumindest mit meiner bisherigen unterschiedlichen Erfahrung da nicht widersprechen kann ist, dass der Kern von agilen Vorgehen der kleinste Kern ist, das die Inspektion und Maßnahmen ergreifen und stetige Verbesserung also quasi Stichwort Retrospektiven, wie auch immer man es nennt, also das regelmäßige Reflektieren über das, wie es läuft und daraus ableiten, was man jetzt irgendwie anders oder besser machen kann oder beibehalten kann, verstärken kann je nach dem. Inspect and Adapt. Machen wir jetzt zur Zeit aktuell auch auf Metaebene bei UnIv, also macht man ja natürlich sowieso ständig, wenn man gut arbeitet; das fällt mir jetzt gerade so ein, weil ich darüber spreche; wir haben halt festgestellt, dass in unserer Zeiterfassungsverwaltung manche Sachen irgendwie nicht so optimal laufen, dass man da halt über mit dem Tooling, was wir da haben manches nicht gut sehen, zuordnen können also das Controllling sagt da, hmm, „fehlt mir Infos“ und Inspect and Adapt okay, was ist das Problem, woran liegts was können wir besser machen, heißt wir müssen in unserem Fall irgendwas mit Projektverfolgungs- / Buchungspositionen oder irgendwie besser werden. Es wurden schon Maßnahmen abgeleitet, aber auf jeden Fall ist es auch so ein Beispiel.

Interviewer: Ja.

Befragter: Du kennst dich doch mit Ägilität aus, Inspect and Adapt.

Interviewer: Ok, ich denke zu einigen Sachen kommen wir später noch einmal, weil du jetzt schon vieles gesagt hast, vielleicht noch einmal so zum An-

fang, wenn man einmal direkt nach Scrum sieht, dort hast du ja schon Rollen übernommen.

Befragter: (Zustimmendes hmm)

Interviewer: Ich weiß, dass du mal Scrum-Master gemacht hast.

Befragter: Ja

Interviewer: Hast du noch mehr gemacht in diesem ganzen Umfeld oder du hast doch die Schulungen dazu.

Befragter: Ich habe die Scrum-Master-Schulung gemacht, die certified Scrum-Master von einem der Anbieter von denen es inzwischen ganz viele gibt und hab natürlich die Rolle des Scrum-Developers inne immer mal, weil man ja als Entwickler in einem Scrum-Team diese Rolle hat, ob man nun gleichzeitig Scrum-Master ist oder nicht. Das war auch die einzige Scrum-Schulung dazu, außer natürlich die diversen Literatur-Sachen zu diesem ganzen Themengebiet, die ich mir reingezogen hab im Laufe der Zeit; ich denke die Rolle des Product-Owners habe ich noch nicht so richtig explizit gehabt, aber dadurch, dass ich bei uns im Thema der also in unserem UnIv-Team der PCO, der Projektkoordinator und mich und meine zwei Kollegen da entsprechend verwalten muss, Aussteuern, wie auch immer leiten, ist auch manchmal so ein bisschen Product-Owner-Proxy Tätigkeit dabei, das Product-Owner-Proxy[T1.1] heißt eben, es gibt jemand anderes, der der Product-Owner ist. Es ist ein ganz, ganz typisches, häufig anzutreffendes, ähm, normalerweise Anti-Pattern[T1.2], aber es muss auch nicht unbedingt ein Anti-Pattern sein, weil es kann auch einfach sein, dass das der einzigste Weg ist, wie man das hinbekommt.

Interviewer: Hmm.

Befragter: Der Product-Owner in unserem Umfeld ist ja im Prinzip, ist natürlich bei UnA das Product-Management im Prinzip, also das ist die offizielle Stelle, die da hingehört und die klassischen Rollen sind dabei ja so ein bisschen anders als bei Scrum, also der Product-Owner bei Scrum, der ist irgendwie total wichtig und zentral und er muss irgendwie auch so ins Technische rein blicken können und auch ins Fachliche, vor allem ins Fachliche, also das ist das Allerwichtigste und das Technische wäre halt schön, aber es geht eben auch ohne und eigentlich ist das Produktmanagement, mit dem wir zur Zeit ganz viel zu tun haben, nämlich die Person A in der Rolle eigentlich sehr sehr gut wie ich finde, auch wenn sie das vielleicht auch

gar nicht weiß, dass sie der Product-Owner ist. Aber sie lebt diese Rolle eigentlich erstaunlich gut und interessiert sich auch immer für die technischen Aspekte, wenn man sagt, „Nee so kann man das nicht umsetzen.“, dann heißt es immer „Warum?“, dann erklärt man es ihr und dann sagt sie „OK“, dann anders.

Interviewer: Ich werde ihnen diesen Ausschnitt mitteilen.

Befragter: Ja, den darfst du ihr sehr, sehr gerne mitteilen. Wir sind sehr zufrieden mit der Person A.

Interviewer: Eine weitere Frage, hast du schon einmal für andere Teams, also du hast ja eigentlich immer für UnA gearbeitet, aber hast du für andere Teams auch schon Rollen übernommen oder da irgendwas gemacht für sie oder wirklich nur in diesem einen Team? Oder dazu geholfen?

Befragter: Also direkt in einer Scrum-Rolle, ich habe mal bei einem anderen Kunden eine Woche lang einen Kollegen vertreten, auch Medizintechnik, ähnliches Umfeld, dann war ich einmal eine Wochen in Hamburg und dort sollte ich eigentlich primär einfach irgendwas mit bezüglich Testen, Testspezifikation, Review, ein paar Tests durchführen und so Sachen, die hatten aber auch gerade eben auch ihre Scrum und waren gerade kurz nach ihrer Scrum-Einführung oder waren innerhalb dieser Transition zum Scrum hin und das war sehr kurz nach meiner, etwa ein halbes Jahr nach meiner Scrum-Master-Schulung, wo ich natürlich noch ganz begeistert und unerfahren, aber das war dort gerade aktuell ein heißes Thema und dann haben wir uns das natürlich auch dort noch einmal beobachtet und haben dort wiederum ähnliche, aber auch andere Probleme festgestellt als bei UnA, z.B., dass dort vom Management her durchaus alles, das kam von oben her, das man Scrum machen soll und es gab einen Scrum-Master, der das Ganze ganz motiviert vorangetrieben hat und interessanterweise war die Akzeptanz in Teilen des Teams überhaupt nicht gegeben. Das ist bei uns eher, im UnA Umfeld anders ist.

Interviewer: Dann nehme ich das nächste Thema auf, und zwar geht es darum, negative Entwicklungen zu analysieren, zu sehen, was es so gibt und zwar erst einmal grundsätzlich. Dafür kannst du auch das Blatt Papier nutzen. „Was kann sich eigentlich negativ auf die Leistung von einem agilen Team, was dir da persönlich so einfällt?“. Ich darf dir auch die kleinen Zettelchen (*Anm.: Haftnotizen*) geben, einfach was fällt dir spontan ein, was es für negative Einflüsse im Team oder von außen, wobei mich eher das im Team interessiert.

Befragter: Also, so Sachen, wie organisatorische Rahmenbedingungen die natürlich sehr viel ausmachen, sind jetzt nicht so wichtig.

Interviewer: Die sind jetzt eher weniger wichtig.

Befragter: Ich brauch doch große Zettel, ich schreib groß.

Interviewer: Ja, dann große Zettel.

Befragter: Die sind aber trotzdem ganz wichtig, ich schreibe es trotzdem auf, also negative Auswirkung. Auf jeden Fall das Thema negative Entwicklungen?

Interviewer: Naja, was kann sich allgemein negativ auf solche ein Team auswirken, also die Frage ist halt, wie definierst du Leistung, ...

Befragter: Erst mal negative Einflüsse, Einflüsse, die sich negativ auswirken auf die Leistung, richtig?

Interviewer: Richtig, genau.

Befragter: (*Anm.: schreibt auf das Blatt*) Schlechte Kommunikation[T1.3], wobei schlecht immer noch so ein schwammiger Begriff ist, also, es ist auch ein weites Feld (*Anm.: schreibt auf Zettel*), ganz grob gesagt heißt das einfach, dass Information die irgendjemand im Team braucht und jemand anderes im Team hat, nicht rüberkommt, das ist das Eine, also auf der fachlichen Ebene[T1.4] und das Gleiche gibt es natürlich auf der persönlichen Ebene[T1.5] (*Anm.: schreibt weiteren Zettel*), das ist dann so was, wenn zwischenmenschlich irgendwelche Probleme sind, wobei ich weiß gar nicht, ob das wie viel man damit besserer Kommunikation beheben kann, ja, doch natürlich da gibts, das sind wieder andere Aspekte drin, dass man halt irgendwelche Sachen zum falschen Zeitpunkt anspricht oder auf die falsche Art anspricht oder nicht anspricht, obwohl man sie ansprechen müsste. Soll ich das auch alles aufschrieben?

Interviewer: Du bist ganz frei, ich lass dich da jetzt einmal gestalten.

Befragter: Achso, du hast ja eh die Tonspur.

Interviewer: Genau. Ich habe die Tonspur, ich das ist für mich so eine Art Landkarte, wo du dir einfach mal, was irgendwie auch so die Orientierung erleichtern kann. Ja, also ich werde auf die Punkte wieder zurückkommen oder wir werden sie vielleicht noch umgruppieren.

Befragter: Ja

Interviewer: Einfach einmal um Sachen, die du so siehst, weil ich sehe ja manches anders, ja aber du siehst es ja auch aus anderer Erfahrung.

Befragter: Ja, genau.

Interviewer: Was dir aus der Praxis einfällt. Vielleicht auch, ich habe noch eine weitere Frage an dich: Welche Probleme im Team hast du schon einmal beobachtet? Das ist die Frage, nur anders gestellt, vielleicht hilft dir das, damit du besser etwas findest. Oder denke an Situationen, die du erlebt hast in der Zeit, was du gelöst hast.

Befragter: Genau, also das war das Eine. Kommunikation. Da ist es ja so, das solche Probleme Einflüsse, die sich negativ auf die Leistung des Teams auswirken, dass hat grundsätzlich überhaupt nichts mit Scrum zu tun. Sondern das ist einfach generell.

Interviewer: Das stimmt, ja. Gibt es vielleicht irgendwelche Scrum-spezifischen Probleme die auftreten können?

Befragter: Also hmm, spezifisch ist es dann so, dass die Sachen, die generell irgendwie ein Problem sind halt bei Scrum einfach sich ein bisschen anders manifestieren. Was macht denn Scrum aus? Scrum ist ja ein Projektmanagement-Framework, das einfach ein paar Rollen und Aktivitäten definiert und die Probleme sind üblicherweise da, wenn die Rollen nicht richtig funktionieren, also Aufgaben nicht wahrnehmen können oder tun und das ist dann, oder eben wenn die Aktivitäten schlecht laufen oder falsch laufen; Also die Rollen sind natürlich der Product-Owner, der Scrum-Master und das Team, die anderen Leute und der Product-Owner ist eigentlich nach meiner bisherigen Erfahrung auch von dem, was ich mich mit anderen Leuten so ausgetauscht habe, oft immer ein Problem, weil diese Rolle sehr sehr wichtig ist und bei einer solchen Transition zu einem Scrum Vorgehen von einem nicht Scrum Vorgehen schwierig ist, da jemanden in diese Position da zu setzen oder weil das oft nicht gemacht wird oder es lang dauert, bis das Verständnis da ist. Andererseits habe ich jetzt auch gelernt direkt in unserer Teamerfahrung, dass man auch durchaus, wie ich finde für begrenzte Aufgaben, also innerhalb eines größeren Projekts, innerhalb eines Teams, das mehrerer Dinge parallel oder nacheinander abarbeitet, das man da durchaus für bestimmte Aufgabe auch bestimmte Leute begrenzt als Product-Owner definieren kann. Das ist dann immer genau derjenige oder diejenige, der für diese Aufgabe irgendwie die Anforderungssicht hat, z.B. aus unserer Sicht als UnIvzulieferer für UnA ist ja sowieso UnA an sich der Product–Owner

also ist der Product–Owner auch immer bei UnA zu suchen, weil ihr immer das Zeug bestellt. Das ist völlig klar, aber da sind auch verschiedene Personen zuständig und wenn es jetzt um DSA / XSA Themen geht, dann bist du immer mein bester PO–Proxy, egal, wo es dann her kommt, meistens kommt es dann aus dem PM, aber manchmal muss man immer denjenigen suchen, der die meiste Ahnung vom Thema hat, oder eben oft üblicherweise das Projektmanagement, also Produktmanagement also Person A oder die Person B oder so; das war auch so ein Problem bei PRO, da gab es eine Zeit lang so eine Phase, wo uns, also Person E und mir, aber ich glaub auch teilweise UnA–intern gar nicht so richtig klar war, wer da jetzt das Sagen hat[T1.6], das hat sich dann nach einer Weile aufgeklärt, nämlich ob man jetzt den Person C oder die Person B fragen muss. Das hat auch ein paar Wochen lang zu Problemen geführt, das schreibe ich jetzt gleich mal hin, Rollen, das ist jetzt wieder Scrum–spezifisch, Rollen unklar[T1.7] besetzt. Das Thema Rollen ist auch ein ganz wichtiger Punkt, das ich von der Architekturschulung kenne, die ich ja halte, die Grundlagenschulung zum Thema Softwarearchitektur, wo es auch um die Rolle des Architekten geht und da ist es auch ganz oft der Fall, dass es eben einen Architekten gibt und der heißt halt so und dem seine eigentliche Rolle ist vielleicht gar nicht so klar oder sie geht in eine ganz andere Richtung und eigentlich müsste der ganz anders heißen weil er was ganz anderes macht. Das hat natürlich auch viel mit Kommunikation zu tun und in dem konkreten Beispiel, mit dem du sicher viel anfangen kannst, ist diese Ding Person C, Person B, die Person B ist ja Produktmanagement, also sie entscheidet letztendlich wie das Ding aussehen soll, also was das Feature, was wir entwickelt haben wirklich tun muss, aus Kunden Sicht, Person C ist halt auf der technischen Seite zuständig und ist natürlich auch sehr stark entscheidungstragend, aber da gabs einfach Situationen, wo wir halt dachten, da reden wir halt mit Person C, die Person B ist halt so, wir reden mit unserem Proxy, den kennen wir, kennen wir die Telefonnummer, fragen wir den was, dann sagt er „ja so machen wir das“, oder „nein, so machen wirs nicht, weil“ und dann zwei Tage später heissts „ähh nee die Person B hat entschieden, wir machens doch anders“. Gab dann auch gute Gründe dann haben wir irgendwann kapiert „Moment mal, es reicht nicht, einen von beiden zu Fragen“, und dann sind wir dazu übergegangen, das wenn er einen Änderungswunsch an uns gemailt hat, dann stand schon immer noch drin: „Ist auch mit Person B abgeklärt und besprochen.“, nachdem wir das auch mehrmals nachgefragt haben. Genau, das ist ein ganz wichtiges, ja, natürlich auch Kommunikation, das sind halt so verschiedene Aspekte: Ich erwarte

das jetzt jemand irgendeine Rolle hat, also ich glaube jemand hat eine Rolle und ich erwarte von dem dann halt, dass er die und die Aufgabe übernimmt und er weiß aber gar nichts davon, weil er die Rolle ganz anders sieht.

Interviewer: Das heißt, er hatte die Rolle selbst und hat es nicht mitbekommen, das er die Rolle überhaupt wahrnehmen sollte, aber jeder andere sieht ihn in dieser Rolle.

Befragter: Genau so. Was wäre das hier, oder umgekehrt jemand hat die Rolle aber die anderen wissen gar nicht, dass das jetzt die Rolle überhaupt ist. Das ist dann auch so ein Thema, dass man durchaus auf der organisatorischen Ebene lösen kann mit Schulungen und sonstige Sachen, wenn man jetzt wirklich sagt, ich mache explizit Scrum, in unserem Fall war das jetzt gar nicht so richtig Scrum, sondern in meinem mentalen Modell haben wir halt erst den Product-Owner erst auf den Person C gemappt und dann fest gestellt: „Moment das müssen wir, bzw. diese Rolle müssen wir auf zwei Personen verteilen." Als wir das gemacht haben, als wir unser mentales Modell angepasst haben, hat es besser funktioniert[T1.8]. Obwohl natürlich noch weder der Person C noch die Person B sich selbst alsProduct-Owner bezeichnen würden, weil sie keine Ahnung haben, dass dies überhaupt existiert[T1.9]. Der Person C schon, dem habe ich das ab und zu gesagt. Genau da ist aber tatsächlich eben dieser Unterschied. Person C weiß nichts vom Product-Owner und deswegen denkt er auch nicht darüber nach, dass er diese Rolle hätte oder haben könnte und was dazu gehört, gar nichts davon wusste und unser Mentalmodell war er aber der Product-Owner und dann haben wir halt Dinge von ihm erwartet, die er nicht bei sich gesehen hat. Dann mussten wir ihm halt sagen: "Wir erwarten jetzt folgendes von dir, damit dies alles klappen kann."

Interviewer: Fallen dir noch mehr Sachen ein?

Befragter: Genau, ja, ich schreibe es trotzdem noch hin: Organisatorische Einflüsse, aber du hast ja gesagt, die sind hier nicht so interessant. (*Anm.: schreibt es auf: Organisatorische Rahmenbedingungen*[T1.10]) Zum Beispiel die Firma sagt: „Wir machen Scrum, aber es darf sich nichts ändern!"[T1.11] oder im Team sagt man: „Wir machen Scrum", aber die Firma drumherum sagt „Nee Scrum gibts nicht, wir machen weiter Wasserfall." Wenn wir sagen, wir machen Scrum, dann kommt auch wieder die Rollen mit ins Spiel und zwar die des Scrum-Masters, also natürlich, wenn die Firma sagt, wir machen Scrum, dann müssen die Leute geschult werden, die Rollen klar besetzt werden und das geht immer schief, zunächst. Da kommt dann der Scrum-Master

ins Spiel, dessen eine Aufgabe eigentlich ist dafür zu sorgen, dass diese organisatorischen Rahmenbedingungen besser passen. Also da irgendwie hin zu arbeiten und dazu muss der Scrum-Master ermächtigt sein, wenn man sagt: „Du bist jetzt der Scrum-Master!"; Bei dir würds vielleicht sogar reichen, aber wenn man z.B. sagt, die Person D ist jetzt der Scrum-Master und die Person D ist jetzt nicht so die Person, die zum Chef hin geht und sagt, das müssen wir jetzt so machen oder sie sagen, ich bin jetzt der Scrum-Master und ich bin ein Externer, ich kann zu niemanden bei UnA hingehen und sagen: „Höh, das müssten wir jetzt ändern." dann bringt das nix.

Interviewer: Ok, das heißt, du brauchst aber von der anderen Seite, du brauchst ein gewisses Standing und gewisse Möglichkeiten und Freiheiten.

Befragter: Genau. Gerade der Scrum-Master braucht gewisse Möglichkeiten auf die organisatorischen Rahmenbedingungen Einfluss zu nehmen, um diese vom Scrum definierte Rolle korrekt zu erfüllen. Das ist halt auch so die Frage: „Wir haben irgendwelche Rollen, die können unklar besetzt sein, die können auch völlig klar besetzt sein, sie können aber nicht ermächtigt sein." Das Ermächtigt ist das schöne Wort (*Anm.: schreibt auf: Rollen nicht ermächtigt*[T1.12]) und das heißt für den Scrum-Master, wie gesagt, für den Product-Owner würds auch heißen, dass er gar nicht die Entscheidungsgewalt über die inhaltlichen Vorgaben hat, die er eigentlich haben müsste. Zum Beispiel als wir dachten, der Person C, aus unserer Sicht ist der Person C unserer Product-Owner für die PRO-Themen und dann haben wir festgestellt: Der ist gar nicht ermächtigt, nur teilweise, nur ein bisschen auf der technischen Seite, aber er entscheidet nicht, ob der Button grün ist, links oben ist, das entscheidet die Person B und der Product-Owner für PRO ist dann letztendlich das Konglomerat aus den beiden; das mussten wir auch erst einmal kapieren. Genau. Dann ähm; (*Anm.: sortiert die Zettel um und denkt nach*) Beim Scrum-Team, wenn wir jetzt schon die Rollen machen, wenn ich schon die Rollen durch gehe, fehlt halt noch der Scrum-Developer, wir als Scrum-Entwickler, ich sehe mich jetzt schon hauptberuflich als agilen Entwickler und nicht als Scrum-Master oder Product-Owner im Moment schieben dann gerne die Verantwortung für Probleme auf die anderen Rollen und die Organisation und überhaupt sind immer alle anderen Schuld, ist ja klar (*Anm.: mit Ironie in der Stimme gesprochen*). Man selber ist ja nie Schuld. Und da ist aber auch eine ganz wichtige Voraussetzung, dass das ja alles funktioniert und das ist überhaupt oft wiederum organisatorisch unklar, wenn man sagt: „So wir machen jetzt agil, wir machen jetzt agil nach Lehrbuch! Wir machen jetzt Extreme-Programming, das ist super, das finden alle ganz toll,

das finden alle Entwickler super, weil sie das viel mehr Freiheit und mehr Selbstverantwortung haben und überhaupt und das machen wir jetzt, wir fangen Morgen damit an.“ Oder es ist, das ist kein Scheiß, das ist Realität, hat mir eine Kollegin erzählt und zwar die, die Testerschulung heute halten wird, bei der UnB haben sie das eingeführt, in dieser Business-Unit bei ihnen. Die UnB ist ja groß, die machen vieles, die wollen jetzt eben auch Richtung agil gehen und in irgendeinem Team haben sie halt beschlossen: „So wir machen jetzt Scrum, ab Morgen!“ Irgendwie so: „Ihr seid jetzt agil.“ (lacht) Möglicherweise haben sie auch gesagt: „Ok ihr kriegt jetzt eine Schulung und dann seid ihr jetzt agil.“ Also nee, es kriegen jetzt alle eine einwöchige Scrum-Schulung, meinetwegen auch das komplette Programm Scrum; es gibt ja inzwischen sogar Scrum-Programme für alles, das ganze Paket dann bist du nach drei Wochen zertifizierter Scrum-Guru; nach drei Wochen und kennst diesen Prozess. Du weißt also genau, welche Schritte dazu gehören und so, aber das ist ja nur ein Framework. Es bringt also erst einmal gar nichts. Der Scrum-Developer, und jetzt schreibe ich es noch einmal generell: Rollen nicht befähigt. (*Anm.: schreibt dies auf*) und das gilt natürlich für den Scrum-Master und Product-Owner auch, man kann halt vielleicht seine Rolle nicht erfüllten, wenn man irgendwie das Know-How dazu nicht hat und für den Scrum-Developer heißt es aber, dass man z.B. keine Ahnung hat von Clean-Code und agiler Softwareentwicklung und Code-Design und Refactoring und man hat überhaupt keine Softwareentwicklungs-Infrastruktur[T1.13], wie z.B. Continous-Integration oder man hat keine Ahnung von Unit-Tests, etc. pp. weil man in einem Wasserfallmodell arbeitet, bis heute aber am Morgen wollen wir anders sein, das diese Techniken eben nicht erfordert, weil man es halt durch Prozessmittel anders löst[T1.14].

Interviewer: Siehst du dann nur diese fachlichen Geschichten, also bei der Befähigung von Rollen, da kann man ja noch weiter denken, dass man nicht nur fachliche Sachen sieht.

Befragter: Richtig! Also da sind wir natürlich wieder beim Thema Kommunikation, ich sag mal bei den Scrum-Entwicklern ist die fachliche Befähigung das Wichtigste. Natürlich muss das Scrum-Team miteinander reden können, also das ist eigentlich der Soft-Skill; man muss sich verstehen, das ist ja, wenn jeder eine fest definierte Rolle hat, ich denke, das kann man bei UnA schon auch beobachten gibt es halt Leute, die haben von Natur aus mehr verschiedene Tätigkeiten und andere mit einem eingeschränkteren Bereich und bei den ganzen Can-Leuten die sind ja noch ein bisschen fester definiert, da kannst du auch als absolut ultra introvertierter Stoffel mit su-

per duper fachlichen Fähigkeiten perfekt durchs Leben kommen, weil du musst dich nicht mit allen austauschen. Als fachlicher Depp hast du sowieso verloren, aber man kann sich dann vielleicht auch ein Feld finden, in dem man dann eben doch zurecht kommt und innerhalb eines Scrum-Teams ist es halt auch wichtig für eine solche Transition, wo man sagt, so, wir machen jetzt Scrum und vorher hatten wir es nicht und das war auch so ein Punkt bei dem anderen Kunden in Hamburg, dass da halt eben manche im Team mit austauschen wollten und collecitve code ownership[T1.15] ist ja auch so eine Sache, dass der Code halt quasi allen gehört, was natürlich aus Firmensicht ganz wertvoll ist, weil dann eben nicht einer nur der einzige ist, der eine Ahnung hat zu einem Thema, du willst ja keinen Flaschenhals und so und dann aber Leute halt sagen: „Nee, das gehört mir, du darfst da nicht drauf schaun." Dann ist dies halt schlecht. Oder wenn man halt nicht so gut in der Lage ist, miteinander zu reden, weil man halt nicht so eine Redeperson ist[T1.16]. Wobei man natürlich, ich behaupte mal, bis auf (*Anm.: kurze Redepause*), das wir schon wieder vielseitig. Also ich sag mal Kommunikation auf der fachlichen Ebene kriegt man grundsätzlich immer hin, vorausgesetzt, die Leute haben fachlich eine Ahnung. Also kriegt man hin im Sinne von: Wenn man feststellt aha, da gibts ein Problem dann kann man dagegen eine Maßnahme ergreifen, wie z.B. man richtet Dinge wie regelmäßig Meetings ein, oder man stellt jemanden ab, dass Leute miteinander reden oder man bringt den Leuten bei: Schreibt bitte eine Mail, wenn irgendwas ist[T1.17]. Auf dieser persönlichen Ebene, also wenn es fachlich ist, ich meine jetzt wirklich den nackten Austausch von fachlichen Informationen, das kriegt man schon hin; man muss aber erkennen dass dies der Fall ist[T1.18]. Auf der persönlichen Ebene, wenn es wirklich darum geht, dass sich Leute auch so auf der Metaebene austauschen und das man wirklich gut zusammen arbeitet und es nicht einfach nur darum geht jetzt Info weiter zu reichen, da gibt es halt verschiedene Aspekte warum das schief gehen kann[T1.19], z.B. dass jemand ganz einfach introvertiert ist; das ist ja völlig ok, gerade bei Software-Entwicklern ist das durchaus üblich und nach meiner bisherigen Erfahrung gerne auch kombiniert mit einer hohen fachlichen Expertise. Also es gibt dann so Leute die so ein bisschen hacken und die sind auch ganz tolle Hacker, es ist nur blöd, wenn man mit denen kommunizieren will. Oder aber auch dass jemand generell nicht so gern mit Leuten spricht. Wenn er dann mal was zu sagen hat ist das ganz wunderbar.

Interviewer: Also das sind dann eher so Persönlichkeitsmerkmale.

Befragter: Genau, das wäre dann so ein Persönlichkeitsmerkmal. Gegenteil ist natürlich auch Laberbacke, bin ich dann eher in der Schublade, dass man den ganzen Tag dann immer quatscht und quatscht und allen erzählt, wie schwierig man es hat oder so, aber irgendwie dann nix macht. Da ist man dann ganz ganz gut vielleicht in einem anderen Bereich aufgehoben, der auch wichtig ist fürs Team. Offtopic Zwischenfrage: Hab ich dir diese Videos schon gegeben?

Interviewer: Nee, welche meinst du?

Befragter: Die zwei Videos von dem einen Professor, der über Arbeit und Führung spricht.

Interviewer: Nee, hast du mir noch nicht gegeben, aber ich habe einen USB-Stick dabei.

Befragter: Sehr gut, wichtig, ganz wichtig. Anderes Thema ist natürlich wie heißt das, wenn sich Leute eben nicht so mögen, ähm,

Interviewer: Disharmonie?

Befragter: Ja genau, eher Abneigung, persönliche Abneigung*T1.20*. Da gibts natürlich auch zwischen Leuten, da gibts natürlich auch ganze Theorien dazu, wie sich so etwas entwickelt und steigert und was man dagegen machen kann. Meistens hilft reden, aber manchmal ist es auch tatsächlich so, das kann sich steigern und manchmal ist es auch so, dass man manchmal nicht miteinander kann und dann feststellt, ok mit dem rede ich nicht so gern und dann muss das einfach auf der Team-Führungsebene gehandhabt werden, dass man sagt, ja ok, dann tu ich die halt nicht in ein Pair-Programming Aufgabe*T1.21*. Dann sollen die halt zwei unterschiedliche Sachen machen und nur miteinander reden, wenn es sein muss und so weiter. Ja, genau da gibt es, glaub ich noch viel mehr. Also da gibts tatsächlich noch viel mehr, was mir jetzt gerade nicht so einfällt. Generell ist es so, dass Scrum und die ganzen agilen Methoden gerade bei dem Thema Kommunikation ansetzen*T1.22*. Wir sagen, dass die Idee ist, dass das Team eben viel mehr miteinander zusammen viel enger miteinander zusammen arbeitet als das das vielleicht in anderen Modellen der Fall ist, dass wir eben nicht klar Aufgaben definieren und verteilen, vielleicht noch von oben herab, sondern dass die Leute sich da zusammenraufen, dann hast halt dieses lustige Storming, Forming, Performing usw. Modell Adjourning oder so*T1.23*.

Interviewer: Meinst du, dass man das im Scrum nicht hat?

Befragter: Doch, doch, das hat man im Scrum natürlich erst recht. Das hast du eher weniger, wenn du klar verteilte Aufgaben hast, vielleicht auch, aber dann ist es vielleicht ein bisschen weniger stark, weil zur Not sagt man halt: „Ich mach hier mein Ding."

Interviewer: Also, im Scrum muss man also mehr dafür tun, dass dieses Modell zum Performing auch kommt.

Befragter: Genau, genau, das muss einem klar sein, dass wenn ich Scrum machen will dieses Modell (*Anm.: zeichnet*) greift, dass es auf jeden Fall eine Rolle spielt und das man es beachten muss. Ich weiß ja, bei dir geht es auch so ein bisschen um das Thema Agilität braucht Führung und das ist so.

Interviewer: Das ist so ein Hintergrund bzw. ein Nebenschauplatz, es geht momentan wirklich nur um das Team selbst, Hauptsächlich um Probleme zu diagnostizieren. Du hast ja jetzt hier dir schon verschiedene Felder erarbeitet. Ich sehe momentan bei dir zwei Zweige, du hast einerseits die ganzen Rollengeschichten hier, du hast auch ein bisschen das organisatorische, das aber auch indirekt wieder mit den Rollen zusammenhängt. Das ist an sich ganz interessant, das hast du auch hier im unteren Teil dargestellt und im oberen Teil haben wir quasi das Thema Kommunikation und Persönlichkeitsmerkmale für einzelne, die aber meistens die Auswirkungen auf die Kommunikation haben. Das beobachte ich bei dir so zusammengefasst.

Befragter: Ja, die Persönlichkeitsmerkmale sind immer die Sachen, die mir als erstes einfallen, wenn ich an Beispiele für schlechte Kommunikation denke, wobei ich ganz bewusst dieses Punktpunktpunkt hingeschrieben habe, vielleicht fällt mir noch etwas sein, aber das sind natürlich nicht die einzigen Einflüsse, das ist nur so das Beispiel, was ich jetzt auch durch die Videos, die du noch schaun wirst, wenn ich sie dir gegeben habe da auch vorkommen, das Schöne an den Persönlichkeitsmerkmalen sind so Sachen, das kann man eben auch als Führungskraft oder als andere Teammitglieder halt so ein bisschen beobachten und bemerken und dann sagt man das Interessante an Persönlichkeitsmerkmalen ist, dass die relativ fest sind, d.h. das sind Dinge, da sagst du, die kannst nicht einfach so ändern, du kannst jetzt nicht sagen: Sei jetzt mal anders. Sondern da muss man im Team damit umgehen, naja der ist halt so, da müssen wir mal schaun, dass das trotzdem klappt.

Interviewer: Vielleicht noch etwas, um das Bild noch zu vervollständigen, Scrum arbeitet ja mit Selbstorganisation ...

Befragter: Ganz kurz ... ach so, spricht weiter, vielleicht kommen wir da jetzt genau hin ...

Interviewer: Ich möchte dich jetzt ein wenig Richtung Selbstorganisation lenken, damit wir noch eine andere Facette, wieder einen Teil in das Bild reinbringen, die noch zusätzlich was bringt. Wo irgendwie Konstellationen oder warum könnte die Selbstorganisation irgendwie versagen kann. Wir wissen aus unserer Erfahrung, bei uns hat sie schon einmal ein bisschen versagt..

Befragter: Ja ...

Interviewer: ... das ist nicht untypisch, die Frage ist nur, woran kann man das festmachen oder welche Situationen, Konstellation oder vielleicht kennst du ja noch andere Beispiele, wo dies ins Stocken geraten ist, nicht mehr funktioniert hat oder problematisch wurde, ... was hast du da drauf geschrieben?

Befragter: Aktivitäten, ich hab ja vorhin gesagt, Scrum ist ein Framework und es definiert Rollen und Aktivitäten, also Aktion / Tätigkeiten die im Rahmen des Scrum-Frameworks durchzuführen sind und die Scrum-Menschen sagen, dass dies wichtige Schritte sind. Und da gibt es ja genau eine Richtung, wie hast du das gesagt, Selbstorganisation ...

Interviewer: hmm, Selbstorganisation ....

Befragter: Schreibe ich mit drauf, aber das ist genau das.

Interviewer: Was verstehst du unter Selbstorganisation? Nur um zu wissen, ob du das gleiche Verständnis von diesem Begriff hast ...

Befragter: Das ist eine sehr gute Frage, weil ich habe nur ein schwaches und schwammiges Verständnis davon, aber das kann man darauf zusammen fassen, dass so ein Team bestehen aus all den Leuten sich selbst im Prinzip sich selbst durch diese lustigen Forming, Stroming, usw. Phasen hindurch durchbringt. Organisation heißt ja, man hat in irgendeine Struktur mit Rollen und Tätigkeiten und das geht natürlich weit über die Scrum-Rollen hinaus[T1.24], diese sind ja nur sehr minimalistisch, damit man anpassbar ist auf alle möglichen Fälle und an sich geht in der Organisation darum, die richtigen Rollen zu erkennen, was für Rollen brauche ich für meine Arbeit und welche Personen setzte ich auf diese Rollen. Das ist eigentlich ein nettes Organisationsmodell, das ich mir gerade ausgedacht habe. (*Anm.: schreibt*) Ich habe es jetzt so aufgeschrieben, Rollen, die für eine Aufgabe notwendig sind und

Personen, die Rollen besetzen, allerdings ist das ein bisschen gefährlich, weil dieses Personen, die die Rollen besetzen ist eigentlich falsch herum formuliert, eigentlich ist es das Mapping von Rollen auf Personen, quasi Zuweisung von Rollen auf Personen. Die Organisation definiert ja nicht die Personen, sie definiert ihre Zuordnung. (*Anm.: Schreibt Zuordnung von Personen auf Rollen*)[T1.25]. Das ist doch ein netter Organisationsbegriff, oder?

Interviewer: Ja

Befragter: Ok, Selbstorganisation würde dann heißen, dass das Team diese zwei Aspekte findet und erfüllt, also man stellt fest, man hat hier eine Aufgabe, man muss hier etwas machen, was mach ich denn da jetzt, ja da brauche ich erstmal einen Entwickler und dann brauche ich einen Architekten, dann brauch ich einen Buildmanager oder ... im Prinzip nee das sind Rollen, wir sprechen jetzt über Rollen, nicht über Personen, ich brauch jemand, der sich um den Build kümmert, ich brauche jemanden, der sich um die DSA kümmer, etc. pp. ...

Interviewer: Genau, das sind jetzt sehr fachliche Rollen.

