

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Interviewstudie zum Verständnis und zur Nutzung von Scrum in der Softwareentwicklung

Bachelorarbeit

im Studiengang Informatik

von

Leon Schlecht

**Prüfer: Prof. Dr. Kurt Schneider
Zweitprüferin: Dr. Jil Klünder
Betreuerinnen: Dr. Jil Klünder und Melanie Busch**

Hannover, 14. September 2020

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 14.09.2020



Leon Schlecht

Zusammenfassung

Scrum ist die meistverwendetste Vorgehensweise in der Softwareentwicklung und hat sich in den letzten Jahre immer mehr verbreitet. Jedoch wird häufig von der Theorie abgewichen, weshalb diese Ausarbeitung im Rahmen einer Interviewstudie Umsetzungen von Scrum in der Praxis untersucht hat, um die Hintergründe etwaiger Abweichungen besser verstehen und nachvollziehen zu können. Um dies zu erreichen, wurden mit zehn Teilnehmern aus acht Unternehmen der Softwareentwicklung Leitfadeninterviews durchgeführt und mithilfe der qualitativen Inhaltsanalyse nach Mayring ausgewertet.

Die Auswertung hat gezeigt, dass keins der interviewten Unternehmen Scrum immer in vollständiger Übereinstimmung zur Theorie verwendet, sondern immer auf irgendeine Art und Weise irgendwo Abweichungen macht. Allerdings gab es drei Unternehmen, die nur selten abweichen und sich, wenn möglich, an die Theorie halten. Von den restlichen sieben Unternehmen kommen mal mehr und mal weniger Abweichungen vor. Zwar versuchen viele der Unternehmen die Theorie einzuhalten, jedoch entstehen eine Vielzahl der Abweichungen nicht wirklich als eigenständige Entscheidung, sondern durch externe Faktoren, die sich schwer beeinflussen lassen. Häufig gibt es Probleme durch den Product Ownern ohne Entscheidungsgewalt oder unerfahrenen Product Ownern, die zu ungünstigen Eingriffen in den Sprint führen. Auf der anderen Seite legen die Unternehmen Wert auf regelmäßiges Kundenfeedback und halten sich größtenteils an die kurzen Iterationszyklen der Sprints.

Insgesamt konnte die Interviewstudie erneut zeigen, dass Scrum in vielen Fällen nicht nach der Theorie umgesetzt wird und diverse Abweichungen gemacht werden. Auswirkungen von den einzelnen Abweichungen sind im Wesentlichen nicht bekannt und daher weiß man nicht, wie schädlich oder vielleicht auch wie förderlich manche dieser Abweichungen sind.

Abstract

Interview Study on the Understanding and Use of Scrum in Software Development

Scrum is the most common approach for the development of software and has become more and more prevalent in recent years. However, practitioners often deviate from the theoretical aspects of Scrum, which is why this thesis examines those in an interview study to find out more about the reasons behind deviations in order to better understand them. To accomplish this, ten participants from eight companies were interviewed with a guideline-based approach and its results were analysed with Mayrings Qualitative Content Analysis.

The analysis has shown that none of the interviewees uses Scrum in full accordance to its theory. Instead, every company deviates from the theory in some way. However, there were three companies that rarely deviate and whenever possible try to avoid deviating from theory. Of the other seven other participants, some deviate more and some deviate less. Although many companies try to act in accordance to theory, many deviations happen not by their own choice, but rather due to external factors, which they have little to no control over. There are often problems with Product Owners who lack the authority to make decisions or are unexperienced, both of which leads to inappropriate changes of the sprints' scope. On the other hand, many companies put emphasis on getting regular feedback from the customer and they try to have short iteration cycles due to their sprints.

To sum things up, yet another interview study could show, that Scrum is in many cases not practiced according to its theory and many deviations emerge instead. Consequences of particular deviations are largely unknown and it is therefore unknown how damaging or perhaps even how helpful some of these deviations can be.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation und Zielsetzung.....	1
1.2 Forschungsfragen und Methodik.....	2
1.3 Struktur der Arbeit.....	2
2 Scrum	4
2.1 Allgemeiner Überblick.....	4
2.2 Scrum Artefakte.....	5
2.3 Scrum Team.....	6
2.4 Scrum Events.....	8
3 Interviews	11
3.1 Teilnehmerakquise.....	11
3.2 Interviewdurchführung.....	12
3.3 Auswertungsmethodik.....	14
4 Ergebnisse	17
4.1 Teilnehmer- und Unternehmensübersicht.....	17
5.2 Auswertung.....	19
5.2.1 Scrum Artefakte	19
5.2.2 Scrum Team	22
5.2.3 Scrum Events.....	24
5 Diskussion	29
5.1 Einschränkungen der Validität.....	29
5.2 Beantwortung der Forschungsfragen	30
5.3 Interpretation.....	34
6 Verwandte Arbeiten	38
7 Fazit und Ausblick	43
Anhang 1: Interviewleitfaden	46

1 Einleitung

Scrum ist seit seiner erstmaligen Veröffentlichung [1] auf der OOPSLA¹ Konferenz im Jahre 1995 mittlerweile die Vorgehensweise der Wahl bei der Entwicklung von Software geworden [2]. Auf Scrum zu verzichten ist schon fast die Ausnahme geworden, wie es die Daten der HELENA² Studie [2] oder der 14. jährliche “state of agile report” [3] zeigen. Des Weiteren wird aus den HELENA Ergebnissen ersichtlich, dass nicht nur Scrum, sondern auch viele andere agile Vorgehensweisen einen hohen Stellenwert in der Softwareentwicklung gefunden haben [2]. Doch diese weite Verbreitung bringt nicht nur Vorteile mit sich, denn immer häufiger hat sich gezeigt, dass Agilität missverstanden und falsch interpretiert wird [4]. Dies hat zur Folge, dass die angestrebten Vorteile, beispielsweise schneller zu entwickeln oder flexibler auf den Kunden reagieren zu können, der Gefahr unterliegen, auszubleiben [4]. Zum einen ist dies auf die explosionsartige Verbreitung von Agilität zurückzuführen, woraufhin viele plötzlich den Eindruck bekommen haben unverzüglich agil werden zu müssen und somit nicht die notwendige Zeit investieren, sich ein tiefgründiges Verständnis von Agilität anzueignen [4]. Deshalb entstehen häufig Abweichungen, dessen man sich vielleicht nicht mal bewusst ist [4]. Obwohl sich anzupassen einen hohen Stellenwert in der Agilität einnimmt, sollten Abweichungen aus einem tiefen Verständnis heraus im Sinne eines Verbesserungsprozesses passieren und nicht wahllos [4, 5]. Generell konnten die agilen Prozesse bei vielen Unternehmen sich noch nicht ausreichend ausreifen [4].

1.1 Motivation und Zielsetzung

Längst ist dies kein unbekanntes Problem mehr und es wurden bereits mehrere Untersuchungen angestellt, um die Hintergründe von Abweichungen in der agilen Welt zu verstehen, beispielsweise von Diebold et al. [6]. Da Scrum das weitverbreitetste Vorgehen in der Softwareentwicklung ist [2] und eine Untersuchung [7] ergeben hat, dass 643 verschiedene Variationen von Scrum möglich sind, fokussiert sich diese Arbeit auf Scrum und nicht auf andere agile Vorgehensweisen. Häufig ergeben sich Abweichungen von Scrum aufgrund traditioneller Unternehmenshierarchien oder aber durch Schwierigkeiten mit den Product Ownern [6, 8]. Das Problem dabei ist, dass es nicht bekannt ist, welches Ausmaß solche Abweichungen haben und ob sie den Entwicklungsprozess nicht vielleicht sogar erheblich schaden können. Wenn die Vorteile der agilen Entwicklung durch Abweichungen von der Theorie ausbleiben oder erheblich gefährdet sind, kann man sich die Frage stellen, warum man dann überhaupt Scrum benutzen sollte. Der Zweck dieser Interviewstudie ist es eine weitere Untersuchung zu dem aktuellen Stand von Scrum in der Praxis zu machen und herauszufinden, welche Aspekte umgesetzt werden, wo abgewichen wird und ob Abweichungen aus einem bestimmten Grund heraus passieren und ob es den Praktikern überhaupt bekannt ist, inwiefern ihr Entwicklungsprozess der Theorie folgt.

¹ Object-Oriented Programming, Systems, Languages, and Applications

² Hybrid Development Approaches in software systems development

1.2 Forschungsfragen und Methodik

Im Rahmen dieser Interviewstudie wurden semi-strukturierte Interviews mit 10 in der Softwareentwicklung tätigen Personen aus 8 deutschen Unternehmen durchgeführt. Aus der eben aufgeführten Zielsetzung lassen sich nun zwei zentrale Forschungsfragen formulieren:

1. Wo und warum weicht man von den theoretischen Aspekten von Scrum in der Softwareentwicklung ab?
2. Welche der theoretischen Aspekte von Scrum werden bei dessen Verwendung in der Softwareentwicklung umgesetzt?

Gerade die erste Forschungsfrage deckt ein zu großes Spektrum ab, so dass vier untergeordnete Forschungsfragen zur Hilfe formuliert werden:

- Bei welchen Rollen weicht man in der praktischen Umsetzung der Scrum Teams von der Theorie ab und wodurch entstehen solche Abweichungen?
- Bei welchen Events weicht man in der praktischen Umsetzung der einzelnen Scrum Events von der Theorie ab und wodurch entstehen die Abweichungen?
- An welchen Stellen weicht man bei der praktischen Umsetzung der einzelnen Scrum Artefakte von der Theorie ab und wodurch entstehen die Abweichungen?
- Wissen die Scrum-Praktiker, welche Aspekte von Scrum in ihrem Entwicklungsprozess verändert oder weggelassen werden und warum man das macht?

Diese Forschungsfragen erleichtern es auf bestimmte Abweichungen aufmerksam zu werden und diese auszuwerten bzw. zu interpretieren. Es ist wichtig auf konkrete Aspekte aufmerksam zu werden, was die Problembereiche sind und welche Aspekte gut laufen, damit man langfristig gesehen dazu in der Lage ist, fundierte Anpassungen und Erweiterungen machen zu können, die sich idealerweise irgendwann empirisch belegen lassen.

1.3 Struktur der Arbeit

Zunächst wird die hinter Scrum stehende Theorie in Kapitel 2 beschrieben. Es werden die einzelnen Teilaspekte von Scrum betrachtet, so dass dieses Theoriewissen später die Unterschiede und Gemeinsamkeiten bei den einzelnen Scrum-Umsetzungen hervorheben kann.

Im anschließenden 3. Kapitel werden Details zu den Interviews beschrieben. Dort wird auf die Akquirierung der Teilnehmer, die Durchführung der Interviews, sowie auf die gewählte Auswertungsmethodik eingegangen.

Die intensive Auswertung der Interviews erfolgt im 4. Kapitel. Hier werden die Unterschiede und Gemeinsamkeiten zur Theorie dargestellt und im darauffolgenden 5. Kapitel werden die Auswertungsergebnisse diskutiert. Dabei werden auf mögliche Einschränkungen der Validität

eingegangen und Antworten auf die Forschungsfragen gegeben. Des Weiteren findet dort die Interpretation der Ergebnisse statt. Daraufhin werden im 6. Kapitel verwandte Arbeiten untersucht, die Forschungen zu ähnlichen Thematiken durchgeführt haben und dessen Ergebnisse in Verbindung mit jenen dieser Interviewstudie gebracht.

Das letzte Kapitel befasst sich zum einen mit Möglichkeiten für tiefergehende Forschung, die sich aus den Ergebnissen der Interviewstudie ableiten lassen, und Aspekten aus einzelnen Interviews, die nicht unmittelbar mit Scrum zu tun haben und somit nicht zentraler Forschungspunkt waren, wie beispielsweise bestimmte Praktiken des Software Engineerings. Zum anderen liefert das Kapitel einen finalen Überblick über die Resultate der Interviewstudie.

2 Scrum

Dieses Kapitel stellt die theoretische Grundlagen von Scrum dar, damit in der Auswertung und Diskussion Abweichungen entsprechend festgestellt und verglichen werden können.

2.1 Allgemeiner Überblick

Scrum ist ein Management Framework in der agilen Softwareentwicklung und umfasst die notwendigen Praktiken, Prozesse und Aufgaben für erfolgreiches Projektmanagement [9]. Im Kern von Scrum stehen die regelmäßig stattfindenden Sprints, in denen man in einem festgelegten Zeitraum zwischen 1 – 4 Wochen an Neuheiten und Verbesserungen der Software, einem sog. Inkrement, arbeitet [9]. Zudem ist Scrum eine iterative Vorgehensweise, so dass ein Sprint von einem weiteren gefolgt wird [9]. Das kontinuierliche Einbeziehen von Kundenfeedback findet große Bedeutung in der Entwicklung, um stets die aktuellsten Anforderungen berücksichtigen zu können [9].

Das Scrum Framework besteht aus verschiedenen Artefakten, Rollen und Events [9], welche im kommenden Teil detaillierter beschrieben werden. Abbildung 1 gibt einen ersten Überblick über die verschiedenen Aspekte von Scrum und wie diese im Zusammenhang stehen. Sofort zu erkennen ist, dass Scrum eine leichtgewichtige Vorgehensweise ist, die nicht besonders viele Dinge braucht [9].

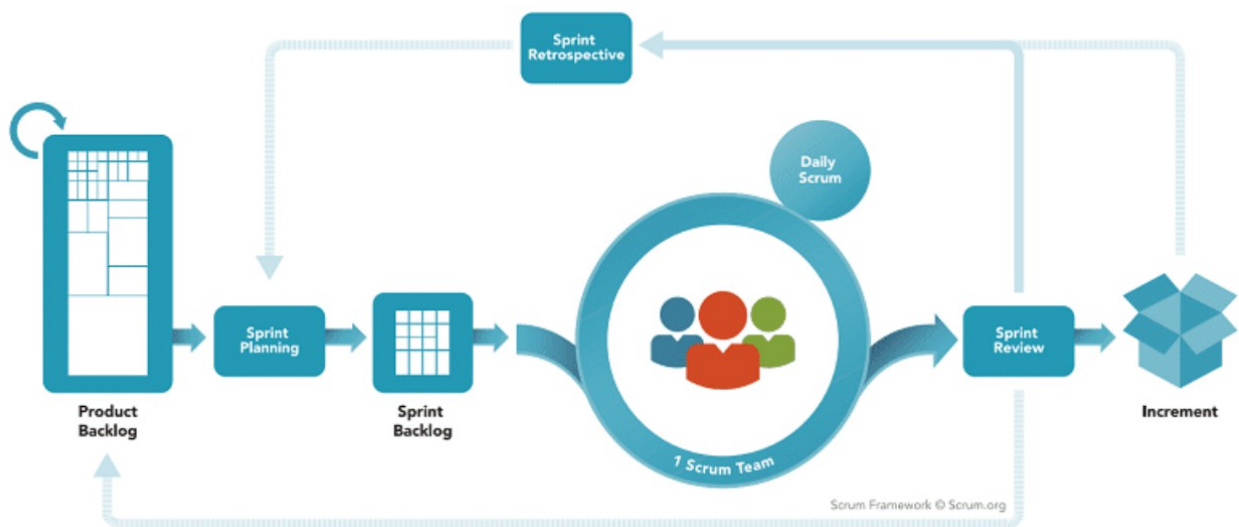


Abbildung 1: Vereinfachte Scrum Übersicht

Quelle: https://scrumorg-website-prod.s3.amazonaws.com/drupal/inline-images/ScrumFramework_2000x1000.png

Ein weiterer Punkt um den wandelnden Kundenanforderungen gerecht zu werden, basiert

Scrum auf Empirismus und vermeidet es deswegen zu weit in die Zukunft zu planen [9]. Durch den Empirismus soll sich das Team innerhalb eines kontinuierlichen Verbesserungsprozesses immer mehr verbessern [9]. Um dies zu festigen, wird es von folgenden drei Säulen umgeben [9]:

- **Transparenz** (transparency): alles ist für alle Beteiligten sichtbar und es herrscht ein gemeinsames Verständnis über die zu erledigende Arbeit,
- **Untersuchung** (inspection): den eigenen Fortschritt beobachten und Handlungsbedarf rechtzeitig erkennen,
- **Anpassung** (adaptation): etwas verändern, falls Probleme erkannt wurden.

Neben diesen drei Säulen gibt es fünf Scrum Werte, die vom Scrum Team gelebt werden, um bestmögliche Ergebnisse zu erzielen und die drei Säulen des Empirismus leichter aufrecht zu erhalten [9]. Die Werte lauten [9]:

- **Einsatz** für das Team und für den Sprint,
- **Mut** das Richtige zu tun und an schweren Problemen zu arbeiten,
- **Fokus** auf den Sprint und das Team,
- **Offenheit** für die anstehende Arbeit und kommenden Herausforderungen,
- **Respekt** untereinander und sich gegenseitig für fähig halten.

Diese Aspekte bieten den ersten Einblick in das Scrum Framework. Im offiziellen Scrum Guide [9] wird die Wichtigkeit, das Scrum Framework nicht willkürlich zu verändern, ausdrücklich betont. Bestimmte Veränderungen könnten zwar funktionieren, wären dann aber laut Schwaber und Sutherland [9], den Erfindern von Scrum, kein Scrum mehr. In der Einleitung wurde bereits thematisiert, dass häufig von der Theorie abgewichen wird [2], weshalb diese Ausarbeitung eine Interviewstudie zu dieser Thematik durchführt.

2.2 Scrum Artefakte

Scrum besteht aus mehreren Artefakten [9]. Ein Artefakt ist dabei eine Liste von zu erledigender oder bereits abgeschlossener Arbeit [10]. Sie werden ständig aktualisiert und verändert [4] und tragen dazu bei, die Transparenz hochzuhalten, so dass alle ein gleiches Verständnis von der zu erledigenden Arbeit haben [9].