Befragter: Genau, das sind sehr fachliche Rollen, ich brauche jemanden, der sich darum kümmert, dass dies alles hier läuft, der sich um die Organisatorische Infrastruktur kümmert, also mit Organisation meine ich die persönliche, man kann ja auch Dinge organisieren, z.B. einen Rechner, aber wir meinen jetzt das hier ... interessant, interessant, Rollen, die für eine Aufgabe nötig sind, das passt schon; genau, wir brauchen jemanden, der sich darum kümmert, dass die Infrastruktur da ist. Man wird dann vielleicht irgendwann fest stellen, der bei Streit schlichtet, vielleicht stellt man auch fest: „Ach nee, wir streiten uns nicht, wir sind alle total harmonisch." Ich sag das jetzt so schrittweise, aber das passiert vielleicht alles ganz schwammig, gerade bei der Selbstorganisation kommt das eher so: Ah was machen wir jetzt? Und dann überlegt man, was brauchen wir jetzt eigentlich und dann legt man mal los, irgendwie und dann stellt man fest: Oh, das brauch ich auch noch! Je nachdem wie viel Vorausplanung man macht und das andere ist dann, ok, du machst jetzt den Buildjob, weil du das schon mal gemacht hast und ich hol uns schon mal Kaffee, weil ich sonst nix weiß, so ungefähr. Was auch immer. Und das gleiche ist natürlich das Gegenteil, ist halt eben nicht Selbstorga, sondern, dass halt jemand anderes die Organisation übernimmt, das ist eigentlich auch durchaus attraktiv, wenn man z.B. jemanden hat, das ist auch wieder eine Rolle, diesen Unterschied finde ich ganz wichtig, weil eine Rolle ist so eine Art Aufgabe, die irgendwie von jemandem irgendwie

besetzt werden muss und diese Rollen und Aufgaben sind ja immer da und das ist aber unabhängig davon kann man die aber so oder so mit Personen besetzten. Das wird gerne, weil man halt so spricht: Ich brauch jemanden der sich um den Build kümmert. Ja und da denkt man ok, einer muss sich um den Build kümmern und dann wird diese Rolle und die Person schnell gleich gesetzt, das ist so ein Thema, was im Architekturseminar halt auch ganz Zentral am ersten Tag in, da geht es eben um die Rolle des Architekten, da geht es um die verschiedenen Arten, wie man diese Rolle erfüllen kann, das ist ein ganz wichtiges Thema, was dann zu Aha-Momenten führt auch. Das es ganz wichtig ist, das zu unterscheiden. Wenn er sagt: Ok, ich habe diese Rolle des Build-Managers und die ist halt so und die können aber auch zwei Leute erfüllen, oder wenn man es richtig macht, kann die auch ein Computer erfüllen, und dann braucht man nur noch jemanden, der den Buildmanager managed und der hat dann vielleicht weniger zu tun (lacht).

Interviewer: Das stimmt. Noch einmal die Frage zurück, was mich interessiert ist, wo liegen da die Knackpunkt oder wo könnte das ganze nicht funktionieren ...

Befragter: Genau, also das Schöne ist, wenn ...

Interviewer: ... oder wo hast du das schon beobachtet, dass es nicht funktioniert?

Befragter: Kurz ausgeholt, wenn du keine Selbstorganisation machst, dann hast du im Idealfall, der sich mit Projektmanagement auskennt und schon Erfahrung hat in der Richtung und nichts anderes tut, als so eine Organisationsstruktur zu machen. Dann weiß der genau, was da alles zu tun ist, was man so braucht, um so ein Projekt erfolgreich durchzuführen, wenn er Glück hat; wenn man Pech hat, dann weiß er das zwar für die Projekte, die er bisher gemacht hat, aber nicht für das tolle neuartige Projekt, das ganz anders ist und dann wundert man sich, warum es nicht funktioniert[T1.26], aber oft ist es halt so, man halt niemand mit diesem Know-How, mit diesem Meta-Know-How, über Organisationsstrukturen. Das ist auch wieder so eine Rolle: Das Know-How muss halt da sein, also es ist hilfreich, wenn dieses Know-How da ist, dann kann man nämlich gleich recht frühzeitig so eine Organisation zusammen strukturieren, die für diese Aufgabe passt. Wenn man sich selbst organisiert, dann ist die Idee, dass man ja lauter Experten hat, fachliche Experten, die vielleicht auch gute Kommunikations-Skills haben, die dann sich die Aufgabe anschaun und sich so hin organisieren, dass das ideal-optimal zugeschnitten ist und für eine neue Aufgabe sich wieder

anders selbst organisiert. Das gilt natürlich nur, wenn dieses Team in der Lage ist, Organisationsstrukturen aufzubauen. Sprich man muss dann auch schon ein bisschen was über Organisationsstrukturen wissen, über Rollen und Aufgaben, über Tätigkeiten, über Projektmanagement quasi. Die Idee vom Scrum ist: Wir geben schon mal so ein Framework vor, wir geben mal drei Rollen vor, die sind aber eigentlich sehr abstrakt und die Aufgaben definieren, die nur dafür sorgen sollen, dass wir schon mal keine Gedanken darüber machen müssen, wer jetzt was auf der Projektorganisations-, auf der Managementebene was tut, sondern wir können uns dann rein um das Fachliche kümmern. Das ist ein ganz wichtiger Aspekt für die Selbstorganisation (*Anm.: Schreibt auf das Papier*): Inspect + Adapt! Scrum gibt Mittel für schnelles Feedback vor auf verschiedenen Ebenen[T1.27]. Das ist eigentlich das Wichtigste an Scrum. Diese Rollen das das ist ok, das ist einfach nur, um schon eine Grundstruktur zu liefern, die sinnvoll ist, aber der Kern von „agile" ist die schnelle Rückkopplungsschleife. Und deswegen gibt es bei Scrum diese verschiedenen Schleifen, die Scrum hat. Die Produktidee, die Sprints, wo man immer so ein Deliverable macht und bei jedem Sprint die Retrospektive und dann kommt halt hinten was raus ...

Interviewer: Ja ...

Befragter: ... und innerhalb der Sprints gibt es dann noch die Daily-Standups, das sind alles so Feedback-Schleifen. Und das ist eigentlich der Kern von Selbstorganisation, also das ist notwendig für Selbstorganisation, dass man, weil wie gesagt, Selbstorganisation heißt, ich schau mir mal die Aufgabe an und organisiere mich dann als Team so hin, dass ich die Aufgabe dann optimal erfüllen kann, was natürlich deswegen viel toller ist, als die von außen gesetzte Organisation, weil die ja vielleicht nur grundsätzlich eine Organisationsstruktur vorgibt, die sich halt in Projektmanagementbüchern bewährt hat und die kann passen, muss aber nicht. Die kann sogar total kontra-produktiv sein. Wenn ich mich aber an die Aufgabe anpassen kann, die Aufgabe kann sich auch im Lauf der Zeit ändern und dann muss ich meine Organisation ändern, das ist z.B. das was ich gesagt habe im UnA-Projekt, wir als UnIv hatten eben mal die eine Rolle, dass wir selbst was entwickelt und dann abgegeben haben und jetzt haben wir im gleichen Projekt die Rolle, dass wir Stückchen zuliefern und das ist eine andere Tätigkeit. Und dieses Stückchen zuliefern, das war auch das Beispiel mit dem Person C und der Person B und solche Sachen, wo man halt fest stellt: Unsere Organisation muss ein bisschen anders sein, als wir das letzte Woche noch dachten. Genau. Deswegen ist hier natürlich auch ganz klar, was schief gehen kann:

Nämlich wenn man dieses Feedback nicht nutzt, oder die Inspektions-Mittel, die einen Scrum gibt[T1.28], man muss auch nicht auf die Scrum-Inspektions-Mittel gehen, aber auf jeden Fall, wenn man das nicht macht, dann kann die Selbstorganisation gar nicht funktionieren. Natürlich kann sie auch nicht funktionieren, wenn die Befähigung und Ermächtigung[T1.29] der Leute nicht gegeben ist. Ich weiß jetzt nicht, ob es jetzt zur Ermächtigung zählt, z.B. auch dass man gar nicht die Zeit hat, sich um Organisationsthemen zu kümmern, weil man viel zu sehr damit beschäftigt ist, Bugs zu fixen oder so was in der Richtung[T1.30].

Interviewer: Das stimmt ja. Das ist schon Interessant, weil du meine Fragen zum Teil schon vorweg nimmst.

Befragter: Ich habe erstaunlich viel über Scrum und agile in den letzten Jahren gelesen.

Interviewer: Das Andere stelle ich mal noch ein wenig zurück, was man vielleicht dagegen tun kann, das machen wir besser am Schluss. Ich möchte noch ein bisschen, ich gebe dir noch ein zweites Blatt, damit wir noch ein bisschen in die Richtung Diagnostik gehen. Du hast bereits viele Probleme irgendwo angesprochen, bei vielen auch schon Ansätze gezeigt, wie kann ich das irgendwie bemerken, dass in meinem Team etwas nicht stimmt, oder mit meinem Output etwas nicht stimmt, im Prinzip, das etwas nicht stimmt. Das würde mich noch ein bisschen mehr interessieren, einfach auch um Leuten einen kleinen Leitfaden an die Hand zu geben, wie kann ich bemerken, dass etwas im Team nicht stimmt. Ich schau mir das Ganze aus der Coach-Sicht an, ich schau mir ja im Prinzip mein Team ja ohnehin jeden Tag mein Team an und überlegst dir: Mensch, da stimmt was nicht! Die Frage ist, woran erkennst du, dass da etwas nicht stimmt. Du hast da oben auf Blatt schon viele Probleme aufgezählt, die dir einfallen, die Frage ist, wie kommst du darauf, dass du das in deiner täglichen Arbeit beobachten kannst.

Befragter: Ich schreib jetzt hier mal Retrospektiven[T1.31] hin. Das ist eine sehr gute Frage.

Interviewer: Oder anders herum, was dir vielleicht leichter fällt, vielleicht hast du schon einmal irgendwann solche Probleme erkannt, das du dich in diese Situation noch einmal rein versetzt, woran du das erkannt hast, oder was die Rahmenbedingungen waren oder was war dafür ausschlaggebend. Vielleicht greifen wir das Beispiel mit der Person B noch einmal auf ...

Befragter: Ja, genau ...

Interviewer: Das ist vielleicht ganz gut. Oder eigentlich dieses Rollenproblem, woran hast du das eigentlich bemerkt, dass es nicht funktioniert?

Befragter: Das überlege ich gerade, warum ist uns das aufgefallen? Das ist immer wenn es weh tut. Also machen wir wieder so wie hier, der Zettel, ich muss ein bisschen Orga machen. Das hier ist sozusagen ...

(*Anm.: Störung durch einen anderen Mitarbeiter der den Raum betritt. Fachfremder Beitrag wurde entfernt.*)

Befragter: Was ist jetzt hier noch mal die Fragestellung?

Interviewer: Im Prinzip, woran man solche problematischen Situationen / Konstellationen erkannt hat und vielleicht noch einmal an das Beispiel mit dem Rollenkonflikt, was es im Team bzw. bei der Implementierung gab.

Befragter: Genau (*Anm.: Schreibt*)

Interviewer: Schreib doch mal PRO-Implementierung hin, das man das einfach am Beispiel analysiert, das ist gar nicht so schlecht gut denke.

Befragter: Das ist deswegen ein ganz hervorragendes Beispiel für das ganze Thema, weil wie ich schon sehr oft bemerkt habe und nicht müde werde zu wiederholen, das bis auf die organisatorischen Rahmenbedingungen der Auftragsgestaltung ist das eigentlich eine, fast schon lehrbuchmäßiges Beispiel für eine agile Entwicklung. Sowohl von der Anforderungsseite, was man erreichen will und wie man es erreichen will und wie man es dann auch tatsächlich umsetzt, nämlich durch Iterationen, kurze Zyklen, häufiges Draufschaun, also aufs Produkt, das Endergebnis, also nicht mal auf der Metaebene, mit dem dann aber auch zusammenkommt ... Oh Gott, das ist es, dadurch haben wir es erkannt. Ich schreibst da her. Am Beispiel PRO-Workflow. Wir haben an diesem PRO-Workflow, weil es fürs Produktmanagement ein ganz heißes, wichtiges, aktuelles Thema war, wo sie selbst ja noch nicht so ganz sicher waren, wie es am Ende sein soll, also so ein richtig klassisches agiles Beispiel, also für einen Fall, wo man agil arbeiten möchten. Und deswegen rein auf dieser fachlichen Ebene hat das Produktmanagement regelmäßig drauf geschaut, weil es für sie ja interessant und wichtig war. Schnelles Feedback zu kriegen.[T1.32] Arbeitsergebnis. Das hat natürlich dazu geführt, dass wir natürlich insgesamt schnelles Feedback bekommen haben, dazu, wie wir arbeiten[T1.33]. Weil man dann eben, wahrscheinlich, innerhalb von ein, zwei Wochen ein Feedback hatte auf der Produktebene: „Das ist schlecht, das sollte bitte anders sein!“, was natürlich dazu geführt

hat: „Häh, wieso ist das denn jetzt plötzlich schlecht, wir haben es doch genauso gemacht, wie Person C es bestellt hat, jetzt will es auf einmal Person B anders.“, da haben wir uns dann ein paar mal drüber geärgert, genau, das war das wichtige Stichwort, woran erkenne ich es: Wenn man sich ärgert.[T1.34] Daran erkenne ich, dass ein Problem besteht, oder auch anders: Wenn es weh tut.[T1.35] Das ist dann immer bei so Infrastruktur-Sachen[T1.36], wenn etwas weh tut, dann stimmt halt die Build-Pipeline nicht oder dann stimmt das organisatorische nicht, dass der blöde Web-Proxy nicht erreichbar ist, oder dann ist man Versionskontrollsystem zu langsam und dann tuts weh und dann ärgert man sich. Und das sind dann auch alles Sachen, die wichtig sind, für die Produktarbeit. (*Anm.: Schreibt auf*) „Wenn es weh tut!“ Das sind natürlich zwei Phrasen für die gleiche Sache. Genau und hier. Das nimmt natürlich auch schon ein bisschen die Antwort vorweg, aber das ist es halt: „Inspect and Adapt“ Also was hier funktioniert hat, das ist ja das Beispiel wo es funktioniert hat, das wir so ein Problem erkannt haben. Also Beispiel (*Anm.: Schreibt auf*) „Problem erkannt: PO-Rolle war falsch interpretiert.“ Also mit interpretiert meine ich, wir haben den falschen als PO interpretiert.

Interviewer: Da war das falsche Rollenverständnis, also die Entwickler hatten das falsche Rollenverständnis.

Befragter: Genau.

Interviewer: Während auf der anderen Seite nicht wichtig war oder keine Gedanken gemacht hat ...

Befragter: Da wurde es einfach nicht, da hat sich einfach keiner Gedanken darum gemacht. Das ist so das. Für die Person B war es glaube ich schon klar, für den Person C war es nicht klar und für uns war es auch nicht klar. Der Person C ist ja auch Entwickler. Also es war , ja war irgendwie unklar. Einfach nur unklar. Problem erkannt und zwar woran, weil es halt, weil gefühlt gefühlt, (*Anm.: Schreibt auf*) „spontane Anforderungsänderungen“ Also es war so, es gibt ja immer mal wieder Anforderungsänderungen, das hat verschiedene Gründe, das hat sich da so gehäuft[T1.37], das war nämlich so: „Gestern war das doch noch anders, also ich gestern den Person C gefragt habe, war es noch anders und heute kommt die Person B daher und will es jetzt so, das kann doch wohl nicht wahr sein. Reden die nicht miteinander?“ Oh, Reden die nicht miteinander, also auf der fachlichen Ebene heißt schlechte Kommunikation meistens mangelnde Kommunikation.

Interviewer: Kann man das auch irgendwie erkennen, das das miteinander Reden nicht passiert?

Befragter: Ja, das erkennt man dann also an dem Fall hat man es erkannt, genau, wir haben das deswegen erkannt, das war so, das war so das woran wir erkannt haben, dass es ein Problem gibt, also es gab spontane Anforderungsänderungen und da haben wir erkannt: Da passt doch irgendwas nicht. Der Weg war so (*Anm.: Schreibt auf*): Es gab spontane Anforderungsänderungen, dann dieses „da passt doch was nicht", dann dieses, die Frage, „Wer hat denn nun das sagen?" und dann ist halt die Lösung „PO-Rolle neu interpretiert". So ungefähr wars. Und was und tatsächlich geholfen hat, diese Erkenntnis zu erlangen und da kommt jetzt tatsächlich die Agilität ins Spiel, ist dass wir häufige Inspektionen hatten, also es waren jetzt keine Retrospektiven, es ist auch scheißegal, welches Mittel man wählt, Hauptsache man hat häufiges Feedback und dass die spontanen Anforderungsänderungen wären nicht passiert, wenn die Person B nicht drauf geguckt hätte.

Interviewer: Die Anforderungen wären dann einfach unklar gewesen.

Befragter: Naja, teilweise gab es echte, sowas wie am nächsten Tag wird dann genau das Gegenteil verlangt. Man hat halt eine Frage, soll ich das so oder so machen, dann sagt der eine machen wir es so und dann sagt der andere nenene, anders herum. Ich weiß nicht mehr was das war, da ging es um eine Konfig von dem Ding, will ich ein Häkchen haben für AFD oder will ich ein Häkchen haben für generisch für wie heißt das, irgendeine Eigenschaft, die zufällig vom AFD-Gerät erfüllt wird. Also das ist quasi eine reine, man könnte halt sagen, das ist ein Implementierungsdetail, es ist aber sichtbar auf der Oberfläche, also ist es auch aus Nutzersicht wichtig und dann hat halt Person C die eine Variante bevorzugt und die Person B oder Person A in dem Fall sogar das PM die andere Variante und dann war man sich da auch nicht einig, das hat dann auch dazu geführt, dass wir uns noch einmal zusammen gesetzt haben, das ist jetzt ein Beispiel, aber da gab es noch mehrere so in der Richtung. Das Beispiel war dann so, das man tatsächlich noch einmal miteinander diskutieren musste und dann ist man zu einer dritten Lösung gekommen, was natürlich auch toll ist. Es gab dann aber auch einfach Sachen, wo er gesagt hat, naja weiß ich nicht, machen wirs halt so und dann kommt die Person B am nächsten Tag, nenene das wollen wir so haben. Wo ihm selbst auch nicht klar war, dass er die Entscheidung halt nicht treffen soll und danach schon. Prozessmittel, das unterstützt hat ist eben: Häufige Demos[T1.38] (*Anm.: Schreibt auf Zettel*). Da waren es jetzt die Demos; die Demo ist ja auch hier irgendwo, wir haben hier irgendwo die Retrospektive und eine Demo und die Retro ist halt schon von vorne herein so dafür da, dass man über diese Meta-Dinge spricht und die Demo ist dafür da, das man

über die fachlichen Sachen spricht[T1.39]. Aber natürlich fallen bei den Demos die Probleme auf, weil wenn das Fachliche immer passt, dann ist es ja auch ok. Und normalerweise ist es ja so, das dann halt irgendwas, häufige Demos haben dann dazu geführt, das wir eben solche Sachen schnell erkannt haben, dass eben spontane Anforderungsänderungen, hat halt drauf geschaut, am nächsten Tag stellt sie fest, nee Moment, was habt ihr denn da gemacht. Häufige Demos und daraus dann dementsprechend häufige Schmerzpunkte ...

Interviewer: ... oder Unstimmigkeiten, was die Gegenseite erwartet hat ...

Befragter: ... und daraus folgt natürlich auch: Häufige, was wollte ich denn jetzt sagen, Anpassungen, Korrekturen, da gibt es so ein schönes Wort, ich sag jetzt mal Anpassungen und das wiederum ist dann Selbstorga. Ja.

Interviewer: Ok, nachdem wir jetzt das Thema abgeschlossen haben, fallen dir noch mehr Probleme an, wo man Probleme erkannt hat, oder wo sie dir aufgefallen sind?

Befragter: Ja, es gibt unendlich viele Beispiele, das ist ein Beispiel war auch da bei meinem einen Wochenausflug in Hamburg, da waren die Standups unendlich lang. Das war bei uns ja auch eine Zeit lang, dass die Standups unendlich lang waren und das ist ein bekanntes Anti-Pattern und das muss ja an sich gar nicht schlimm sein, aber es hat einen Grund, warum das ein Anti-Pattern ist, es war auch so, das die auch lang und langweilig waren und da waren die Leute halt gesessen und nicht gestanden und sie haben nicht das gemacht, was wir jetzt im Standup machen, was ich sehr gut finde, wir sagen halt knackig, was man macht und man kommt dazu so ein Standup ist natürlich auch eine total wichtige Inspection-Sache, die die Kommunikation verbessert, mindestens auf der fachlichen Ebene, dass da Infos weitergereicht wird, das man überhaupt erkennt, das es ein Problem gibt, oder ein Verbesserungspotential, es gibt ja nicht immer nur Probleme, es gibt ja auch so Sachen: Ich mach hier an was und ich weiß nicht so gut, ich muss jetzt dann halt mal zwei Tage lang googeln und der andere sagt, ah da kenn ich mich mit aus. Genau und das war halt so ein Standup, da hat halt jeder erstmal so eine halbe Stunde 10-Minuten lang erzählt, wie im Detail was er gestern gemacht hat. So ganz im Detail auf Code-Ebene. Dann haben wir uns halt alle hingestellt und dann hat man schon schneller gesprochen.

Interviewer: Ok, was bei mir noch hier steht: Wie unterstützt das Scrum-Framework beim Erkennen? Da hast du ja auch schon viel genannt, also gerade was du hier oben geschrieben hast, eigentlich diese Kontrollpunkte hast

du als sehr wichtig hervorgehoben und du hast auch oft was von den Rollen geredet. Das habe ich jetzt so mitgenommen.

Befragter: Noch ein Stichwort, das ich gerade genannt habe, aber vielleicht noch einmal explizit, diese Punkte und so und was das reine, nackte Framework noch hinausgeht, die Smells. Es gibt ja so typische Scrum-Antipatterns[T1.40]. Da gibt es auch ganz viele und die nennt man auch alle im Netz und so und das wie bei Code-Smells, der Code schaut an der Stelle so und so aus, das muss jetzt nicht schlecht sein, ist aber einen zweiten Blick wert und das Gleiche gibt es auch in Scrum und das sind zum Beispiel so Sachen wie ein zu langes Standup[T1.41] oder andere Sachen. Und die helfen einem dabei, so Probleme zu erkennen.

Interviewer: Eigentlich sind das hier zwei Gruppen, die du hingeschrieben hast: Sachen, die andere Leute auch schon erkannt haben, die irgendwo in solchen Dingen wie Satire zusammengefasst sind und hier da hast du jetzt persönliche Gefühle hingeschrieben. Das ist vielleicht auch ganz interessant, die Frage ist, kann man solche Gefühle auch versachlichen, weil ansonsten, um Gefühle aufkommen zu lassen muss man oftmals sehr weit drinnen sein, dass man sich irgendwie ärgert oder das es weh tut bei seiner täglichen Arbeit, die Frage ist kann man das auch von einem zurück gezogenen Blick, jemand der nicht so häufig reinschaut oder der ein bisschen zurückgesetzt vom Team ist, also ich sag jetzt mal dein Chef, dass er auch so was erkennen kann. Fällt dir dazu etwas ein? ... oder siehst du das eher als schwierig, das man das von außen erkennen kann? Weil, es sind ja jetzt Sachen, da muss man eben im Team verankert sein, du musst also bei sein, selbst als Scrum-Master tust du dich da ja unter Umständen schwer. Also wenn du jetzt ein Scrum-Master bist, der nicht immer bei der täglichen Arbeit komplett dabei ist, es gibt ja auch so wechselnde.

Befragter: Ja, hier; Genau dafür bietet Scrum eben wieder explizit Mittel, nämlich bei dem was ich als Aktivitäten genannt habe, da kommt dann die explizite Retrospektive in Spiel. Also das ist nämlich der Grund, warum, wir machen ja zur Zeit keine, was ich ganz schlecht finde, aber es funktioniert trotzdem weil wir alle im Team sind, wir sind alle im Team verankert[T1.42] und deswegen tauchen so Probleme auf. Also deswegen erkennen wir solche Probleme, das ist eigentlich eine ganz schöne Unterscheidung. Es ist zwar trotzdem schlecht, dass wir keine Retros machen, aber das ist noch mal ein anderer Punkt. Scrum, ich nenns mal Werkzeug (*Anm.: Schreibt auf*) Retro, wohingegen die Demo eigentlich für diese fachlichen Themen gedacht ist. Wenn

man im Team so verankert ist, dann reichen schon die Probleme, die man in den Demos findet, um eine inoffizielle Retro anzustoßen. Die Retro ist ja auch gar nicht, zwecks stetiger Verbesserung, die hat ja keinen Selbstzweck, die ist ja dafür da, das man sich ständig weiter verbessert, bzw. auch nicht nur Verbesserung, das ist ja gar nicht mal der Knackpunkt, sondern Anpassung. Als der Ami spricht ja immer von Verbesserung und „Making a change", die wollen ja toller und besser haben, als es vorher ist, das solls immer. Ich finde es immer ganz komisch, das kann man ja auch machen und wollen, man will ja immer besser werden, aber ich finde es auch ok, wir sind doch eigentlich ganz wunderbar, aber auch wenn wir ganz wunderbar sind ist es halt so, das die Realität, die Welt drum herum bleibt ja nicht stehen, deswegen muss man auch immer die Möglichkeit haben, sich auch anzupassen. Auch wenn man sagt, wir sind schon total toll, kann es auch sein, dass das Toll von heute die Kacke von Morgen ist. Genau das ist das Mittel von Scrum, nämlich die Retrospektiven. Die haben nämlich den Effekt, dass wenn du als Scrum-Master z.B. nicht im Team bist, sondern nur halt so ins Team angeschlossen dann ist aber die offizielle, Scrum-Master ist dann derjenige, der die Retrospektive auch managen soll, vielleicht auch moderiert, vielleicht auch jemand anderes moderieren lässt, das ist dem Team selbst überlassen, das ist alles nicht vorgeschrieben, aber auf jeden Fall dabei ist. Also da gibt es ja das sind ja Sachen vorgesehen, am Daily-Standup sollte der Scrum-Master am Besten auch mit dabei sein, damit man halt auch schnell Sachen bemerkt, die nicht passen, da wird man vielleicht Sachen ansprechen wie der Build schlägt schon wieder fehl, weil irgend so ein Test flaky ist oder unser Build-Server ist zu langsam, wir brauchen einen schnelleren Rechner, das wäre dann etwas, worum sich der Scrum-Master dann kümmern könnte und wenn der Scrum-Master dann nicht beim Standup dabei ist und nicht bei der Retro dabei ist, dann kann bekommt er das nicht mit und kann seine Rolle dann nicht erfüllen.

Interviewer: Dann sind wir wieder da oben beim Rollenproblem. Dann eine etwas theoretische Frage, was gibt es aus deiner Sicht an Grenzen vom Scrum-Framework, gerade wenn es um die Erkennung geht. Bis jetzt hast du ja gesagt, es ist alles super an diesem Framework. Ich denke, jeder weiß, dass Scrum nicht die Allerwelts-Lösung ist und die Frage ist, wo ist da vielleicht die Grenze von diesem Framework, wo man dann auch vielleicht noch andere Methodiken braucht.

Befragter: Also tatsächlich, genau, wie soll ichs sage, tatsächlich ist das Scrum-Framework, das reine, reine Scrum-Framework sehr klein, bewusst und hat also viel Anpassung, nein nicht Anpassungsbedarf, es ist einfach nur

ein Framework, ein Rahmenwerk, ein Rahmen, der befüllt werden muss mit dem konkreten Projekt, mit der konkreten Arbeit und deswegen lehne sich die Scrum-Berater, zumindest der eine, der eine, der ein zwei, die ich kennen gelernt hab zurück: „Das schreibt Scrum nicht vor!“[T1.43] Wenn man einen konkrete Frage, wie mach ich das in dem und dem Fall, das schreibt Scrum nicht vor, das müsst ihr machen, wie es auch passt. Das ist natürlich eine Stärke, es soll ja anpassbar sein, aber auch eine Schwäche, weil es solche Antworten nicht von sich aus liefert. D.h. aber nicht, dass es dafür nicht geeignet ist, das ist nur so eine Sache, die dazu führt, dass man in der Praxis meint: „Wir machen jetzt Scrum!“ Es aber gar nicht richtig umsetzt[T1.44], weil man da halt noch sehr viel mehr dazu tun muss, zusätzlich und das vielleicht gar nicht versteht, das man da halt noch mehr hinzufügen muss und dann denkt man, Scrum ist nicht geeignet. In der Theorie sollte es aber für sehr sehr viel geeignet sein, wenn man es nur richtig befüllt. Das einzige ist, wo es sicherlich nicht passt, ist wenn die Grundvoraussetzungen nicht stimmen und man gar nicht die Möglichkeit hat, diese Inspect-and-Adapt-Schleife zu fahren, also wenn man von der Organisationsstruktur die Rollen gar nicht besetzten kann, das kann sehr leicht passieren. Wenn man die Aktivitäten nicht durchführen kann, das ist jetzt eher unwahrscheinlich, weil normalerweise das kriegt man schon hin, dass man eine Retro macht, oder ein Standup oder eine Demo, ob man die jetzt gut oder schlecht macht, das ist eine ganz andere Frage, das ist ja jetzt nicht grundsätzlich von Scrum, aber dass man vielleicht gar nicht sinnvoll iterativ arbeiten kann, weil es heißt, wir haben z.B. einen festen Scope und einen festen Zeitraum[T1.45], wir fangen jetzt an und wir wollen genau in 3 Monaten fertig sein, es muss genau das drinnen sein und es ist mir wurscht was ihr währenddessen macht, Hauptsache es ist fertig. Dann kann man zwar Scrum fahren und es wird sich dann aber so wie ein sinnloser, nutzloser Overhead anfühlen. Aber in einem solchen Fall hat man ohnehin verkackt. D.h. Scrum wird einem nicht helfen, es wird einem aber sonst auch nichts helfen, außer ein sehr gut vorausgeplantes Projektmanagement, was dann aber schon alles ganz toll definiert hat. Wir haben ja auch bei UnA über Kanban nachgedacht oder auch über so Mischformen wie Scrum-Ban und mir ist dann auch erst irgendwie später klar geworden, das man durchaus auch Scrum machen kann und dabei ein Kanban-Board verwenden kann. Kanban ist nämlich noch minimalistischer, denn das sagt einfach nur: Visualisiere deinen Fortschritt und achte auf die Auslastung, also auf das Work-In-Progress. Also minimiere Work-In-Progress, um den Durchfluss zu verbessern, zu maximieren. Oh und da kommen wir aber tat-

sächlich zu einem Punkt, der bei beiden Fällen nicht adressiert wird, der einfach nicht im Fokus liegt von denen: Scrum und auch Kanban sind eigentlich Frameworks, Projektmanagementframeworks, Scrum ist ein Projektmanangementframework, Kanban ist eigentlich ein Produktivitätsverbesserungsdings. Das sind eigentlich alle zum Produktivitätssteigerung. Auch Scrum hat den Fokus darauf, dass wir darin besser werden[T1.46], diese Inkremente, einen besseren Output zu liefern, auch einen qualitativ besseren Output, dadurch das wir Retrospektiven machen und häufige Inspektionen, aber auch einen schnelleren Output liefern. Ja, wobei Scrum ist da schon, also bei Kanban ist es auf jeden Fall, da geht es einfach drum, das wir unsere Arbeit so strukturieren, dass wir besser raus kommen. Das tangiert aber nicht die Frage, ist das was wir machen das richtige. Also, da gibts ein anderes Thema, Impact-Mapping, das man halt sagt, was erreiche eigentlich mit dem was ich mache, also was hat es den für einen Wert für den Endkunden, für die Welt oder für wen auch immer. Sprich ich habe hier so eine Anforderung und setze die halt möglichst effizient um aber ist diese Anforderung überhaupt wichtig, oder spielt die mit anderen Sachen zusammen, die vielleicht noch wichtiger wären, also es geht da um Priorisierung, in Richtung Nachfragen, was will ich überhaupt erreichen, das ist eigentlich klassische Anforderungsmanagementsache, was will ich denn überhaupt, warum will ich das denn machen. Wenn er sagt ich will den Button da haben, dann kann ich den Button implementieren in meinem nächsten Sprint in der Demo zeigen, dann sagt er: „Jawoll er ist da.“, aber ich kann auch vorher fragen: „Warum willst du ihn da haben und willst du nicht vielleicht doch eher eine Drop-Down-Box ist das nicht viel besser oder umgekehrt?“ Genau, das ist in Scrum nicht so sehr adressiert, außer dass man im Idealfall die Rolle des Product-Owners so gut besetzt, das der das halt bewerten kann und dadurch und durch die schnellen Feedback-Zyklen, also eigentlich ist Scrum in der Theorie total super, weil es alles adressiert, in der Praxis dann halt doch wieder nicht wegen der Anti-Pattern, die aber in der Praxis immer mit rein kommen. Und Kanban zum Beispiel des sagt halt, mach es so, dass man möglichst effizient, daher kommt es ja von der Produktionsliniensache und da kann halt sowas auf der Strecke bleiben und das kann bei Scrum auch auf der Strecke bleiben, weil es gerne entweder vom Team her kommt einfach nur um so ein bisschen bei der Organisation zu helfen oder wenn es vom Management kommt und sagt wir machen jetzt Scrum, wir sind jetzt agil, dann denken die immer nur an Kostenersparnis, weil sie irgendwo mal gelesen haben, weil dadurch alles besser wird, billiger und schneller und dann wundern sie sich, weil es doch

nicht schneller wird sondern plötzlich langsamer weil die ganzen Probleme gesehen werden, die man vorher ignoriert hat. Genau, soll ich das auch noch hier irgendwo hin schrieben?

Interviewer: Gute Frage, es ist nur die Frage, welche Probleme sie dann ignorieren?

Befragter: Alle die man ignorieren kann natürlich. Genau und das ist natürlich diese Feedback-Zyklen am Scrum sollen ja Probleme sichtbar machen, das ist ja auch so ein Ding, was hier vielleicht auftauchen sollte, ist jetzt vielleicht auch egal. Nee das ist eigentlich das Prozessmittel. Scrum hat eben diese Demos und die schnellen Feedbackzyklen und die sorgen dafür, dass Probleme früher sichtbar werden. So ist es gedacht. Und wenn man das nutzt, sieht man früher Probleme, d.h. wenn man mit Scrum anfängt, stellt man fest, das ist ja alles kacke, weil man die Probleme sieht, die schon vorher da waren.

Interviewer: Ok, lass mich mal gucken, haben wir jetzt noch etwas.

Befragter: Hoffentlich nicht denn es ist ja schon fortgeschrittene Zeit.

Interviewer: Ich würde jetzt auch sagen, das da hinten lasse ich jetzt weg , das war quasi eine Frage, wo man als Scrum-Master einmal besonders gefragt warst oder wo eine besondere Herausforderung hattest.

Befragter: Ja, aber es ist schon so lange her, aber definitiv.

Interviewer: Das ist bei dir jetzt wohl ein bisschen falsch, weil du momentan nicht aktiv in der Tätigkeit drinnen bist. Das ist was für Leute, das man sich einfach noch an Situationen rein versetzen kann.

Befragter: Also da wo ich Scrum-Master gemacht habe und auch noch davor, wo die Person G das gemacht hatte, da war das definitiv so, das wir da mal gefragt waren. Das sind dann halt auch so bestimmte Problemchen mit der Kommunikation zwischen Person F und Person H zum Beispiel oder so Sachen, dass man halt festgestellt haben, wir müssen dann halt doch mal die Standups irgendwie anderes machen.

Interviewer: Wie habt ihr das zum Beispiel, weil zwischen Person F und Person H das ist ja so ein Thema, das müsst ihr ja auch erkannt haben.

Befragter: Das haben wir frühzeitig erkannt, das haben die beiden eigentlich selbst gelöst, weil das sind ja beides sehr ... nicht dominante Personen, also der Person H ist überhaupt auch eine dominante Person, aber beide sind

so im Team-Kontext haben ein hohes Ansehen, einen hohen Stand, also von der Organisation her. Der Person H war ja unser Projektleiter und Chefarchitekt von der Software damals und Person F ist im UnA-Software-Kontext, also die Autorität Nummer eins und denen kannst halt schlecht was sagen. Du kannst ihnen ganz konkret was sagen, was se anders machen sollen. Dem Person F sogar noch mehr, weil er da nicht auch noch so dominant ist und dem Person H schon gar nicht oder teilweise ja schon, aber das muss dann wirklich ganz konkret sein, da kannst du nicht grundsätzlich was sagen: Hey mach das doch mal bitte generell anders. Und sie gegenseitig natürlich auch nicht. Sie haben es natürlich versucht, beide, es war schwierig, dann haben sie sich halt ausgekaspert, interessanterweise war das so, dass die persönlich eigentlich ganz wunderbar miteinander zurecht gekommen sind. Also beim Mittagessen haben sie sich immer ganz wunderbar unterhalten, aber fachlich gab es halt immer Knatsch. Dann haben sie halt immer Sachen ausdiskutiert und haben dann halt beschlossen, sie machen es dann mal so oder so, meistens hat der Person F einfach nachgegeben, nach dem Motto: „Der Klügere gibt nach." und hats dann irgendwie U-Boot mäßig doch anders gemacht*T1.47*. [1]

Interviewer: Ok, das ist dann aber auch nicht im Sinne der Teamleistung, das ist dann ja auch schon ein Problem.