Inkrement

Ein Inkrement beinhaltet die neuen Bestandteile der Software, an denen im vergangenen Sprint gearbeitet wurde [9]. Pro Sprint entsteht ein potentiell releasefähiges Inkrement [9], was direkt bei dem Kunden eingesetzt werden kann. Ob das Inkrement tatsächlich releast wird, entscheidet nicht der Product Owner [9].

Das Inkrement muss in Übereinstimmung mit der vom Scrum Team festgelegten definition of done sein [9]. Zu ihr gehört üblicherweise, dass etwas in einem benutzbaren Zustand ist und ausreichend getestet wurde [11] und dient dazu, die Transparenz hochzuhalten [9].

Jedes Inkrement sollte einen echten Mehrwert für den Kunden haben und tatsächlich einsetzbar sein, so dass nicht nur an unsichtbaren Aspekten gearbeitet wird [9].

Product Backlog

Der Product Backlog ist eine priorisierte Liste aller Features, Funktionalitäten und Verbesserungsmöglichkeiten, die die Software betreffen [9]. Er besteht aus Product Backlog Items, die je eine Beschreibung haben, welche ausdrückt worum es geht [9] und wie sie dem Unternehmen einen Mehrwert bieten [11]. Bei der Beschreibung ist es wichtig, dass nicht beschrieben wird, wie etwas umgesetzt wird, sondern nur, was und warum es umgesetzt wird [11]. Höher priorisierte Items sind i.d.R. detaillierter ausgedrückt, damit diese bald eingeplant werden können [11]. Zudem sind die Items priorisiert und haben eine Schätzung über die Umsetzungsdauer [9]. Für wie das Schätzen passieren sollte, gibt es keine konkreten Vorgaben von Scrum [9].

Der Product Backlog unterliegt regelmäßigen Refinements (Verfeinerungen), wo die Product Backlogs Items detaillierter ausgedrückt werden, die Priorisierung angepasst wird und ggf. geschätzt werden, damit sie problemlos für die Planung von Sprints eingesetzt werden können [9].

Sprint Backlog

Der Sprint Backlog besteht aus den ausgewählten Product Backlog Items für den aktuellen bzw. anstehenden Sprint und beinhaltet den zugehörigen Umsetzungsplan [9, 11]. Der Umsetzungsplan entsteht, indem die ausgewählten Product Backlog Items in überschaubare Aufgaben zerkleinert werden, die i.d.R. in 1 Personentag schaffbar sein sollten [9]. Zu Beginn eines Sprints wird der Sprint Backlog immer neu erstellt und nur das Entwicklungsteam ist für dessen Verwaltung zuständig [9].

2.3 Scrum Team

Ein Scrum Team besteht immer aus drei Rollen: dem Product Owner, dem Entwicklungsteam und dem Scrum Master, die sich alle eine wichtige Rolle im Entwicklungsprozess einnehmen [9]. Neben dem Scrum Team sind auch noch Stakeholder am Projekt beteiligt, wie beispielsweise Kunden und Nutzer, die größtenteils für Feedback zuständig sind [9].

Product Owner

Der Product Owner (kurz: PO) ist für die Wertmaximierung der Software zuständig [9]. Um dies zu erreichen verwaltet er den Product Backlog und ist insbesondere für dessen Priorisierung zuständig [9]. Somit legt der Product Owner fest, woran in den Sprints als nächstes gearbeitet wird [9]. Darüber hinaus gibt er dem Entwicklungsteam über die entwickelten Funktionalitäten Feedback, um sicherzustellen, dass die Entwicklung immer mit den aktuellen Anforderungen übereinstimmt [9]. Der Product Owner steht im ständigen Austausch mit den Stakeholdern des Projekts und berücksichtigt dessen Feedback, jedoch trifft er die finale Entscheidung letztendlich immer selbst [9]. Außerdem wird von ihm die Entscheidung getroffen, ob das fertiggestellte Inkrement tatsächlich releast wird.

Um klare Entscheidungen über den Product Backlog treffen zu dürfen, benötigt der Product Owner Befugnisse solche Entscheidungen überhaupt treffen zu dürfen [9].

Außerdem ist es äußerst wichtig, dass der Product Owner nur eine einzelne Person ist, damit es nicht zu Einigungsschwierigkeiten kommt, die den Entwicklungsprozess verlangsamen könnten [9].

Entwicklungsteam

Das Entwicklungsteam ist für das Erstellen des Inkrements zuständig [9]. Es entscheidet selbständig, wie es seine Arbeit erledigt und wer welche Aspekte erledigt [9]. Niemand schreibt dem Entwicklungsteam das vor, da es letztendlich selbst für die Umsetzung des Sprint Backlogs zuständig ist [11]. Idealerweise arbeitet das Entwicklungsteam ungehindert und ohne Eingriffe von außen an dem Sprintziel [9]. Die Entwicklung geschieht stets unter Berücksichtigung der Priorisierung des Product Owners [9].

Ein Entwickler ist dabei eine Person, die für alle Aufgaben des Sprints eingesetzt werden kann im Sinne eines crossfunktionalen Teams [9]. Dazu besteht ein Entwicklungsteam auch aus Entwicklern mit verschiedenen Fähigkeiten, damit sie ohne Outsourcing von Aufgaben selbständig am Inkrement arbeiten können [9]. Außerdem bedeutet das, dass das Entwicklungsteam stets die Priorisierung des Product Backlogs berücksichtigen kann [9].

Ein Entwicklungsteam sollte aus 3-9 Entwicklern bestehen [9], häufig wird aber auch von einer typischen 7 ± 2 Größen gesprochen [10]. Ein zu kleines Team ist nicht in der Lage viel in einem Sprint zu schaffen [11]. Wenn das Team allerdings zu groß wird, verliert man leicht den Überblick aufgrund des steigenden Koordinationsaufwands und die Effektivität der Kommunikation sinkt [9]. Der Product Owner und der Scrum Master zählen nicht mit zum Entwicklungsteam, es sei denn, sie sind direkt an der Erstellung des Inkrements beteiligt [9].

Da sich nicht immer alles mit einem Team von 9 Leuten erledigen lässt, besteht bei größeren Teams die Möglichkeit Scrum in skalierten Varianten zu verwenden, jedoch wird dies in dieser Ausarbeitung nicht weiter behandelt. Für Skalierungsmöglichkeiten sei beispielsweise auf Scrum of Scrums [12], Nexus [13] oder Large Scale Scrum (LeSS) [14] verwiesen.

Scrum Master

Der Scrum Master sorgt für die korrekte Umsetzung von Scrum und bietet die notwendigen Schulungen inner- wie außerhalb des Teams an [9]. Zum großen Teil unterstützt der Scrum Master den Rest des Scrum Teams [11]. Er zeigt Stakeholdern, welche Interaktionen mit dem Scrum Team hilfreich und welche weniger hilfreich sind [9]. Insbesondere gehört hierzu, dass durch Stakeholder nicht willkürlich der Sprintumfang verändert wird [9]. Außerdem dient der Scrum Master dem Entwicklungsteam als „servant leader“ und versucht immer das zu tun, was das Entwicklungsteam gerade braucht, um es in der Selbstorganisation zu stärken [9].

Hinzu kommt, dass er den Product Owner bei dem Management des Product Backlogs unterstützt, so dass der Product Backlog stets sinnvoll priorisiert wird und für das Entwicklungsteam verständlich ausgedrückt ist [11].

Eine weitere wichtige Aufgabe des Scrum Master ist es sicherzustellen, dass alle Events im vorgesehenen Rahmen stattfinden, er sie moderiert und auf die Zeitbeschränkungen achtet [9].

2.4 Scrum Events

Innerhalb eines Sprints gibt es Events, die sich jeden Sprint wiederholen [9]. Ein Event ist dabei meistens eine Art Meeting, das allerdings nicht nur der Kommunikation dient, sondern auch die Möglichkeit bietet, Kundenfeedback einzuholen und innerhalb des Teams auf Probleme aufmerksam zu werden [9]. Alle Events finden innerhalb eines vorgegebenen zeitlichen Rahmens statt, den sog. Timeboxes statt [9].

Sprint

Ein Sprint ist ein zeitlich begrenztes Vorhaben, in dem das Entwicklungsteam an einem neuen Stück Software, einem Inkrement, arbeitet [9]. Dabei ist es von essentieller Bedeutung, dass am Ende jedes Sprints ein potentiell releasefähiges Inkrement entsteht [9]. Bis das Projekt abgeschlossen ist, werden Sprints von weiteren Sprints gefolgt [9]. Der Sprint selbst ist auch ein Event, indem alle anderen Events enthalten sind [9].

Ein Sprint hat eine feste Länge von 1 – 4 Wochen, die nicht verändert werden kann [9]. In zu kurzen Sprints bleibt zu wenig Zeit zum eigentlichen Entwickeln und der Aufwand für Meetings wäre zu groß [9]. Wird ein Sprint jedoch zu lang, verliert man schnell den Überblick und die Wahrscheinlichkeit steigt, dass sich die Anforderungen über diesen Zeitraum verändert haben [9].

Für jeden Sprint einigt sich das Scrum Team auf ein konkretes Sprintziel, was dessen Umfang beschreibt [9]. Während eines Sprints dürfen im Normalfall keine Änderungen am eigentlichen Umfang vorgenommen werden, damit sich das Entwicklungsteam auf die Umsetzung fokussieren und hingeben kann [9]. Im Notfall kann der Sprint durch den Product Owner abgebrochen werden, was aufgrund der kurzen Sprintzyklen allerdings nur in wenigen Ausnahmefällen wirklich erforderlich ist [9].

Daily Scrum

Das Daily Scrum (kurz: Daily) findet jeden Tag innerhalb eines Sprints mit dem Entwicklungsteam und Scrum Master statt [9]. Primär geht es um die Einsatzplanung bis zum nächsten Daily Scrum [11]. Solange das Entwicklungsteam nicht gestört wird, können der Product Owner oder manche Stakeholder sich als stille Teilnehmer selbst einen Überblick verschaffen, [11]. Ein Daily findet der Einfachheit halber jeden Tag am selben Ort zur selben Zeit statt [9].

Scrum schreibt nicht direkt vor, wie ein Daily auszusehen hat, allerdings ist eine typische Vorgehensweise, dass man die folgenden 3 Fragen beantwortet [9, 11]:

- „Was habe ich seit gestern gemacht, um das Sprintziel zu erreichen?“,
- „Was habe ich heute vor für das Sprintziel zu tun?“ und
- „Sehe ich irgendwelche Hindernisse, die mich oder das Team daran hindern, das Sprintziel zu erreichen?“.

Pro Daily beträgt die Timebox 15 Minuten [9]. Meistens setzen sich einzelne Entwickler direkt nach dem Daily zusammen, um sich über die Umsetzungsdetails des anstehenden Tags auszutauschen [9].

Sprint Planning

Das Sprint Planning (kurz: Planning) passiert zu Beginn des Sprints und dient dazu den Sprint konkret zu planen und sich auf ein Sprintziel zu einigen [9]. Beteiligt sind das komplette Scrum Team [9] und Fachexperten können bei Bedarf für spezifische Fachfragen eingeladen werden [11]. In ein Sprint Planning werden folgende Faktoren berücksichtigt: der aktuelle Stand des Product Backlogs, das letzte Inkrement, die verfügbare Kapazität des Entwicklungsteams, sowie Erfahrungswerte über die Leistung der letzten Sprints [9].

Im Wesentlichen passieren im Planning zwei Dinge [9]:

- Zunächst nimmt sich das Entwicklungsteam so viele der höchst-priorisierten Product Backlog Items vom Product Backlog, wie es glaubt schaffen zu können [11]. Es ist wichtig, dass das vom Entwicklungsteam übernommen wird, weil es letztendlich für Umsetzung zuständig ist [9]. Im Anschluss kann dann aus den ausgewählten Product Backlog Items das Sprintziel formuliert werden [11].
- Anschließend erstellt das Entwicklungsteam einen Plan, wie sie die ausgewählten Product Backlog Items in die Tat umzusetzen wollen [9]. Dazu wird also der Sprint Backlog erstellt [9]. Zudem besteht hier noch mal die Möglichkeit Unklarheiten über gewisse Product Backlog Items mit dem Product Owner zu klären, um Missverständnisse frühzeitig zu vermeiden [11].

Die Timebox des Sprint Plannings beträgt 2 Stunden pro Sprintwoche [11].

Sprint Review

Am Ende des Sprints wird in einem Sprint Review (kurz: Review) das neue Inkrement dem Product Owner und den Stakeholdern vorgestellt [11]. Für die Stakeholder und vor allem für den Product Owner besteht hier die Möglichkeit, dem Entwicklungsteam Feedback zu geben [9].

Das Scrum Team kommuniziert den groben Verlauf des Sprints und berichtet dabei welche Dinge geschafft wurden und berichtet grob, wo Probleme aufgetreten sind [9]. Die detaillierte Reflexion des Scrum Teams erfolgt erst in der Sprint Retrospektive [9].

Außerdem besteht im Review immer die Möglichkeit, Aussichten für den weiteren Verlauf des Projekts und der nächsten Sprints zu besprechen [11]. Hierbei ist natürlich wieder wichtig, dass der Product Owner die finale Entscheidung trifft, wie es weitergeht [9].

Als Timebox gilt hier ca. 1 Stunde pro Sprintwoche [11].

Sprint Retrospektive

Am Ende des Sprints gibt es neben dem Review die Sprint Retrospektive (kurz: Retrospektive), die sich primär um die detaillierte Reflexion des letzten Sprints kümmert [9]. Dementsprechend nimmt das gesamte Scrum Team teil [9]. Bei der Retrospektive untersucht man, wie der letzte Sprint insgesamt verlief und bespricht, welche Dinge gut und schlecht liefen [9]. Dabei versucht man die Hintergründe zu verstehen, warum bestimmte Dinge so passiert sind, wie sie passiert sind, um am Ende die Retrospektive mit konkreten Maßnahmen verlassen zu beenden [9]. Diese Maßnahmen beinhalten, wie man die problematischen

Aspekte verbessern und wie man die positiven Aspekte beibehalten möchte [11] und sollten im nächsten Sprint unbedingt sichtbar gemacht werden [9]. Der Scrum Master sorgt dafür, dass eine konstruktive Atmosphäre herrscht, in der sich jeder problemlos einbringen kann oder zumindest die Chance dazu erhält [11].

Die Timebox beträgt ca. 1 Stunde pro Sprintwoche. [11]

3 Interviews

Insgesamt wurden 10 Interviews mit Mitarbeitern aus 8 verschiedenen Unternehmen der Softwareindustrie durchgeführt. Dieses Kapitel beschreibt, wie die Akquirierung der Teilnehmer erfolgte und gibt einzelne Details zu der Interviewdurchführung.

3.1 Teilnehmerakquise

Um ausreichend viele Teilnehmer für die Interviewstudie zu finden, wurde Kontakt mit 115 Unternehmen aufgenommen. Die Kontaktaufnahme erfolgte in zwei Wellen, da zwei Wochen nach den ersten 47 kontaktierten Unternehmen erst fünf Unternehmen zugesagt haben. Daher wurden weitere 68 Unternehmen kontaktiert, um eine größere Datengrundlage zu erhalten. Initial wurde eine Teilnehmerzahl zwischen 8-10 angestrebt, um eine akzeptable Datengrundlage zu haben. Alle Teilnehmer wurden anhand zeitlicher Verfügbarkeit und Bereitschaft an einem Interview teilzunehmen, selektiert. Konkret heißt das also, dass jeder, der zugesagt hat, interviewt wurde und nach einer Zusage keine weitere Filterung erfolgte.

Teilnehmersuche

Gesucht wurde nach Softwareunternehmen im deutschsprachigen Raum, die Scrum verwenden. Die meisten der kontaktierten Unternehmen wurden über Stellenausschreibungen auf Jobportalen, in denen das Stichwort „Scrum“ mindestens 1 mal aufgetaucht ist, gefunden (meistens stepstone und indeed). Für die Teilnehmerwahl wurden die Suchergebnisse nach zeitlicher Aktualität sortiert und Unternehmen für Unternehmen durchgegangen.

Aufgrund der Tatsache, dass eine 2. Runde der Kontaktaufnahme notwendig war, wurden die Kriterien bei der Suche etwas heruntergesetzt und eine Kontaktaufnahme erfolgte in manchen Fällen auch, wenn das Wort "agil" mehrfach in einer Stellenausschreibung aufgetaucht ist, da es nicht unwahrscheinlich ist, dass dann auch irgendwo Scrum zum Einsatz kommt (siehe Kapitel 1). Natürlich hat man dadurch auch mehr Unternehmen kontaktiert, die gar nicht als Interviewteilnehmer in Frage kommen, jedoch relativiert sich diese Zahl, nachdem eine ausreichend hohe Zahl an Unternehmen kontaktiert wurde.

Neben den Stellenausschreibungen wurden manche Unternehmen auch direkt über Google gefunden, weil sie auf ihrem Webauftritt sofort ihre Agilität bzw. Scrum-Verwendung betonen. Beispielsweise wurden mit einer Suchanfrage der Art "Softwareunternehmen Scrum <Ort>" einige potentielle Interviewkandidaten gefunden.

Statistiken zur Kontaktaufnahme und Zusagequote

Die 10 Teilnehmer sind aus 8 Zusagen entstanden, wodurch sich eine Zusagequote von knapp 7% ergibt. Abgesagt wurde von insgesamt 19 Unternehmen (16,5%) und sind in aller Regel aus Kapazitätsgründen passiert. Von den restlichen 96 Unternehmen (76,5%) gab überhaupt es keine Antwort. Es ist gut möglich, dass zumindest ein Teil dieser Unternehmen noch mit Corona-bedingten Umstellungen zu tun, so dass wenig Zeit für andere Sachen übrig bleibt

und nicht geantwortet wird. Natürlich kann trotz der spezifischen Suche nicht vollkommen ausgeschlossen werden, dass manche der kontaktierten Unternehmen tatsächlich keine geeigneten Kandidaten für ein Interview wären und sich deshalb nicht um eine Antworten bemühen wollten.