Befragter: Genau und damals, ich glaub da war ich noch kein Scrum-Master, das war alles noch früher, da haben die anderen halt einfach nur versucht, das irgendwie in begrenzte Bahnen zu bringen, in gelenkte und dem einen zu erklären: „Komm, wies der andere macht, ist doch gar nicht so schlecht, der Aspekt ist doch gut." Ja, gebe ich zu, der Aspekt ist gut. Und umgekehrt, das man so auf einen Kompromiss. Ein-, zweimal war es so, da haben sie sich dann im Nebenraum getroffen und haben dann eine halbe Stunde was ausdiskutiert und waren dann halbwegs zufrieden mit dem Ergebnis. Da haben sich beide zumindest aufeinander zubewegt gehabt. Das war ein klassischer Fall von Selbstorganisation, weil es halt zwei so Führungspersönlichkeiten waren, da konntest nicht ernsthaft, noch nicht einmal der Chef konnte da dem Person H was groß sagen.*T1.48*

Interviewer: Das war dann so das Thema Persönlichkeit, wo das her kam.

Befragter: Ja. Genau, da ist es so, das war gar nicht schlechte Kommunikation auf persönlicher Ebene, aber es war halt wirklich so ein Aspekt,

[1] Anmerkung: Der Absatz handelt von einer Situation, bei der es Probleme mit den Persönlichkeiten von Teammitgliedern gab.

Persönlichkeitsmerkmale, das wäre jetzt irgendwie dominante, das wäre jetzt ein Merkmal, das dazu geführt hat, dass die eine Person sehr durchsetzungsstark[T1.49]. Das ist ja ein positiver Aspekt und Persönlichkeitsmerkmal das von einem Architekten erwartet wird. Das hat auch oft Vorzüge gehabt, in dem Moment vielleicht gerade nicht so offensichtlich waren, wenn er dir halt irgendwas sagt, nee, das iss aber blöd, und dann sagt er: „Du machst das jetzt so!“, dann grummel und dann stellst du hinterher fest, nee so blöd war es dann gar nicht. Und an manchen Stellen wars dann halt doch blöd. Genau, das sind Persönlichkeitsaspekte. Und wenn es dann Richtung Agilität braucht Führung geht, dann ist das so ein Aspekt, wo eine Führungskraft solche Dinge kennen und beachten muss.

Interviewer: Ja, aber kann sie das dann auch bemerken? Gut, bei dem Konflikt hat man das mit Sicherheit auch von außen gesehen?

Befragter: Das hat man gesehen.[T1.50] Ja, da geht es eigentlich aus dem agilen schon wider raus. Generell, wenn man in einer Art Führungskraft-Rolle ist, z.B. als ermächtigter Scrum-Master, ein Scrum-Master kann auch einfach völlig machtlos sein, aber es kann ja auch sein, dass er wirklich sagt so, du darfst Sachen, was weiß ich, zur Chef-Etage bringen oder so, nee ich will das jetzt bitte anders gestalten, oder wenn man auch zufällig akzeptiert ist vom Team als jemand der Entscheidungen treffen kann oder so, es gibt ja auch oft, dass ein Projektmanager, der irgendwie Befehlsgewalt hat, zum Scrum-Master umdefiniert wird, was dann zu manchen Anti-Patterns führt in der Richtung, aber der hat dann wenigstens nicht das Ermächtigungsproblem. Und dann muss man auf solche Sachen achten. Z.b. in Retros und in Standups.

Interviewer: Ok, dann danke ich dir für das Interview.

Notizen des Befragten

Während des Interviews wurde durch den Befragten 2 Schaubilder angefertigt, die er während des Interviews zum Teil umsortierte. Diese sind in Abbildung C.1 und Abbildung C.2 zu finden.

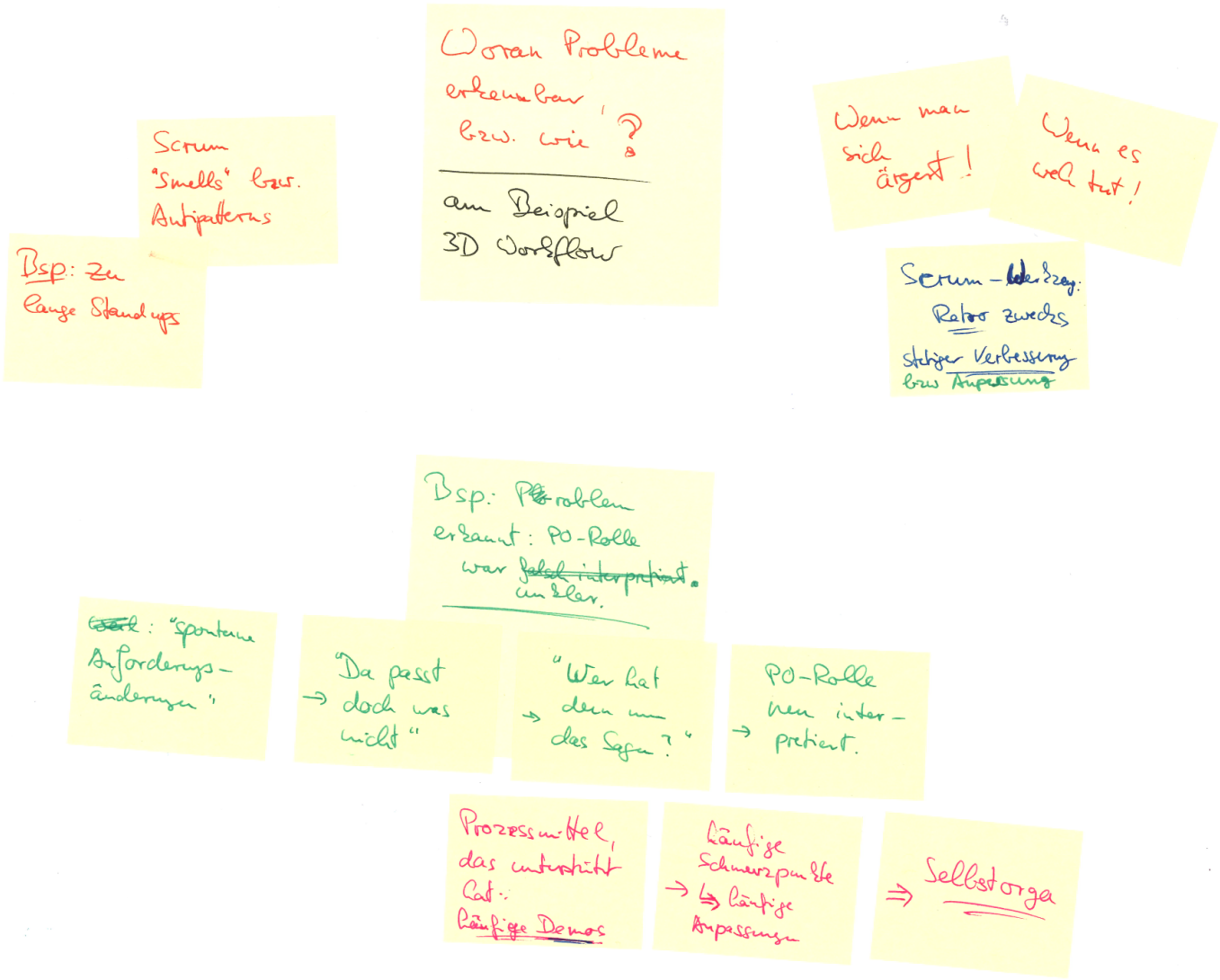


Abbildung C.1: 1. Blatt vom 1. Interview



Abbildung C.2: 2. Blatt vom 1. Interview

C.2 Interviewtranskription vom 2. Interview

Datum 11.08.2016

Fachlicher Hintergrund Der Interviewpartner ist bei einem Softwaredienstleister angestellt, der Kunden bei ihrer Softwareentwicklung durch Consulting unterstützt. Der Fokus seiner Beratertätigkeit liegt dabei auf den Teams und deren Zusammenarbeiten während der Softwareentwicklung. Er begleitet Veränderungsprozesse aus Sicht des Teams und berät das Management währenddessen. Er hat langjährige Erfahrung sowohl mit klassisch arbeitenden Teams, als auch seit ca. 6 Jahren im Coaching mit agilen Teams. Neben dem Studium der Informatik hat er zusätzlich einen Bachelor in Psychologie erworben.

Interview

Interviewer: Hallo, ich würde von dir gerne den Hintergrund erfahren, zum einen wie lange du dich mit Agilität beschäftigst, wie dies bei dir begonnen hat, welche Hintergründe du dazu hast und was du schon für Aufgaben, Rollen in diesem Kontext übernommen hast.

Befragter: Meine erste Berührung mit agilen Geschichten, konkret mit Scrum hatte ich um 2004, wo ich selbst meine damalige Diplomarbeit geschrieben habe bzw. ich habe mir überlegt, ob ich über das Thema schreiben möchte, habe mich dann allerdings dagegen entschieden und bin dann eher bei den klassischen Prozessen, RUP raus gekommen; hatte dann erst einmal eine Weile relativ wenig damit zu tun und bin dann so im Jahre 2008, war ich bei einem Kunden unterwegs im Automobilbereich, wo ich damals recht viel mit Projektmanagementmethoden zu tun hatte und mein Eindruck war, bei ihnen würde Scrum super reinpassen, das würde ihnen helfen, viele Probleme zu beseitigen und habe ihnen dies damals vorgeschlagen, es vorgestellt und diskutiert; im Endeffekt wurde es dann vom mittleren Management abgelehnt, also das Team bzw. die Teams wollten dies alles, es ging um einen größeren Laden, beteiligt waren so zwei, drei Teams mit denen wir dies besprochen haben, die wollten das, die fanden das super, ein paar kannten das auch schon von vorherigen Arbeitgebern und wollten das gerne eingeführt haben, aber das mittlere Management hat sich damit unwohl gefühlt und hat das sozusagen abgeblockt. Ich weiß aber inzwischen sind sie teilweise auch

schon dabei Sachen in die Richtung zu machen, aber es hat damals nicht funktioniert. Das war so für mich das erste Projekt, ich bin drauf gekommen über mehrere Kollegen hier, die sich mit dem Thema hier auseinandergesetzt hatten und wo es hier teilweise Vorträge und Schulungen mit gab, hab mich dann, ab 2010 war das, deutlich intensiver mit beschäftigt, ich glaub es war 10 oder 11, da war ich bei einem anderen Automobilzulieferer unterwegs und da ging es explizit drum, eine Methode, die sie entwickelt hatten mit einem eigenen Werkzeug das sie entwickelt hatten, weltweit auszurollen. Also sie haben das im Prinzip hier an einem Standort schon umgesetzt, haben da sozusagen eine Adaption an ihr Automotive-Umfeld gebaut, die ich bis heute für sehr sehr vernünftig und sinnvoll finde und haben dafür auch ein selbstgeschriebenes Werkzeug für das ganze Thema Planung gebaut, also Planung plus Forecast, Metriken; letztendlich haben sie es aus einem Excel mit einer Server-Schnittstelle hinten dran gebaut, sprich man konnte die Daten direkt auf einen Server hoch bzw. runter laden, direkt in Excel, was für die ganzen Manager super angenehm war, weil die

Interviewer: ... kennen das alle ...

Befragter: ... genau und Excel hat auch ein paar nette Eigenschaften, die sehr einfach und übersichtlich sind. Sie haben sich damals auch ganz bewusst gegen Jira als Werkzeug entschieden. Genau, und da ging es wirklich bei mir drum vor allem zu schaun, was muss man am Werkzeug machen bzw. auch was sind so die Methoden, Ideen, Konzepte hinten dran und das zu versuchen in zwei andere Divisions zu tragen, die eben auch weltweit Standorte hatten, d.h. ich war da auch in Japan, Moment bei dem Kunden war ich in China, in Krakau, also Polen, Schweden, USA unterwegs.

Interviewer: ... d.h. du hast das auch international auch, ich sage jetzt mal im kulturellen Umfeld dann auch erlebt.

Befragter: Ja, jetzt zwar nur in jeweils kurzen Ausschnitten, aber ja. Die habe ich durchaus mitgekriegt und da würde ich auch sagen, das passt nicht überall so super gut hin, also je nach Kultur passen verschiedene Aspekte von Agilität besser oder schlechter[T2.1]. Also je autoritärer eine Kultur ist, desto weniger passen halt auch solche agilen Vorgehensweisen mit Delegationskonzepten sehr gut rein[T2.2]. Aber auf den Kulturaspekt können wir gerne noch einmal separat eingehen, weil das ist ein großes Thema.

Interviewer: Ja aber das finde ich interessant, weil das habe ich natürlich bis jetzt von wenigen gehört; klar dazu muss man ein Stückchen rumkommen, dass man das betrachtet, aber das habe ich mir jetzt schon angemarkt.

Befragter: Können wir gerne näher drauf eingehen. Genau, so in diesem Rahmen habe ich mich auch generell in das Thema mit eingearbeitet und hab dann damals meinen Scrum-Master während dieser Zeit gemacht gehabt und hab dann dort im Team auch ein bisschen verschiedene Rollen eingenommen und hab mich danach in verschiedenen anderen Projekten auch immer wieder Richtung Coaching-Technisch weiterentwickelt, war dann im wesentlichen bei Einführungsprojekten mit dabei, wo auch in einem weiteren Projekt ein größerer Automobilhersteller und Zulieferer auch agile Methoden bei sich einführen wollte und dort 2 Teams hatte, wo ich sehr viel getan hab, beteiligt waren auch ein paar Kollegen, die dann versucht haben das Management abzuholen und Stichwort: Transition-Team zu bilden, wir reden hier von einer Größenordnung von 2000 Leuten, wie man die halt umstellen kann, weil ich war dort vorwiegend mit 2 Teams beschäftigt, die von der klassischen Entwicklung immer noch in einem klassischen Umfeld sozusagen in Richtung einer agilen Entwicklung weiterentwickeln wollten. Noch weitere Kunden, wo ich seit dem unterwegs bin, wo ich dann auch diverse Einführungsgeschichten habe, also es sind inzwischen noch so drei, vier weitere Projekte. Im wesentlichen der Normalfall ist immer der Punkt, wo ich hinkomme, ist das die Leute schon irgendwas vorher erleben, allerdings noch nichts richtig agiles und dann versuche ich mich mit den Teams zusammen zusetzen, wie kann man das sinnvoll umbauen bzw. in manchen Fällen auch mit dem Management zusammen setzen, wie muss man das ganze System aufsetzen, um das umzusetzen und jetzt gerade habe ich noch ein sehr spannendes Projekt hinter mir, wo ein Kunde, der schon seit etwa einem Jahr angefangen hat auf agile Methoden umzustellen, jetzt mal von außen jemanden haben wollte, der sagt, schau doch mal bitte drauf und sag uns, ob das was wir hier machen in die richtige Richtung läuft oder wo sind eben die Sachen, wo es dann Änderungsmöglichkeiten oder Bedarf gibt um sozusagen da auch ein bisschen ganz bewusst aus dem eigenen Saft raus zu kommen.

Interviewer: Um sozusagen noch eine dritte Sichtweise zu bekommen. Was schätzt du, wie viele verschiedene Teams du schon betreut hast?

Befragter: Ich komme wohl so auf ungefähr 5-6 Kunden und etwa 10 Teams, so in der Größenordnung. So in etwa; das kann man sicher so grob durchrechnen, für ein Team braucht man ein halbes Jahr, so ein drittel bis halbes Jahr zur Betreuung, und wenn ich seit 2011 im wesentlichen in diesem Thema unterwegs bin mit noch ein paar Kleinigkeiten die ich auch noch nebenher mache, dann hast 5 Jahre, d.h. da sind maximal 10 Teams, die da sinnvoll zu betreuen sind. Es kommt ein bisschen darauf an, bei den internationalen

waren es tatsächlich mehr Teams, die ja hinten dran standen, aber mit einigen davon hatte ich ja sehr wenig Kontakt, d.h. dort habe ich eine Vorstellung gemacht, die Sachen kurz erzählt und dann war ich weg. Da war dann keine große Betreuung da.

Interviewer: Da hast du dann auch keinen tieferen Einblick bekommen ...

Befragter: Da kann ich dir auch nicht sagen, ich weiß in manchen Fällen dann nach zwei Monaten das die Leute es wieder sein haben lassen, ich weiß bei anderen, dass sie es weitergelebt haben, manche es sich ein bisschen angepasst, aber ich kann dir nicht genauer sagen, was in dieser Zeit passiert ist.

Interviewer: Ok, ich glaube das reicht mir jetzt so als einen guten Überblick ergeben. Gut, dann wollte ich ein wenig ausloten, was gibt es denn für negative Entwicklungen, gerade dich befragen, was denkst du so aus deinem Hintergrund heraus, was kann sich negativ auf die Leistung von so einem agilen Team auch auswirken, also da wäre es gut, wenn du ein wenig mit der Zettelwirtschaft arbeitest, wenn du möchtest, dass man da halt, ich sag jetzt mal so, alles was dir einfällt da auf dem Zettel schreiben, am Besten auf das Blatt Papier.

Befragter: Also was ich erstmal grob unterscheiden möchte sind die Einflussfaktoren, ob sie von innen oder außen kommen. Also Thema ist dir jetzt sozusagen, Team, du kürzt es jetzt auf Teamzufriedenheit oder Teamstimmung,

Interviewer: eigentlich die Leistung, das ist eigentlich, ich sag mal, die Firma interessiert sich ja oft nur, ja das hört sich jetzt blöd an, aber ich muss, ...

Befragter: ja, das sind aber für mich ganz ganz wichtige zu unterscheidende Punkte ...

Interviewer: ich sag jetzt mal so, als Firma interessiere ich mich ja, meistens das das Team, also mir ist eigentlich ziemlich egal, wie das Team arbeitet, ich möchte natürlich schon irgendwie die Leute zufrieden sind; warum mache ich das: Ich möchte ja eine gewisse Leistung von diesem Team haben. Also das ist so ein Schluss, in dem ich auch in meinen Voruntersuchungen dazu gekommen bin und deswegen habe ich mich damit auch beschäftigt. Leistung messen ist ja schwierig aber man hat ja trotzdem irgendwie so einen Begriff oder ich sage nennens mal, viele Leute haben ein Gefühl dafür: Mensch wir sind jetzt gut oder wir sind schlecht in der Leistung, also wir sind im Perfroming oder auch nicht, deswegen habe ich dies auch so ein wenig auf das

Leistungsthema, weil das ist ja eigentlich auch, ich nenne es für den Vorgesetzten oder Unternehmen primär interessant ist.

Befragter: Also meine Aussage ist, wenn ein Kunde zu mir sagt, er will Agilität einführen, um typischerweise krieg ich das Stichwort Effizienz zu erhöhen, was ja letztendlich für Leistung ein zumindest ein Teil des Begriffes ist und der am Meisten vom Management am stärksten zu beachten ist, dann sag ich ihnen: „Nö, das kriegt ihr nicht, lasst es bleiben." Mit Effizienz will ich nix am Hut haben. Agile Methoden führe ich aus anderen Gründen ein. Ich führe agile Methoden deswegen ein und ich will agile Teams aus den Gründen haben, das ich, wenn dann die Effektivität hoch kriege, dass sie weniger falsches machen. Das Team wird nicht mehr arbeiten, aber sie werden mehr am richtigen arbeiten[T2.3], d.h. sie werden mehr an den entscheidenden Punkten arbeiten. Sozusagen dann das zweite ist schlicht und einfach zufriedene Mitarbeiter zu haben. Die werden nicht unbedingt mehr arbeiten für sich genommen, aber die werden weniger wechseln, die werden weniger Fehlzeiten haben und damit natürlich, wenn ich sozusagen die Wechsel mit berechne, wenn ich jemanden 10, 12, 15 Jahre im Unternehmen halten kann, hat er für mich im Normalfall einen ganz anderen Wert, als wenn alle 3 Jahre die Leute weg sind[T2.4] und dieses dauerhaft über längere Zeit die Leute einfach eine Umgebung zu schaffen, in der die Leute sich so wohlfühlen, dass sie trotz vielleicht Projektproblemen oder sonstigen Dingen, die immer da sind, dabei bleiben und motiviert sind, weiterhin motiviert arbeiten, nicht diese innere Kündigung haben und die Zeit absitzen, das ist für mich ein entscheidender Punkt und der 3., was tatsächlich der Hauptgrund ist, weswegen die Kunden eigentlich das Thema bei mir einführen ist das Thema, ja responding zu Change[T2.5], die Umgebungsbedingungen verändern sich und die verändern sich immer schneller; d.h. sowohl die Marktanforderungen, also auch die technologischen Anforderungen verändern sich im Moment in einem Zyklus mit dem andere Methoden und Arbeitsweisen nicht mehr hinterher kommen. Das Team wird nicht mehr leisten, aber es ist in der Lage, schneller auf beispielsweise neue Technologien zu reagieren, es ist in der Lage schneller mit einem gewissen Komplexität, Stichwort: Je höher sich das ändert wird auch die Komplexität und überhaupt noch mit dieser Komplexität zurecht zu kommen, das ist was, was ein Team macht und es gibt noch einen Punkt, wo das Thema Leistung tatsächlich ein bisschen rein kommt, das ist die Qualität der Outputs. Die Qualität der Outputs sollte bei einem gut funktionierenden Team, agilen Team hoch sein und im Schnitt auch höher, als es bei anderen Teams ist, was aber nicht sozusagen hoch geschraubt wird, das

kann mal ein Nebeneffekt sein, es gibt auch Fälle bei denen es sicher passiert ist, aber es ist nicht der Default-Fall, dass das Team plötzlich viel mehr leisten wird, plötzlich viel mehr produzieren, mehr Ergebnisse machen wird, es wird eher die richtigen und bessere Ergebnisse produzieren[T2.6].

Interviewer: Das meinte ich jetzt auch nicht, ich meinte mit Leistung eigentlich das Gesamtkonzept, das es auch das richtige ist, was sie machen, in einer akzeptablen Zeit hinbekommen und auch auch gerade das mit der Qualität ist für mich auch Leistung, denn nicht nur die Masse machts. Das meinte ich schon mit Leistung; ich brauch da halt irgendeinen Begriff, wie man sich das vorstellen kann.

Befragter: Ich hack nur drauf rum, weil das ist tatsächlich was, was ich ganz typischerweise als aller erstes dem Management bei den Projekten klar zu machen und wenn ich merke, das sie immer noch auf der Effizienzschiene sind, dann kann ich hier von vornherein sagen, das die Projekte nicht funktionieren werden, die werden nicht sehr agil werden. Da ist auch ein bisschen das Thema Kurz- vs. Langfristig ein ganz wichtiger Unterpunkt. Weil ich habe jetzt ein Team, ein Team das ich mir näher angeschaut habe, die haben aus meiner Sicht sehr gut verstanden, was sie da gerade tun und die haben jetzt in diesem Team das Thema Crossfunktionalität ganz ganz extrem gepusht.

Interviewer: Was verstehst du unter Crossfunktionalität?

Befragter: Also Crossfunktionalität, die klassische Corssfunktionalität[T2.7] heißt beispielsweise das Gegenteil davon wäre, du hast einen der schreibt die Anforderung, du hast einen, der entwickelt für Sprache was weiß ich den Client, der andere entwickelt für den Server und der Dritte macht die Tests. Das ist etwas, das haben sie sehr stark gepusht das aufzulösen, insbesondere zwischen den verschiedenen Technologien und Projekten, das sie also von diesen Elfenbeintürmen, einer ist der Experte für ein bestimmtes Thema für ein bestimmtes Projekt wegkamen, hin zu sie haben eine relativ homogene Gruppe. Das heißt nicht, dass jeder das gleiche Wissen hat, aber d.h. wenn du einen aus der Gruppe raus nimmst, dann wird die Gruppe immer noch ziemlich gut funktionieren, weil immer noch genügend andere den einen ersetzen können. Das war davor nicht der Fall; da war es so, sie hatten gewisse Technologien, da gab es genau einen, der kannte sich damit aus und wenn der im Urlaub war, dann ist in diesem Bereich nichts passiert. Und wenn der vom Laster überfahren worden wäre, dann hätte man ein gigantisches Problem gehabt, weil es dann erst einmal 2 Leute ein halbes Jahr gebraucht hätten, um sich einzuarbeiten. So kurzfristig war das Ergebnis, dass die Leis-

tung von diesem Team runtergegangen ist. Die haben weniger Output geliefert und die Ergebnisse waren nicht unbedingt besser als zuvor, die waren teilweise sogar schlechter. Nach einer gewissen Zeit, nämlich jetzt nach ungefähr einem halben, dreiviertel Jahr ist es so, dass sie dadurch viel flexibler sind. Sie können jetzt besser darauf reagieren, wenn in irgendeinem Themengebiet eine neue Zusatzaufgabe, so die klassische Überlastsituation: Im einen Gebiet kommen plötzlich noch ein paar Anfragen rein und jetzt sind sie in der Lage, dass andere das abfangen, wenn der eine, der halt dafür der Experte ist, die gibt es immer noch bis zu einem gewissen Grad, wenn der Experte gerade keine Zeit hat oder überlastet wäre, können andere das abfangen. Das konnten sie vor einem halben Jahr nicht. Dadurch können sie auf Änderungen, auf Requests von außen, auf zusätzliche Anfragen viel besser im Team Sachen hin und herschieben und damit auch dementsprechende Sachen reagieren. D.h. aber nicht, dass die Performance, die Leistung, der Output in irgendeiner Form nach oben gegangen ist.

Interviewer: Ja, da gebe ich dir Recht, aber natürlich ist die Leistung für das eine Projekt, ist jetzt, wo es dann gar nicht vorangegangen wäre, das ist natürlich gestiegen im ..

Befragter: Das ist das Stichwort Risikomanagement.

Interviewer: Genau, das Risiko ist dann einfach gesunken.

Befragter: Genau. Es ist im Prinzip eine Risikominimierung an der Stelle passiert.

Interviewer: Dann reden wir nicht mehr von Leistung.

Befragter: Ja, genau deswegen, aber ich hab glaube ich grob verstanden, worauf du hinaus willst.

Interviewer: Ok.

Befragter: Also, zunächst mal gibt es aus meiner Sicht im wesentlich da 2 Einflussgrößen, die man unterscheiden muss beim Team. Das eine, was ich als Kontext bzw. Umgebung bezeichnen würde, also in dem das Team sozusagen arbeitet und das andere ist sozusagen das Innere, dass das Team als sich, das Team selbst, also aus von innen heraus und vorn außen heraus gibt es da Einflussgrößen, die da wirken und Effekte haben können und die muss man da auch sehr deutlich trennen. Das ist auch ein bisschen gerade an Scrum, aber auch an einigen anderen von den agilen Ansätzen, die fordern ja sehr stark, sie haben sehr starke Anforderungen an den Kontext, der muss etwas

leisten, sonst kann das Team gar nicht super arbeiten. Und sie fordern, sie stellen sehr stark diese Forderungen, stellt uns bitte diesen Kontext her, sonst haben wir hier ein Problem. Beispiel, da sich nenne, das ist ein Beispiel, das ich bei allen Kunden erlebe und bei allen Kunden erstmal schwer fällt, zu sagen, Scrum sagt, ich habe ein Team und ich arbeite an diesem einen Projekt; Punk. Jemanden in 2 Scrum-Teams gleichzeitig drinnen zu haben geht nicht, das ist kein Scrum-Team mehr. Jemand, der unter 80% für ein Projekt verfügbar ist, ist ein Teilzeitteammitglied, jemand der unter 60% verfügbar ist, ist kein Teammitglied mehr. D.h. ich kann keine Leute in 2 bis 3 Scrum-Teams rein setzen. Das ist aber das, was im ganz normalen klassischen Projektmanagement ständig gemacht wird[T2.8]. Leute werden irgendwie werden kreuz und quer über 7 Projekte, also ich habe Kunden die haben teilweise 13, 14 Projekte parallel und springen von einem in das andere hin und her. Und das ist eine Anforderungen aus Scrum heraus, aber ich würde fast sagen aus allen agilen Methoden, wir können auch ein Kanban nehmen mit einem Work-In-Progress-Limit, die sagen ich reduziere das, was ich gleichzeitig tue. Kein Mensch kann sinnvoll in mehreren gleichzeitig arbeiten, ansonsten springe ich die ganze Zeit hin und her; ich kann das an manchen Stellen ein bisschen zuarbeitend tun, ok, aber nicht mich hauptsächlich darauf zu konzentrieren. Und das wäre sozusagen von den Kontextfaktoren einer, der ganz klar gefordert ist, sprich das Thema Verfügbarkeit bzw. ich finde den Ausdruck aus dem Kanbahn den WIP ganz schön, den kann ich auf verschiedene Ebenen setzen, aber als eine einzelne Person getroffen kann ich immer nur an einem Projekt zeitgleich arbeiten. Alles andere was ich mache führt zu Ineffizienzen. Das ist natürlich auch was, was ganz klar zur Motivation beiträgt, umgekehrt ist es, das erlebe ich selbst an mir auch, dieses Springen, dieses ständige Springen zwischen mehreren Projekten führt zu zusätzlicher kognitiver Last und das führt halt dazu, das man irgendwann aus- bzw. überlastet ist und Überlast, da kann ich dann wieder die ganzen Stresstheorien anwenden, d.h. das führt dann wieder zu Burnout, Motivationsverluste, usw. alles hinten dran[T2.9].

Interviewer: Da sinkt dann auch im Endeffekt die Leistung, das muss man ja auch mal sagen, es gibt ja auch Untersuchungen, wenn man an mehreren Sachen gleichzeitig arbeitet, dass dann die Leistung irgendwo nach unten geht.

Befragter: Das was ich kenne ist so, dass man am effektivsten ist, wenn man nach Aufgaben geht bei zwei Aufgaben ist. Bei einer hat man 80%, bei zweien 100% und dreien noch irgendwie 60-70% und dann gehts massiv runter.

Naja, bei zweien habe ich die Umschaltzeiten, manchmal muss ich über was nachdenken, bevor ich wieder an was weiterarbeiten kann, in der Zeit kann ich das andere zweite Projekt bearbeiten. Das ist aber auf Aufgaben bezogen und nicht auf Projekte. Weil in einem Projekt habe ich ja meistens auch 7, 8 Aufgaben.

Interviewer: Da gab es auch eine Veröffentlichung dazu.

Befragter: Genau, das wäre dann auch einer der Kontextfaktoren gewesen. Ich kann mal kurz durchgehen; ich habe da auch ein psychologisches Modell, was ich auch immer ganz gerne meinen Kunden zeige, was aus meiner Sicht sehr stark hinter diesem psychologischen, es nimmt nicht alles, aber einige Aspekte, die in agile Methoden eingebaut sind; basierend aus meiner Sicht auf diesem Modell letztendlich, ob die Leute das wussten oder nicht wussten sei mal dahingestellt, aber es passt sehr gut zusammen. Sekunde. Job-Characteristics-Modell[T2.10] nennt sich das. Mal schaun ob ich es schnell gefunden kriege (Anm.: Sucht an seinem Laptop nach diesem Modell)

(*Anm.: Hier wurde ein für die Arbeit nicht relevanter Teil entfernt, da er persönliche Details zu Personen enthielt.*)

Befragter: Hackman and Oldham[2] und es ist schon aus den 70er Jahren und ist damit nichts, was irgendwie fancy, neu oder modern ist und relativ alt und gut wissenschaftlich bestätigt und was man eben hat und das ist das was sozusagen man in jedem Kontext haben will sind die Outcomes, das sind sozusagen die Ergebnisse die man gerne haben möchte, was wahrscheinlich der Begriff Leistung subsumiert.

Interviewer: Genau, der das zusammen fasst, (Abgelesen von der Präsentation auf dem Laptop) high internal motivation, high quality performance, high work satisfaction, low absenteeism, low turnover; was letztendlich natürlich dazu führt, dass der Mensch eigentlich schon motiviert wird er auch mehr leisten. Zumindest bessere Qualität abliefern.

Befragter: Das möchte ich im Prinzip haben, als Manager sollte ich mich bemühen, das zu kriegen. So, was man dazu braucht sind entspreche psychologische Zustände, nämlich die Leute müssen der Arbeit eine Bedeutung zumessen, sie müssen sich für die Ergebnisse verantwortlich fühlen und sie müssen auch die Ergebnisse tatsächlich sehen und mitkriegen, was die sind[T2.11]. Und dementsprechend gibt es dann Eigenschaften des Jobs, die man

[2]Es handelt sich beim gezeigten Modell um das Job-Characteristics-Model aus Hackman und Oldham 1976.

dafür haben muss, das im Prinzip das zustande kommt; das ist es das Spannende des Modells, dass die 3 Bereiche sind multiplikativ, d.h. wenn einer davon scheiße ist, dann können die anderen zwei noch so toll sein, es bringt mir nichts während die 3 hier untereinander additiv sind. Beispiel für so was wäre, wenn ich jetzt irgendwo einen Job habe, meinetwegen in einer Flüchtlingsunterkunft und ich habe einen scheiß Job, wo ich nur Essen ausgeben muss, das ist jetzt kein sonderlich toller Job, der hat keine hohe Skill variety, ich brauche nicht allzu viele Fähigkeit, Ganzheitlichkeit, es ist ein Schritt aus dem Ganzen, ich sehe nicht das große ganze, aber er hat eine unglaublich hohe Bedeutung, denn wenn ich das nicht tue dann sterben da Leute oder verhungern Leute vor mir. Insofern sind die auch austauschbar, weil in diesem Fall ist die Bedeutung so hoch ist, dass sie sich ersetzen. Während es auch anders ist, wenn ich einen Job habe, wo ich irgendwie am Ganzen beteiligt bin, wo ich alle meine Fähigkeiten einbringen kann, das Ergebnis ist am Ende aber gar nicht mal so bedeutend, häufig in Forschungsbereichen z.B. wo ich sag ich habe da so ein kleines bisschen an dem ganzen Riesending irgendwo mitgemacht aber ich hab da alles von mir gefordert, dann kann es das ausgleichen. Während so etwas wie Autonomie und Feedback nicht ersetzbar sind. Autonomie: Ich muss einen möglichst großen Einfluss auf die Art und Weise wie ich meine Arbeit mache haben d.h. wenn mir die sehr stark vorgegeben ist, ganz klar mach den Schritt dann den Schritt, den Schritt und ich kann nichts daran ändern selbst wenn ich merke, das es blödsinnig ist, da habe ich kein Autonomie[T2.12]. Das geht dann auch auf zeitliche Autonomie, ich kann nicht Arbeiten wann es für mich am sinnvollsten ist oder am Besten passt, ich kann auch nicht mit anderen Abteilungen reden z.B. von einem großen Automobilhersteller, die haben auch mal so eine Angebotspräsentation hatten zum Thema Agilität, da kam eben auch die Aussage ja, wo ich meinte hier Crossfunktionale Teams, es muss ganz klar sein, die Entwickler und die Testteams müssen zusammenarbeiten können. Die meinten dann: „Ja Moment halt die sind zwei ganz verschiedene Hierarchien die können nicht direkt zusammen arbeiten."[T2.13] Ja, das ist mir eigentlich egal, was die Hierarchien sind, wie Moment halt, Hierarchien. Das ist so ein Beispiel, wo man die Leute dürfen nicht miteinander reden in den Abteilungen neben dran, weil sie dann über die Hierarchiestufen hinweg gehen müssten, dann habe ich keine Autonomie in deren Bezug.

Interviewer: Da wird man dann immer wieder blockiert, verstehe.

Befragter: Genau und d.h. dieses Thema Autonomie brauche ich und ich brauche Feedback zu meinen Ergebnissen, im Idealfall so schnell wie mög-

lich ein Kundenfeedback, wo ich merke da habe ich das richtig gemacht oder das Falsche. Das muss kein gutes Feedback sein, es muss aber ein möglichst präzises inhaltliches Feedback sein[T2.14], wo ich sehe, ob was ich gemacht habe das richtige oder war es was falsches, was war gut daran, was war schlecht daran. Das schlimmste was passieren kann im Bezug auf Feedback: „Ah ja vielen Dank für die letzten 4 Wochen von Arbeit, ich brauche es leider nicht mehr." Das ist dann halt das worst-case, aber auch wenn es dann nur rumliegt und keiner möchte gar kein Feedback davon. Ich schick meine Sachen irgendwo hin, ich hab was gemacht, einen Bericht erstellt schick den irgendwo hin und krieg gar kein Feedback oder ein halbes Jahr später. Das ist sehr schlecht. D.h. sehr zeitnahe möglichst inhaltlich präzises Feedback hilft mir meine Ergebnisse zu verstehen und auch darüber eine Möglichkeit zu schaffen, dass ich mich selbst verbessern kann[T2.15]. So das sind eigentlich die Eigenschaften und wenn ich mir jetzt agile Modelle anschaue, jetzt Scrum als der Klassiker, aber kann man auch auf andere verwenden, was ich da dann hab, ich hab das corssfunktionale, deswegen sich Sachen wie skill variety, ich habe sozusagen das Thema Ganzheitlichkeit (task idendity), wo ich sozusagen nicht nur einen Schnipsel mache, sondern wir haben am Ende ein Ergebnis und ich hatte auch Scrum-Teams, die sich teilweise aus Hardware, Software, Mechanik zusammensetzten und die hatten halt nach jedem Sprint tatsächlich einen Prototypen mit einem 3D-Drucker ein neues Gehäuse gebaut, hatten ein Sample von der Elektronik mit eingebaut und haben zumindest ein paar Screenshots angezeigt. Vielleicht funktioniert das Programm noch nicht, aber die haben zumindest das Ding schon einmal zeigen können, schau mal her hier, in unser Produkt verbaut sieht das ganze Ergebnis so aus. Die waren an dem Ganzen beteiligt, haben das Ganze auch gesehen, diese task significane, gut das hängt jetzt ein bisschen davon ab, in welchem Bereich man unterwegs ist, da können die agilen Methoden nur bedingt etwas dafür tun, das ist eher der Kontext für relevant. Beispiel: Ich hatte einen Medizintechnik Hersteller, wir haben hier gerade Operationsmethode, die wir hier entwickeln, die dazu führen, dass jemand, der in Zukunft eine Augen-OP hat, nach einem Tag wieder nach Hause kann. Bisher musste er mindestens 20 Tage mit verbundenem Auge im Krankenhaus bleiben. Und da merkst du dann wenn die Leute sagen: „Hey an so was arbeite ich, ich bin ein Teil davon!" Da ist natürlich eine sehr, sehr hohe Eigenmotivation da. Wenn ich dabei einen anderen nehmen, der meinetwegen Waffen herstellt: „Hey cool, ich entwickle Waffen, dann können wir noch mehr Leute umbringen." Das wird nie ein Grund sein.