3.2 Interviewdurchführung

Interviewform

Da sich diese Interviewstudie thematisch mit den Hintergründen für gewisse Scrum-Umsetzungen und Abweichungen befasst, wurden die für die Interviews eine qualitative Form gewählt. Bei einer rein quantitative Form würde man nicht so viel Informationen über die Hintergründe einzelner Abweichungen erfahren.

Deswegen traf die Wahl dann auf semi-strukturierte Interviews, die mithilfe eines Leitfadens (Anhang 1) durchgeführt werden und bei der Durchführung ein gewisses Maß an Flexibilität bieten [19]. Es wurde sich für diese Form entschieden, weil semi-strukturierte Interviews die Möglichkeit bieten, tiefer auf einzelne Aussagen eingehen zu können und aufgrund des Leitfadens ein ausreichendes Maß an Vergleichbarkeit für die Auswertung gewährleistet wird [19]. Im Vergleich zu einem strukturierten Interview, was ausschließlich an einem statischen Leitfaden bzw. Fragebogen orientiert ist und der Interviewumfang genau im Vorfeld festgelegt ist [19], bietet ein semi-strukturiertes Interview eine umfangreichere Möglichkeit tiefergehende Fragen zu stellen.

Ausführung

Aufgrund der SARS CoV-2 Pandemie wurde selbstverständlich kein Interviews in Person durchgeführt. Stattdessen wurden sie per Videotelefonie durchgeführt und jeweils aufgezeichnet. Vorab gab es eine Einwilligungserklärung, die beinhaltet, dass der jeweilige Teilnehmer mit der Interviewteilnahme, dessen Aufzeichnung, der anschließenden Transkription sowie der darauffolgenden Auswertung einverstanden ist. Des Weiteren klärt die Einwilligungserklärung ausdrücklich über die Anonymisierung auf. Vor Beginn jedes Interviews lagen sie in unterschriebener Form vor, so dass jeweils ohne Bedenken mit dem Interview gestartet werden konnte, was auf der Plattform BigBlueButton durchgeführt wurde.

Interviewleitfaden und -ablauf

Der Interviewleitfaden (Anhang 1) wurde mithilfe der einzelnen Scrum-Aspekte erstellt und beinhaltet somit Fragen zum Scrum Team, zu den Events und zu den Artefakten. Außerdem wurde darauf geachtet, dass zwei Fragen vorkommen, die sich speziell um Abweichungen und dessen Hintergründe kümmern. Des Weiteren ist für den Leitfaden wichtig, dass möglichst keine puren Ja-/Nein-Fragen verwendet werden, weil sie meistens keinerlei Tiefe für die einzelnen Hintergründe bestimmter Vorgehensweisen bieten und sind dadurch nur in wenigen Fällen sinnvoll [16]. Außerdem gilt zu beachten, dass auch keine Suggestivfragen verwendet werden, da sie die Echtheit der Antwort beeinflussen können, da der Teilnehmer bereits durch die Fragestellung in eine gewisse Richtung gelenkt wird [16]. Solche Fragen können in aller

Regel in offene und neutrale Fragen umgewandelt werden, die den Teilnehmer dazu die Möglichkeit bieten viel von einem neutralen Standpunkt aus zu einem gewissen Aspekt erzählen.

Normalerweise begann jedes Interview mit einer kurzen Begrüßung, wobei die Rahmenbedingungen des Interviews erwähnt wurden und gefragt wurde, ob der Teilnehmer mit der Aufnahme einverstanden sei. Nachdem diese Frage bejaht wurde, konnte die Aufnahme gestartet werden und die Frage konnte für die Aufzeichnung wiederholt werden. Durch die Anonymität, worauf jeder Teilnehmer noch mal explizit hingewiesen wurde, erhöht sich die Wahrscheinlichkeit, dass die Antworten offen und ehrlich sind, da sie nicht befürchten müssen, dass die Interviewteilnahme negative Konsequenzen für sie hat und nicht aufpassen müssen, was sie wie sagen.

Um einen generellen Überblick zu bekommen, wurde zunächst mit Fragen zu allgemeinen Aspekten des Unternehmens und der jeweiligen Person begonnen. Dies hat zur Folge, dass in einen Redefluss kommt, da diese Fragen keine kognitive Herausforderung darstellen sollten. Dieser Redefluss ist vorteilhaft für den weiteren Verlauf des Interviews, da es dann wahrscheinlicher ist, dass bei den wirklich relevanten Fragen ausführlich geantwortet wird. Gerade in diesem Teil kommen schnell die Vorteile der semi-strukturierten Interviews hervor, da die Reihenfolge nicht exakt festgelegt ist [16]. Der offene Anfang bietet nämlich eine Vielzahl an Antwortmöglichkeiten, so dass manche noch nicht gestellte Fragen bereits beantwortet wurden oder sich die Möglichkeit ergeben hat, tiefergehende Rückfragen zu stellen. Falls dies nicht geschah, wurde einfach mit den nächsten Fragen weitergemacht und auf diese Art und Weise langsam der Fokus auf die Untersuchung der Scrum-Umsetzung gelenkt.

Nachdem alle relevanten Fragen beantwortet waren, wurde am Schluss den Teilnehmern noch die Möglichkeit gegeben Ergänzungen zum Interview anzusprechen, falls bestimmte Aspekte nicht thematisiert wurden oder sie den Eindruck bekommen haben, dass eine Aussage falsch aufgefasst wurde.

Interviewdauer

Über die 10 Interviews hat sich ein Durchschnitt von 47:14 Minuten pro Aufnahme ergeben. Die ersten Interviews neigten etwas unter dem Durchschnitt zu liegen, wohingegen die letzteren Interviews neigten etwas über dem Durchschnitt liegen. Teilweise lassen sich Variationen in der Zeit am Detaillierungsgrad der Antworten begründen. Während sich manche bei ihren Antworten sehr kurz gehalten haben, gaben manche ausführliche Beispiele für bestimmte (Fehl-)Praktiken, obwohl inhaltlich trotzdem der gleiche Rahmen behandelt wurde.

Technische Schwierigkeiten

Fast alle Interviews verliefen ohne größere technische Schwierigkeiten, wodurch 9/10 Interviews problemlos mit der Webcam durchgeführt werden konnten. Die Verbindung war in den meisten Fällen stabil und die Tonqualität der Aufzeichnungen war fast überall einwandfrei. Das Interview mit den technischen Schwierigkeiten fand in verkürzter Form

statt, was zur Folge hatte, dass der Interviewleitfaden nicht vollständig thematisiert werden konnte. Aus diesem Grund wird dieses Interview nur teilweise in der Auswertung berücksichtigt.

3.3 Auswertungsmethodik

Da bereits geklärt wurde, dass es sich um qualitative Interviews handelt, muss nun geklärt werden nach welcher Methodik bei der Auswertung vorgegangen wird. Es stellt sich außerdem die Frage, ob deduktiv, induktiv oder eine Mischung bei der Vorgehensweise gewählt werden sollte. Da die abschließenden Kategorien nicht im voraus bekannt sind, wird eine induktive Methodik gewählt. Dadurch entstehen die Kategorien erst schrittweise bei der Auswertung des Interviewmaterials [17]. Im Laufe der Auswertung werden die Kategorien immer wieder angepasst, je nachdem auf welche Ergebnisse und Aussagen man stößt, so dass die Kategorien immer das Material abdecken [17].

Da es sich um eine induktive Vorgehensweise handelt, wurde dabei die zusammenfassende Inhaltsanalyse nach Mayring [17] gewählt. Bei der Auswertung wurde wie folgt vorgegangen [17]:

- 1 – Transkription
- 2 – Transkripte verkürzen & Sprache vereinheitlichen
- 3 – Kodieren
- 4 – Inhalte zusammenfassen
- 5 – Kategoriebildung

1 – Transkription

Alle Interviews wurden nach der Durchführung transkribiert. Die Art und Weise, wie bestimmte Dinge gesagt wurden, welche Wörter häufig verwendet werden oder wie die Mimik dieser Person zu einem bestimmten Zeitpunkt war, wurde nicht berücksichtigt, weil das Wichtige der Transkripte für die Auswertung dieser Arbeit ist der inhaltliche Teil ist. Diese Details aufzunehmen vermag zwar in manchen psychologischen Studien sinnvoll sein, hier jedoch nicht. Bei dem Transkribieren wurden allerdings keine inhaltlichen Dinge verändert oder auf irgendeine Art und Weise manipuliert, damit die Echtheit der Aussagen garantiert bleibt.

Zunächst wurde das Gesagte Wort für Wort im Transkript festgehalten. Nachdem eine Transkribierung abgeschlossen wurde, wurde sie sprachlich korrigiert, damit die Sprache verständlicher ist. Dies geschah, indem grammatikalisch inkorrekte Sätze so verändert wurden, dass sie hinterher einen grammatikalisch korrekten Satz ergeben. Bei Sätzen, die mehrfach angefangen werden, bis der tatsächliche Satz anfängt, wurde der nichts-aussagende Anfang gestrichen und außerdem wurden Füllwörter vollständig entfernt.