Interviewer: Ja, das kommt natürlich ganz auf den Job an.

Befragter: Genau, da ist es wirklich die Frage, des dessen was ich tue und habe ich wirklich einen Job, wo am Ende was raus kommt und ich merke es bringt der Welt eine Verbesserung. Es macht etwas, ich habe etwas bewegt.

Interviewer: Ja.

Befragter: Was man natürlich in den agilen Methoden teilweise drinnen ist, das dafür mehr Bewusstsein dafür geschaffen wird, da man natürlich auch teilweise mit dem Kunden interagiert, den Kunden auch sieht und mehr mit kriegt. D.h. durch solche Methoden krieg ich dadurch auch wiederum mehr Bezug, kriege ich mit wie er Sachen einsetzt und welche Bedeutung es für ihn hat. Insofern indirekt haben wir das auch wieder dabei. Autonomie ist ja auch eine der Kontextfaktoren, da kann man das gleich mal mit dazu schreiben, als eines der ganz wichtigen Dinge. Ich muss letztendlich dem Team als solches eine sehr hohe Autonomie zubilligen, was in der Praxis wieder schwer ist, weil ich muss da zwei Sachen ausbalancieren: Nämlich es gibt dieses Stichwort Autonomie und Alignment[T2.16]. Heißt ich kann ein Team nur dann autonom arbeiten lassen, wenn ich davor ein sinnvolles Alignment habe. Sprich ich muss mir als Unternehmen über die Zielrichtung, „Wo wollen wir eigentlich hin?" vorher im klaren sein. Nicht über die Details aber im groben. Ich muss über den Kontext geteilt aber inklusive dem Team muss ich sozusagen eine gemeinsame Vorstellung haben, wo wollen wir eigentlich hin, und dann kann ich die Leute loslaufen lassen. Den Weg finden sie selber, den muss ich nicht vorgeben. Aber wenn wir 3, 4 verschiedene Vorstellungen haben, was eigentlich das Ziel ist, die sozusagen ganz unterschiedliche Ideen haben wo wir eigentlich hin wollen und Umfeld haben und dann sagt man dem Team: „Jaja, lauft ihr mal!" Und von draußen steht der eine Manager: „Ich möchte aber hier hin gehen", der Kunde sagt aber: „Ich möchte da hin gehen." und ein Produktmarketing sagt: „Ich möchte aber in die Richtung gehen.", dann ist das Team hilflos, völlig Überfordert und wird sich extremst unwohl fühlen und wird dann eben auch durch die Freiheit in diesem Fall frustriert werden. Dann wäre es besser, sie hätten weniger Autonomie als das das ich ihnen eine Autonomie gebe, aber nicht die Voraussetzungen schaffe, damit um zu gehen.

Interviewer: Da ist dann vielleicht auch die Frage, was ist hier unter Autonomie genau zu verstehen, wie Autonomie, dass ich alles selber festlegen darf, die Autonomie der Arbeitsweise vielleicht, dass war jetzt mehr die Autonomie, wie das Endprodukt aussehen soll, wo es hingehen soll.

Befragter: Das was ich meine ist die Autonomie über den Weg, über die Schritte. Das Alignment brauche ich über das Ergebnis, dann kann ich die Leute sozusagen die Schritte selbst gehen lassen. Wobei mit Schritten natürlich, ich beziehe hier jetzt so, wenn ich in Scrum-Rollen spreche, den Product-Owner mit ein. Also den Product-Owner ist letztendlich derjenige der dafür sorgt, dass ein sinnvolles Alignment besteht und die Zielrichtung klar ist. Wenn der das nicht kann, kann ein PO seinen Job nicht machen und wenn er den Job nicht machen kann, dann wird es das auch nicht sinnvoll an das Team weitergeben und dann kann er das Team nicht mehr sinnvoll schützen. Er muss in der Lage sein, dieses Alignment hinzukriegen, um sozusagen dann sinnvoll irgendwo die Schritte in die Richtung gehen zu können. Klar kann ich natürlich trotzdem noch den Leuten in der Wahl ihrer Arbeitsweise gewisse Freiheiten lassen, das macht auch sonst noch Sinn, aber es wird dann ggf. schwieriger, es entsteht dann halt an manchen Stellen Frustration. So ein Klassiker was ich letztens bei einem Kunden hatte, wo genau so eine Situation ziemlich großen Frust im Team erzeugt hat, die wirklich sehr sehr gute Arbeit geleistet haben, regelmäßige Prototypen zu bauen und dann kam eben relativ spät von den Stakeholdern: Ja aber hier die Produktstückpreise müssen wir senken, Hersteller von Produkten und die Produkte sind einfach zu teuer, die müsst ihr billiger machen und ein anderer der neben dran stand, aber die Produkte müssen irgendwie geiler werden, wir brauchen hier irgendwie coolerer Touchbedienfunktionalitäten hinten dran. Das Team sagte dann was jetzt: „Entweder wir können es billiger machen und hier eine scheiß Technik verwenden oder wir können es halt so wie wir es jetzt so geplant haben ein bisschen teurer machen und dafür eine geile Bedienung haben. Wir können nicht beides machen, geht nicht. Sorry. So, was wollt ihr jetzt?“ Und die sind halt die ganze Zeit zwischen den beiden hin und her geeiert.

Interviewer: Dann wirklich einmal in die eine Richtung und dann wieder zurück?

Befragter: Genau und das war dann halt immer vor allem sie sind dann vor allem immer nur geeiert, weil dann sozusagen kam: Ja jetzt müssen wir noch eine Kostenrechnung haben und dann müssen wir doch sagen ob wir es nicht doch billiger hinkriegen kann und dann müsst ihr noch mehr Recherchieren, ob es nicht doch noch billigere Zulieferer gibt und ja ich möchte aber da noch einmal das Konzept überhaupt ausgearbeitet haben und die haben auch keine Ergebnisse mehr geliefert, sondern waren nur noch das, was sie eh schon gemacht haben zu rechtfertigen und mehr Information für diese Entschei-

dung herbei zu schaffen, die aber eigentlich völlig irrelevant war, weil das Wissen, die Fakten, die Daten waren ausreichend. Es war einfach nur, dass zwei verschiedene Stakeholder waren, die zwei verschiedene wahrscheinlich sogar Ziele hatten und die sich halt sozusagen diesen Konflikt über das Team ausgetragen haben[T2.17].

Interviewer: Das ist auch interessant: Konflikte über das Team austragen

Befragter: Jo. Da hätte ich noch einen schönen Punkt, der mir gerade in diesem Kontext einfällt: Keine Zielvereinbarung mit Geld. Also wenn ich in einem Kontext bin, in dem an persönliche Ziele Geld gehängt wird, egal wer das ist in diesem ganzen Kontext, also Product-Owner, oder sonst was jemand im Team ist. Wenn Geld an persönlichen Zielen hängt, dann wird das einen agilen Prozess kaputt machen, weil im Zweifelsfall werden in diesem Moment die persönlichen Ziele über das Team gestellt, also über die Teamziele gestellt[T2.18] und es wird früher oder später passieren, im Idealfall ist es natürlich abgestimmt, aber in der Praxis gibt es halt immer wieder Konflikte, wo dann z.B. jemand Fortbildungsziele hat: Bis Ende des Jahres muss ich noch eine Schulung belegt haben, keine Ahnung. Und gleichzeitig hat das Team aber noch ein Release auszuliefern und braucht dafür aber eigentlich die Person. Diese Situation wird immer in irgend so einer Form vorkommen. Und wenn ich jetzt Geld an dieses Fortbildungsziel gesetzt habe, dann werde ich das durchziehen; dann gehe ich da einfach hin, scheiss egal, dann hat das Team einfach Pech gehabt. Wenn ich so was aber habe, dann wird der Teamgedanke als ganzes sehr massiv beeinträchtigt. Damit wird sozusagen dieses „Wir" als Team sind die eigentlich entscheidende Instanz, die in der Lage ist zu entscheiden und auch genau über solche Sachen hinwegzusetzen, die wird damit zerstört. Also das ist so ein weiterer Umgebungsfaktor. Gut.

Interviewer: Vielleicht schaun wir uns jetzt mal die andere Seite an, das Team innen.

Befragter: Da sind es wahrscheinlich viele Sachen, die ich noch nennen könnte. Für mich ist vielleicht noch klar zum Abschluss in der Wahrnehmung ist der Kontext, deswegen habe ich damit auch angefangen, der viel problematischer ist, weil es die Unternehmen nicht schaffen, diesen Kontext bereit zu stellen. Also Manager, der sich plötzlich raus hält, also beispielsweise ich darf auch nicht einfach in das Team rein regieren[T2.19]. Negativbeispiel von einem Kunden, wo beim Automobilzulieferer, OEM kam her, also der Hersteller kam her: „Hey, was sagt ihr mir das ihr meinen Liefertermin nicht

halten könnt, ihr arbeitet noch nicht mal Samstags!“ Was macht ein Management: Gibt das nach unten durch und sagt: „Ihr müsst jetzt auch Samstags kommen.“ Die Leute unten sagen: „Das macht gar keinen Sinn, wir sind nicht das Bottleneck, Nachbarabteilung Ok, da macht es Sinn, aber wir, an uns hängt es nicht.“ Management sagt: „Egal, der Kunde fordert es, also ihr kommt Samstags.“ Du kannst dir vorstellen, was mit Team-Motivation, mit Autonomie, das ist in einer Sekunde, mit einer Entscheidung ist das Ganze hinweg. Da kannst du hinterher noch so viel rumeiern, du hast keine Chance das wieder halbwegs sinnvoll eingefangen zu kriegen. Wenn das jetzt inhaltlich sinnvoll ist: „Hey, ja, das bringt wirklich was.“, dann hätte das wahrscheinlich anders ausgesehen, aber in dem Moment, wo sie sagen: „Hey halt, das bringt nichts!“ Und das Management sagt: „Uns egal, macht trotzdem!“ In diesem Moment ist das Ganze kaputt[T2.20]. Genau solche Bedingungen zu schaffen, wo ein Management seine Finger raus zieht und die Leute arbeiten lässt, die Leute selbst die Entscheidung treffen lässt, das ist was, was ich selten erlebe. Das ist genau der Unterschied wo ich sehe, wo ich so verschiedene Arten des Managements kenne: Das klassisches Management ist unheimlich kontrollbedürftig[T2.21], alles mitzukriegen, alles eingreifen zu können und da eben viel zu überreguliert, während ein agiles Management betrachtet sein Team als Regelkreise. Als agiles Management baue ich einen Regelkreis, ich sorge dafür, das ich ein System baue, das sich selbst verbessert und dann: „Get out of the way!“ Dann lass ich dieses System in Ruhe[T2.22]. Ich schaue nur, ob diese Regelkreise funktionieren und optimiere ggf. die Regelkreise, aber ich rede nicht in die tägliche Arbeit, in die tägliche Veränderung rein [T2.23]. Und wenn das Team irgendwie Freitag mittags um drei nach Hause gehen will und das für das Team im Sinne des Regelkreises funktioniert, dass sie hinterher gute Arbeit, bessere Arbeiten leisten sollen sie das machen. Wenn sie ihre Meetings draußen auf der Wiese machen wollen, ist mir wurscht. Im Endeffekt als Management interessiert mich das Ergebnis am Ende. Aus allem anderen muss ich mich raus halten. Wo ich meckern darf[T2.24]: Klar, wenn die Ergebnisse nicht stimmen, wenn die nicht passen, wenn die schlechter werden. Dann ist es ok. Dann darf ich auch was sagen, dann sollte ich auch was sagen. Aber aus allem anderen habe ich mich herauszuhalten im agilen. Das kriegen die meisten nicht hin.

Interviewer: Das ist glaube ich auch ganz schwierig als Manager, das ist auch so die Grundansicht eines Managers, dass er das zu tun hat, oder?

Befragter: Von manchen ja, als Manager aber damit stell ich mich über die Leute, ich sage ich als einzelne Managementperson weiß viel besser, was zu

tun ist als ihr Fachexperten. Das ist das was er nicht sagt, aber das steckt hinten dran und das ist oftmals eine gefühlte Kontrolle[T2.25], die sie sich dadurch behalten. Während im agilen, wenn sie das erstmal freilassen, dann bekommen sie eine echte Kontrolle, weil die Kontrolle haben sie vorher auch nicht. Sie haben nur das Gefühl sie hätten die Kontrolle. Wenn jemand krank wird, können se auch nix machen. Sie haben nur das Gefühl sie könnten da etwas machen; sie werden ein bisschen Druck machen und kurzfristig können sie auch vielleicht reagieren. Das Problem ist allerdings das alles was da kurzfristig funktioniert macht langfristig ihr System kaputt. Auf dieses langfristige Denken wird von mir viel Zeit verwendet, das macht auch den Unterschied aus, ob es ein Unternehmen schafft, agile Methoden erfolgreich einzusetzen oder nicht. Das ist der Hauptkern, deswegen auch da der Fokus. Jetzt den Wechsel zum Team intern. Da ist es so, dass das aus meiner Sicht zunächst einmal muss ein Team überhaupt mal in die Lage kommen sozusagen also agil, häufig ist vor allem diese Verantwortungsübernahme[T2.26]. Das muss aus dem Team kommen. Ich kann ja nicht sagen: „Hey du bist jetzt dafür verantwortlich." Das funktioniert nicht, sondern ich muss sozusagen die Leute dazu kriegen, das sie von sich aus sagen: „Ja ich bin bereit, ich übernehme die Verantwortung dafür[T2.27]." Darüber vielleicht auch noch einen Kontextfaktor, ich springe noch einmal zurück: Das Thema Fehlerkultur[T2.28]. Das ist sozusagen die andere Seite der Verantwortungsübernahme, wenn ich eine Fehlerkultur habe, wo ich bei jedem Fehler eine aufs Maul kriege, dann übernehme ich keine Verantwortung.

Interviewer: Das heißt im Prinzip zwischen den beiden existiert so eine Art Wechselwirkung, wenn man das darstellt.

Befragter: Das interessante ist jetzt aber auch hier wiederum gibt es sehr verschiedene Personen bzw. Persönlichkeiten[T2.29]. Es gibt Leute, die das als Bedürfnis sehr stark haben und denen es auch sehr leicht fällt und die werden sich einerseits auch Umfelder suchen, in denen sie das auch kriegen und es gibt aber auch Leute, die wollen gar nicht so viel Verantwortung übernehmen die wollen da gar nicht so viel Autonomie. Die fühlen sich viel wohler, wenn ihnen jemand sagt, was sie zu tun haben und sie sich dann darauf zurückziehen können: „Ich hab ja nur das gemacht, was mir gesagt wurde[T2.30]." Das sind wirklich unterschiedliche Eigenschaften von Menschen hinten dran, also das sind jetzt zwei Extreme, im Prinzip ist es aber wie immer in der Psychologie, in allen Fällen gibt es nie zwei Schubladen, sondern es sind immer irgendwelche Kontinuums. Es gibt alles dazwischen. Aber sozusagen in dieser Extrembetrachtung ist es eben auch so, dass es tat-

sächlich Leute gibt, die sich im komplett agilen Umfeld sehr schwer tun[T2.31]. Wenn ich so jemanden im Team habe, eine Person die läuft ein bisschen mit, das geht normalerweise ganz gut, wenn ich allerdings zwei Drittel des Teams aus Leuten besteht, die sich schwer damit tun, Verantwortung zu übernehmen und das auch nicht wollen, dann tu ich mich tatsächlich sehr schwer in einem solchen Kontext agile Methoden einzuführen und zu verwenden[T2.32]. Der Vorteil deswegen im IT-Bereich sind die meisten Leute durchaus so, das sie nicht unbedingt mit schwimmen wollen. Es gibt auch die, aber es ist nicht die Mehrheit der Leute. Wenn ich jetzt an andere Sachen denke, so was wie Behörden, würde ich vermuten ist der Anteil der Leute die einfach sagen: „Sagt mir was zu tun ist, ich wills einfach nur machen!" Viel viel größer. Auch Verwaltungsbereiche typischerweise, auch da sind die eher anzutreffen. So, d.h. diese Fähigkeiten, dieser Wille ist eine wichtige Sache. Das andere ist aber auch, in Leuten die das per se erst einmal wollen, das dieser Zusammenhang, die müssen das erstmal lernen. Ich muss erstmal lernen, Verantwortung zu übernehmen[T2.33]. Das ist nichts was ich per se aus kann. Obwohl die Bundeswehr nicht unbedingt mein Lieblingsunternehmen ist, das scheinen sie tatsächlich ganz gut zu können; so diese Ausbildung dieser informellen Führung. Verantwortungsübernahme ist auch immer eine Form von informeller Führungsübernahme[T2.34]. Also das Setting spielt ja meistens auch mit zusammen. Gut aber das ist sozusagen ein Thema, das in der Gruppe sehr wichtig ist. Dann die zweite Schwierigkeit: Ich brauche im Agilen einigermaßen homogene Teams[T2.35]. D.h. wenn ich da einen Superexperten und fünf Vollnoobs hinsetze, dann wird der eine Experte ziemlich schnell genervt sein und sagen: „Hey, ich sitzt nur da und bring anderen nur ständig was bei und ich kann mich gar nicht mehr mit interessanten Sachen beschäftigen." und die Noobs sagen: „Ja der will alles selbst machen, alles Entscheiden und lässt uns gar nichts machen, was soll denn das?" Ich brauche ein einigermaßen, das muss nicht super homogen sein, ein einigermaßen homogenes Team, damit das funktioniert. Und dann brauch ich natürlich auch ein paar Sachen, die eigentlich selbstverständlich sein sollten, aber nicht immer sind: Die Umgangsformen untereinander. Agile Teams zeichnen sich vor allem dadurch aus, dass sie nicht nur ein: „Wir arbeiten irgendwie miteinander." sind, sondern meistens darüber hinaus die anderen Mitarbeiter als Menschen sehr stark respektieren schätzen[T2.36], kennen lernen und das führt eben sozusagen dazu, man kann das Teamspirit nenne[T2.37]. Das führt zu einer gewissen Resilenz gegenüber Misserfolgen. Misserfolge passieren immer, Dinge gehen immer schief, aber wenn ich einfach weiß, wenn ich den anderen als Person

schätze und den auch sozusagen in anderen Kontexten als nur in diesem Arbeitskontext und ich weiß der hat jetzt gerade Mist gebaut, vielleicht spielt da auch das Thema Vertrauen auch mit rein, dann bin ich in der Lage anders damit umzugehen. Dann sagst du halt, ok, das war nicht böse gemeint, es ist halt Mist passiert, kein Problem, das kriegen wir gefixt, während wenn es das einzige ist, was ich von ihm wahrnehme und es dummerweise auch noch zweimal am Stück passiert ist[T2.38]. Jetzt zum zweiten mal und der macht jetzt hier schon wieder so eine Scheiße, Vollidiot, was soll der hier, dann bin ich auf einer emotionalen Konfliktebene, wo es viel viel schwieriger ist, den Konflikt zu lösen. Während wenn ich diese emotionale Ebene: „Hey, wir mögen uns!" Dann kann ich viel viel krasser, heftiger fachlich diskutieren, was auch sinnvoll und hilfreich ist, als das ich das sonst könnte[T2.39]. Dann brauche ich auch nicht diesen Aufwand dieses emotionale, vorsichtige mit Handschuhen anfassen und auch umgekehrt es ist sehr schnell zu reagieren, denn sonst ist man versucht, ja, ok, komm ist net so gemeint; während mit dem anderen, wenn ich jemanden hab von dem ich sozusagen keine emotionale vertraute Basis habe, dann reagiere ich sehr schneller auf irgendwelche Punkte; irgendwo ist dann z.B. eine Formulierung falsch, wo ich mich nicht gewertschätzt fühle und dann reagiere ich sofort und explodiere: „Hey, was soll des hier?" und reagiere dann auch entsprechend dann auch im schlimmsten Fall passiv aggressiv[T2.40]. Ich arbeite dann dagegen aber nicht dadurch, dass ich aktiv was kaputt mache, weil das würde man dann sehen, sondern dadurch, dass ich einfach gewisse Dinge nicht tue. Ich berichte keine Missstände mehr. Das gibt es durchaus häufiger solche Kontexte wo die Leute einfach sagen: „Du, ich war jetzt schon 3-4 mal angepisst, ich sag jetzt nichts mehr, wenn das Projekt gegen die Wand läuft. Ich weiß das es gegen die Wand läuft, nicht mein Problem. Dann sehen es die Leute, ich halte mich zurück." Also so etwas habe ich durchaus auch schon erlebt. Das ist durchaus nichts ungewöhnliches.

Interviewer: Naja, also das ist dann halt so eine Art, wie soll man sagen, Trotzreaktion nicht. Die Leute wissen sich dann teilweise nicht mehr anderes zu helfen. Ich kenn das Problem auch, wenn man dann irgendwann eingedeckt ist, dann sagt man halt auch irgendwann mal, ja du wenn du mir nicht glaubst das es so ist, dann werde ich halt Tatsachen schaffen. Punkt. Das ist das was ich so aus meinem Bereich kenne.

Befragter: Genau und dann Tatsachen. Das ist schon fasst noch das Gute, denn dann sehen die Leute tatsächlich aktiv, schlimmer ist die passiven Tatsachen schaffen.

Interviewer: Ja, ich wollte gerade sagen, passive Tatsachen kann auch sein, wenn du meinst wir sind bis dahin fertig und du meinst wir müssen bis in zwei Wochen, ich habe dir gesagt wir brauchen sechs Wochen wir werden dafür sechs Wochen brauchen.

Befragter: Genau, dann werden wir nur so acht Wochen brauchen.

Interviewer: Oder so, weil ich dann auch keine Lust mehr habe.

Befragter: Genau das. Das ist halt genau das, was dann entsteht.

Interviewer: Das ist dann aber meistens auch, wenn ein Autorität mit ins Spiel kommt?

Befragter: Was zu hinterfragen ist ja ok, aber zu sagen: „Ich glaub dir nicht." Das ist dann genau der Punkt, wo das persönliche Verhältnis nicht mehr stimmt, dann habe ich da ein Problem. Wenn das persönliche Verhältnis noch stimmt, dann kann ich so was immer noch diskutieren: „Hey du sorry, das ist quatsch, was du redest, plan das bitte nicht so, sonst blockieren wir hier." So was sage ich, wenn das Verhältnis noch funktioniert. Wenn das Verhältnis nicht mehr funktioniert, dann sage ich es nicht mehr, sondern dann ist es mir einfach wurscht[T2.41]. Dazu zählen natürlich auch noch wieder so eine Reihe von Sachen, die sind eigentlich auch wieder sozusagen Kontextfaktoren, die spielen immer wieder zusammen, sowas wie das Colocated[T2.42]. Also es ist sehr schwer Teams zusammen zu halten, wenn sie verstreut, wenn sie teilweise verstreut sind. Es geht tatsächlich einfacher wenn alle verstreut sind, weil man sich dann nur online trifft, nur online kommuniziert oder wenn man eben möglich beieinander ist und sich direkt im Büro trifft. Das finde ich immer noch das Beste, wenn Teams zusammen im selben Büro oder zwei Büros nebeneinander, die sich aber immer direkt miteinander austauschen können. Je weiter ich dies auseinanderziehe, desto schwieriger wird es, da dann viel Kommunikation verloren geht[T2.43]. Gerade auch diese informellen, persönlichen Geschichten. Die gehen verloren, die hat man nicht in dem Maße, dadurch ist es dann viel schwieriger was nachzufragen, ist es viel schwieriger was unangenehmen anzusprechen, deswegen ist das auch wieder so ein Umgebungsfaktor.

Interviewer: Was noch ein ganz interessantes Thema ist, die Selbstorganisation, das durchaus durch die Autonomie über die du schon gesprochen hast im Team verankert. In welchen Situationen / Konstellationen kann die Selbstorganisation versagen? Du hast zum Teil schon über den Kontext ge-

redet und davon abhängig ist, gibt es auch noch andere Dinge, die dir dazu einfallen, gerade aus dem Team heraus kommen?

Befragter: Ich würde tatsächlich zurück auf einen Punk, der dafür ganz wichtig ist, damit eine Selbstorganisation überhaupt funktioniert, ist der Feedback-Punkt. Die Gefahr ist, dass wenn ich sozusagen kein gutes Feedback, das ist wieder eine Schnittstelle nach außen, das ist wieder ein Kontextpunkt eigentlich, aber wenn ich es nicht schaffe das ein sinnvolles inhaltliches Feedback regelmäßig zu bekommen, daher auch immer dieses hochfrequente immer wieder deutliches Feedback zu kriegen, dann ist schon auch die Gefahr da, dass das Team irgendwelche Sachen sich da selbst organisiert aber diesen Kundenfokus, diesen Ergebnisfokus, auch Qualitätsfokus, die gehen dann sehr schnell verloren[T2.44]. Das ist was, das kommt, das sind meistens Dinge, die kommen nicht aus dem Team selbst heraus, manchen Menschen schon aber nicht per se, sondern die Leute machen halt im Zweifelsfall eher das, was ihnen Spaß macht, was sie gut finden, was sie richtig finden, aber nicht unbedingt das, was der Kunde, der halt sozusagen das Produkt am Ende haben möchte, das Ergebnis haben möchte als Wichtigstes braucht[T2.45]. Das ist eine gewisse Gefahr, wenn dieser Disconnect zwischen Team und Kunden besteht. Also wenn es da keine sinnvollen Schleifen gibt bzw. gebaut sind. Im Team ist es eine Sache, was da oft ein Problem ist, da gehe ich noch einmal auf die Persönlichkeiten zurück, da ist so ein bisschen, wie nennen wir es, ich nenne es mal equalistisch, also dieses es gibt ja unterschiedliche Motive beim Menschen. Unter anderem gibt es auch sehr stark machtmotivierte Menschen. Es gibt Leistungsorientierung, Machtmotivierung und Sozial, der Begriff fehlt mir gerade, Bindung, soziale Interaktion. Das sind so die drei typischen Motive. Es gibt verschiedene Theorien, aber von einer der klassischen Motivtheorien sind das die Motive, die uns antreiben, warum tun wir Dinge. Die sind zwar nie so ganz 1:1 ausgeprägt, aber es gibt eben schon Menschen, bei denen sind gewisse Motive sehr deutlich überwiegend und wenn man Menschen hat, die ein sehr hohes Machtmotiv aufweisen, denen also die persönliche Macht und zwar nicht im Sinne von einer Macht im Sinne des Ganzen im Sinne von einer Verantwortungsmacht im Sinne von ich möchte hier Macht haben, das ich für das Team das Beste tun kann, sondern im Sinne von ich möchte die Macht haben, um für mich selbst Vorteile zu verschaffen. Wenn ich so jemandem im Team habe, dann stört er massiv das Team[T2.46]. Das sind Menschen, die sich oftmals sehr gerne auch im Management ansiedeln, weil als Typ hat man mehr Macht, aber das ist natürlich auch für diese Leute auch das Problem diese Macht aufzugeben. Da sehr hoch

machtmotivierte Menschen stören natürlich wollen und werden immer ein agiles Team stören. Das andere, was natürlich da auch rein spielen kann sind auch Leute mit, das spielt auch ein bisschen damit zusammen, mit gewissen Persönlichkeitsstörungen. So als mein Lieblingsthema: Narzissmus[T2.47], auch die Chefkrankheit genannt. Sagt dir die Krankheit was?

Interviewer: Hab ich schon ein mal gehört.

Befragter: Sie ist ein psychologisches Störungsbild, das davon aus geht, im Prinzip fehlt jemand narzisstisch veranlagten ein aus sich heraus kommende Selbstwertgefühl. Ich fühle mich nicht als aus mir heraus als einen wertvollen Menschen, sondern ich benötige dafür einen konstanten Input von außen. Ich bin darauf angewiesen, dass mir andere Menschen immer sagen wie toll, wie schön, wie super, wie großartig ich bin, damit ich mich irgendwie als was Wert fühle. Das führt natürlich dazu, das narzisstische Menschen viel Zeit und Energie, unglaubliche Energie darauf verwenden erfolgreich zu sein, um eben genau diese Wertschätzung, diese Rückmeldung zu kriegen. Auch teilweise auf ihr äußeres, wobei sie sich auch immer überschätzen, sie halten sich immer für schöner als sie sind, sozusagen dieses unglaublich viel auf diese Außendarstellung wert legen. Sachen als besser verkaufen als sie sind usw.. Das Problem was dabei entsteht, so im gemäßigten Sinne sind gemäßigte Narzisten sind gar nicht so schlimm, die Strahlen halt ein bisschen, sind gute Vertriebler, stellen Dinge halt ganz gut dar, können meistens auch gut reden, sind meistens auch recht charismatisch, haben aber das Problem und da wird es dann schwierig, dass sie ihr Umfeld als Mittel zum Zweck erachten, Menschen als Ressourcen sehen. Dementsprechend aber auch immer wieder schaffen andere Leute zu missbrauchen, vor allem psychologisch an manchen Stellen fertig zu machen und damit eben auch eine solche Teamstruktur, Teamformung massiv stören können. Sie sind auch die, die tendenziell... Jetzt gehen wir mal einen Schritt zurück! Im agilen gibt es eine Aussage, wie ein guter Chef im agilen ist: „Take a little bit less of credit than you share and a little bit more of the blame than your share.“[T2.48] Das ist die Beschreibung eines guten Chefs im agilen. Narzisten sind davon genau das Gegenteil. Ich nehme mehr vom Ruhm an als mir zusteht aber will dafür weniger vom Blame, von den Vorwürfen oder wie auch immer. Also das ist so als ein Beispiel für eine Persönlichkeitsstörung die man leider häufiger findet. Das ganze nennt sich Chefkrankheit, je höher man im Management steigt umso mehr findet man davon, weil die Eigenschaften auch sehr gut zu Chefeigenschaften passen. Die sind dann auch wirklich bereit dafür 16, 20 Stunden am Tag dafür zu arbeiten, das zu erreichen weil ihnen ist das auch

wichtiger, wie alles andere und deswegen sind sie dann auch oftmals erfolgreich. Wenn ich erfolgreich sein will, dann bin ich es auch, was dafür tue; das hängt zusammen. So jemand im agilen Team ist dann nicht so gut.

Interviewer: Ja, da organisiert sich das Team dann nicht mehr selbst, sondern da wirds dann organisiert.

Befragter: Genau, bzw. da wird da wird sich da dann sehr schnell rebellieren an dieser Stelle. Ansonsten, ich hab ab und zu mal Leute, die von sich aus sagen, nee sie wollen das nicht. Meistens sind das Fachexperten, die, das ist jetzt ein bisschen eine Unterstellung aber ich würde ihnen Unterstellen, ein bisschen Angst haben ihren Expertenstatus zu verlieren und die einfach sagen sie wollen nicht, sie sind nicht bereit ihre Kontrolle soweit an ein Team abzugeben. Sie wollen sich selbst aus steuern können, sie wollen nur für sich verantwortlich sein und sie trauen keinem Team, sie wollen das nicht, sie machen bei sowas nicht mit. Das sind weniger aber so ich sag mal so in der Größenordnung 1% der Bevölkerung hat man einfach. Ich hab bis jetzt einen deutlichen und einen zweiten mitgekriegt in meiner Laufbahn, die so drauf waren, Die hat man dann auch tatsächlich auch aus dem Team rausgenommen, alles andere hilft nicht. An dieser Stelle noch eine Anekdote, die habe ich zwar selbst nicht mitgekriegt, aber mein Chef mal. Und zwar kam mal ein also es war ein Vortrag von einem Team, das über mehrere Jahre sehr erfolgreich mit agilen Methoden gearbeitet hatte und das Team als solches hat sich dann dazu entschieden zurück zu gehen zu klassischen Projektmanagementmethoden. Dann noch die Frage, warum, es hat doch super funktioniert kam dann die Aussage: Jaja, es funktioniert auch alles Super, aber inzwischen haben wir so keine Trennung zwischen Freundschaften, persönlichen Beziehungen und Arbeiten mehr, so das es komplett ineinander überläuft und sie haben jetzt einfach andere Lebensumstände, sie sind jetzt ein bisschen älter geworden, sie haben da einfach ein Bedürfnis da eine Trennung zu haben. Sie wollen nicht mehr, dass sich die Arbeit und ihr Privatleben so sehr vermischt, sondern sie wollen das wieder als zwei verschiedene Seiten haben und das würde im agilen Kontext so nicht mehr so gut funktionieren. Deswegen wollen sie zurück in den klassischen Kontext, wohlweislich aller Vorteile die sie ansonsten haben, aber dann können sie aber besser diese Sachen getrennt halten. Fand ich, ich habs nur erzählt gekriegt aber ich fand es eine sehr spannende Story.

Interviewer: Damit hat das Team für sich selbst entschlossen sich selbst umzuorganisieren.

Befragter: Ich meine, wenn man dem Team, das ist eine spannende Frage, erlaubt man einem Team die Macht sich auch dazu entscheiden zu können, nicht mehr agil zu sein? Also wenn ich sozusagen die Verantwortungsübergabe selbst für die eigenen Prozessverantwortung, Selbstorganisation, erlaube ich dem Team zu sagen: Nö, wir wollen nicht agil sein.

Interviewer: Eigentlich schon, weil das gehört auch dazu.

Befragter: Ich bin der Meinung: Ja.

Interviewer: Du musst dem Team eigentlich, zumindest wenn du es nicht nur an Scrum fest machst, sondern allgemein am agilen Denkansatz, dann musst du ihnen das auch erlauben zu sagen: „Ich möchte so arbeite. Ende." Weil es mir vielleicht sogar Vorteile bringt.

Befragter: Ich bin auch der Meinung, man muss es dem Team offen lassen.

Interviewer: Das ist ein interessanter Faktor: Man kann dem Team eigentlich nicht vorgeben, dass es dann agil sein soll.

Befragter: Ich kann einem Team nicht vorschreiben, dass es agil sein soll, das geht nicht von außen. Das einzige, was ich als Chefs machen kann ist es das vorzuleben, ich kann ihnen das Wissen, das Know-How geben, ich kann ihnen vielleicht einen Coach für eine Zeit lang an die Seite stellen[T2.49], der ihnen vielleicht mal zeigt und ein bisschen predigt, die Sachen erklärt, aber im Endeffekt wenn ich dem Team die Voraussetzung geschaffen hab, dass es agil arbeiten könnte, dann muss ich die Entscheidung dem Team überlassen.

Interviewer: Das ist schon fast ein wenig philosophisch. Was haben wir noch, eine ganz spannende Frage noch an einen Coach: Wie löst man dann solche Situationen auf? wenn man jetzt solche Kunden hat. Bei der Persönlichkeitsstörung hast du ja schon gesagt, dass man versucht denjenigen aus dem Team zu bekommen, wie macht man das dann bei den anderen Sachen, das man solche Probleme dann auch irgendwo aufgelöst kriegt, weil du hast ja schon gesagt, das liegt im Team selber die Lösung, aber ...