2 – Verkürzung & sprachliche Vereinheitlichung

Als nächstes muss das Auswertungsmaterial deutlich verkürzt und vereinheitlicht werden. Deswegen wurde nach der Transkription jedes Interview verkürzt, d.h., dass beschmückende

Adjektive entfernt wurden. Dabei wurde jeder Satz einzeln durchgegangen, damit nicht aus Versehen Inhalte zusammengefasst werden. Natürlich können bei diesem Vorgehen nicht alle Wörter einfach als irrelevant gesehen werden, da manche Wörter wirklich benötigt, um eine Aussage klar auszudrücken. Außerdem wurde nach Möglichkeit versucht die Sprache zu vereinheitlichen, wodurch die Vergleichbarkeit zwischen den Interviews schon in diesem Schritt gesteigert wird. Bei Aussagen, die mehrfach in fast exaktem Wortlaut aufgetaucht sind, wurden die Dopplungen gestrichen.

3 – Kodieren

Nachdem dieser Schritt abgeschlossen wurde, konnte mit der Kodierung begonnen werden. Bei der Kodierung geht es darum, Aussagen auf eine sehr kompakte Art und Weise zu abstrahieren, um sie später in Kategorien gruppieren zu können. Auch hier wurde wieder Satz für Satz alles vorgegangen, bis jeder Satz mindestens einen Code erhalten hat [18]. Pro Satz können mehrere Codes verwendet werden, da manchmal mehrere Sachen zutreffend sind [18]. Statt Aussagen, die zunächst unwichtig erscheinende Aussagen enthalten, zu löschen, bekamen sie zunächst den Code „Sonstiges“. Wann immer möglich, wurde ein bereits existierender Code verwendet. Nachdem ca. 30-50% eines Interviews kodiert war, wurde eine Übersicht geschaffen und Codes wurden, falls nötig angepasst, damit die Kodierung eines Interviews halbwegs einheitlich erfolgte [18]. Zwischendurch wurden manche Codes aus verschiedenen Interviews bereits verglichen, um schon auf erste Gemeinsamkeiten oder Unterschiede für die Auswertung aufmerksam zu werden.

4 – Inhalte zusammenfassen

Nachdem die Kodierung eines Interviews abgeschlossen war, wurde es inhaltlich zusammengefasst, denn viele der getroffenen Aussagen bringen in der Tat nicht viele Erkenntnisse in Bezug auf Scrum in der Praxis. Außerdem hatte das Auswertungsmaterial meistens immer noch eine utopische Länge, so dass dieses noch überhaupt nicht überschaubar war und zusammengefasst werden musste. Bei der Zusammenfassung wurden entweder bestimmte Aussagen fusioniert oder Beispiele gelöscht, da diese im Normalfall viel zu spezifisch sind.

Die Zusammenfassung wurde zwei mal durchgeführt, aber natürlich wurde immer darauf geachtet, dass keine wesentlichen Informationen verloren gehen. Das Material wurde nun schon deutlich überschaubarer als vorher, allerdings fehlt noch Struktur in dem Material und dementsprechend wurde nun inhaltlich sortiert. Die Sortierung erfolgte zunächst nach den einzelnen Scrum Bereichen, also beispielsweise dem Daily Scrum, dem Sprint Review oder dem Product Owner. Logischerweise befanden sich dann bereits sehr ähnliche Codes in den einzelnen Bereichen, da beispielsweise, die Beschreibung der Vorgehensweise im Daily Scrum einen ähnlichen Code bekommen hat, wie der Zeitrahmen eines Daily Scrums (z.B. „Daily Vorgehen 3 Fragen“ und „Daily überziehen“).

5 – Kategoriebildung

Nachdem das Material ausreichend zusammengefasst und sortiert wurde, konnte mit der

finalen Kategorienbildung begonnen werden. Hierbei geht es primär darum Gemeinsamkeiten und Unterschiede in den Kodes der Interviews zu finden. Es wurden also beispielsweise die Kodes der 10 Interviews zu den Product Ownern verglichen und analysiert und daraus dann die finalen Kategorien gebildet. Dies wurde für jeden einzelnen Scrum-Aspekt durchgeführt, wodurch die Abweichungen langsam zum Vorschein kommen und gruppiert werden konnten.

Insgesamt sind hierdurch 34 Kategorien entstanden, welche im Wesentlichen die Abweichungen beinhalten.

4 Ergebnisse

Dieses Kapitel beinhaltet die Auswertung der Interviews. Zunächst wird ein grober Überblick über allgemeine Daten der Interviewteilnehmer sowie über ihre Unternehmen gegeben. Danach werden die mit der Vorgehensweise des Methodikkapitels gewonnenen Kategorien detailliert beschrieben.

4.1 Teilnehmer- und Unternehmensübersicht

Tätigkeiten der Interviewteilnehmer

Tabelle 1 gibt eine kurze Übersicht der verschiedenen Tätigkeiten der Interviewteilnehmer. Ein Großteil der Teilnehmer ist entweder Scrum Master oder Entwickler. Ein Interview wurde mit einem Product Owner durchgeführt. Die anderen beiden Interviews wurden mit einem Entwicklungsleiter und einem CTO durchgeführt, was also keine direkte Rolle eines Scrum Teams in dem Sinne sind.

Rolle im Unternehmen	Anzahl
Scrum Master	4
Entwicklungsteam	3
Product Owner	1
Andere	2

Tabelle 1: Tätigkeiten der Interviewteilnehmer

Einordnung der Unternehmen

In Tabelle 2 werden die verschiedenen Tätigkeitsbereiche der Unternehmen dargestellt. Die meisten Unternehmen der Interviewstudie sind für das Erstellen von Individualsoftware verschiedener Kunden zuständig und übernehmen sozusagen den kompletten Entwicklungsprozess. 3 der Unternehmen entwickeln hauptsächlich bei ihren Kunden mit, indem sie Kundenteams durch eigene Mitarbeiter unterstützen. Sie schicken Mitglieder für das Entwicklungsteam und Scrum Master bzw. agile Coaches zu den Kunden. Die verbleibenden 3 Unternehmen entwickeln in regulierten Branchen und sind ein mal in der Energie- und Wasserwirtschaft tätig und zwei mal entwickeln sie Software für Berufsgenossenschaften.

Tätigkeitsbereich	Anzahl
Individualsoftware	4
Bei Kunden entwickeln	3
In regulierter Branche entwickeln	3

Tabelle 2: Tätigkeitsbereiche der Unternehmen

Die verschiedenen Unternehmensgrößen der Interviewteilnehmer werden in Tabelle 3 veranschaulicht. Dabei bezieht sich die Unternehmensgröße auf alle Angestellten Personen des Unternehmens, nicht nur auf die Softwareabteilungen. Die meisten Unternehmen sind im mittelgroßen Sektor bei einer Mitarbeiterzahl zwischen 250 und 500 vertreten. Weniger verbreitet waren etwas kleinere Unternehmen mit einer Mitarbeiterzahl zwischen 50 und 249 Mitarbeitern und die restlichen beiden Unternehmen haben eine Unternehmensgröße von unter 50 Personen. Unternehmen mit mehr als 500 Angestellten sind in der Interviewstudie nicht vertreten.

Unternehmensgröße	Anzahl
< 50	2
50 – 249	3
250 – 500	5
> 500	0

Tabelle 3: Unternehmensgrößen der Interviewteilnehmer

Die verschiedenen Vorgehensweisen der Unternehmen für die Entwicklung von Software werden in Tabelle 4 beschrieben. Eine Hälfte der Unternehmen geht ausschließlich mit agilen Vorgehensweisen an die Softwareentwicklung heran und verwenden dabei in keinem Projekt traditionelle Methoden. Pro Projekt wird die Vorgehensweise zu Beginn festgelegt. Diese Unternehmen legen Wert auf die kurzen Iterationszyklen und Flexibilität der agilen Softwareentwicklung, weshalb komplett auf den Einsatz traditioneller Vorgehensweisen verzichtet wird.

Bei der anderen Hälfte der Unternehmen kommen auch beispielsweise das Wasserfallmodell zum Einsatz oder leichte Abwandlungen davon in Form von hybriden Ansätzen, also Kombinationen von agilen und traditionellen Ansätzen. 3 dieser Unternehmen möchten komplett aufhören das Wasserfallmodell zu benutzen, da sie von der Agilität und des regelmäßigen Feedbacks überzeugt sind.