Befragter: ... nein, nicht immer. Man braucht tatsächlich noch eine Umgebung dafür. Der spannende Punkt, da unterscheidet sich kein agiles Vorgehen von anderen Vorgängen, im Prinzip hast du ein Team und vielleicht auch einen Scrum-Master, die erste Instanz die sich um solche Sachen kümmert, sich kümmern kann aber nicht immer können die alle Sachen lösen. Z.b. für das Team ist es sehr schwer, dass das Team jemanden aus dem Team herausnimmt[T2.50], das kann es zwar irgendwie vorschlagen, das ansprechen,

aber im Prinzip muss sowas außerhalb passieren. Und dementsprechend können nicht alle Konflikte im Team gelöst werden. Der Grundsatz Ansatz davon aus meiner Sicht ist es das beste ist, um Konflikte zu lösen, ist das präventiv zu tun. D.h. ich muss a) schon mal vorab die Leute kennen lernen, zu verstehen, wie sie ticken, was sind ihre Bedürfnisse, was ist ihnen wichtig, was sind ihre Eigenschaften, typischerweise empfehle ich das auch den Scrum-Mastern das sie einfach regelmäßig mit den Leuten solche 1on1 machen, sich mit ihnen zusammensetzen, um sie einfach als Menschen kenne zu lernen, als Bedürfnisse kennen zu lernen. Ich mach das bei jedem Coaching am Anfang, das ich mich so eine Stunde etwa mit jedem aus dem Team einmal alleine hinsetze, meistens gehe ich raus, wo es irgendwie nett ist, in ein Cafe oder ein bisschen spazieren und stelle ihnen eine Reihe von Fragen, um sie kennen zu lernen. So die Frage: „Was motiviert sie selbst zum arbeiten? Was ist ihnen besonders wichtig? Was soll ich auf gar keinen Fall tun?“[T2.51] Einfach solche Sachen, um zu verstehen wie sie ticken, weil jeder hat, jeder Mensch ist ein Individuum, jedes Individuum hat andere Bedürfnisse und es geht drum, dass wir im agilen aus diesen einzelnen Bedürfnissen ein gesamtes, funktionierendes Konstrukt bauen[T2.52]. Das heißt natürlich das manchmal auch manche Bedürfnisse nicht so gut zur Geltung kommen, weil ich halt im take-one-for-the-team mich zurücknehmen muss, aber das kann ich nicht bei allen Sachen machen. Es muss halt irgendwo eine Balance geben. Dafür muss ich erst einmal verstehen, was die Leute können, was ihnen wichtig ist und ich muss da eben auch wiederum auf einer, außerhalb dieser Arbeitsebene kennen lernen. Ich muss verstehen, ich muss die Leute einfach eine Wertschätzung entgegen bringen und auch, dass sie eben auch wertvolle Menschen per se erst einmal sind. Wenn wir auf so einer Ebene, das versuchen auch gegenseitig hinkriegen, dass sie mich genauso wahrnehmen und wenn man auf dieser Ebene ist, dann kann man die meisten Konflikte, die auftreten, lösen, weil man dann eben nicht auf der persönlich-emotionalen-Ebene ist, sondern das ist es halt irgendein Bedürfnis, Konflikte ist halt immer das irgendein Bedürfnis verletzt ist und das kann mal ein Bedürfnis nach Anerkennung sein, das mal ein Bedürfnis nach Autonomie sein, das kann mal ein Bedürfnis nach Sicherheit oder Stabilität sein. Es ist immer irgendein Bedürfnis, das nicht erfüllt wird aus dem der Konflikt entsteht. Man braucht meistens zur Lösung, wenn er erstmal auftritt, am besten sollte er gar nicht auftreten, aber wenn er dann doch auftritt, Zeit als eine der kritischen Ressource, also wenn man versucht Konflikte unter Zeitdruck zu lösen, das funktioniert nicht. Ich muss ganz bewusst Zeit, Raum geben[T2.53].

Im agilen hat man da sehr häufig das Thema Retrospektiven, wo man auch bewusst auch ein bisschen dem Team Luft und Zeit gibt, Sachen ansprechen zu können, aber man hat auch einen Scrum-Master, der nicht sieben Projekte auf einmal betreuen kann. Typischerweise ist der Scurm-Master auch so ein bisschen der Team-Puffer und als Scrum-Master sollte ich eben auch ein bisschen Zeit haben mich um solche Themen kümmern zu können. Wenn ich merke, oh da ist gerade jemand unzufrieden, der grüßt nicht mehr richtig oder brummelt zu viel oder hat ziemlich viele Augenringe[T2.54], dann setzt ich mich mit ihm mal zusammen und versuche herauszufinden was denn da los ist[T2.55]. Wenn es dann halt wirklich mal eskaliert, dann muss ich mich auch mit den Leuten hinsetzen und erstmal aus der Situation herausgehen, erstmal zu schaun, was sind da die Bedürfnisse hinten dran und dann zu schaun, ok, was hat denn Ressourcen um diese Bedürfnisse zu lösen und dann auch mal ein Verständnis zu schaffen, wenn es zwischen Personen die Konflikte sind, zu schaun, ok, warum haben denn gewisse Leute anders so gehandelt und was kann man vielleicht tun, dass sowas in Zukunft nicht wieder auftritt. Normalerweise ist das ganz klassisches Konfliktmanagement, das hat eigentlich nicht spezifisch mit dem agilen zu tun.

Interviewer: Damit haben wir so diesen Block abgeschlossen, jetzt möchte ich ein wenig über Diagnostik wissen. Eine ganz allgemeine Frage, wie bemerkst du das im Team etwas nicht stimmt? Du gehst da ja zum Teil hin, guggst dir das auch an und dann, ich kenn es von anderen Sachen, ich schau mir das an und dann schrillen schon die Alarmglocken, also ich merke, da ist etwas nicht so da oder ich habe einen Ansatz, wo ich schon entdecke da das etwas im Busch ist.

Befragter: Also, ich kann darauf zwei Antworten geben. Der häufigste Fall ist es, die Leute sagen es einem[T2.56]. In dem Moment, wo ich, darauf basieren alle meine erfolgreichen Projekte, nicht alle meine Projekte sind erfolgreich, aber die, die erfolgreich sind, basieren immer darauf, dass ich zum Kunden eine sehr gute persönliche Beziehung aufgebaut habe[T2.57]. Meistens bin ich mit den Kunden auch Abends was Essen gegangen, man kennt sich, man hat mit ihnen auch mal zwischendurch gequatscht und in allen diesen Fällen ist es so, dass wenn solche Konflikte irgendwo da sind, dann sagen die Leute das. Wenn da Sachen schief gehen, dann sprechen die das auch untereinander oftmals an, in einigermaßen gut funktionierenden Teams werden dies eh oft angesprochen, in manchen Fällen ist es aber auch so, dass ich mit einem mal zum Essen gehe und er dann sagt: „Hey du, was da gestern gelaufen ist, ich hab da ein Problem mit dem, das kann da so nicht weiter gehen.“ Wie

gesagt, das ist tatsächlich der häufigste Fall, das passiert auch regelmäßig, das ist nichts was ganz selten vor kommt, sondern eher fast immer wieder. Natürlich habe ich da auch den Vorteil der Rolle von außen kommend und auch dem, dass ich mein Verhalten, ich werde auch niemals wenn mir die Leute solche Sachen sagen, das gegen sie benutzen. Das kriegen die Leute meistens relativ schnell mit. Und aus der Kombination, ich bin der von außen, ich stelle für sie keinerlei Gefahr da durch meine Handlungen, sondern ich stell für sie eine Ressource dar, die ihnen vielleicht eine Möglichkeit bietet das zu läsen, sprechen sie in vielen Fällen solche Sachen an. Der zweite Punkt ist sozusagen, ich würde immer sagen, wenn man nicht völlig blind ist, merkt, sieht man das. Man merkt das. Wenn man in ein Meeting reinschaut und die Leute sitzen dann da so scheiss egal, dann merkst du da was nicht richtig ist. Wenn die Leute irgendwie unmotiviert sind, ein Meeting ist rum, sie könnten jetzt zur Arbeit gehen und sitzen immer noch da: „Naja, müssten wir eigentlich wieder was schaffen (Atmet tief ein und wieder aus) ... müsstmer erst mal ein Eis essen gehen.“[T2.58] Dann merkt man schon, irgendwie ist da gerade etwas faul, dann stimmt da irgendwas nicht so ganz. Was es dann ist weiß man dann vielleicht noch nicht, aber man merkt, da ist irgendwas und dann kann man immer noch nachfragen, nachhaken[T2.59]. In der Idealvorstellung hat man natürlich im Scrum dafür eine Retrospektive wo das gelöst wird, de facto funktioniert die ja bei eigentlich, ja Retrospektiven funktionieren nur gut bei ohnehin schon funktionierenden Teams. Bei einem disfunktionalen Team funktioniert auch eine Retrospektive nicht so gut, weil die Leute sich nicht trauen, die Sachen anzusprechen[T2.60]. Sie müssten ja das Vertrauen in das Team haben, dass man die Probleme gelöst kriegt und wenn dieses sie Vertrauen aber nicht in das Team haben, warum sollte ich dann mit Problemen ankommen, damit mach ich mich ja nur verletzbar, angreifbarer. Wenn ich davon aus gehe, man würde mich angreifen, dann sage ich die Sachen lieber nicht. (kurze Pause im Gespräch) Wir waren gerade noch mal beim Thema, hilf mir doch mal, ich grad noch mal den Kontext verloren.

Interviewer: Also du hast jetzt da für mich zwei Sachen gehabt, also entweder ich erkenne es sofort, dass da zwar nicht irgendwas richtig schief läuft, aber wenn ich mir die Situation anschau, und es irgendwo bemerke, weil die Leute du hast das Beispiel genannt unmotiviert irgendwo drin sitzen oder gerade keine Lust mehr dazu haben. Und das andere war das die Leute wirklich offensiv auf die zugekommen sind.

Befragter: Wobei da gibt es auch den umkehrten Fall, ich frage auch offen und aktiv regelmäßig nach. Also ich frag eigentlich immer, das ist die erste Frage, die ich bei jedem Kunden, jedes mal wenn ich sie treffe ist: „Und wie gehts?“ Und die zweite Frage ist: „Und wie läuft es so im Projekt?“ Das sind meine zwei ersten Fragen die ich stelle, das tue ich jetzt aber nicht weil ich das für richtig halte, sondern weil mich das einfach interessiert. Ich hole mir aktiv dann auch regelmäßig, und das ist denke ich mal in dem Kontext, was a ein Scrum-Master-Job ohnehin ist aber auch Management-Job ist, regelmäßig das Feedback einholen. Einfach mit den Leuten ab und zu in der Raucherecke stehen oder beim Kaffee stehen oder wo auch immer sich die Leute treffen und einmal kurz Frage stellen: „Wie läufts bei euch?“ Das reicht[T2.61]. Das ist der Push vs. Pull und beides funktioniert.

Interviewer: Hast du da eigentlich irgendwie, da du gesagt hast, viel Zeit als Scrum-Master verbracht einen Unterschied gehabt zwischen der Zeit, wo du Scrum-Master warst und immer im Team mit dabei warst ...

Befragter: ... viel Zeit als Scrum-Master habe ich tatsächlich selbst nicht viel verbracht, ich mach es meistens im Coaching das ich den Job für eine Zeit lang ein bisschen übernehme, aber so für ein Team über eine lange Zeit, da können wir in einem Jahr noch einmal drüber reden, ...

Interviewer: du bist dann trotzdem für die Leute noch ein externer ...

Befragter: Ja es gab dieses eine Projekt, wo das über ein Jahr hinweg lief, wo ich glaub ich nicht mehr als extern wahrgenommen wurde, da war es dann schon eher intern, da war ich nicht offiziell, sondern nur so bedingt die Scrum-Master-Rolle inne gehabt. Du hast halt einfach mehr Historie miteinander. Das kann sowohl nützlich als auch hinderlich sein. Also ich mein so positive Historien, wenn ich es mir in dieser Firma hier anschaue, es gibt hier Leute, die kenne ich seit 13 Jahren mit denen war ich etliche male Skifahren, zusammen in der Sauna, Wandern, mit denen hab ich schon irgendwie was weiß ich, gemeinsam besoffen, das ist halt völlig klar, das man dann auf einer ganz anderen Ebene unterwegs bist in so einem Kontext. Es kann aber auch negativ sein, auch das gibt es immer überall, es gibt natürlich auch hier Leute, die mich schon mal verletzt haben in meinem Leben, Konflikte der halt nicht richtig gelöst wurde, wo man dann halt mal sagt: „Ok, wir kommen schon wieder irgendwie miteinander zurecht, wir können miteinander so arbeiten aber persönlich vertrauen tu ich denen dann nicht wirklich.“ Irgendwelche intimen Details werde ich dann nicht unbedingt erzählen. Und das

kann sich natürlich in beide Richtungen auswirken und dementsprechend das dann auch erleichtern oder erschweren.

Interviewer: Jetzt habe ich da noch so ein paar Fragen, ich weiß nicht ob das noch passt: Zu Situationen / Themen die Probleme nicht mehr erkennt das hast du vorhin schon mal angeschnitten gehabt, gerade so mit der Retrospektive wenn das Team nicht mehr richtig zusammen spielt. Hast du vielleicht Erfahrung auch wenn Teams sehr lange miteinander zusammen arbeiten, ob sich da irgendwie auch was, gerade bei diesem Problem erkennen irgendwie ändert oder immer gleich bleibt?

Befragter: Jein, ich habe nur indirekt Erfahrung mit, weil ich halt meistens als Consultant eine gewisse Zeit da bin und dann bin ich wieder weg. Teams über lange Zeiträume begleiten ist eher der Außergewöhnliche Fall, es ist eher so, dass ich später wieder reinkomme um mal wieder reinzuschaun. Natürlich verändert sich das, es schleifen sich Sachen mehr ein. Beim neuen Team hat man so diesen Storming, Norming, Perfroming, kennst sicher auch *T2.62*

Interviewer: Ja

Befragter: so wenn sich ein Team ändert, wenn ein Umbruch ist, dann sind da relativ viele Freiheiten und ein Team verändert sich sehr stark. Nach einer gewissen Zeit nimmt das aber ab und ein lang zusammen arbeitendes Team, haben sich halt gewisse Methodiken, Interaktionen, die haben sich einfach eingeschliffen, die sind selbstverständlich, die werden auch gar nicht mehr diskutiert, nicht mehr hinterfragt, die sind dann halt einfach da. Was jetzt per se zunächst nichts schlimmes ist. Was aber dadurch passieren kann, ist das man disfunktionale Verhaltensweisen hat, manche die sogar in der Vergangenheit sinnvoll waren, jetzt aber in einem geänderten Umfeld nicht mehr sinnvoll sind, die Welt dreht sich ja weiter und das gar nicht mehr erkennen*T2.63*. Ich habe gerade auch einen Kunden, den ich nicht so gerne habe, der mich ziemlich nervt und da ist es so, der hat intern ganz massive Probleme untereinander und zum Teil versuchen sie die Probleme dann über solche Externen zu lösen, wobei sie dann auf Art und Weisen gelöst werden, die teilweise dazu führen, dass man sozusagen also ... man hat unterschiedliche Vorstellung, sagt unterschiedliche Dinge, die wir tun sollen und macht uns hinterher den Vorwurf, dass wir sozusagen nicht das was die jeweilige Seite haben wollte, gemacht haben. Du verstehst das Prinzip?

Interviewer: Du bist quasi zwischen die Räder gekommen.

Befragter: Genau, das Kunde wirft einem, man sieht genau, hey da und da hat der Kunde ein Problem, da müsste er was ändern, aber er lässt es gar nicht zu das da was geändert wird, er will es gar nicht. Das ist so gleich wie in der Psychologie, du brauchst erstmal eine Krankheitserkenntnis, vorher kannst du den Leuten nicht helfen. Wenn Leute es nicht einsehen das sie krank sind, dann probierst du es erst gar nicht. Und so ist es eben auch in manchen Kontexten, die Leute haben massive Probleme aber sie wollen das gar nicht sehen.[T2.64] Ihnen ist es, Menschen haben so eine Veränderungsresistenz und es ist ihnen scheinbar lieber in einem disfunktionalen Umfeld zu leben, dass ihnen aber vertraut ist, als in einem unvertrauten, das vielleicht besser funktionieren könnte, sich darauf einzulassen. Es ist immer so, selbst bei sehr guten und sinnvoll agierenden Teams, dass man auch von außen noch mal Dinge anders sieht. In manchen sieht man auch, warum Dinge funktionieren, aber häufig aber auch warum manche Sachen einfach schief laufen. Jetzt hat er gerade ein sehr schönes Projekt, auch ein sehr guter Kunde wo das genau der Fall war, die sagten schaut euch das bitte mal an wir arbeiten seit einem Jahr und haben an den und den Stellen Probleme und da kann ich natürlich von außen, gerade wenn ich auch andere Firmen kenne, wo ich auch andere Sachen gesehen habe natürlich leichter mal sagen: „Hey, die Stelle das halte ich nicht so für sinnvoll, was ihr da macht, überlegt euch mal, ob ihr da dran was ändern könnt." Das ist dann halt auch so ein Punkt, wo die Leute dann auch sagen: „Ja ok." Da wären sie vielleicht gar nicht drauf gekommen. Sie sehen zwar das Problem, aber sie sehen halt nicht dieses eigentlich harmlose aussehende Teil hinter dem Problem, wo das Problem her kommt. Das kann man von außen häufigerweise besser sehen[T2.65].

Interviewer: Also die Ursache sozusagen zu finden ....

Befragter: Genau. Ob es die endgültige Ursache ist sei mal dahingestellt, aber zumindest ist es sozusagen näher an der Ursache dran, so diese Betriebsblindheit. Da kann man von außen schon mehr Sachen aufzeigen und manchmal ist es auch so das in manchen Projekten noch nicht mal nur im agilen, sondern generell, dass man von außen dafür benötigt wird, ihnen den Spiegel vorzuhalten. Eigentlich weiß das Unternehmen als Ganzes wo das Problem ist. Aber Teile, die Teile des Wissens haben, sind nicht in der Lage diese Aussage so zu transportieren, das es vom Rest des Unternehmens gehört wird.[T2.66] Von außen mach ich nichts anderes, als mir anhören was die Leute sagen, das zu einem kohärenten Bild bringen und es dann noch einmal, meistens der Ebene oben drüber, ein bisschen deutlicher zu sagen.

Interviewer: Auch eine interessante Frage in diesem Zusammenhang: Wer ist meistens der Initiator der dich von außen als Coach ruft?

Befragter: Es sind meistens Teamleiter, Abteilungsleiter, so Team, Gruppen, Abteilungsleiter. Es gibt so zwei unterschiedliche Fälle, der eine funktioniert aus meiner Sicht relativ gut, der andere eher schlecht. Der schlechte Fall ist von oben, so wir wollen jetzt Agilität einführen, wir haben keine Zeit, also mach du. Der andere Fall, der deutlich besser funktioniert ist wenn es eher weiter unten ist: Wir finden das sind interessante Konzepte, wir wollen von Anfang an die Sachen lieber gleich richtig machen, wir wollen da auch jemanden Wissen, der die richtigen Stellen schon mal sagt und der uns einfach als Coach auf der Seite steht. Das eine mal bin ich eher in der Rolle: Das ist der große Experte und der sagt uns, wie es geht. Das andere mal bin ich eher in der Rolle: Wir haben da einen Sparringpartner, der auf der selben Ebene ist, der schon relativ erfahren ist und der mit seiner Einstellung ein bisschen helfen kann. Das zweite funktioniert um ein vielfaches besser.

Interviewer: Weil du da das Team erst auf auf deine Seite ziehen musst.

Befragter: Vor allem ich kann von außen keine Änderung bewirken, die Leute können sich nur selbst ändern. Wenn ich dann schon in einer hierarchischen Rolle drinnen bin oder wahrgenommen werde, dann funktioniert das nicht. Es ist damit auch das Selbstverständnis, was damit korreliert, dass ich mir intern für Veränderungsprozesse Zeit nehmen muss. Ich muss mich drum kümmern, ich muss das intern managen, ich muss mir intern die Zeit nehmen und für jede Stunde, die ich beim Kunden bin, muss der Kunde mindestens eine Stunde, eher mehr selbst Zeit noch zusätzlich nehmen für Vor-, Nach- und sonstige Bereitung. Bei manchen Kunden hat man das Gefühl, sie kaufen einen externen ein, weil sie einfach selbst keine Zeit habe, um sich um was zu kümmern. Das funktioniert halt auch nicht. Das funktioniert beim Engineering super, weil mir die die Arbeit abnehmen, aber nicht bei Veränderungsprozessen.

Interviewer: Das macht schon irgendwie Sinn, nur viele denken darüber nicht nach.

Befragter: Nicht jeder, manche schon, sogar einige denken darüber nach und haben das tatsächlich auch verstanden, aber es gibt immer wieder zwei, drei Kunden wo man so denkt: „Nee, ihr habts nicht verstanden."

Interviewer: Klar, weil man sich natürlich auch von dem externen erhofft, dass er einem einfach sagt, wie es geht.

Befragter: Genau, der weiß wie es geht, der sagt uns wies geht, der macht das. Das funktioniert bei keiner Sache gut, aber bei agilen geht das schon zweimal nicht.

Interviewer: Da habe ich noch eine Frage, die ich mir irgendwann gestellt habe: Wo gibt es aus deiner Sicht Grenzen vom Scrum-Framework bzw. die praktische Ausübung gerade beim Erkennen von Problemen. Wobei, das hast du jetzt bereits schon angeschnitten. Wo eigentlich da die Grenzen von diesem Scrum-Framework, gerade in der Praxis, um Sachen zu erkennen. Wobei, wenn ich wieder so darüber nachdenke, geht das wieder in eine ähnliche Richtung.

Befragter: Ja, ich kann dazu noch ein bisschen mehr dazu sagen. Weil was mir zumindest in den klassischen Scrum-Sachen noch fehlt, das mache ich eigentlich überall zusätzlich bei Kunden dazu, so Mile-High-Retrospective oder Release-Retrospektive[T2.67], wie man es nennen kann. Weil was sehr häufig ganz gut funktioniert sind diese Sprints und das Team, wie es untereinander und miteinander arbeitet. Diese Sachen zu verbessern, sind relativ gut eingebaut und im funktionierenden Kontext mit einem funktionierenden Scrum-Master mit einem funktionierenden Team funktioniert das. Da ist schon ein Verbesserungsprozess innerhalb des Teams da. Aber es gibt darüber hinaus längerfristige Themen, die auf einer größeren Ebene sind. Die kann man innerhalb eines Teams gar nicht klären. So Sachen wie aus einem Team mit 8 Leuten machen wir zwei Teams draus und stocken noch jeweils einen auf und haben dann zwei Fünferteams. So was kann ich im Team nicht klären. So die Frage über verschiedene drei, vier, fünf Iteration hinweg, funktioniert das eigentlich, wie wir über die Iteration hinweg die Sachen machen, so diese Indikation die wir unserem Kunden langfristig geben: Was kriegt er in einem halben Jahr. Alles was so dieses Big-Picture, diese übergreifenden Themen betrachten, vielleicht auch mal Zusammenarbeit nach außen. Natürlich kann ich schaun, ob eine Sprint-Demo als solche funktioniert, aber weiß ich überhaupt ob ich die richtigen Kunden eingeladen habe, oder ob da sogar mit den falschen Leuten geredet habe. Wenn ich das Produkt entwickelt habe, ist es hinten ein Markterfolg oder nicht. Das sind alles so größere Themen und diese sind in der kleinen Iteration so nicht drin. Man braucht auch immer diese Release-Ebene dazu und man braucht auch immer wieder mal in vierteljährlich, maximal halbjährlichen Taktung einen Rückzug, wo man sich mal wieder zurückgeht und sagt: „Wir schaun uns jetzt mal unsere ganzen Sachen auf einer Ebene oben dran an." Deswegen auch so dieser Begriff Mile-High, ich flieg drüber, ich schau mir das ganze nicht so im klein-

klein an, sondern über ein Big-Picture und schau, ob wir hier auch noch auf dem richtigen Weg unterwegs sind. Das ist mir eine Sache, die mir je nachdem, in manchen Scrum-Beschreibungen ist das tatsächlich so ein bisschen mit drin, im klassischen Scrum-Werk kenne ich es nicht, das ist aber etwas, was ich für fast unerlässlich halte, dies zu machen. Andere Sachen sind halt alles was über die Teams hinaus geht, da gibts halt auch viele Ansätze, wie Community of practice Sachen, auch Scrum-of-Scrum-Ansätze, aber so dieses organisatorische Lernen als Unternehmen ist da halt auch nicht eingebaut und was da auch nicht eingebaut ist, das wird so ein bisschen als Voraussetzung gestellt, ist es aber selten, auch da kann sich etwas weiterentwickeln ist das Managementverhalten, Firmenkultur[T2.68] als Stichwort. In was für einer Firmenkultur ich bin, wie sich auch so ein Management dort verhält das ist halt was sich im Laufe der Zeit ändern, anpassen sollte, sich auch weiter entwickeln sollte. Scrum sagt eigentlich nur, da gibt es gewisse Voraussetzungen, die müssen so und so sein, aber liefert auch keinerlei Instrumente, Mittel, Werkzeuge, um da eine Weiterentwicklung zu haben.

Interviewer: Das man das Weiterentwickelt, oder das man überhaupt in diese Richtung kommt.

Befragter: Oder um überhaupt mal nur eine Feedback-Loop zu haben, wie es denn läuft. Braucht aber eigentlich auch jedes Unternehmen.

Interviewer: Man kann eigentlich sagen, nach innen hin ist das Scrum-Framework, also wenn man jetzt mal die kleine Ebene betrachtet relativ gut geschlossen ...

Befragter: Jo

Interviewer: ... nur wenn man dann räumlich, zeitlich oder hierarchisch nach außen will, da muss man noch etwas anbauen, um auch zu schaun das das vernünftig funktioniert.

Befragter: Genau ich habe da eine Fragestellung, die muss ich lösen. Das Anbauen hat halt immer von den Begrifflichkeiten so die Gefahr, ich bau mir dann so ein Prozessframework drum herum, nennen wir es SaFe oder so was, ... ja aber im Prinzip gibt es Themen, die löst ein Scrum nicht.

Interviewer: Ich muss mir halt auch Verhaltensweisen angewöhnen, die außen herum noch sind.

Befragter: Es sind halt auch inhaltliche Sachen, Portfolio-Management. Scrum sagt mir nicht, wie ich mir mein Portfolio-Management als Unternehmen machen muss. Vertriebs, Marketing-Prozesse, da ist ja nichts davon.

Das ist halt wirklich nur ein kleiner Teil der auf die Entwicklung fokussiert ist und das Bewusstsein ist ganz klar, natürlich hab ich drum herum immer andere Bereiche, andere Teile, die auch im Idealfall agil werden sollen. Aber da ist nicht unbedingt ein Scrum-Framework das richtige. Da kann ich mir vielleicht ein paar Ideen heraus ziehen, aber ich würde jetzt nicht in jeder Abteilung in einem großen Unternehmen, selbst im kleinen Unternehmen, ich würde nicht in jeder Abteilung per se Scrum einführen. Scrum ist eine Implementierung des agilen Gedankens, der relativ viele sinnvolle Dinge für einen Entwicklungskontext liefert. Aber es ist nicht „die Antwort", sondern dahinter stehen die agilen Werte, die agilen Gedanken und die muss ich eben rein kriegen und was ich als Methode auswähle ist egal. Scrum ist nur eine Möglichkeit, ein sinnvoller Anfangspunkt für Entwicklungskontexte, aber mit Sicherheit nicht für jeden Kontext geeignet.

Interviewer: Gut, dann wäre ich am Ende, ich danke dir!

C.3 Interviewtranskription vom 3. Interview

Datum 2.9.2016

Fachlicher Hintergrund Der Befragte entwickelt seit seinem Berufseinstieg 2009 in agilen Teams Software. Im Laufe der Zeit hat er sich als Scrum-Master weitergebildet und zertifizieren lassen und übernahm diese Aufgabe bis jetzt in zwei unterschiedlichen Teams und Unternehmen. Dabei bleibt er immer selbst noch Entwickler neben seiner Scrum-Master-Aufgabe. Er hat ca. 5 Jahre Erfahrung als Scrum-Master aus der täglichen Praxis.

Interview

Interviewer: Hallo, für mich die Einstiegsfrage: Wie lange du dich mit agilen Teams befasst, wie dein Werdegang ist, wie du dazu gekommen bist, einfach, das ich deinen fachlichen, wie auch deinen praktischen Background habe. Ich lass dich jetzt einfach mal so ein bisschen erzählen, was du gemacht hast, wo du her kommst.

Befragter: Also ich bin seit 2009 dabei und bin dann dort recht schnell in ein Projekt rein gekommen wo es Scrum gelebt wurde und in dem Projekt habe ich dann ungefähr 1,5 Jahre als Teammitglied in einem Scrum-Team gearbeitet, sprich mein Background ist eigentlich überhaupt nicht im klassischen Entwickeln, sondern ich habe sofort in der agilen Welt angefangen. Und dann nach 1,5 Jahre kam ein ganz spannender Punkt und zwar hat da unser Scrum-Master, der auch, wir waren beim Kunden unterwegs, der Scrum-Master war auch von uns hier, der wollte eine andere Aufgabe unternehmen und sein Amt sozusagen zur Wahl freigestellt und hat gesagt: „Ok, das Team kann aus dem Team heraus einen neuen Scrum-Master wählen." Vorher haben ein wir ein paar Leute auch die Scrum-Master-Schulung bekommen, also die Zertifizierung dazu und aus den drei Leuten durfte das Team dann wählen. Das Team war relativ groß, es waren damals so bestimmt 10 Leute und da ist dann die Wahl auf mich gefallen. Das war dadurch recht spannend, weil es tatsächlich eine Team-Entscheidung war oder eine Team-Wahl und das hat natürlich das Standing im Team als Scrum-Master ganz anders dargestellt, als wenn du von außen rein kommst[T3.1]. Vielleicht aus Team-aspekten ganz interessant dieser Vorgang. Da bin ich dann 3,5 Jahre unterwegs gewesen als Scrum-Master, das war bis vor kurzem und jetzt bin ich in einem anderen Projekt unterwegs, wo ich als Scrum-Master-Coach unterwegs bin. Das

kann man sich so vorstellen, das ist ein großes Unternehmen mit mehreren tausend Entwicklern und die haben eine Abteilung, die sich um den Softwareentwicklungsrahmenprozess kümmern, also wie soll die Softwareentwicklung statt finden und seit mehreren Jahren sind die dabei, einen agilen Fokus rein zu bringen. Im Team überlegt man sich dann, wie man eine agile Transition weitere führen kann und die bestehende vorantreiben kann und da bin ich jetzt unterwegs.

Interviewer: Ok. Welche Branchen ungefähr.

Befragter: Medizintechnik und das wo ich jetzt bin ist Governance.

Interviewer: Ich wollte nur wissen ob Automobil oder Medizintechnik, das ist ja in der Region so der Fokus, einfach das ich das mit den Leuten vom Background her einordnen kann. Gut. Du hast die Scrum-Master-Schulung gehabt und bist dann hauptsächlich nur als Scrum-Master ...

Befragter: Nee ich habe, das war ja so eine fifty-fifty-Geschichte, ich bin auch im Team noch weiter als Entwickler gewesen.

Interviewer: Ok, damit warst du kein Vollzeit-Scrum-Master, sondern du hast ein bisschen mehr die Aufgaben vom Scrum-Master übernommen.

Befragter: Ja, wenn dann mehr für das Scrum zu tun war, dann habe ich mehr gemacht, dann weniger. In einem eingeschwungenen Team ist dann auch der Status erreicht, wo eigentlich ein Vollzeit-Scrum-Master gebraucht wird.

Interviewer: Dann zum Block der für mich ganz interessant ist, das sind negative Entwicklungen. Da ist natürlich die Frage, was ist eine negative Entwicklung, das kann man unterschiedlich definieren, ich habe es an der Leistung fest gemacht, wobei der Begriff mit Vorsicht zu genießen ist, für mich ist Leistung nicht nur, das ganz viel Output generiert wird, sondern da gehört auch die Qualität dazu, das vernünftig gearbeitet wird und das vor allem das Richtige heraus kommt. Das ist das, was wichtig ist für die Leistung. Da wäre es für mich interessant, was aus deiner Sicht kann sich von Einflussfaktoren, die man aus dem Team heraus, aus der Umgebung hat, auf ein agiles Team auswirken. Dafür darfst du auch gerne das Blatt hernehmen darfst und einfach ein bisschen was erzählst.

Befragter: Da habe ich verschiedene Sachen gesehen. Eine Sachen, die recht häufig passieren kann ist das man im täglichen Doing, im Alltag, wir haben jetzt schon seit Jahren Scrum gemacht und wir sind da jetzt schon so fest,

das es irgendwann ein bisschen fahrlässiger verfolgt wird, das also die Disziplin nicht mehr stimmt[T3.2], den ich schon ab und zu gesehen habe und dann geht es darum, das der Scrum-Master sagt: „Ok Leute, passt auf, wir wissen warum wir diese Meetings haben, wir wissen warum wir diese Artefakte haben, das Big-Picture noch einmal anzeigen und alle noch einmal ins Boot holen." Aber das kann schon passieren, wie gesagt das Team ist in der Kernbesetzung mit relativ wenig Fluktuation weit über fünf Jahre[T3.3] zusammen gewesen. Und da kann das dann schon ab und zu passieren, da versucht man dann als Scrum-Master neue Impulse einzubringen[T3.4]. Was ich auch gesehen habe ist, na gut, das entwickelt sich natürlich, das wirkt sich negativ aus auf so Sachen, wie Standup-Disziplin[T3.5] und solche Geschichten, wenn so was passiert, wie das die Leute doch mal ein bisschen weg bleiben, wir können ja sowieso untereinander reden[T3.6], wir sitzen ja im gleichen Büro, dann merkt man schon das hie und da die Kommunikation schlechter Läuft, Lücken entstehen, wichtige Informationen nicht so sauber durch die Gegend getragen werden[T3.7], also es hat natürlich überall Auswirkungen wenn Disziplin fehlt, in welchem Bereich, aber das ist glaube ich logisch. Was großes, das mir aufgefallen ist, wenn die Projektumgebung oder die Organisation nicht komplett agil aufgestellt ist, weil man sich noch in relativ starken Regelwerken oder Hierarchien befindet, dann kann es schnell passieren: Hey, wir sind jetzt in unserer Umsetzung zwar agil, aber wir merken regelmäßig, das mit unseren Wünschen, die aus der Agilität heraus kommen, Änderungswünschen, Impediments die aufgezeigt wurden, das da relativ wenig passiert, weil außen herum alles so starr ist und eigentlich das Gefüge um uns herum nicht agil ist und das schlägt massiv auf die Stimmung im Team und kann dann wiederum dazu führen, dass das Team sich sagt: „Warum machen wir eigentlich Agilität, wenn der Rest um uns herum auch nicht agil ist?"[T3.8] Das habe ich in dem Team vor allem immer wieder gesehen, wenn es Retrospektiven gab, wenn es Diskussion gab zu bestimmten Themen. Wir waren da in einem großen Konzern und wir wussten schon wo die Probleme stecken und wo jetzt Probleme auf Firmenebene sind, z.B. wo es darum ging das die Konzepte nicht sauber waren und solche Geschichten, uns war schon immer bewusst, wo wir gehemmt sein konnten und wir haben uns dann auch als Scrum-Master eskaliert an die entsprechenden Stellen, ja und dann ist halt nichts passiert oder es wurde gesagt: „Ja das ist dann halt einfach so." oder: „Das können wir nicht entscheiden, weil da sind noch 20 Hierarchien[T3.9] über uns, gefühlt." und solche Geschichten. Und da habe ich gemerkt, das kann sich ganz schön negativ auf ein Team auswirken. Man kann versuchen in seiner Insel so gut wie mög-

lich zu arbeiten, aber wenn das restliche Umfeld nicht mitzieht, auch wenn man das nur relativ selten sieht, weil man relativ abgeschottet ist von dem ganzen drum herum im Konzern, dann gibt es immer wieder Situationen, wo das dann doch so durch strahlt[T3.10]. Ein bisschen schwierig zu erklären, aber ich glaube du verstehst, was ich meine.

Interviewer: Ich weiß schon was du meinst an dieser Stelle.

Befragter: Das ist ein wichtiger Punkt gewesen. Also es ging hauptsächlich darum, wo negative Effekte im Team auftreten.

Interviewer: Genau, oder was sich allgemein auch negativ auf das Team auswirken kann, also von innen heraus, wie auch von außen. Du hast ja jetzt schon Sachen gesagt, die von innen sind, weil das mit der Disziplin ist typischerweise von innen heraus ...

Befragter: ja das ist von innen.

Interviewer: oder auch die Standup-Disziplin, oder das andere, was du gesagt hast, das man die Organisation nicht so einfach ändern kann, das ist dann etwas was von außen kommt, was sich dann nach innen auswirkt. Das geht dann zwar auch wieder auf das Thema Disziplin zurück, weil die dann auch sagen: „Was soll das hier, wenn ich nicht weiter komme und ich soll machen, dann geht es halt nicht mehr." Dann lässt halt die Disziplin und die Motivation nach.

Befragter: Ahh, ich hab noch ein andere gutes Beispiel. Menschen. Es gibt die Menschen im Team, die total begeistert sind an dem agilen Vorgehen und das auch für sich mit entdecken und lieben und auch das total für sich entdeckt haben, das verstehen und auch Freude dran haben und dann gibt es die Menschen, die gerne für sich brodeln oder für sich arbeiten[T3.11], ich will nicht sagen, das sie nicht teamfähig sind, aber sie stellen dann irgendwie alles infrage, jedes mal und es gibt wieder große Diskussionen. Meistens sind das auch die Menschen, die absolut unverzichtbar sind, die sozusagen die Genies sind im Team und die auch auf Grund dessen vom Team geschätzt werden. Man ist dankbar, das sie da sind, technisches Know-How ganz groß, ohne die würde es wahrscheinlich ganz schlecht aussehen, aber das Team weiß jedes mal, wenn es in die Retro geht, kann es sein, das er nicht da ist oder kann es sein, das er dann einfach desinteressiert in der Ecke sitzt. Wenn das Team damit umgehen kann und es weiß, das ist unser Pappenheimer, der ist einfach so, dann ist es ok, aber da kann sich natürlich schon auf die Bereitschaft des Teams negativ auswirken, sich zu engagieren. Er wird toleriert, weil wir wis-

sen warum wir ihn tolerieren, aber irgendwie ist es trotzdem schade, das er sich nicht einbringt. Das kann glaube ich auch gravierender sein. Bei uns war es nicht so schlimm, aber es war schon ein interessanter Faktor. Und dann Thema Mensch, wir haben einen Kollegen im Team gehabt, sehr respektiert, auch ein sehr wichtiges Mitglied, schon sehr kritisch aber nicht so kritisch, das er nicht Destruktiv wäre, sondern um Lösungen zu finden. D.h. die meiste Zeit war es überhaupt kein Problem mit ihm, sondern eher Konstruktiv. An einem Tag hat er wirklich einen schlechten Tag gehabt, einen wirklich schlechten Tag und da hat er die Retrospektive so dermaßen blockiert und zerschossen, dass das Team wirklich fertig war[T3.12]. Ich als Scrum-Master in der Rolle auch, total und das hat sich massiv auf die Stimmung im Team ausgewirkt und alle haben gesagt, das kann nicht sein, das geht so nicht und wir haben wirklich Mediation gebraucht um das wieder glatt zu ziehen, das Thema.

Interviewer: In dem Fall habt ihr euch dann Hilfe von außen besorgt, oder?

Befragter: Nee, ich hab mir als Scrum-Master Hilfe geholt, wie ich mit der Situation umgehen kann und mir ein paar Experten bei uns in der Firma haben gesagt, wie ich mit Mediation in der Rolle des Coachs das Team wieder dazu bringe[T3.13]. Es war kein Problem, aber es war irgendwie ein schwarzer Block[T3.14], der im Team war, der vieles gehemmt hat im Team, das so etwas passieren konnte.

Interviewer: Wie hat das dann ausgesehen, das das gehemmt war, wurde dann derjenige irgendwie ausgeschlossen?

Befragter: Er wurde nicht ausgeschlossen, aber .... wie wars denn genau ... also die Arbeitsmoral und zwar es waren einfach Dinge, die angesprochen werden mussten, die innerlich so gegärt haben. Es ist schwer zu erklären, aber irgendwie war das Team gehemmt, es wurde nicht mehr so viel geredet, alle wussten, dass da noch ein Thema ist, das da noch auszudiskutieren ist, aber keiner hat sich gewagt es anzusprechen und somit war auch das untereinander im Team so ein bisschen, ja, schwer an der Stelle. Ich glaube besser kann ich es nicht erklären. Aber mir fällt noch was anderes ein. Auch wieder Thema Schwarze Schafe, man sagt ja wieder, es wird häufig in Büchern, auf Konferenzen suggeriert, wie geht man mit Schwarzen Schafen in Scrum um, wie kann man die wieder ins Boot holen. Ist das überhaupt möglich? D.h. jetzt nicht, das in dem Team nur schwarze Schafe waren das waren überhaupt nicht der Fall, das war vollkommen ok, aber es gab eine Situation, wo auch

wieder, nee, das sogar wieder die gleiche Situation, es ging um ein bestimmtes Thema und ich hab überlegt und die Person hat dieses Thema total geblockt und wollte im wesentlichen den ganzen Prozess zerstören an der Stelle. Und ich hab überlegt, wie kann ich ihn wieder ins Boot holen, was kann ich tun, damit er sich mit seiner Kritik abgeholt fühlt[T3.15], aber gleichzeitig das Team einen positiven Effekt spürt durch irgendeinen Input und ich habe gesagt ok, ich werde mit dieser Person reden und ich werde mit ihm zusammen einen Vorschlag für dieses Problem ausarbeiten. Und das habe ich ihm dann vorgeschlagen, ob er sich das vorstellen könnte, das wir uns zusammen setzen und zusammen das Problem bearbeiten weil ich seinen Input da jetzt wertvoll finde, auch wenn er sehr harsch und sehr unverschämt, nicht gerade fair formuliert wurde in der Retro, aber trotzdem ich sehe die Punkte, können wir reden drüber. Da ist er plötzlich sehr aufgeblüht und hat sich geöffnet und einen sehr guten Vorschlag mit mir ausgearbeitet, der auch vom Team sehr akzeptiert wurde[T3.16].

Interviewer: Ok, d.h. du bist mit ihm aus dem Problem in die Lösung gegangen.

Befragter: Genau. Ich weiß nicht ob das immer klappt, aber jedes mal, wenn eine Diskussion kommt in irgendwelchen Fachgruppen zum Thema schwarzes Schaf, dann nenne ich immer dieses Beispiel, weil es für mich ein totales Aha-Erlebnis war. Also es war so schön zu sehen, wie sich diese wirklich massive Blockade aufgelöst hat, in dem Moment, wo ich ihn abgeholt habe: „Sag komm her, ich verstehe dich, lass uns drüber reden und welche Vorschläge hast du, wie würdest du es besser machen?“ Und das hat wirklich gut geklappt. Das hat aber auch dazu beigetragen, dass das Team dann wieder gut funktioniert hat an dieser Stelle; abgesehen von ein paar anderen Reaktionspunkten. Das war ein großartiges Erlebnis, das würde ich immer wieder so tun. Also wenn ich jetzt irgendwelche Leute im Team habe, die sich gegen ein Thema stellen, das vom Rest des Teams einigermaßen akzeptiert wird würde ich versuchen das in der Gruppe zu diskutieren und wenn das nicht möglich ist, wie in dieser Situation, würde ich sagen: „Komm her, wir pairen jetzt, wir überlegen uns jetzt was zusammen.“ Ich glaube die Menschen wollen einfach abgeholt werden. Wenn man ihnen dann den Respekt und ein Ohr schenkt, dann glaube ich das die meisten darauf eingehen und die meisten Menschen auch ein Interesse daran haben, ihre Meinung kund zu tun[T3.17].

Interviewer: Also das ist eine interessante Geschichte, das ich nämlich auch schon mal gelesen, nicht über das Problem reden, sondern halt auf die Lösung, wobei du wahrscheinlich im Team selbst nicht die Möglichkeit hattest, quasi in diese Lösungsrichtung zu gehen, weil das sich einfach komplett blockiert hat, weil es dann immer in diesem Konflikt festgesteckt war.

Befragter: Eigentlich hätte ich die Retro abbrechen müssen, weil es ging nicht weiter, es ging nicht weiter und es wurde immer schlimmer und hat auch Gefühle verletzt. Und zwar massiv und ja in solchen Situationen ist man festgefahren, da muss man dann abbrechen und sagen, komm das klären wir dann. Wir haben häufiger mal, passt vielleicht nicht ganz rein, aber wenn wir Retrospektiven irgendwelche Probleme identifiziert haben und versucht haben in den Lösungsraum zu gehen und dann stecken geblieben sind, habe ich dann irgendwann, wenn die Time-Box abgelaufen war gesagt, wir brauchen, wie haben wir es genannt, eine Task-Force die sich um dieses Problem kümmern sollte und es noch einmal kurz nachbereiten und dann nehmen wir das in die nächste Retrospektive oder ins Daily wenn ihr dann fertig seid rein und dann erzählt ihr mal, was ihr da für Gedanken hattet[T3.18]. Nimm die Leute mit, enorm und ja gut, der Scrum-Master ist für Impediments da, aber d.h. nicht, das er die alle lösen muss. Es gehört dann auch zur Selbstbestimmung des Teams und zur Selbstorganisation.

Interviewer: Sind dann da schon Probleme beobachtet worden, denn du gehst mit dieser Vorgehensweise her und nimmst entweder einen einzelnen raus oder eine gewisse Gruppe aus dem ganzen Team, das sich das dann wie sagt man, manifestiert, das man dann solche Sachen nur noch einzeln löst oder das sich dann Grüppchen bilden ...

Befragter: Nö.

Interviewer: Also die Gefahr hast du nicht gehabt.

Befragter: Nee die Gefahr hab ich nicht. Dieses Mittel habe ich nur genommen, wenn wir festgefahren waren.

Interviewer: Ok, wenn sowieso keine Möglichkeit mehr da war.

Befragter: Es ist keine Lösung gekommen oder es war nicht möglich eine Lösung zu finden, aus irgendwelchen Gründen oder die Zeit war um. Aber sonst ging es darum, dass das ganze Team die Lösung, sich an der Lösung beteiligt. Und das war alles auch nicht, jetzt ist das so, sondern das war einfach nur eine Art Lösungsvorschlag, der ausgearbeitet wurde und vom ganzen Team getragen werden sollte[T3.19].

Interviewer: Ok d.h. die Akzeptanz musste dann trotzdem eingeholt werden. Wurde das dann meistens so akzeptiert wie das vorgestellt wurde?

Befragter: Ja, meistens, doch.

Interviewer: Da gabs dann eigentlich ....

Befragter: Ja, das Team hatte ein grundsätzlich absolut bombastische Stimmung und war wirklich fair, ich kann mir eigentlich kein besseres Team vorstellen, aber selbst in solchen Teams gibt es dann mal Probleme, von daher (lacht).

Interviewer: Kein Wunder, acht Stunden am Tag aufeinander sitzen, passiert das zwangsläufig, da braucht man sich nicht vormachen. Genau, ok. Da war jetzt schon ein bisschen dabei, was ich als nächste Frage: Welche Probleme hat man schon einmal beobachtet, das hast du jetzt schon ein bisschen angesprochen, gibts noch mehr, wo du Probleme gesehen hast?

Befragter: Andere Probleme, die man so im Team sehen kann oder wie, generell?

Interviewer: Genau, oder im ganzen agilen Umfeld ...

Befragter: Hmm; im Team jetzt, oder auch von außen?

Interviewer: Hauptsächlich im Team, das finde ich jetzt wichtig.

Befragter: Ja, ich glaub, was man immer wieder sieht, dass ein Team schnell konsterniert wirkt, wenn man merkt, dass von außen Probleme hereinkommen, an denen man relativ wenig tun kann[*T3.20*]. Das habe ich vorhin schon beim Umfeld angesprochen, wir hatten noch ein anderes Problem, wo der Product-Owner noch nicht so gut funktioniert hat, da gings drum, das ich den irgendwie coache und dann noch helfe, das er uns gute Stories liefert, sozusagen, das wir Dinge haben, mit denen wir wirklich arbeiten können[*T3.21*]. Wir haben lange gebraucht, um das Problem zu identifizieren, wenn solche Probleme immer wieder wiederholen und irgendwie nicht abschaltbar sind, selbst wenn es etwas kleines ist, dann wirkt sich das schon negativ auf das Team aus, aber das ist ja eigentlich auch logisch. Andere Dinge, die ich beobachtet habe, also jetzt nur negative, oder auch positive?

Interviewer: Eigentlich es sind eher so die negative, aber wenn du positive Sachen hast ist das auch interessant, da man damit ja auch negativen Sachen vorbeugen kann oder vielleicht auch drauf hinarbeiten.

Befragter: Was ich bei uns als positiven Effekt gesehen habe ist das Team war fachlich und handwerklich auf einem ganz, ganz hohen Level und wie

es halt in der Medizintechnik halt ist, Qualität steht ganz weit oben aber dieser Fokus, das war etwas, was das Team ganz stark angenommen hat, was es wirklich geliebt und gelebt hat. Dinge wie Clean-Code und alles was mit Unittesting zu tun hat, also das ganze Handwerk, da haben sie sich wirklich drin verliebt und man konnte immer beobachten, wie miteinander, wenn ein Review statt gefunden hat oder Fragestellungen waren, wie sie es genossen haben sich gegenseitig sozusagen zu befruchten mit Ideen und anderen Meinungen und anderen Vorschlägen und anderen Designgedanken usw. und mit Hilfestellung[T3.22]. Das ist eigentlich was natürliches, wenn das Team sich zusammen hilft, aber ich glaube, wenn man so ein gewissen Aspekt, so einen gewissen Fokus im Team schafft, an dem alle gemeinsam arbeiten wollen, z.B. wir wollen nach Clean-Code leben oder so Unittests sollen Sinn machen, nicht so wie die von der Konkurrenz dann schweißt das zusammen und machte eine super Stimmung und kann gleichzeitig dazu führen, dass das Produkt auf einem sehr qualitativ hohen Stand ist. Auf der anderen Seite kann ich mir vorstellen, wenn so etwas nicht gegeben ist, wenn man keinen gemeinsamen Fokus hat, keine gemeinsame Vision, dann kann das recht schnell zu Unzufriedenheiten führen[T3.233] glaube ich. Man hat dann nichts, an dem gemeinsam festhalten kann und keinen gemeinsamen Weg und das führt dazu, das die Leute vielleicht auch gegeneinander arbeiten, weil jeder eine andere Meinung zum Thema hat. Was ist noch positiv? Ja, bei uns war das Thema Qualitätsbewusstsein ganz hoch, dadurch das wir bei Medizintechnik waren, das hat viel Fokus gegeben. Ich habs in anderen Teams, die ich gecoacht habe jetzt in der neuen Rolle ab und zu mal gesehen, das wenn Beispiel der Qualitätsstandard nicht festgelegt war und jeder da eine andere Meinung dazu hatte, dann haben die sich ein bisschen zerfleischt, zumindest die Testabteilung und das war Uneinigkeit an der Stelle[T3.24].

Interviewer: Ok.

Befragter: Ich glaub, das wesentliche ist wirklich, dass das Team diese Phasen, Norming, Storming, Performing usw. sauber durch geht und wirklich einen Stand hat, wo die mit Freude zusammen arbeiten[T3.25].

Interviewer: Wo sie wissen, was sie vom anderen auch erwarten können.

Befragter: Ja und dann glaube ich, das Probleme weniger werden, zumindest aus dem Team heraus. Aber sonst aus dem Team heraus, habe ich relativ wenige Negativbeispiele muss ich ehrlich zugeben. Nee.

[3] Normen / Standards / Kultur / Vorgehensweisen wirken sich positiv aus.

Interviewer: Gut, dann gibt es im Scrum das Thema Selbstorganisation, was dort ganz wichtig ist. Du hast jetzt ein Team gehabt, das sehr gut funktioniert, die Frage, die ich mir irgendwann gestellt habe, gibt es einen Punkt, wo diese Selbstorganisation versagen kann oder sich abstellt, so das dies nicht mehr passiert und was für Situationen oder Konstellationen notwendig sind, falls du so etwas schon einmal erlebt oder gesehen hast, das es da runter ging damit?

Befragter: In Stressituationen[T3.26], ganz klar. Wenn die Leute überarbeitet sind, zu viele Baustellen sind, an denen sie arbeiten müssen, oder wenn Druck von oben kommt zu bestimmten Themen, dann ist einfach keine Kraft, keine Energie und keine Kapazität da, um die Artefakte und Attribute von Scrum zu leben, die zur Selbstorganisation führen: z.b. Stand-Ups, die Retrospektiven, den Prozess, den sich das Team vereinbart hat, irgendwie Status umsetzen, alles was das Team sich auferlegt hat, um zu fließen, um einen Flow zu bekommen, das wird dann relativ schnell über Bord geschmissen. Und die Akzeptanz ist dann auch nicht da, weil die Leute dann sagen: Äh Leute, ich hab so viel zu tun gerade, ich hab da keinen Kopf für gerade. Geht das in die Richtung, was du meinst?

Interviewer: Ja, das ist das was ich meinte.

Befragter: Kannst du die Frage noch einmal wiederholen, für ein anderes Beispiel vielleicht?

Interviewer: Also ganz konkret habe ich: Welche Situation bzw. Konstellationen im Team kennst du, bei denen die Scrum-Elemente zur Selbstorganisation versagen?

Befragter: Ok, Stress auf jeden Fall. Resignation hatte ich schon angesprochen, wenn sich das Team extrem viel Mühe gibt und merkt, das es bestimmte Dinge nicht ändern kann, weil sie von außen unveränderbar oder scheinbar unveränderbar ein gehen.

Interviewer: Kollidiert dann unter Umständen auch ich sag jetzt mal die Organisation mit dem eigentlich im Team verankerten agilen Vorstellungen un der Stelle?

Befragter: Richtig, ich nenn mal ein Beispiel. Nehmen wir mal an, das ich weiß gar nicht, ob das so selten ist, aber nehmen wir mal an das Unternehmen in dem das Team steckt hat ein Produkt, das zu bestimmten Releasedates heraus kommt. Also weil es ein Massenprodukt ist und kein individuelles Produkt und kein Individualprogramm, das jetzt sehr agil an den Kunden

heraus geschmissen werden kann, es ist ein Massenprodukt. Dann hast du aber im wesentlichen einen festen Termin und die ganzen Iterationen, die du im Scrum-Team machst, die potential shipable Increments, die sind zwar potential shipable, aber werden niemals geshipt[T3.27].

Interviewer: Die sind eigentlich uninteressant.

Befragter: Also nach außen hin für den Endkunden sind sie uninteressant, nach innen hin nicht, weil du hast an der Innenseite innerhalb des Unternehmens so etwas wie einen Kundenversteher, der den Kunden widerspiegelt und der kann sozusagen dein Kunde dann sein, so ein Proxy-Kunde. Das ist vielleicht nicht mal der Product-Owner, wer auch immer halt, aber der weiß angeblich, was die ganzen Kunden wollen. Den kannst du natürlich regelmäßig liefern in deinen Sprints und der wird dir dann hoffentlich auch vernünftiges Feedback geben, damit du halt auch agil weiterentwickeln kannst. Und das funktioniert auch und das ist auch machbar, aber trotzdem, dieses Wissen, das der Kunde eigentlich ein ganz anderes Bild ist, das führt dazu, dass das Team sich manchmal sagt: „Warum machen wir das eigentlich, das ist doch gerade ein bisschen lächerlich." Das ist schwierig, weil da wird die Methode angewendet, der Einsatz der Methode ist zwar sinnvoll an sich, aber der eigentlich Nutzen, den diese Methode bieten sollte ist verschattet, ist in Wahrheit nicht real[T3.28].

Interviewer: Ja. Ich sag mal ein vorgeschobener Grund, ...

Befragter: Ich weiß nicht ob der nur vorgeschoben ist, der kann schon wirklich valide sein.

Interviewer: Aber man hat nicht so den Druck dahinter. Man weiß nie, der Kunde, das braucht jetzt nur unsere Testabteilung, deswegen ist es nicht so schlimm, wenn es nicht richtig funktioniert.

Befragter: Richtig. Sowas in der Art, das ist zum Glück nicht ganz so passiert, aber es hat dazu geführt, also es hat sich nicht negativ auf die Qualität ausgewirkt, aber es hat dazu geführt, dass das Team hinterfragt hat, ob wir wirklich uns diese festen Sprints ans Bein binden sollen und sagten: „Hey, warum müssen wir uns vor dem Sprintreview so einen Druck machen, das wir alle Stories noch fertig kriegen, ganz schnell, weil das muss dann noch gezeigt werden. Ja ok. Am Ende des Tages werden wir leider die Qualität drücken, die ersten zwei Tage des neuen Sprints sind wir dann damit beschäftigt, das wieder glatt zu ziehen, damit die Integration wieder sauber flutscht, oder irgendwelche Tests. Die waren dann nicht wirklich Done. Warum ma-

chen wir uns diesen Druck, wenn wir zwar potentially shippen, aber gar nicht wirklich shippen." Der Gedanke ist nicht schlecht und nicht falsch und das hat jetzt auch gar nichts damit zu tun, das dadurch die Selbstorganisation des Teams negativ beeinflusst wird, aber das führt dazu, das sich das Team Gedanken macht und ähm, ja ...

Interviewer: Es macht sich dann ja Gedanken darüber, ob sie richtig arbeiten, ob es so sinnvoll ist zu arbeiten. Sie hinterfragen an der Stelle eigentlich das Scrum-Konzept.

Befragter: Das Scrum-Konzept für diese Situation, was nicht schlecht ist, was vollkommen valide ist. So was kann dazu führen.

Interviewer: Ok, was hast du dann in solchen Situationen gemacht, in deiner Rolle als Scrum-Master, das du solche Probleme aufgelöst hast, also solche Stressituationen, zu viel gleichzeitige Arbeit, Resignation, Anzweifeln des eigentlichen Scrum-Prozesses, was hast du da aktiv getan?

Befragter: Ich hab mir Gedanken gemacht und zusammen gefasst würde ich sagen, viele Probleme, die entstehen basieren darauf, das man Scrum, ich will jetzt nicht wie ein Scrum-Missionar klingen, aber das man Scrum nicht korrekt, also nicht in seiner Gänze gelebt hat[T3.29], weil ich glaube, viele Probleme lassen sich dadurch beheben oder kommen daher, das bestimmte Aspekte vernachlässigt werden, d.h. das die Disziplin nicht 100%ig stimmt[T3.30] oder so. Das Problem mit dem Stress: Das agile Manifest sagt ganz klar, was man da tun soll, das eigentlich kein Druck ausgeübt werden darf, sprich das Problem ist nicht das agile Vorgehen, sondern das Mindset von der Außenwelt zum Beispiel. Da wäre es dann die Aufgabe des Scrum-Masters für das Team gerade zu stehen: „Leute, nein, Sprint ist Sprint, das zählt und alles andere was ihr reinkippt gilt nicht." und solche Geschichten. Wie gesagt, ich glaube, viele Probleme lassen sich mit Scrum selber regulieren.

Interviewer: D.h. du hast dann also versucht die Scrum-Umgebung herzustellen, indem du dann positiv darauf eingewirkt hast, das diese Einflussfaktoren abnehmen.

Befragter: Genau, also was ich versucht habe, zu schaun: Was ist das Problem, wie kommt so etwas zustande und was bietet eigentlich Scrum, dass das Problem nicht auf tritt oder was ist der Optimalzustand. Das kann man dann relativ leicht entschlüsseln, wenn man sich überlegt, was sagt der Scrum-Guide oder was sagen agile Prinzipien dazu. In den meisten Fällen habe ich dann das Problem in irgendwelchen Schwächen gesehen, wo

der Prozess nicht sauber gelebt wurde. Aber, um jetzt mal von der Scrum-Missionars-Rolle kurz weg zu kommen, ich habe meine Rolle als Scrum-Master auch gesehen, das ich gewisse Prozesse auch hinterfrage, wenn es sinnvoll ist. Wir haben auch tatsächlich aufgrund dieser Anfrage vom Team, oder diesen Gedanken, warum liefern wir das eigentlich, haben wir uns ernsthaft Gedanken über Scrumban gemacht und auch eine gewisse Scrumban-Implementierung durchgeführt. Wir haben Scrum an einer Stelle tatsächlich ein bisschen aufgeweicht[T3.31], weil wir für uns gemerkt haben, das wir nicht, es hatte noch ein paar andere Hintergründe, aber das wir auf diese Art ...

Interviewer: Das war von deinem Vortrag damals?

Befragter: Genau, ich glaube da warst du auch da ...

Interviewer: Ja, da war ich da.

Befragter: ... das wir auf diese Art und Weise, wie wir in unserem Setup sind, nicht agil arbeiten mit Scrum, nicht agil werden können, also an bestimmten Aspekten, das ging einfach an der Stelle nicht. Und dann, selbst wenn du als Scrum-Master eingesetzt bist und Scrum-Master bist, viele Probleme lassen sich mit Scrum lösen, aber nicht alle, sonst würde es nicht Methoden wie XP oder Kanban geben. Und dann ist man halt plötzlich nicht Scrum-Master, sondern Scrumban-Master, oder agiler Coach, keine Ahnung. Ich weiß jetzt gar nicht, wo wir waren.

Interviewer: Im Prinzip, wie du mit den Situationen umgegangen bist, um die Probleme aufzulösen.

Befragter: Ah ja, es ist klar, generell ging es um den Prozess, zu kucken ob da etwas nicht ganz gestimmt hat, aber eigentlich hat es dann wiederum mit den Menschen zu tun und im wesentlichen kommt es dann wieder auf den Menschen zurück, selbst wenn es daran lag, das irgendwie der Prozess nicht sauber gelebt wurde, gibt es einen Hintergrund warum er nicht sauber gelebt wurde. Sei es der Aussetzer von der Person, der dazu geführt hat und klar, wenn man ein Problem identifiziert hat, geht es nicht darum, mit dem Hammer zu kommen und zu sagen: „Leute hier na.“, sondern die Leute abzuholen[T3.32], wirklich ein Coach zu sein. Und das war eigentlich immer der Lösungsansatz. Natürlich habe ich mir als Scrum-Master Überlegungen gemacht, wie ich vorgehen würde, aber es ging niemals darum zu sagen: „So, dann machen wir das jetzt so!“, sondern es ging darum, Coach zu sein, in der Retrospektive das Team reden lassen und vielleicht auf bestimmte Dinge hinweisen[T3.33]: „Habt ihr schon mal das gesehen?“, vielleicht indem man

bestimmte Informationen speziell aufbereitet, so dass ein Problem sichtbar wird[T3.34], das habe ich dann schon in meiner Rolle gesehen, aber im wesentlich ging es darum, das man dem Team hilft, die Probleme zu erkennen, den Schmerz zu spüren und dann aber dem Team hilft, vielleicht in diese Richtung zu gehen, wenn man keine andere sieht, oder vielleicht hat das Team auch bessere Ideen. Das war mir ganz, ganz wichtig: Das Team, Team sein zu lassen und dem Team wissen zu lassen, das es sein Prozess ist und das es auch ihre Verantwortung ist[T3.35].

Interviewer: Wie bist du dann mit ich nenne es mal Zwiespalt an der Stelle umgegangen, dass du ja eigentlich selber Team-Mitglied bist ...

Befragter: ... schwer ...

Interviewer: ... weil ich stelle mir das schwer vor. Ich weiß nicht, wie du vorher im Team verankert warst, also fachlich eine sehr wichtige Person warst oder auf gleichem Level warst.

Befragter: ... auf gleichem Level, ich war eher der Neuling.

Interviewer: ... du warst eher der Neuling.

Befragter: Da habe ich viel darüber nachgedacht, über die Zwitterrolle. Zum einen, um im Handling muss man schon fast schizophren sein, man muss ganz klar auch zeigen, das man die Rollen, wann man welche Rolle ist, aber ich bin immer noch nicht sicher, was ich empfehlen würde. Ob ich sagen würde: Wenn die Scrum-Master-Rolle nur zur Hälfte ausgefüllt werden kann oder soll, weil das reicht, weil das Team schon den Mindestlevel hat, dann sagt: „Der ist jetzt Entwickler in dem Team oder Entwickler in einem anderen Team, oder noch zusätzlich Scrum-Master in einem anderen Team.“ da gibt es ja mehrere Optionen. Ich glaube, das es nicht ganz so gut ist, wenn er im gleichen Team Entwickler ist, wie in einem anderen Team, weil dann die Konflikte nicht auftreten und man ja auch in der Retrospektive hat man ja selber auch Bedürfnisse, das kann man, habe ich nicht, aber hätte man lösen können, in dem man jemand anderes oder externes die Retrospektive moderieren lässt und für die Rolle dann den Scrum-Master-Hut absetzt. Einen großen Vorteil hatte es aber und den würde ich immer wieder verteidigen, man kann ein ganz gutes Vorbild sein[T3.36]. Nicht nur, indem man dem Team hilft, sondern die Beschlüsse, die gefasst wurden, das man die vorbildlich befolgt. Und das ist jetzt wieder Servant-Leader, oder eine Führungsperson oder eine Führungskraft ist dann wesentlich effizienter und anerkannter, wenn sie natürlich auch das mitträgt und aktiv mit lebt und bei-

spielhaft zeigt[T3.37]. Das war ein großer Effekt. Dadurch, das ich dann auch als Entwickler dem Prozess vorbildlich gefolgt bin, habe ich gezeigt, dass es geht und habe auch gleichzeitig motiviert, dass die anderen sagen: „Ok, der macht das auch und dann zieh ich da auch mit[T3.38]." Ich konnte auch schnell Erfahrung damit sammeln, wenn ich gemerkt habe, ja das klappt nicht, weil ich das am eigenen Doing gemerkt habe. Also das ist ein großer Vorteil: Vorbild sein ist ein großer Vorteil. Aber wie gesagt, ich habe mich noch immer nicht entschlossen, was der bessere Weg ist.

Interviewer: Ok, dann lass ich jetzt mal den Themenblock mit den Entwicklungen bei Seite, was mich weiter interessiert ist die Diagnostik. Gut, du bist jetzt im Team drinnen, aber man hat ja trotzdem ein gewisses Gefühl: Mensch da stimmt etwas nicht im Team, entweder als Scrum-Master oder als Teammitglied. Kannst du einfach mal erzählen, wie denkst du denn, das du so etwas mitbekommst oder auf welche Zeichen achtest du?

Befragter: Ich bin ein Mensch, der sehr auf die zwischenmenschliche Beziehung achtet[T3.39] und auf die Gefühle von den Menschen und ich versuche die Gefühle der Menschen zu lesen[T3.40] und da kann man relativ viel mitbekommen. Ob es jetzt ein schlechter Tag ist, ob es jetzt wirklich enttäuscht oder genervt sein ist von einem bestimmten Aspekt der Arbeit und wenn man das Vertrauen der Leute genießt, dann kann man, wenn man so etwas sieht, auch auf die zugehen und sagen: „Hey, was ist da los? Kann ich irgendwas für dich tun oder stimmt irgendwas nicht?" Das ist ein Punkt, das ist aber nicht leicht. Ich kann sagen, das es mir mit bestimmten Menschen nicht so leicht fällt, das irgendwie zu entschlüsseln. Natürlich kann man sagen vom Management her, an den Zahlen[T3.41], keine Ahnung, klar, da kann man auch Probleme entdecken, wenn z.B. das Sprint-Ziel regelmäßig gerissen wird oder der Burndown noch nie die untere Linie gesehen hat bei einer gewissen Anzahl von Sprints, dann kann man mal darüber reden. Man kann Probleme bei bestimmten Personen, also jetzt nicht aufs ganze Team bezogen, bei bestimmten Personen sehen, wenn sie z.B. häufiger zu spät kommen zum Stand-Up oder wenn sie bei bestimmten Sachen nicht da sind oder einfach von sich aus gefühlt kein Commitment zeigen, das sieht man.[T3.42] Das kann zum einen sein, das sie persönlich gerade eine schwierige Phase haben, dann ist das so. Dann war es bei uns im Team tatsächlich, weil wir da sehr offen waren, wenn ich das mal angesprochen habe: „Hey, was ist da los?", dass sie das auch von sich aus dem Team gesagt haben: „Hey Leute, ich hab gerade eine schlechte Phase, weil das und das, ...", oder man wusste das schon, das er eine schlechte Phase, weil keine Ahnung, er ist gerade Vater geworden

und hat dadurch chronisch zu wenig Schlaf oder so was in der Art, aber es kann natürlich auch sein, das er mit bestimmten Aspekten im Team oder mit der Arbeit nicht zufrieden ist und da kann man ihn vielleicht abholen, wenn man so etwas sieht. Perfektes Beispiel, zwei Leute, die immer sehr knapp vor dem Stand-Up da waren und häufig auch ein bisschen zu spät im Stand-Up dabei und dann sozusagen die Hälfte verpasst haben, war das Team nicht glücklich drüber, weil auch häufig auch Dinge von denen als Input wichtig waren und auch Dinge, Information verloren gegangen ist natürlich und was war deren Grund, ganz eindeutig, das Stand-Up war zu früh für die.

Interviewer: Um wie viel Uhr war das?

Befragter: Das war viertel vor zehn.

Interviewer: Ah ok, ein ähnliches Problem hatten wir nämlich auch einmal.

Befragter: Ja (lacht). Wir haben es dann auch verlegt, ich glaube auf halb elf, das Problem ist halt gewesen, das Team ist von den Arbeitszeiten sehr stark, also da gab es wirklich zwei Lager. Da gab es das eine Lager, das schon um halb acht da war, spätestens um acht und das andere war das, das am liebsten erst um elf gekommen wäre und dann ging es halt darum einen Mittelweg zu finden, die einen haben gesagt, ja toll, wenn wir das Stand-Up um elf machen, dann ist aber schon mein halber Arbeitstag rum, wenn ich dann dringend Infos brauche, dann habe ich schon einen halben Tag Impediment gehabt und ja[T3.43]. Aber wie gesagt, das die Leute zu spät kommen, das kann dann schnell ein Hinweis dafür sein, die könnten sich ja auch drauf einstellen, aber gefühlt war das jedes mal ein Affront. Also ich hatte nie das Gefühl, dass die sich wirklich Mühe geben wollten, zu kommen. Oder das offensichtlichste Signal, wenn die Leute auf einen zu kommen[T3.44].

Interviewer: Passiert das häufig, dass die Leute dann auf dich zukommen?

Befragter: Ja, ja. Hat mit der Teamleitung zu tun und mit dem Vertrauen zueinander, weil das war halt ein extrem starker Zusammenhalt, schon wirklich freundschaftliche Beziehungen, ganz ganz eng[T3.45] und dann sind die halt einfach oder haben zum Stand-Up ganz offen und ehrlich gesagt oder sind auf einen zu gekommen[T3.46]. Deswegen ich hab nicht wirklich Härtefälle erlebt, wo ich aktiv noch mehr Signale hätte suchen sollen. Vielleicht habe ich deswegen auch nicht so viel Erfahrung, was für Signale man noch sehen könnte. Aber ich glaube die zwischenmenschliche Ebene ist eine ganz wichtige.

(*Anm.: Hier wurde ein nicht relevanter Teil wegen persönlicher Details entfernt.*)

Befragter: Ich glaube auch wenn man das Vertrauen nicht hast, dann kannst du auch nicht sinnvoll Scrum machen, das geht einfach nicht, weil solche Dinge wie Shared-Code-Ownership ist ja dann nicht möglich. Ich glaub Vertrauen und Motivation ist auch in den agilen Prinzipien drinnen mit „Build Environments where people can trust each other“[T3.47].

Interviewer: Eine andere Frage, vielleicht ist dir das auch einmal untergekommen, wo das Team ein Problem nicht mehr selbst erkannt hat, wo es vor sich hin gelebt hat, wo jemand dann plötzlich auf das Team zugekommen ist und meinte: „Könnt ihr mal die Augen auf machen, da habt ihr doch ein Problem!“

Befragter: So was hatten wir. Das hast du vielleicht auch aus dem Vortrag raus genommen, den ich da gegeben hatte. Wir haben tatsächlich ein Problem gehabt, das haben wir auch, die Root-Cause haben wir nicht gesehen, aber die Auswirkungen haben wir ständig gesehen. Auf wirklich eine ganz lange Zeit, wir wussten nie was wir machen sollten. Das war regelmäßig, das war wirklich regelmäßig in Retrospektiven thematisiert worden, aber wir haben nie eine Lösung gefunden. Das war wirklich hart und haarig. Irgendwann haben wir dann ein bisschen resigniert und haben gesagt: „Das ist einfach so.“, das sind wahrscheinlich, das ist einfach speziell für uns Software oder für uns, das ist halt so und haben es dann hingenommen, bis wir eine spezielle Art von Retrospektiven eingeführt hatten, diese 10000-Meter-Retrospektiven[T3.48] haben wir sie genannt, sprich wo wir den Abstand, den wir genommen haben, erhöht haben, deutlich erhöht haben. Ich glaube es ist wirklich so, wenn man nur in den einfachen Sprint-Retros arbeitet ist man noch zu nah am täglichen Doing ist, um Probleme zu identifizieren, zu sehen, weil man dann den Wald vor lauter Bäumen nicht sieht oder weil man zu sehr befangen ist. Manchmal muss man einfach ein, zwei Schritte weiter zurück gehen, um den Gesamtkontext zu sehen, oder andere Dinge. Und so haben wir das auch in dieser Retrospektive gehabt, die haben wir immer am Ende eines Releases gemacht und dieser bewusste Schritt zurück hat uns dann die Augen geöffnet, hat uns dann gezeigt, hat uns ein Element gezeigt, das wir ein bisschen blöd gelegt haben, das ist uns aber wirklich nicht aufgefallen, was war das denn, das wir eigentlich ganz schlecht im Grooming waren, weil das Grooming, das Refinement das war immer die direkt vor dem Sprintende, wo alle Leute den Stress hatten, ihr Zeugs noch fertig zu kriegen, sprich

die Refinement-Aufträge, die an das Team gegangen sind, die wurden total Stiefmütterlich behandelt. Da ist uns aber nicht aufgefallen, ganz ehrlich und das hat natürlich dazu geführt, das die Qualität der Stories so schlecht waren, das sie immer wider explodiert sind und dann in der Retrospektive, wo wir dann aus dem Sprint-Stress haben wir gemerkt: „Mensch, das ist ganz schön blöd, wie wir das machen, warum nehmen wir nicht uns mehr Zeit, die Stories besser vorzubereiten, um sie gründlicher vorzubereiten.“, das hat ganz viel Auswirkung haben, also ja. Ein Mittel ist Dinge zu sehen, also die man sonst nicht sieht, wenn man ein bisschen Abstand nimmt oder diese Art von Retrospektive. Ich glaub Abstand nehmen, würde ich sagen. Da gibt es ja auch viele gute Methoden, selbst auch für die Sprint-Retrospektiven, wie die Fußball-Retrospektive oder so, wo man versucht ein bisschen zu abstrahieren, das Geschehen zu abstrahieren und dadurch ein bisschen mehr um die Ecke zu denken und Dinge besser zu erkennen. Man kann natürlich sagen, man kann den Sprint natürlich rekapitulieren, was ist alles passiert, hier und da, aber man kann es auch so mit einer Fußballretrospektive[T3.49], ich weiß gar nicht, ob das offiziell so vorgesehen ist, aber so haben wir es gelebt, wir haben dann gesagt, ok, wir erstellen einen Ticker, wie so einen Live-Ticker von einem Fußballspiel, das Fußballspiel war der letzte Sprint und wir machen jetzt so Tickerereignisse, aber mit Fußballmetaphern. Wie ein brutales Foul, Pfostentreffer, oder eine gelbe Karte, eine rote Karte und solche Elemente und dann haben wir die Sprint-Ereignisse auf diese Metaphern gemappt. Klingt erst mal einfach, ist aber ganz nicht so leicht und man muss wirklich ein so bisschen um die Ecke denken, aber dabei versucht man das Problem automatisch schon ein bisschen mehr mit zu analysieren, da kann man ein bisschen mehr Insights bekommen. Das habe ich gerne gemacht, um mehr, um Probleme besser zu finden.