Vorgehensweise bei der Entwicklung	Anzahl
Ausschließlich agil	5
Nicht ausschließlich agil	5

Tabelle 4: Vorgehensweisen der Unternehmen

5.2 Auswertung

Die erstellten Kategorien der qualitativen Inhaltsanalyse nach Mayring [14] werden zur Ergebnisdarstellung wieder in die einzelnen Scrum-Bereiche für einen schnelleren Überblick über die Theorie und dessen Abweichungen eingeordnet. Viele der Kategorien sind nicht disjunkt, so dass manche Unternehmen zu einigen Bereichen, wie beispielsweise dem Product Owner, von mehreren Abweichungen betroffen sein kann. Abbildung 2 gibt eine Übersicht, über die die häufigsten Kategorien der jeweiligen Scrum Aspekte auf einen Blick. Hervorgehoben wurden alle Kategorien, die in mindestens 5 Fällen aufgetreten sind. Auffällig ist, dass die häufigsten Kategorien bei den Product Ownern, Scrum Mastern und den Sprinteingriffen sind.

5.2.1 Scrum Artefakte

Die Auswertungskategorien der Artefakte werden in Tabelle 5 dargestellt.

Artefakt-Kategorie	Häufigkeit
Inkrement nicht jeden Sprint	3
Kein Product Backlog Refinement	3
Product Backlog nicht priorisiert	2
Unzureichende definition of done	3
Anforderungen mit fester Spezifikation	1
Sprint Backlog unzureichend geschützt	6

Tabelle 5: Kategorien der Artefakte



Abbildung 2: Die häufigsten Abweichungskategorien

Quelle: Eigene Darstellung

Inkrement

Bei 3 der interviewten Scrum Teams entsteht nicht in jedem Sprint ein auslieferbares Inkrement. In diesen Fällen werden die Inkrements alle zwei bis vier Sprints in einen potentiell releasefähigen Zustand gebracht, so dass sie bei dem Kunden eingesetzt werden können. In einem Team passiert dies aufgrund der Reviews, die nur jeden zweiten Sprint stattfinden. Ein anderes Team gibt an, dass nur ein mal pro Jahr etwas veröffentlicht wird und ein anderer Grund ist, dass das Testen und die Entwicklung einen Sprint auseinander liegen. In den anderen 7 Fällen wird Wert darauf gelegt, dass ein potentiell releasefähiges Inkrement am Ende des Sprints entsteht.

Product Backlog

Statt eines Product Backlog Refinements werden in 3 Unternehmen die Product Backlog Items erst im Planning detaillierter gemacht und geschätzt. 1 anderes Unternehmen gab an, dass in dem Refinement nur die neuen Anforderungen geschätzt werden. In den anderen Unternehmen kümmert sich das Team im Refinement um die Verfeinerung der Product Backlog Items. Sie werden detaillierter gemacht, zu große Items werden aufgeteilt und anschließend geschätzt. 5 Teams benutzen eine „definition of ready“, die festhält, ab wann ein Item bereit ist für einen Sprint eingeplant zu werden, damit potentielle Verzögerungen verhindert werden.

2 Teilnehmer berichteten davon, dass der Product Backlog nicht priorisiert ist, so dass die Entwickler sich selber um die Priorisierung kümmern müssen bzw. diesbezüglich ständig Rückfragen mit dem Kunden machen müssen. Bei einem Teilnehmer werden die Anforderungen mithilfe einer Spezifikation verwaltet, woraus die zu erledigenden Aufgaben erstellt werden. In allen anderen Fällen werden die Anforderungen hauptsächlich mit User Stories verwaltet. In einem Unternehmen haben die übergeordneten Themen der User Stories aufgrund gesetzlicher Regulierungen strenge Deadlines.

Das Schätzen von Items des Product Backlogs erfolgt in 6 Fällen mittels Story Points, welche ein abstraktes Maß für Schätzungen sind. In 3 Unternehmen kommen klassische Schätzungen mit Personenstunden zum Einsatz, weil man sich entweder selber nicht gut hineinversetzen konnte in das abstrakte Maß oder Story Points Probleme für den Kunden darstellten.

3 Teilnehmer berichten davon, dass keine wirkliche definition of done benutzt wird, in den 6 anderen Fällen gab es eine festgelegte definition of done.

Sprint Backlog

6 der Sprint Backlogs sind nicht sonderlich geschützt, bzw. an ihm können gravierende Änderungen durch Personen außerhalb des Entwicklungsteams, wie beispielsweise dem Product Owner vorgenommen werden (mehr dazu unter dem Aspekt Sprint).

Manche Teilnehmer verwenden das Wort „Backlog“ und meinen mal den Product Backlog und mal den Sprint Backlog.

Für die Verwaltung des Product Backlogs und Sprint Backlogs kommt immer eine Software zum Einsatz. In 7 Fällen ist das Jira und in den anderen Fällen wurde je etwas unterschiedliches verwendet. Physikalische Boards wurden in wenigen Fällen als ergänzendes

Mittel benutzt.

5.2.2 Scrum Team

Tabelle 6 stellt die einzelnen Kategorien der Product Owner, der Scrum Master und der Entwicklungsteams dar.

Team-Kategorie	Anzahl
PO ohne Entscheidungsgewalt	6
PO versteht Rolle nicht	5
Mehrere PO	3
Kein PO	2
PO priorisiert Product Backlog nicht	2
Scrum Master überlastet	5
Scrum Master schützt Sprint schlecht	4
Entwicklungsteam zu klein	2
Entwicklungsteam dauerhaft zu groß	1
Entwicklungsteam temporär zu groß	2
Entwicklungsteam mit externem Mitglied	1
unflexible Rollen im Entwicklungsteam	5
Entwicklungsteam nicht selbstorganisiert	2

Tabelle 6: Kategorien der Scrum Teams

Product Owner

Bei 6 Teilnehmer kommt es bei den Product Ownern zu Befugnisproblemen, so dass sie ihre Entscheidungen nicht immer selbst treffen dürfen und des Öfteren Anweisungen des eigenen Managements weiterleiten müssen. Häufiger ist dies wohl auf mangelnde Agilität in dem Kundenunternehmen zurückzuführen. 5 Teilnehmer erwähnen explizit, dass Probleme mit dem Product Owner zum Großteil daraus entstehen, dass der PO sehr unerfahren ist und die Rolle nicht ausreichend versteht, so dass er sie nicht optimal umsetzen kann.

In 2 Projekten gibt es mehrere PO, bzw. keinen klaren Ansprechpartner und in einem Unternehmen gibt es Probleme bei den Product Ownern durch zu häufige Vertretungen, weshalb die Prioritäten der Entwicklung durcheinander geraten. In 2 anderen Projekten gibt es

teilweise gar keinen PO, woraufhin diese Rolle vom Entwicklungsteam übernommen wird, was dann versucht, im Interessen des Kunden zu handeln. 2 Unternehmen haben Probleme damit, dass der Product Backlog vom Product Owner vernachlässigt wird und ihn nicht priorisiert.

2 Teilnehmer arbeiten mit einem internen Product Owner und 7 Teilnehmer mit externen Product Ownern zusammen, die durch den jeweiligen Kunden gestellt werden. Das andere Unternehmen hat mal mit externen und mal mit internen Product Ownern zu tun. In einem Fall weist der interne Product Owner die Aufgaben des anstehenden Sprints im Planning an das Entwicklungsteam zu. Gemeinsam haben eigentlich alle Product Owner, dass sie dem Entwicklungsteam Feedback für die Entwicklungsergebnisse geben.

Scrum Master

In 5 Unternehmen sind die Scrum Master deutlich überlastet, weil sie für zu viele Teams zuständig sind oder in einer Hybridrolle eingesetzt werden, also zeitgleich Teil des Entwicklungsteams sind. In diesem Kontext geben 3 Teilnehmer an, dass der Scrum Master in maximal zwei Projekten gleichzeitig sein sollte und in 4 Fällen werden die Scrum Master für ein Projekt eingesetzt. Der Sprintrahmen wird von 4 Scrum Masters nicht ausreichend geschützt. Zwar wird versucht Eingriffe teilweise zu verhindern, jedoch gebe man recht zügig nach, wenn der PO bzw. das Unternehmen etwas Druck ausübt. Dahingegen versucht etwa die Hälfte der Scrum Master deutlich stärker einen Eingriff zu verhindern, damit sich das Entwicklungsteam besser auf die Entwicklung fokussieren könne.

In einem Unternehmen verteilt der jeweilige Scrum Master des Projekts die Aufgaben im Daily Scrum an die Entwickler, jedoch unter teilweiser Rücksprache zu dem jeweiligen Entwickler.

Mithilfe des Scrum Masters versuchen viele Unternehmen den Product Owner sowie dessen Unternehmen zu coachen, damit es zu weniger Problemen in seiner Rolle und dem Product Backlog komme.

Entwicklungsteam – Teamgröße

Das exakte Quantifizieren der Teamgröße gestaltet sich als schwierig, da diese Zahl bei vielen Teilnehmern sehr projektabhängig ist. In 2 Projekten besteht das Entwicklungsteam aus zu wenigen Personen, was aufgrund der Unternehmensgröße passiere. Auf der gegenüberliegenden Seite des Spektrums ist ein Entwicklungsteam dauerhaft mit 14 Mitgliedern zu groß. Diese Größe wurde vom Management festgelegt. Bei 2 Projekten gibt es temporär zu große Teams, die erst aufgeteilt werden, sobald sie die Notwendigkeit

darin sehen. Das Aufteilen in diesen Fällen wurde ein mal bei 14 und ein mal bei 12 Mitgliedern vorgenommen.