Interviewer: Ok, das sind dann halt Probleme, ich sags mal vom Problemtyp sind das Probleme gewesen, welche Art von Probleme das sind ist da vielleicht auch interessant.

Befragter: Was gabs denn da, was waren das für Probleme, die verdeckt waren? Ich hab jetzt nur dieses eine da gesehen. Ich glaub das sind Dinge, die man, gefühlt schon immer so waren und die man nicht infrage gestellt hat, weil man nicht gemerkt hat, das die problematisch sind. Ja, vielleicht das und ich glaube das kommt aus der Gewohnheit heraus.

Interviewer: Gewohnheit, das ist dann wahrscheinlich ein großes Problem, weil das ist eigentlich das, wo man mit einem Team eigentlich hin möchte,

man möchte ja, das es eingeschliffen funktioniert, aber gleichzeitig ist es das größte Problem.

Befragter: Ja, es ist ein bisschen das Wort Gewohnheit ist tatsächlich ein bisschen schwierig, aber vielleicht dieser Alltagstrott, wenn der Alltagstrott zur Gewohnheit geworden ist. Man will ja eigentlich keine Gewohnheiten etablieren, weil man ja ständig open minded sein möchte und ständig den Prozess hinterfragen möchte, um effizienter zu werden. Da ist Gewohnheit dann schwierig, aber es sollte wiederum Gewohnheit sein, sich regelmäßig zu hinterfragen, weist du was ich meine?

Interviewer: Ja, aber man braucht ja auch gewisse Gewohnheiten, z.B. wenn man ein Release macht, dann will man da was bauen, was hinliefert und noch ein Dokument schreibt, ein Changelog z.B..

Befragter: Das müssen wirkliche Standards sein.

Interviewer: Das sind Standards, aber das ist dann ja auch gefährlich.

Befragter: Ja, das haben wir schon immer so gemacht, jaja das stimmt.

Interviewer: Das habe ich unter dem Thema Gewohnheit verstanden.

Befragter: Ja da haste recht, Gewohnheiten können auch gefährlich sein.

Interviewer: Sie sind vielleicht hilfreich in der Routine, aber das Hinterfragen fällt einem dann schwer. Hast du da irgendein Mittel, gut das mit der Fußballretrospektive, einfach um die Gewohnheit ganz anders darzustellen, ...

Befragter: genau, zu abstrahieren

Interviewer: zu abstrahieren oder einfach, das sie was denken müssen, das sie wirklich darüber nachdenken müssen.

Befragter: Also wenn ich das Problem gesehen habe, aber wenn ich dem Team helfen wollte, das Problem zu sehen, dann habe ich versucht eben Information die ich habe, aufzubereiten, dass es das Team selbst erkennen kann[T3.50]. Weil dieser Aha-Effekt des Erkennens ist extrem mächtig und führt dazu, das es einen starken Drang, einen starken Wunsch gibt das zu ändern, also auch vom Team heraus. Sprich wenn es ein Problem ist, das der Scrum-Master entdeckt aber das Team selber nicht sieht dann ist ist das ein Mittel um dem Team da zu helfen.

Interviewer: Das kann dann auch ein gefährliches Mittel sein, nicht dass sich das Team dann darauf verlässt.

Befragter: Das stimmt. Wenn es das merkt (lacht). Ich hab auch mal von Methoden gehört, ja das könnte gut sein, das man in den Retrospektiven nicht nur über Probleme redet, sondern sich für jede Retrospektive ein anderes Element des Prozesses vornimmt und darüber kurz philosophiert. Selbst wenn man sagt, der funktioniert doch eh gut, wir sind super in den Reviews, total super, aber wir reden trotzdem mal drüber, denkt ihr da gibt es noch etwas das man besser machen kann. Das man bestimmte Aspekte beleuchtet, selbst wenn man denkt, das die gut laufen und zwar regelmäßig; das fand ich auch ein gutes Mittel. Habe ich selber nicht benutzt, habe ich aber von gehört.

Interviewer: Gut, dann hab ich da noch eine Frage, ein wenig hast du darüber erzählt, es ist wichtig, das man noch so einen größeren Überblick kriegt, wo siehst du selber die Grenzen vom Scrum-Framework, gerade in der praktischen Ausübung oder beim erkennen von Problemen im Team?

Befragter: Wo ich denke, das Scrum einem nicht hilft, Probleme zu entdecken?

Interviewer: Genau, ...

Befragter: Hindert Probleme zu entdecken ...

Interviewer: Hindern jetzt nicht, sondern wo du dir eine Möglichkeit schaffen musstest oder wie du schon gesagt hast, diese 10000-Meter Retros, das ist etwas was im Scrum so direkt nicht vorgesehen ist, das man das so übergreifend macht, aber was ich schon von vielen Leuten gehört haben, das sie das einfach sehr positiv finden, weil ganz neue Aspekte erkannt werden.

Befragter: Ich sag mal so, Vorschreiben wie Scrum wirkt, ist es an vielen Stellen dann doch nicht. Bestes Beispiel ist so etwas wie ein Burndown. Ein Burndown ist offiziell gar nicht vorgeschrieben in Scrum, sondern da steht das man eine Möglichkeit haben sollte, den Fortschritt des Teams zu visualisieren. Ich weiß nicht, ob das in einer früheren Version drinnen stand, aber ich glaube es ist nicht direkt vorgeschrieben, trotzdem machen es alle, in jedem Scrum-Buch wird halt Brundown vorgeschrieben. Es gab eine Phase für uns im Team, wo der Brundown eher kontraproduktiv war, aus unterschiedlichen Gründen und da habe ich dann nach anderen Möglichkeiten gesucht, unseren Fortschritt zu visualisieren und damit haben wir dann unser Problem gelöst. Das heißt jetzt nicht, dass da Scrum nicht genügend gegeben hat, war nur relativ undeutlich an der Stelle, was man da machen kann. Das ist ein Aspekt, aber sonst. Nö. Es gibt halt sonst höchstens noch die Fälle,

wo man Scrum typischerweise nicht so gut anwenden kann, z.B. wenn die Aufgaben ständig auf Zuruf kommen, aber das sind dann halt die typischen Grenzen von Scrum.

Interviewer: Du brauchst halt ein gewisse Kontinuität bei Scrum, auch in der Anlieferung der Aufgaben. Seid ihr dann da vollständig isoliert, was dies an geht, kommen dann bei euch keine Querfragen oder Anfragen rein?

Befragter: Doch die kommen auch rein, die kommen rein, werden versucht rein zu bringen. Dadurch das unsere Sprintlängen relativ lang sind, also vier Wochen, also ich bin nicht mehr in dem Projekt aber sie waren damals vier Wochen, kannst du die halt nicht so lange liegen lassen. Bei ein Wochen Sprints ist das ok, da kannste das in den nächsten Wochensprint einqueuen wenn du möchtest, aber wir haben dann einen Puffer geschaffen für solche Aufgaben und den dann von der Availability abgezogen.

Interviewer: Ok, ihr habt da dann vorgesehen.

Befragter: Wir haben dann halt gesagt, wir können weniger in den Sprint mit rein nehmen.

Interviewer: Wie viel war das?

Befragter: Es gab immer die Frage bei uns, ob Defects, also Issues die bei uns Hochpoppen, ob die jetzt User-Stories werden sollen oder ob das Zusatzaufwand ist. Das ist relativ schwierig, ob man das jetzt zu Stories zählen würde, aber es gab auch Themen, wie wenn jetzt andere Teams eine Anfrage hatten oder dringend oder irgendwas schnell untersucht werden sollte. Wie viel war das, also wenn ich die Defects weg nehme, war ein vielleicht 10%.

Interviewer: Also schon relativ wenig.

Befragter: Es gab mal Zeiten, wo es mehr war, wir haben dann angefangen das zu tracken, mit einem Lego-Board[T3.51] haben wir das genannt, da haben wir die Aufwände, die wir, eigentlich ist es mehr gewesen 10 ist viel zu wenig, ich schätze mal schon, das es um die 20% waren, es gab halt große Abhängigkeiten zur Plattform und wenn die mal was geändert haben, dann hat es halt geknallt und dann sind das Dinge, die man nachziehen musste und da konnte man nicht warten und das waren dann unsere Zusatzaufwände. Die waren dann aber ungeplant. Ja doch, das sind Dinge, die einfach rein kommen, die gehören auch dazu. Und da haben wir dann, um uns das zu visualisieren haben wir so Legosteine mit unterschiedlichen Farben, mit unterschiedlichen Kategorien, die wir dann im Stand-Up je Stunde sozusagen gesetzt und da kamen ganz witzige Bilder bei raus. Da konnte man dann auf den ersten Blick

sehen das ist ziemlich gelb d.h. keine Ahnung, die Plattform hat uns ziemlich viel Aufwand gekostet oder so.

Interviewer: Ok. Gut vielleicht noch so eine Frage, kennst du noch Situationen, in denen du als Scrum-Master besonders gefordert oder gefragt warst?

Befragter: Also außer die ich jetzt schon genannt habe?

Interviewer: Genau.

Befragter: Ich hatte es glaube ich kurz angesprochen ich weiß nicht ob es so deutlich raus kam, aber wir hatten, der Product-Owner musst sozusagen noch einiges lernen, da war ich sehr gefordert, den zu coachen, um den auf die Füße zu treten. Meinst du jetzt persönlich herausgefordert war?

Interviewer: Naja, vor allem in der Rolle, wo die Rolle besonders gefordert war, oder das du die dann besonders ausfüllen musstest oder wo halt viel Zeit drauf gegangen ist.

Befragter: Naja, das eigentlich schon. Da würde mir jetzt nichts außergewöhnliches einfallen.

Interviewer: Ja, das wäre es, was mich interessiert hat. Du hast mir da einige gute Anregungen und einige Bestätigungen gegeben. Du hast mir halt mal einen guten Blick als Scrum-Master geben können, von dem her war das schon sehr hilfreich für mich.

Befragter: Das freut mich.

Interviewer: Und dann danke ich dir mal dafür.

C.4 Interviewtranskription vom 4. Interview

Datum 15.9.2016

Fachlicher Hintergrund Der Interviewpartner beschäftigt sich seit 2005 mit agilen Methoden. Im Rahmen seiner Arbeit hat er sich als Scrum-Master wie auch als Product-Owner zertifizieren lassen. Er ist weiterhin Teamleiter und disziplinarischer Vorgesetzter von aktuell 22 Mitarbeitern. Er selbst betreut mehrere Projektgruppen, bei denen er für zwei Gruppen die Rolle als Scrum-Master wahrnimmt.

Interview

Interviewer: Für mich ist erst einmal wichtig zu wissen, was du für Erfahrungen hast in dem Bereich, wenn du mal erzählst, wie lange du dich schon mit agilen Sachen befasst, mit Teams, wo du damit in Berührung gekommen bist, einfach das man deine Erfahrung einordnen kann.

Befragter: Ich denke, das hat in etwa im Jahre 2005 angefangen, da haben wir in einem Entwicklungsprojekt mittlerer Größe so mit etwa, also da war ich selbst als Entwickler und Projektleiter in dem Projekt mit etwa 4 bis 6 Leuten damals und da ist der Trend in der Industrie aufgekommen, da haben wir also ein Buch in die Hand bekommen, „Scrum from the trenches“, der Autor, den kannst du dann nachher noch nachschlagen ...

Interviewer: Das Buch sagt mir was.

Befragter: ... das sagt dir was und da fanden wir das ganz toll, weil ich hatte schon immer ein bisschen anderes Verständnis von Projektleitung und Führung von Mitarbeitern und da schien dieses Thema, dass also die Leute mehr selbst befähigt werden oder Selbstverantwortung für die Planung übernehmen, auch für das Projektcontrolling schien uns ganz interessant und da haben wir einfach gesagt, wir probieren mal das ein oder andere aus, was da drinnen steht. Tatsächlich habe ich einmal vorher in einem Projekt als Entwickler selber auch schon Kontakt gehabt, bei einem großen Medizintechnikkonzert, als in einem Projekt auch Scrum angeblich eingeführt wurde, das hat sich aber niedergeschlagen, indem man nur einmal am Tag sich getroffen hat und einen kurzen Status gemacht hat. Das war für mich Scrum damals, da habe ich gedacht: „Ok, jeden Tag eine Statusrunde ist schon auch

gut.“, bei 15 Leuten aber hatte man mehr das Gefühl das ist ganz schön aufwändig. Es wurde auch, das habe ich dann nachher gesehen, falsch angewandt, weil halt viel diskutiert wurde usw.. Also den Begriff kannte ich vorher, nur dann haben wir tatsächlich angefangen im Projekt, erstmal so als U-Boot-Aktion, der Kunde hat das gar nicht so richtig gewusst, mal auszuprobieren, wie ist denn das mit so kurzen Iterationen usw.. Das waren so die Anfänge und das hat sich bei uns eigentlich in dem Bereich, wo ich eigentlich arbeite, im gesamten Engineering-Bereich inzwischen durchgesetzt, dass wir Kundenprojekte nur noch agil anfangen. Das ist jetzt der Stand heute, so war die Entwicklung. Also das können wir natürlich nicht immer selbst entscheiden, sondern der Kunde muss da natürlich mitspielen, gerade im agilen Bereich, ja, aber wenn wir gefragt werden, oder wenn wir es vorschlagen dürfen, schlagen wir immer, jetzt für Softwareentwicklungsprojekte eigentlich den Scrum-Ansatz vor, inzwischen. Ja, ich hab seit dem eigentlich meistens als Scrum-Master dann gearbeitet in den Projekten, ich mach das heute auch noch, ich habe aktuell zwei aktive Projekte, eins von idealer Größe, wechselnde Mitarbeiterzahl 4 bis 7 und ein kleineres, das haben wir jetzt gerade neu gestartet, wir haben immer so Projekte, die über viele Jahre laufen und streng nach Scrum arbeiten, wie in dem aktuellen Projekt, von dem ich sicherlich nachher noch ein bisschen näher berichten kann, haben wir jetzt gerade den 70. Sprint, wobei wir mit 3 Wochen angefangen haben und dann auf 2 Wochen runter gegangen sind. Das hat schon eine gewisse Prozessreife erlangt.

Interviewer: Kontinuität auch ...

Befragter: Absolut, ja, da ist, werden wir nachher sicherlich noch drauf eingehen, da zeigt sich auch, wenn der Kunde an der Stelle, also der Product-Owner derjenige der die Anforderungen liefern muss, das ist die Schlüsselperson, das ist für mich der Schlüssel zum Erfolg und wenn das auf der Seite verstanden wird, wenn die Anforderungen richtig eingekippt werden, dann kann so ein Team auch performen. Das ist für mich eigentlich der Hauptpunkt, oder sagen wir mal die Quintessenz. Wenn das nicht klappt, dann kann auch ein, es gibt ja den Begriff Hyper-Performance bei solchen Scrum-Teams, das funktioniert natürlich nur, wenn man am Anfang das richtige rein kippt.

Interviewer: Ok, ich habe jetzt herausgenommen, du bist jetzt in zwei Teams aktiv als Scrum-Master tätig ...

Befragter: richtig

Interviewer: ... gibt es da noch mehr Teams, jetzt so als ich sag jetzt mal als Teamleiter?

Befragter: Bei uns ist es so, ich verantworte ungefähr jetzt aktuell 22 Leute und ich bin nicht für alle Projekte, in denen die arbeiten bin ich verantwortlich. Also es sind auch Leute in Projekten, die jemand anderes Verantwortet oder der Kunde Scrum-Master ist, oder so was. Das sind aktuell zwei aktive Projekte, die ich als Scrum-Master leite. Und ja, ansonsten ist es immer so über die Zeit, immer 2 bis 3 Projekte. Ich hab dann auch irgendwann, wenn es um die Erfahrung geht, mich entschieden diese Kurse zu machen, d.h. ich hab den Scrum-Master und den Product-Owner auch gemacht, damit man diese Seite mal versteht, also certified Scrum-Master.

Interviewer: Ok, aber den Product-Owner lebst du jetzt nicht, du bist keiner, oder?

Befragter: Ich bin keiner, das stimmt nicht ganz, je nachdem, wie der Kunde es versteht, also mein aktuelles zweites kleinere Projekt, da mache ich de facto den Product-Owner auch, weils der Kunde nicht macht. D.h. ich interviewe, ich hole mir die Sachen raus was gemacht werden soll, aber diese entscheidende Tätigkeit des Storyschneidens und definierens, Akzeptanzkriterien definieren usw. das mache ich voll. D.h. ich leg die Sachen dann nur noch vor, ob sie richtige Priorisierung haben usw..

Interviewer: Ok, du bereitest es quasi vor, stimmst es beim Kunden noch ab und dann hast du die Arbeit eigentlich gemacht, vom Product-Owner.

Befragter: Genau. Da ist auch immer so ein bisschen ein Lehreffekt dabei, weil wir wollen natürlich, bei uns ist es so in unserer Arbeit das immer die Zielsetzung auch ist, die Kunden ein Stück weit mit zu nehmen in ihren Prozessen, sie zu verbessern, ja das ist oft der Wunsch und deswegen ist es auch jetzt nicht schlimm, weil meistens die Leute, die klassischen Produktmanager da keine Erfahrung haben, wie man für ein Scrum-Team die Stories schneidet, dass die halt effektiv arbeiten können[T4.1]. Ja, also die denken in Features und meistens zu groß oder ein Market-Requirement oder wie man es nennt, jeweils und gerade wenn man ein Projekt neu anfängt, Greenfield, muss man natürlich mit dem Schneiden aufpassen, dass die halt natürlich jede Story, die Herausforderung ist natürlich, das die einen gewissen Wert beiträgt, also vertikal definiert ist, aber gleichzeitig auch nicht zu groß wird, so das sie vom Team auch vernünftig gemacht werden kann[T4.2]. Und da ist es sicherlich immer notwendig, dass man als Scrum-Master unterstützt, wie man das richtig macht.

Interviewer: Ok, quasi als Training dann für den Product-Owner.

Befragter: Absolut, klar.

Interviewer: Wobei du dann in die Rolle von einem Coach reinfällst an der Stelle, wobei Scrum-Master und Coach ...

Befragter: Nee, das ist für mich eigentlich die Definition, der ist der agile Coach. Für das Entwicklungsteam ist er da, um agile Entwicklungspraktiken zu etablieren, ja wenn das Team schon läuft ist das praktisch eine Nullarbeit, aber wenn die jetzt noch nicht wissen, was ist testgetriebene Entwicklung und wie man das machen kann, welche Tools braucht man, dann macht man da Vorschläge und kann die anleiten und so ist es natürlich auf der Produktseite genauso, auch ein Product-Owner kann Unterstützung gebrauchen.

Interviewer: Das war es dann zur Fachexpertise, der nächste Block dreht sich bei mir so, ich habe es mal negative Entwicklungen genannt. die Frage ist, was versteht man darunter, es ist schwierig, das an einer Sache fest zu machen, ich habe es ein bisschen an der Leistung fest gemacht. Wobei dieser Leistungsbegriff jetzt nicht unbedingt heißt, Output liefern, sondern der umfasst alles was das Team nach außen bringt. Also das umfasst genauso die Qualität, genauso wie reibungslos die Zusammenarbeiten können, wie effektiv die arbeiten. Das muss man sich darunter vorstellen nicht immer nur die klassische Leistung, in dem Sinn, dass man Output pro Zeiteinheit rechnet, sondern alles was da auch so aus einem Team raus kommt. Da wollte ich dich Fragen, was du aus deiner Erfahrung weißt, was sich negativ auf die Leistung von so einem agilen Team auswirken kann. Da darfst du auch gerne das Papier nutzen, wenn du möchtest, oder einfach nur erzählen.

Befragter: Also da gibt es sicherlich viele Aspekte, einen hab ich vorhin sicherlich schon genannt, also das entscheidende sind gute Stories, gute Anforderungen[T4.3]. Also das sind für mich die Schnittstellen, das Team arbeitet einen Sprint ab und da gibt es ein Gate vorher, da muss die Qualität stimmen, dass die Stories stimmen, dann kann man se bearbeiten, wenn se auf der Hälfte merken man muss halt komplett viel noch klären oder nachfragen oder so, dann können sie natürlich nicht performant sein, das ist ganz klar. Das Gate hinten raus ist dann, was geben sie an Leistung ab. Das ist sicherlich ein wesentlicher Punkt, wenn der Scope halt ständig geändert wird. Dann das Thema Colokation[T4.4], wie eng das Team zusammen sitzt. Ich mein ich hab jetzt ein Projekt, was ich vorhin erwähnte mit den 70 Sprints, da hatten wir am Anfang ein Team, das auf 2 Länder verteilt war, also England und hier Deutschland, das hatten wir in einem Team zusammen, das

haben nachher geändert, indem wir 2 Teams daraus gemacht haben, also ein Scrum of Scrums und da hat man halt wirklich den Performance-Hub messen können[T4.5]. Ja weil wenn du Praktiken hast wie ständiges Review und die Stand-Ups, und so weiter und so fort, ist das halt einfach ohne direkte Kommunikation, wir haben natürlich technisch alles versucht, also mit Hangouts und was man alles macht und Videokonferenzen, aber das ist einfach nicht das gleiche, wie wenn man sich schnell mal zusammen an den Bildschirm setzt und das Review per nachfragen, einfach zusammen macht. Also Colokation wäre ideal, wir versuchen auch die Teams dann auch in ein Zimmer zu setzen, nicht mal über verschiedene Zimmer verteilt. So was wirkt sich noch negativ aus.

Interviewer: gut, du hast jetzt Sachen genannt, die viel von außen kommen, weil das sind ja meistens Entscheidungen die von außen für das Team gefällt werden, was für mich auch wichtig ist, sind auch die Sachen aus dem Team von innen heraus.

Befragter: Naja gut, die Teamzusammensetzung[T4.6], also wenn man halt erwartet, dass ein Inkrement komplett, Definition of Done entsprechend bearbeitet wird, dann muss man ja verschiedene Themen abdecken, insbesondere Test halt auch und so was. Wenn da jetzt wirklich, man muss halt schon bei der Teamzusammensetzung darauf achten, gut das würdest jetzt auch vielleicht als Außeneffekt wieder beschreiben, aber wenn ich, ich schau halt immer drauf, dass jemand dabei ist, entsprechende Requirements und Tester-Skills hat, der halt ständig diese Entwicklung begleitet mit, fängt am Anfang an, Auto-Tests zu schreiben und macht am Schluss die Verifikation, wenn das nicht dabei ist, oder die Skills in dem Team nicht drinnen ist, dann ist es natürlich auch schwierig. Wenn jetzt ein Haufen Vollblutentwickler bleibt das Thema halt auch auf der Strecke, wirkt sich halt auch auf den Output aus, aber könnte man jetzt wieder als externes Problem.

Interviewer: Das kann man so sehen, aber man kann es auch so sehen, dass das Team aber trotzdem mit solchen Sachen zurecht kommen muss. Das muss sich ja auch mal weiterbilden, schulen oder Know-How aufbauen. Man kann ja nicht erwarten, man setzt fünf Leute zusammen und das die all das wissen, was sie brauchen.

Befragter: Ganz klar. Ich mein, das gilt natürlich für alle Skils, oder auch für unterschiedliche Skillevels in den Teams, das kann man sich natürlich nicht immer aussuchen, aber da haben wir im agilen ja jeden Mechanismus, um das auszugleichen. Wenn jetzt ein Superperformer drinnen ist und halt

zwei, die noch Anfänger sind, dann müssen die halt so organisieren, dass der Gute die anderen mit nimmt, das er sie mit verbessert, da hilft nichts wenn jetzt einer für sich da vorprescht, aber damit habe ich in der Praxis jetzt nur gute Erfahrung. Liegt vielleicht auch daran, dass wir nur Leute haben, die so ticken, keine Ahnung. Ich sehe echt vom, wenn wir, das muss man jetzt bei uns nicht anleiten, dass die Leute die schlechteren mitnehmen oder so, das passiert einfach von selber. Ich habe da keine schlechten Erfahrung.

Interviewer: Ich glaub da dazu sind auch die Informatiker sehr kooperativ gestrickt.

Befragter: Jaja, genau, die wären vielleicht beim Roten-Kreuz auch ganz gut aufgehoben (lacht).

Interviewer: Na gut, das ist dann so die zweite Karriere ab 45 ...

Befragter: ja, genau, wenns dafür nicht mehr reicht. Nee aber ich überlege jetzt mal, ich brauch ein bisschen Überlegung ob mir noch was einfällt ...

Interviewer: Oder vielleicht auch eine andere Frage in diesem Zusammenhang, vielleicht macht es das ganze ein bisschen leichter, welche Probleme im Team hast du schon beobachtet? Erinnere dich mal an Situationen, wo du gemerkt hast, das da ein Problem vorhanden war, was waren das für Probleme; so Probleme wirken sich natürlich auch immer auf das ganze Team aus.

Befragter: Ja, das sind ja oft die klassischen Anfangsdinger, wie Pünktlichkeit[T4.7], das ist natürlich klar, wo man sagt, dafür gibt es aber auch Mechanismen. Man will als Team zusammen arbeiten, deswegen gibt es aus meiner Sicht das Stand-Up, das man halt, und wenn man das z.B. mittags legt als Scrum-Master ist man meistens nicht gut beraten, weil die Leute die drinnen sind, also die Performance ist hoch, wenn die Leute möglichst viel zusammen arbeiten, möglichst lange Zeiten. Es gibt halt Leute, die gerne um halb sieben anfangen und manche erst um 11, das kann man Messen sowas, wenn sich des so verstreut und dann hat er halt abends keinen Partner mehr zum Reviewen und dann bleibt der Change halt liegen. Also so was ist schlecht, wenn die Leute beliebig gestreut über den Tag halt da sind[T4.8]. Gegenmaßnahme ist, man fängt beispielsweise um 9 mit einem Stand-Up an und das dann verpflichtend und dann nervts halt alle, wenn einer dann jedes mal eine viertel Stunde zu spät kommt, das geht halt nicht. Dann wird halt ein immer höherer Betrag gezahlt in die Kasse zum Beispiel, da braucht man sich nicht aufregen, der zahlt dann 5 Euro,

des ist auch schön und er lässt es dann irgendwann oder bei 10. Dann gibts halt bestimmte Praktiken oder auch Einstellungen, gerade wenn jemand aus anderem, mit jungen Leuten hat man damit normalerweise kein Problem, aber es gibt zum Beispiel sagen wir mal ältere Entwickler, die halt lange anders entwickelt haben und solche Begriffe, wie „common ownership of code“[T4.9] ist für manche Leute halt ein Problem. Die basteln lieber Wochen lang an ihrem Stück eigenen Häusschen, das sie da im Code bauen und das ist ihr Ding und das wird dann da abgegeben und nach dem Motto wenn einer da was ändern will muss er mich schon fragen. Das ist jetzt nicht so die Einstellung in so einem agilen Entwicklungsteam. Man soll oft einchecken und eigentlich sollte das Verständnis sein, dass jeder wenn er etwas sieht, das kommentarlos auch verbessern und ändern kann. Da gibts schon Reibereien oder Reibungspunkte, wenn du solche, damit man da ein gemeinsames Verständnis bekommt. Das merken wenn wir ins besondere gemischte Teams mit Kunden machen, die haben dann vielleicht eine andere Einstellung, unsere Leute kommen dann ganz selbstverständlich frisch rein, machen Reviewkommentare da ran, wo ihm erst mal die Hutschnur hoch geht. Dieser Kommentar ist ein technischer Verbesserungsvorschlag und kein persönliche Kritik und das können manche Leute nicht unterscheiden.[T4.10] Das ist auch bei uns so gewesen, als wir neu in diesem Projekt Gerrit eingeführt haben, ich weiß nicht, ob du das kennst?

Interviewer: Gerrit, ja.

Befragter: Das kann schon durchaus erstmal zu einem bremsen der Produktivität führen, weil die Leute natürlich erstmal Reviewpartner brauchen und warten müssen, also wir haben es jetzt so gemacht, dass kein Change gemergt werden darf, wenn er nicht durch den Jenkins und nicht mindestens durch einen Peer-Reviewer durch ist. Das ist jetzt seit 2 Jahren da die Regel und für Leute, die das nicht gewohnt sind ist das schon nicht leicht. So was sind Probleme, mal Knirschereien, aber man weiß eigentlich genau, das führt auf Dauer zu einer Qualitätsverbesserung und damit wirds auch günstiger, weil man halt die Probleme früher findet.

Interviewer: Klar.

Befragter: Ich weiß nicht.

Interviewer: Vielleicht mal so zum anschneiden, was ich noch habe, Konflikte, oder wo Rollen infrage gestellt werden, das sind noch so Themen.

Befragter: Damit habe ich noch keine schlechten Erfahrungen gemacht. Ich denke mal, wenn man die Rolle des Scrum-Masters richtig lebt, so wie sie gemeint ist, dann wird man da auch kein Problem mit der Rolle haben. Wenn du natürlich der Projektleiter aus der alten Garde bist und jetzt sagst: „Ich heiß jetzt anders." und genauso weiter arbeitest, dann wirst du vielleicht schon mal auf Granit beißen an der Stelle, weil es ist ja nicht die Idee, die Leute sollen sich ja selbst organisieren[T4.11], du bist ja nicht mehr der Dispatcher weiter hin, ja, und du darfst dich natürlich auch nicht so verhalten, dass du der Chef von der Truppe bist[T4.12]. Bei mir oft eine ganz spannende Situation, weil ich meistens auch noch der Personalvorgesetzte bin von den Leuten aber ich versuch das ganz klar zu trennen. Wenn ich da Scrum-Master bin in dem Projekt, dann bin ich halt ein Teammitglied, das dafür sorgen muss, dass alles läuft, dann eine bestimmte Rolle habe. Eigentlich ist es positiv, finde ich es als Vorgesetzter positiv, weil wenn ich mit den Leuten zusammen arbeiten kann, weil es ist für mich viel leichter natürlich auch deren Leistung nachher zu beurteilen.

Interviewer: Ist das dann nicht gefährlich, wenn du jetzt quasi die Leistung beurteilst von Leuten die du in deiner täglichen Arbeit siehst, die dann vielleicht auch mit Problemen zu dir kommen, weil das ist ja auch ein Zwiespalt, einerseits brauchst du als Scrum-Master ein vertrauensvolles Verhältnis zu den Leuten und auf der anderen Seite bist du dann hinterher der sie beurteilt.

Befragter: Jaja, wie gesagt, das muss man halt trennen an der Stelle. Was heißt beurteilt, ja klar, Verbesserungspotential identifizieren oder Zielgespräche führen oder so was in der Art, aber ich hab noch nie ein; sicherlich man findet immer wieder den Rat, ein Scrum-Master soll kein Linienvorgesetzter sein, aber ich mache es schon immer und habe noch nie ein Problem damit gehabt, bei uns.

Interviewer: Ich denke, das kommt auch ein bisschen drauf an, wie es in der Firma selbst ....

Befragter: Firmenkultur, ganz klar. Und Konflikte hast du ja auch gesagt unter den Leuten, ja, also, das kann ich mir schon vorstellen, dass wenn die sich persönlich nicht riechen können, dass das nicht befördert, aber ich hab jetzt selber in den Scrum-Teams in den vielen, die ich hatte, keinen wesentlichen Punkt gehabt, wo die, also ich mein, es wird schon mal was angesprochen in einer Retro oder so, aber jetzt nie, also ich musste jetzt noch nie so Um-

strukturierung oder so was machen, dass man sagt, das wäre nicht tragbar oder so.

Interviewer: Also es war quasi nie so festgefahren, dass man es nicht mehr lösen konnte, sondern man konnte das immer noch klären.

Befragter: Genau.

Interviewer: Ok, dann lassen wir die Probleme mal, ein anderes wichtiges Thema ist auch die Selbstorganisation, gerade im Scrum und im agilen ist das besonders wichtig, weil da möchte man das sich das Team selbst organisiert, jetzt ist halt die Frage gewesen, ein Team organisiert sich manchmal sehr gut und manchmal stellt es das ein, je nachdem wie es auch gelebt wird. Jetzt ist dann die Frage an dich, ob du irgendwie Situationen oder Konstellationen kennst, wo so etwas versagen kann?

Befragter: Also wenn ich denke mal, irgendjemand ist an so einem Produkt interessiert, nenne wir ihn mal Product-Owner, Kunde oder wie auch immer und wenn der aufhört, das zu überprüfen und das Interesse an den einzelnen Inkrementen einzustellen[T4.13], dann wird auch das Team aus meiner Sicht in den Sprints oder in den Iterationen keinen Aufwand treiben, sich da zu kontrollieren oder das Ziel zu erreichen. Das ist klar, wenn man kein Ziel hat und das nicht kontrolliert wird, wird man auch nix tun, um das zu tun. Das ist für mich eben der Gegenpart oder für den man Arbeit, das ist das entscheidende, wenn das ausgemacht vorher, was getan werden soll und erst mal das ist das, auf dieses Ziel müssen sich beide Commiten oder einigen für die Iteration und wenn das am Ende keinen Interessiert, es wird nicht überprüft, es kommt keiner zur Demo, es schaut keiner wie habt ihr die Zeit eingesetzt und so weiter und so fort, dann wird das auch nicht passieren; das ist für mich der einzige Punkt. Klar, warum soll ich was machen, wenn es keinen interessiert oder es keiner einfordert. Ob ich das jetzt geschafft habe oder nicht, wenn es nicht überprüft wird oder es keine Konsequenzen hat, dann wird das nicht getan. Und genau das Gegenteil ist es, wenn es halt so ist, ich habe ein so ein Musterprojekt für mich, das ist das, wo wir im 70. Sprint sind und da ist es halt so, das hat halt für mich der Product-Owner wirklich die, da sehe ich halt auch den Unterschied zu anderen Projekten. Das ist der einzige, den ich kenne, der das so ernst nimmt, der reist dann alle 2 Wochen hier an, der kommt 2 Tage, alle 2 Wochen hier her, weil das ist an einem anderen Standort, dann muss nicht das ganze Team dorthin fahren, das ist praktisch von den 70 Sprints 5 mal nicht passiert, weil dann mussten wir das per Videokonferenz machen und dann wird das halt alles angeschaut, abgehakt, dann

lässt er sich die Dokumentation noch einmal zeigen, die zwei, drei Stunden Demo, die dann passieren, die sind auch intensiv und dann gibt es am Ende noch mal, es gibt auch jemanden, der Hersteller in USA für den wir da arbeiten, gibt auch noch einmal eine Kurzzusammenfassung, also das ist schon, einfach es wird gewürdigt aber es wird auch den Leuten unangenehm, wenns am Ende nicht funktioniert, das ist ganz klar, das ist nicht wurscht[T4.14]. Das ist der einzige Punkt und da, das treibt die Leute auch einfach, dann kucken die sich das an, ich muss als Scrum-Master, ich führe da kein Burn-Down mehr oder sonst irgendwas, ich hab am Anfang ein paar Vorschläge gebracht und hab das gemacht, immer so an der Tafel, inzwischen habe sie sich das automatisiert, mit so einem google-sheet oder sonst irgendwas, ich kuck mir das auch nicht mehr an jeden Tag mit denen, das kucken die sich an und der eine sagt wir sind hier im Burn-Down schlecht und so weiter und so fort. So ist das, das finde ich halt optimal, aber das ist halt wie gesagt am Ende muss sich einer dafür interessieren.

Interviewer: Ok.

Befragter: Und ich denk mal, da ist auch, da könnte man in andere Projekte auch Kraft rein bringen, wenn man sagt, ok, das ist halt das Problem oft, wenn das Team oft oder in unserem Fall, da hieß es ja mal Scrum ist ja prima, macht wenn ihr das unbedingt wollt, aber da muss man aber dann auch das ernst nehmen und einfordern.

Interviewer: Ok, was ist dann halt die Frage, was das das für die Selbstorganisation bedeutet, weil das Team kann sich ja dann trotzdem umorganisieren und kann dann sagen, dann mach ich halt einfach tagelang bemale ich dann die Tafel oder ich gehe nur noch kickern und organisiere ein Kickerturnier; das ist ja auch ein Teil von Selbstorganisation.

Befragter: Ob es noch andere Gründe gibt, meinst du, dass die Selbstoriganisation schwindet.

Interviewer: Genau, oder das sie mal mehr wird, vielleicht das es auch mal weniger wird, vielleicht das man da feststellt, wir machen da zu wenig.

Befragter: Guter Punkt, ja. Das es immer mal wieder schwankt und sich Sachen einschleifen, das ist ganz normal, da braucht es auch mal wieder einen externen Anrüttler, nach dem Motto, die Kundenseite ist ja die eine, für uns als Dienstleister gibt es natürlich auch noch eine andere Seite, nämlich die wirtschaftliche[T4.15], wenn du sagst die gehen die halbe Zeit Kicker spielen, irgendwann schlägt sich da ja auch in der Produktivität wieder, dann wird

natürlich, sag ich jetzt mal, die Sachen zögern sich dann im Endeffekt hinaus, weil es dann nicht fertig wird und man kann später Rechnung stellen und damit sinkt natürlich auch der Deckungsbeitrag; dann komm ich auf den Plan, in einem Projekt wo ich jetzt nicht Scrum-Master bin, was aber in meinem Bereich läuft, irgendwann guggt man dann mal auf die Zahlen und sagt dann: „Hey Freunde, was ist denn jetzt hier los und wie macht denn ihr eigentlich." Da brauchts mal wieder einen externen Anstoß.

Interviewer: Ok, man kann natürlich sagen, du nutzt natürlich auch die Zahlen, die bei euch aus dem Controlling raus kommen, um die Wirtschaftliche Seite zu betrachten, um auch Probleme zu identifizieren.

Befragter: Ganz klar...

Interviewer: .. das ist für dich dann ein Indikator, ob es da ein Problem gibt.

Befragter: Natürlich und das ist ziemlich einfach.

Interviewer: Die Frage ist natürlich, du hast natürlich den Vorteil, weil du ein Dienstleister bist, weil du ja für jede Aktion bezahlt wirst, wenn du jetzt aber intern im Unternehmen bist, ist das eher schwieriger, weil da wird ja vorher keine Aufwandsabschätzung gemacht.

Befragter: Warum eigentlich nicht ....

Interviewer: ... und ein Preis abgegeben.

Befragter: Das sehen ich nicht, das kann ich nicht gelten lassen als Argument, man kann das genauso, es gibt da ja auch interne Verrechnungssätze für die Arbeitszeit, also wenn ich jetzt als Firma ein Produkt mache, dann kann genauso eine Kalkulation machen. Ich kann Features abschätzen, kann überlegen was wie viel Arbeitseinsatz habe ich da, was sind die internen Personalkosten, dann kann ich das Controlling genauso machen. Ich meine, das kann ich mit dem normalen Projektmanagement mit einer earned value Analyse beispielsweise, das ist eigentlich der Klassiker; ich schau an, wie bin ich im Fortschritt mit der Zeit, v.s. dem geschaffenen Value sozusagen. Vorangekommen und kann dann Extrapolieren, wie ich dann am Ende lande. Das ist eigentlich das klassische Controlling. Und das kann man natürlich feingranularer mit Punkten und Sprints auch machen. D.h. der Mechanismus wäre natürlich für einen Produkthersteller genauso verfügbar. Und normalerweise ist das so, das diese Art von Controlling die Teams selber nicht machen, also das wirtschaftliche und finanzielle Controlling.

Interviewer: Das interessiert die meistens nicht.

Befragter: Ja, weiß ich nicht, das ist dann immer mal wieder die Diskussion wenn ich dann praktisch einen Scrum-Master habe, der ein normaler Mitarbeiter ist, das versuche ich natürlich diese Dimension ihm auch klar zu machen, was das halt bedeutet. Weil wenn wir insbesondere natürlich an einen Festpreis arbeiten, muss das funktionieren an der Stelle.

Interviewer: Klar, ja. Ok. Du hast vorhin noch das Stichwort Kontrolle angesprochen, das finde ich in dem Sinne interessant, weil die agilen Ansätze eigentlich keine Management-Kontrolle haben möchte, aber du in dem Sinne gesagt hast, es gibt jetzt hier doch ein Kontrollelement und zwar ist das die Demo.

Befragter: Naja, was heißt Kontrollelement, irgendeiner, irgendwann muss man ja mal entscheiden, ob das, was da raus gekommen ist nach dem Inkrement des ist, was man erwartet hat, oder nicht. Ich sehe diesen Sprint oder das was da aus gemacht wurde, als Vertrag, Minivertrag zwischen dem Team und der dafür zahlt, Product-Owner.

Interviewer: Die Frage ist nur, siehst du das als Kontrolle in dem Sinne, dass du das Team daran misst an der Stelle, oder ist es eher du möchtest eher über den Mechanismus in diesem Projekt das Feedback darüber geben. Von der Aufstellung ist das ja ein unterschiedlicher Ansatz, ich gebe dem Team ein Feedback zurück und gebe ihnen nicht gleich die Watsche, weil sie s nicht richtig gemacht haben und schmeiss zwei Leute raus, ja

Befragter: Ja gut, das ist jetzt extrem ....

Interviewer: das ist jetzt extrem, aber da gibt es schon ein Unterschied, ob ich versuche ein Team zu kontrollieren und dann versuche einzugreifen und einzuhaken oder ob ich das sein lasse, indem ich das nur Kontrolliere und ein Feedback gebe und sage, das hat mir nicht gefallen. Eich passiert jetzt an der Stelle nichts.

Befragter: Also, ich meine, das würde ich noch einmal ein bisschen unterscheide, in welcher Reife oder wie lange so ein Projekt läuft. Scrum ist immer ein evolutionärer Prozess[T4.16], d.h. es ist in dem Prozess-Framework eingebaut, dass man gewisse Zeit braucht, bis man herausgefunden hat wie es funktioniert. Das ist halt in der Software, ist jedes Projekt anders. Man kann die Erkenntnisse von einem auf das nächste Projekt nur bedingt mitnehmen.[T4.17] Es sind vielleicht andere Teamleute, der Kunde ist anders, Technologie ist anders, Qualitätsanforderungen sind anders. D.h. am Anfang denke ich schon

ist der Lernaspekt am wichtigsten. Wenn die ihre ersten Schätzungen machen, ist eigentlich normal, dass es am Ende schief geht und wichtig ist dann da der Punkt, es raus zu arbeiten, wie man sich da verbessern kann oder wie man es besser in den Griff bekommt. Klar wenn dann das eingespielt ist und eigentlich ein Team; Kontrolle ist so ein komisches Wort, man überprüft hinterher das Ergebnis und analysiert man immer erstmal wertneutral, warum es nicht so war, vielleicht gibt es ja gute Gründe und ich sag dir in der Regel ist es nicht so, dass es heißt ok, der eine war jetzt ein wenig faul in dem Sprint. Das kommt bei uns eigentlich nie vor, also d.h. es wird nie irgendwie dafür verantwortlich gemacht. Es gibt immer gute Gründe, wichtig ist es am Ende, das heraus zu arbeiten. Erwartung ist natürlich schon, dass wenn so ein Ding schon mal ein, zwei Jahre läuft, dass man eine gewisse Zuverlässigkeit da drin hat. Das ist auch so, deswegen ja, da gibt es personelle Konsequenzen gibt es ja nie.

Interviewer: Ich meine ja nur, das kann man ja auch unter Kontrolle verstehen oder wenn dich der Chef kontrollieren kommt ist das eine ganz andere Art von Kontrolle, als wenn ich sage du machst eine Kontrolle auf Augenhöhe. Das ist eigentlich mein Verständnis, wie es in einem agilen Umfeld sein muss.

Befragter: Das ist es genau 100% deswegen das Wort.

Interviewer: Deswegen habe ich da noch einmal nachgehakt, weil du gesagt hattest, da kommt der Kunde zu euch und er macht dann eine Kontrolle, das hört sich für mich schon fast so wie ein Audit an, ...

Befragter: In dem Fall ist es so, dass man das im direkten Kontakt machen kann, weil er das richtige Mind-Set dafür hat. Ich hatte schon Projekte, die sind von Kundenseite her in einer klassischen Denkweise gewesen, da hat es sich nicht empfohlen eine direkte ....

Interviewer: ... da war es dann wirklich eine Kontrolle und nicht ....

Befragter: ... und vor allem auch eine entsprechende, da liefs auch schlecht in dem Projekt, zumindest am Anfang, sind wir da weit unter unserer Schätzung gewesen, sagen wirs mal so, müssen wir weiter ausholen, wenns dich interessiert.

Interviewer: ... kein Problem, es muss jetzt nicht so weit ausgeholt werden, ich wollte nur mal wissen, die Situation hast du auch schon erlebt, wo es eine reine Kontrolle war.

Befragter: Ja, klar, da muss man das anders, im klassischen Teil, den Einlauf abholen, des muss dann halt der Teamleiter machen.

Interviewer: ... der filtert das wichtige für das Team raus.

Befragter: genau.

Interviewer: Du hast ja auch schon einiges genannt, was du gegen die Probleme getan hast, du hast ja schon erzählt, wie du die Colocation aufgelöst hast. gibt es noch mehr in den anderen Situationen, wo du noch andere Mittel eingesetzt hast?

Befragter: Ein Ansatz sind halt technische Dinge, wenn man merkt halt, es sind halt Qualitätsprobleme, die natürlich auf die Leistung niederschlagen, dann muss man halt mit dem Team zusammen überlegen, was man halt verbessern kann, um am Ende bessere Qualität herzubringen, also automatisierte Tests einführen und so weiter und so fort. Also ansonsten ist halt grundsätzlich die Retrospektive aus meiner Sicht das A und O und das ist für mich gleichzeitig der größte Bad-Smell[T4.18] und die sagen: „Nee, Retrospektive haben wir mal am Anfang gemacht und das finden manche ein bisschen lächerlich.“[T4.19], kann auch eine gewisse Routine bekommen, aber es ist trotzdem ein wichtiges Meeting[T4.20]. Es muss nicht eine Stunde dauern, wenn man in 10 Minuten alles wichtige Adressiert ist, dann ist das auch Ok. manchmal ist auch alles prima und dann lässt man es halt, lässt man es nicht, aber macht es sehr kurz. Das ist aber der wesentliche Mechanismus, um solche Sachen, diese Dinge abstellen kann und die Einzelmaßnahmen definieren, so generell muss man das dann halt tun. Das sind halt zwei wichtige Erfolgsfaktoren, ich mache es immer so, dass wir schon eine Liste machen, aber dann suchen wir uns drei Punkte raus von den Verbesserungsmaßnahmen und von denen sagen wir dann konkret, was wird gemacht und wer machts. Das wird dann mit in der nächsten Retro mit angeguggt, ansonsten ist das halt nur drüber geredet[T4.21].

Interviewer: Und dann passiert nix. Wie du auch schon gesagt hast, da gibts den Bad-Smell, dass man keine Retrospektive mehr macht, das ist ja auch schon ein Stück, wo die Selbstoriganisation vom Team verloren geht. Vielleicht hier die Frage, warum wurde die Retrospektive nicht mehr gemacht? Was waren da die Gründe?

Befragter: vielleicht, ja, warum wollen sie es nicht machen ... gute Frage .... ich kann mir nur erklären, dass sie darin keinen Nutzen gesehen haben. Dafür ist ja der Scrum-Master da, dass er das noch einmal motiviert und das ist

insbesondere mit frischen Teams, die das noch nicht so gewohnt sind oder aber erfahrene Leuten die das auch noch nie gemacht haben, die finden das am Anfang kindisch. So jeder sagt jetzt mal so ein bisschen, lustig mit anfassen und so, da gibt es ja verschiedene Auflockerungsmoderationen, da muss man auch aufpassen, dass es nicht zu lustig ist, aber wenn sich die Leute mal drauf einlassen, macht es schon Sinn; aber da muss man als Scrum-Master auch dafür sorgen, dass dies auch statt findet, also das wird dann oft einfach fallen gelassen, warum auch immer; ich weiß es nicht.

(*Anm.: Es gab ein kurzes Privatgespräch, das zum Inhalt nicht beigetragen hat. Dies wurde entfernt.*)

Befragter: Mir fehlt jetzt noch ein Punkt von vorne bezüglich negative Auswirkung auf Performance, vorher ist er mir heruntergefallen. Gift ist, wenn jemand immer wieder aus dem Kontext heraus gerissen wird. Also sie sollen auch voll in diesem Projekt, in dem sie arbeiten, bewegen können. Das ist ja klar, es ist ja nicht nur, das da die Zeit fehlt, wenn er einen halben Tag irgendwas anderes machen soll, sondern es ist halt dieser Kontextwechsel und der Flow geht damit verloren[T4.22]. Das ist schon, ich halte nicht viel von mehreren Projekten, mindestens mal in dieser Iterationszeit sollte man die Planung stehen lassen[T4.23]. Damit zerstört man auch wieder die Selbstkontrolle. Ich verlange von den Leuten sie sollen ihre Zeit ausplanen für diese zwei Wochen, sie geben sich mühe und schaun auch, dass sie das erreichen und dann kommt irgendeiner und sagt jetzt machst du mal einen Tag das. Dann verlieren sie logischerweise die Motivation für die Selbstplanung, wenn sie die dann auch nicht einhalten können. Ja, Ziele, für die du nicht selbst verantwortlich bist, die sind für dich auch nicht attraktiv[T4.24]. Das ist ein psychologisch ein ganz wichtiger Punkt, dass man die dann auch lässt, dass man die auch machen lässt, wie sie sich das geplant haben, weil dann geht die Selbstorganisation auch flöten. Das ist jetzt nicht schwarz, weiß, es gibt immer mal Notfälle, wo man was machen muss, das versteht jeder, aber wenn das zur Regel wird, dass der Plan gestört wird, dann wird das als sinnlos empfunden. Das ist mir jetzt gerade noch eingefallen.

Interviewer: Das stimmt, das stimmt schon. Du kannst auch gerne zurück springen. Mich interessiert natürlich auch die Diagnostik, wie bemerkt man, dass in einem Team etwas nicht stimmt. Mir geht es darum Möglichkeiten zur Diagnose zu finden. Woran bemerkst du eigentlich, ich sag mal, der Mensch kommt in einen Raum rein und bemerkt da stimmt etwas nicht, alleine vom Gefühl her. Wenn ich mir ein Team anguggt und es zwei Stun-

den beobachtet, dann merke ich schon ob das passt, was die Leute gerade machen oder ob jetzt alle schweigen und überhaupt nicht kommunizieren, oder wenn ich mir eine Besprechung anschaue, dann habe ich ein Gefühl. Die Frage ist, wie bemerkst du in deiner Arbeit, dass da Probleme da sind, die du dann auch mal ansprichst?

Befragter: Das ist mehr oder weniger theoretisch, weil bei uns meistens alles funktioniert. Das hört sich blöd an, aber ich würde es schon merken, weil wir haben bestimmte Rituale, die wir einfach machen und wenn sich jemand da raus ziehen würde, ist das ganz offensichtlich, dass er sich da nicht wohl fühlt[T4.25]. Was weiß ich, wir gehen nach dem Stand-Up Kaffee holen und dann wird erstmal gequatscht. Wenn einer da keine Lust hat mitzugehen und es gehen sogar die mit, die keinen Kaffee trinken, beispielsweise. Oder wir gehen halt zusammen Essen, normal oft, wenn sich jemand raus ziehen würde und auch Abends, in der Regel gehen wir alle ein, zwei Sprints halt Abends nach dem Sprint halt noch eine Pizza essen halt so was in der Art, wenn sich da jemand raus zieht aus diesem sozialen Kontext, da merkt man natürlich, dass er oder sie da ein Problem hat[T4.26]. Retro ist ganz klar, ist auch ein wichtiger Punkt. Wenn man eine Kultur etabliert hat, wo sich normalerweise jeder äußert, würde er das dann da normalerweise tun. Oder wenn er sein Verhalten ändert und sich nicht mehr äußert oder anders äußert würde man das natürlich auch merken[T4.27]. Ich führe normalerweise, weil es meistens so ist, dass ich Teamleiter bin, individuelle Gespräche eh mit den Leuten[T4.28] und wenn man eine Kultur etabliert hat, in der man sich auch offen Feedback gibt, wird das auch normalerweise angesprochen. Ich geh davon aus, dass normalerweise Leute zu mir kommen und sagen, hier das und das passt mir nicht[T4.29]. Das passiert auch, das ist jetzt nichts so dramatisches, aber das kann man dann eben entsprechend beeinflussen. Wir hatten jetzt mal so eine Situation, da ist vom Kunden aus dem Team einer sozusagen ein bisschen herausgehoben worden. Es war ein älterer Entwickler bzw. ein erfahrener Entwickler in dem Team, dem hätte vielleicht diese Rolle zugestanden, so eine Art Architekt, aber der Kunde ist halt mit dem anderen anscheinend besser zurecht gekommen oder hat den halt präferiert. Dann hatte er halt gesagt, da kommst und mal zweit Tage mit zu dem und dann müssen wir das mal klären. Da hat man dann schon gemerkt, der andere ist halt vielleicht ein bisschen ruhiger, dann spricht man das halt mal an, am Besten mit beiden zusammen, wie ist das denn jetzt mich euch, ich habe so das Gefühl. Man kann es ja nicht ändern, aber man merkt schon das der mal

ein bisschen komisch guggt oder so, aber sind alles keine großen Sachen. Tut mir leid, dass ich nicht mit mehr Problemen dienen kann.

Interviewer: Du hast halt nicht so viele, vielleicht einmal anders herum gefragt, was denkst denn du, welche Voraussetzungen in einem solchen Team vorhanden sein müssen, dass es so problemlos abläuft, vom Team innern. Das ist eigentlich schon auch so ein Schlüssel, dass man sagt das man solche Sachen besonders fördert oder darauf achtet, ob die das dann nicht mehr tun. Du hast ja vorhin erzählt, die gehen dann gemeinsam mal eben alle zwei Sprints zum Essen, das sind ja dann schon solche Rituale.

Befragter: Die sollte man natürlich einführen, das ist schon sehr viel wert, dass es nicht nur drum geht Arbeit, Arbeit und dann tschüss und dann hat man nichts mehr miteinander zu tun. Aber so eine Teamzusammensetzung, das ist natürlich klar, es ist schon wichtig, dass alle so das Mindset haben, das sind halt schon bestimmte, das hab ich vorhin schon mal gesagt, wenn jetzt halt angenommen das ist jetzt einer der jetzt schon seit 20 Jahren klassisch Software entwickelt und den man da rein mischen würde, da würde es mit Sicherheit Probleme geben. Das passiert auch, wenn man das mit Kunden zusammen machen. Bei uns ist eigentlich eher so, da sind die Leute halt auf einem Alterslevel und auf einem Verständnislevel, darf ich in einem anderen Code rumwursten, wie offen darf ich jemand anderes sagen, das es jetzt suboptimal ist, also von der Technik her oder sonst irgendwas. Das wird dann schnell deutlich, wenn man so Reviewtools einführt, wie gut das funktioniert, also ja Leute mit ähnlichem Mindset, das tut dem Team natürlich schon auch gut, wenn nicht alle gleich ticken, ja es gibt halt immer, die müssen sich ja irgendwo auch ein bisschen reiben können, an bestimmten Punkten, ich sag jetzt mal, es gibt ja jetzt so mehr von der Teampsychologie, ich nenne es mal so kreative Chaoten, die haben halt super Ideen, sind jetzt aber nicht so brillant im fleißigen Ausarbeiten, den brauchst jetzt nicht als Testspecschreiber einsetzen, der wird sich in diese Rolle nicht einfinden. Ich teile jetzt auch keinen ein in dem Sinn[T4.30]. Aber dafür brauchst du schon auch so einen konservativen Bewahrer-Typ in so einem Team, oder so einen Bedenkenträger[T4.31], der aber auch immer mal schaut oh, habt ihr an das gedacht, das ist doch gefährlich und so weiter und so fort. Wenn du nur solche drinnen hast ist es auch schwierig, weil einer muss auch mal Ideen entwickeln. So wenn man da die Wahl hat, dann finden sich auch die richtigen Rollen. Der eine ist dann dieser Bewahrer[T4.32], der ist dann der Fels im Reviewprozess, der findet auch noch die letzten Problemchen, das ist der, der

privat abends die Changes von der git-Entwicklung selbst mal durchliest und so was, weil es halt so toll ist oder von der Kernel-Entwicklung.

Interviewer: Da kann man viel draus lernen.

Befragter: Kann man viel lernen und der ist natürlich auch ein brillanter Partner. Die Leute müssen aber auch verstehen, dass er da seine Rolle spielt. Wenn er natürlich das Image des Meckerers, des notorischen bekommt, das bringt natürlich auch nichts. Das müssen sie schon auch Wert schätzen. Und dafür ist auch genau so der da irgendwelche wüsten Ideen entwickelt genauso wertvoll. So eine gewisse Tamzusammensetzung ist schon wichtig. Aber hat man halt nicht immer im Lösungsraum verfügbar[T4.33]. Man hat Anforderungen von Kunden und Verfügbarkeiten die man hat, eigentlich meistens ist man froh, wenn man es irgendwie hin kriegt.

Interviewer: Gut, das optimale Team kann man sich ohnehin nur auf der Tafel aufmalen.

Befragter: Klar

Interviewer: weil das wir man nicht bekommen; man muss es halt nur hinbekommen, dass es zusammen funktioniert. Nachdem du ja so wenig Probleme hattest, ist das natürlich auch eine schlechte Frage für dich, ob dir irgendwelche Situationen oder Probleme bekannt sind, wo ein Team das nicht erkannt hat. Gerade wenn es um das Stichwort Selbstorganisation geht, wo die immer weiter gemacht hat und jeder von außen sich gewundert hat und das Team war davon überzeugt, dass das alles genau das richtige ist, was sie da tun.

Befragter: Im agilen Kontext.

Interviewer: Eigentlich im agilen Kontext, wo das Team auch die Möglichkeit hat, sich auch zu verändern, anders zu organisieren und solche Probleme abzustellen. ... Oder wo man Probleme ignoriert hat und gar keine Lust hatte, sich darum zu kümmern ....

Befragter: Das sind Fragen ...

Interviewer: Ich versuche jetzt ein bisschen dein Erfahrungswissen anzuzapfen.

Befragter: Das ist ja legitim (*Anm.: denkt wider länger nach*) Ich weiß nicht, jedenfalls nicht dauerhaft, ja ich mein es gibt immer Situationen, aber ich wüsste nicht, wo man es dann nicht wieder durch Reflexion hin gebracht

hatte. Wenn halt das Ergebnis der Außenerwartung nicht entspricht, keine Ahnung, kommt viel zu langsam voran, dann wird das halt auch irgendwie gesagt, gerade wenn du nach Stunden bezahlt wirst, denn dann zahlt der Kunde natürlich jede Stunde und der hat sich das eigentlich viel schneller vorgestellt, das wird dann diskutiert und wieder gerade gerückt, was heißt gerade gerückt oder man argumentiert es halt, aber das Ziel ist jetzt nicht die Antwort auf deine Frage.

Interviewer: Das ist jetzt weniger das, woran ich hier denke. Ich denke halt jetzt wirklich mal an das Team selbst, wenn man jetzt mal Bezahlung außen vor lässt.

Befragter: Jaja, ist schon klar, da habe ich jetzt irgendwie keine Idee, was ich darauf sagen sollte.

Interviewer: Bist du ratlos. Gut, dann eine andere Frage, das ist zum Teil schon ein bisschen theoretisch, wo unterstützt die das Scrum-Framework besonders bei deiner Diagnose?

Befragter: Bei meiner Diagnose?

Interviewer: Also von Problemen, von Situationen, wo etwas nicht stimmt.

Befragter: Das gute ist halt, das du halt diese fest definierten, also du musst nicht was großes anstreben oder Punkte definieren, das Framework gibt ja ständig die Möglichkeiten, das anzuschaun. Du hast ja jeden Tag das Stand-Up und kannst anschaun, wie das abläuft[T4.34], wie die Leute halt sich berichten oder wem sie berichten. Du hast halt immer engen Kontakt mit den Leuten. Das ist halt anders wie bei einem Projektmanager oder Produktmanager, der hat normalerweise relativ wenig Kontakt. Da kannst du Veränderungen mit der Zeit relativ gut wahrnehmen. Du hast natürlich viele Meetings mit den Leuten, wie die Retro usw. wo du sehen oder diagnostizieren kannst[T4.35].

Interviewer: Noch eine Frage, gibt es aus deiner Sicht Grenzen des Scrum-Frameworks bzw. dessen praktische Auswirkung hinsichtlich dem Erkennen von Problemen? Wo siehst du, dass dir das Scrum-Framework nicht ausreicht, wo hättest du gerne mehr Erklärungen im Framework gehabt oder wo du Sachen oder Methoden eingesetzt hast, die nicht so typisch sind. Wo du aus deiner Sicht sagst, das ist total essentiell für deine Arbeit und du aus deiner Sicht sagst: „Warum ist das da nicht drinnen?“

Befragter: Sagen wir es mal so, das Grundproblem ist, dass es im Kern sehr generisch definiert ist. Man kann es nicht einfach her nehmen und tun, weil es wird nicht genau gesagt, wie man Schätzen soll und die ganzen Geschichten; es ist einfach nur der Rahmen erklärt. Es gilt halt, das mit dem Kontext auszufüllen. Wir haben einige Erweiterungen oder Anpassungen in unserem Bereich gemacht, also Medizintechnik-Software. Wenn man es da so einfach gestaltet, reicht es nicht aus. Es ist jetzt kein Widerspruch, sondern eine Ergänzung. Beispielsweise gibt es im Scrum immer nur Sprints. Es gibt am Ende immer eine Definition of Done. Gut das ist recht generisch ja. Wenn du jetzt ein Medizintechnikprojekt machst, wirst du am Anfang andere Sprints haben, als in der Mitte und am Ende, also Arten von Sprints. Deswegen haben wir das erweitert, dass es am Anfang so Vorbereitungs und Planungs-Sprints, da wird auch weniger geschätzt, weil es müssen bestimmte Dokumente entstehen, die regulatorisch gefordert sind, die muss man auch vorher machen. Das hilft nichts, man kann nicht alles, jedes Feature einfach nur per Sprint entwickeln, es muss gewisse Vorplanungsdokumente geben, wie den Entwicklungsplan, Risikoanalyse muss vorbereitet werden, da kann man nicht sagen, das macht man bei jedem Feature iterativ mit, wie die Denkweise wäre. Die Denkweise wäre, man macht für jedes Feature immer quasi alles. Das haut halt so nicht hin, das widerspricht nicht Scrum, aber es braucht halt ein wenig eine Erweiterung. Also diese Planungsdokumente, eine gewisse Architekturelle Vorüberlegung, das kann man dann eben als Spikes machen, das ist auch ganz sinnvoll und bestimmte Dokumente müssen angefangen werden. Dann gibt es halt so die größere Phase der Umsetzung, das ist so der klassische Sprint, wo es halt auch am Ende eine gewisse Qualitätskontrolle gibt, aber die regulatorisch keine Bewandtnis hat. Das dient allerdings nur dafür, dass die Qualität immer hoch ist, aber es befreit nicht davon, dass am Ende bei der Fertigstellung noch verifiziert und validiert wird. Deswegen gibt es bei uns am Ende so typischerweise eine Reihe von Hardening- und Releasesprints, wo dann auch Releasedokumentation erstellt wird. Das kann man alles in diesem Scrum-Rahmen erledigen, wo man sagt, dass die verschiedenen Dokumente finalisiert werden, es muss Approvals geben und dann wird auch formal verifiziert und validiert. Auch da wiederum ist kein besonderer Wert, dass man das vorher abschätzt. Bei der Validierung hat man ein Gefühl, aber es hilft mir nicht, das noch weiter runter zu brechen, wie Kapitel 1 des Validierungsplans sind 3 Punkte, sondern man weiß ungefähr so eine Validierung dauert halt 2 Wochen und dann setzt man einen Sprint dafür an. Da hilft mir Scrum nicht besonders, da bräuchte ich es auch nicht,

das ist ganz klar. Da gehen wir dazu über, wenn so etwas ist, wenn man sagt: „Ok, wir sind fertig mit der Entwicklung und es gibt ein Releasedatum, da weiß man, was zu tun ist, das muss einfach erledigt werden." Da würde man dann eher in so einen Kanban-Modus übergehen, wo man sagt, das sind die Pakete, die noch gemacht werden müssen, die müssen wir einfach noch abarbeiten. Dann sieht man anhand der offenen Punkte noch, ob das jetzt funktioniert oder nicht, aber da hilft mir das nichts, wenn ich das jetzt noch in 3 Iterationen einteile.

Interviewer: Habt ihr da dann Iterationen mit einem offenen Ende, oder die dann keine bestimmten Scope haben?

Befragter: Naja, der Scope ist dann halt einfach länger gefasst, man kann halt sagen: „Ok, jetzt sind wir fertig, jetzt sind es noch vier Wochen bis zum geplanten Release." Da würde man dann keine zweifache Unterteilung mehr machen. Man macht schon normal Stand-Up, um den Fortschritt zu checken und sich abzustimmen, „Wie geht es voran?" du hast es ja am Ende nicht komplett unter Kontrolle, weil wenn bei der Verifizierung oder Validierung nochmal ein Problem auftritt, muss man halt ran, das hilft halt nichts. Da hilfts am Ende relativ wenig und am Anfang ist es natürlich auch schwer planbar bzw. sagen wir mal so, die ersten Sprints haben typischerweise keinen Wert der entsteht und das es ein shipable Product ist, das ist natürlich Quatsch, weil du fängst halt am Anfang viel mit Infrastruktur an. Ich schreibe auch dann keine Stories wie: „Als Entwickler möchte ich ein continous integration system haben, damit ich ständig über die Qualität des Produktes informiert bin.", das ist natürlich Blödsinn. Wir sagen einfach, wir setzen einen Jenkins auf, so aus die Maus. Also da hat das natürlich seine Grenzen und wie gesagt die Definition of Done, das ist natürlich ein ganz wichtiges Instrument, gerade wenn du am Ende mehr benötigst, als funktionierende Features, da haben wir den Mechanismus vorgegeben im Scrum, aber da bedarf es einige Erfahrung, was man da rein schreibt. Die sieht pro Phase, in der man ist auch ein wenig anders aus. In der Entwicklungsphase braucht man am Ende kein unterschriebenen Test-Report, aber da heißt es dann: „Es muss halt im Test-Tool erkennbar sein, das es einen Smoke-Test gegeben hat.", sowas in der Art wäre eine Definition of Done am Ende vom Sprint. Nachher in der Releasephase heißt es dann, es ist dann ein unterzeichneter Test-Report. Ja, Grenzen, Hardwareentwicklung ist auch immer mit den typischen kurzen Zyklen eine Herausforderung. Das haben wir jetzt auch mal gemacht, da hat man halt nicht alle zwei Wochen eine sinnvolle neue Version. Das hat auch ein bisschen seine Grenzen, wenn man insgesamt an

Systementwicklung denkt. Scrum hat schon seine Stärke in der Softwareentwicklung.

Interviewer: Das war es dann. Danke dir.

D

Anhang D

Abkürzungsverzeichnis

BTRSPI Belbins Team-Roles Self-Perception Inventory

TMS Team Management System *Glossary:*Team Management System

XP Extreme Programming *Glossary:* Extrem Programming

E Anhang E

Glossar

Backlog (im Scrum auch Product Backlog) ist eine Liste von Items (z.B. Features oder Aufgaben) welche eine Priorität anhand ihrer Ordnung erhalten und zur Umsetzung bereit sind. [53]

Behavior Driven Deleopment (Verhaltens-getriebene Entwicklung) ist eine Vorgehensweise in der Softwareentwicklung, bei der das Verhalten der Software vor Erstellung einzelner Komponenten beschrieben wird. Meist wird das Verhalten in "Wenn-Dann" Sätzen erfasst, z.B. Wenn der Kunde eine Karotte gekauft hat, dann haben wir eine Karotte weniger im Lager. [46]

Collective Code Ownership Code Ownership gibt an, wie verfahren wird, wenn Quelltexte der Software zu ändern sind. Bei einer Collective Code Ownership wird davon aus gegangen, das der Code einem Team oder einer Gemeinschaft gehört und jeder Entwickler an diesem Code Änderungen vornehmen kann, ohne vorher um Erlaubnis zu fragen. [100]

Definition of Done In der Softwareentwicklung wird dieser Begriff genutzt, um ein einheitliches Verständnis zu schaffen, wann eine Aufgabe als abgeschlossen anzusehen ist. Wie diese definiert wird ist sehr stark vom Kontext und dem Team abhängig. Oftmals sind dort Element zu finden wie: Es muss übersetzbar sein, es muss dokumentiert sein, es muss ein review statt gefunden haben, ... [250, 265, 266]

Extrem Programming ist eine leichtgewichte Methode der agilen Softwareenwicklung, bei der viel Wert auf die direkte Kommunikation mit dem Kunden und Feedback gelegt wird. (vgl. Beck 1999) [46, 269]

Feature Driven Development (Feature getriebene Entwicklung) ist eine agile Methode der Softwareentwicklung und wurde durch Jeff De Luca im Jahre 1997 entwickelt (vgl. Nebulon Pty. Ltd 2015; Coad, Luca und Lefebvre 1999). Im Zentrum der Methode steht das Feature, anhand derer die Entwicklung geplant und entlang eines Feature-Plans organisiert wird. [46]

Impediment (Hindernis) Impediment ist ein Begriff aus der agilen Welt. Er bezeichnet ein Behinderung, so dass die aktuelle Arbeit beeinträchtigt wird oder es unmöglich ist, diese zu vollenden. [90]

Pair-Programming Pair-Programming ist eine Methode aus dem agilen Methodenkoffer, bei der sich zwei Entwickler gemeinsam mit einem Programmieraufgabe befassen und dabei als Paar an einem Rechner sitzen. Einer der beiden bearbeitet den Quelltext, während der andere nachdenkt und den geschriebenen Quelltext überprüft. Entdeckte Probleme oder Verbesserungen können sofort diskutiert und eingearbeitet werden. [104]

Refactoring Refactoring, kann als Restrukturierung übersetzt werden. Dies ist eine Technik, welche bei der Software-Entwicklung angewandt wird, um die Struktur eines Quelltextes zu verbessern. Der Kernpunkt ist, dass sich das Verhalten des Programmes nicht ändert, aber die interne Struktur soweit verbessert wird, das es z.B. besser Lesbarer, Erweiterbar oder Wartbarer wird. [100]

Retrospektive (lat. retrospectare „zurückblicken“) bezeichnet einen Rückblick. In agilen Teams wird damit die eigene Arbeitsweise reflektiert. Dabei kommen unterschiedliche Methodiken zum Einsatz (vgl. Derby, Larsen und Schwaber 2006). [48, 51]

Servant Leader siehe Servant Leadership [58]

Servant Leadership wird auch als dienendes Führen bezeichnet. Es ist eine von Greenleaf beschriebene Philosophie der Führung (Greenleaf 1977), bei dem der Führende die Geführten anleitet und unterstützt, damit diese sich entwickeln und eine hohe Leistung erbringen können.

"'Servant leadership is a holistic approach to leadership that engages both leaders and followers through its (1) service orientation, (2) authenticity focus, (3) relational emphasis, (4) moral courage, (5) spiritual motivation, and (6) transforming influence such that they are both

transformed into what they are capable of becoming."' (Sendjaya 2015, S. 1) [272]

Team Management System ist eine Methode zur Organisationsentwicklung welche durch Charles Margerison und Dick McCann in den 1980er Jahren entwickelt wurde. Es schließt von der Persönlichkeit auf die Rolle im Team. Dazu bildet es ein Modell der Arbeitsfunktionen auf die Persönlichkeitsmerkmale ab (vgl. Margerison 2002; Tscheuschner und Wagner 2008; Tscheuschner 2012). Die Weitergabe des Wissens erfolgt hauptsächlich durch Kurse der Beraterfirma *TMS-Zentrum* 2015. [27, 269]

Test Driven Development (Test getriebene Entwicklung) ist eine Vorgehensweise in der Softwareentwicklung, in der vor der Entwicklung und Implementierung der Komponente die Tests formuliert werden. [46]