Bei 3 Teams ist die Größe des Entwicklungsteams variabel während des Projekts, so dass sie sich je nach Bedarf mal vergrößert und mal verringert. 7 Unternehmen haben angegeben, dass sie versuchen sich innerhalb der empfohlenen Scrum Teamgröße zu bewegen, die genaue Zahl jedoch von Projekt zu Projekt unterschiedlich ist.

Entwicklungsteam – Crossfunktionalität

Jedes der Entwicklungsteams besteht aus verschiedenen Rollen, so dass sich ohne fremde Hilfe um das Erstellen des Inkrements gekümmert werden kann. Allerdings bilden sich bei 5 Unternehmen meistens eine Art Sub-Teams im Entwicklungsteam, die außerhalb des Spezialgebiets keine Aufgaben übernehmen. Der Grund für diese festen Rollen ist, dass die Personen z.B. als Tester eingestellt werden, und dann auch nur die Aufgaben dieser Tätigkeit im Entwicklungsprozess übernehmen. 4 der Teams versuchen in diesem Kontext ihre Spezialisten gleichmäßig auszulasten, berichten jedoch häufig von Leerlauf für bestimmte Rollen am Ende oder Anfang des Sprints und von eventuellen Anpassungen an der Priorisierung, so dass statt des höchstpriorisiertesten Items ein möglichst hochpriorisiertes Item für den Sprint ausgewählt wird. Deswegen sind diese 5 Teams nur teilweise crossfunktional.

Die restlichen 5 Entwicklungsteams haben diese Probleme nicht und legen viel Wert darauf, dass die Entwickler auch außerhalb ihres Expertengebietes Aufgaben erledigen können, je nach aktueller Priorisierung. Laut diesen Teilnehmern bringe Crossfunktionalität ein niedrigeres Ausfallrisiko einzelner Entwickler und einen höheren Wissenstransfer innerhalb des Teams.

Entwicklungsteam – Selbstorganisation

In 2 Fällen bekommen die Entwicklungsteams die zu erledigenden Aufgaben zugewiesen. In einen Fall entsteht dies unter Rücksprache mit dem jeweiligen Entwickler, im anderen Fall geschieht dies durch den PO zu Beginn des Sprints. In den übrigen 8 Fällen sucht sich das Entwicklungsteam seine zu erledigende Arbeit selbst aus.

Bei 1 Team gibt es ein externes Mitglied, welches die Qualitätssicherung übernimmt. Diese Entscheidung wurde vom Kunden, bei dem entwickelt wird, getroffen.

5.2.3 Scrum Events

Tabelle 7 stellt die unterschiedlichen Kategorien der Scrum Events dar.

Sprint

In fast allen Fällen ist die Dauer eines Sprints projektabhängig, jedoch wird sich fast immer zwischen Sprintlängen von 2 – 4 Wochen bewegt und die am häufigsten vorkommende Sprintlänge sind 2 Wochen. Bei einem Unternehmen gibt es manchmal Sprints, die zeitlich nicht begrenzt sind, da der Kunde nur an bestimmten Funktionalitäten interessiert wäre und mit Teilen der Gesamtfunktionalität nichts anfangen könne, so dass ein zeitlich begrenzter Sprint durch ein Funktionalitäten-fokussierter „Sprint“ ersetzt wird.

Bei 3 Unternehmen lässt sich der Sprintumfang nachträglich erweitern. Von diesen 3 Unternehmen ist das Sprintende bei 2 Unternehmen verschiebbar, damit die vereinbarten Funktionalitäten trotz des Sprinteingriffs ausgeliefert werden können. Frühzeitig wird der Sprint manchmal in einem Unternehmen beendet, wenn das Inkrement vorzeitig fertiggestellt werden konnte. Als Alternative dazu holen sich 6 Teilnehmer ein Item vom Product Backlog

unter Rücksprache mit dem PO nach.

Event-Kategorie	Anzahl
Sprintumfang einfach so veränderbar	6
Sprint verlängern	2
Sprint zeitlich nicht begrenzt	1
Inkrement vorzeitig ausliefern	1
Review ohne Stakeholder	6
Retrospektive ohne konkrete Verbesserungsmaßnahmen	5
Retrospektive ohne PO	4
Retrospektive unregelmäßig	1
Kein Daily	1
Daily nicht täglich	1
Daily regelmäßig überziehen	3
teamfremde Mitglieder im Daily aktiv	2
Planning nicht regelmäßig	1
PO legt Sprintumfang fest	1

Tabelle 7: Kategorien der Scrum Events

Tabelle 8 gibt einen Überblick über die verschiedenen Ursachen der Änderungen am Sprintumfang.

Eingriffsgrund	Anzahl
Fehler in laufendem System	9
PO reicht Anweisungen weiter	3
PO hat etwas vergessen	3
Übergeordnete Deadlines gefährdet	2
Kunden nicht widersprechen	2

Tabelle 8: Gründe von Sprintumfangsänderungen

9 Unternehmen gaben explizit an, dass der Sprintumfang veränderbar ist, falls ein kritischer Fehler in einem laufenden Kundensystem auftritt, der sofort behoben werden müsse. Dafür wird dann in der Regel etwas niedrigpriorisiertes - mit Rücksprache zum PO - aus dem Sprint entfernt. In diesem Kontext betonen 2 Unternehmen auch, dass es sehr wichtig ist, sich an die Prioritäten des PO zu halten, damit bei unerwarteten Problemen wenig negative Auswirkungen haben, da das Wichtigste bereits entwickelt wurde. Ein anderes Unternehmen betont, dass Sprinteingriffe durch ständige Kundenkommunikation, verfeinerte Product Backlog Items und ein gut durchgeführtes Planning weitestgehend vermieden werden können. Falls kritische Fehler zu oft auftauchen, lässt sich hierfür 1 Team extra einen Puffer, damit nicht jedes mal der Sprintumfang verändert werden müsse.

3 Unternehmen erlauben es nachträglich Product Backlog Items dem Sprint Backlog hinzuzufügen. Beispielsweise passiert das, wenn der PO etwas beim Planning vergessen habe oder intern in seinem Unternehmen sich Prioritäten plötzlich geändert haben, die er dann weiterleiten müsse. Eingriffe und Adjustierungen von internen PO sind bei 2 Unternehmen erlaubt, wenn wichtige Deadlines gefährdet sind, die wegen gesetzlicher Regulierungen zwingend einzuhalten sind.

2 Unternehmen erlauben es, dass Änderungen an bereits angefangenen Sprint Backlog Items vorgenommen werden können ohne große Kompensation dafür, da der Entwicklungsprozess den Kunden zufrieden machen soll.

Kleinere Änderungen am Umfang werden von 7 Unternehmen erlaubt, solange es den Sprintumfang nicht wesentlich verändern. Allerdings müsse der PO das dann kompensieren, indem er eine andere Sache aus dem Sprint herausnimmt, damit das Entwicklungsteam auch das Inkrement fertigstellen kann. 2 Unternehmen brechen den Sprint ab, falls der PO gravierende Änderungen am Sprintumfang vornehmen möchte, die beispielsweise eine bereits angefangene User Story betreffen würden.

In 2 Fällen wird der Sprintumfang nachträglich reduziert, indem ein Item aus dem Sprint entfernt wird, da die initiale Schätzung des Entwicklungsteams um einiges daneben lag oder andere unvorhergesehene Probleme aufgetaucht sind.

Lediglich 1 Teilnehmer berichtet davon, dass fast nie Veränderungen an dem Sprintumfang vorgenommen werden.

6 mal wurde angegeben, dass häufige Eingriffe negative Auswirkungen auf die Softwarequalität und das Entwicklungsteam haben und beispielsweise das Entwicklungsteam störe sowie die Planbarkeit eines Sprints erheblich einschränke.

Review

Bei 6 der Teilnehmer nehmen Stakeholder normalerweise nicht oder nur in Ausnahmefällen am Review teil. Dies liege zum Großteil an der zeitlichen Verfügbarkeit der Stakeholder und der Tatsache, dass oft mit externen Kunden gearbeitet wird, so dass es sich nicht immer leicht gestalten, regelmäßigen Kontakt mit den Stakeholdern zu haben. In einem Team findet die Retrospektive und das Review gleichzeitig statt, so dass die Stakeholder auch teilnehmen. In einem Unternehmen findet aufgrund von teilweiser unklarer Sprintenden das Review nicht regelmäßig statt, so dass sich eher auf eine finale Auslieferung der Software fokussiert wird.

Das neue Inkrement wird in 3 Fällen anhand einer PowerPoint Präsentation vorgestellt und die anderen 7 legen Wert auf das tatsächliche Ausprobieren der Software, durch die Stakeholder bzw. den Product Owner, um das Feedback zu maximieren.

Retrospektive

Bei 4 Teilnehmern findet die Retrospektive ohne PO statt. Als Grund wurde hierfür wieder die zeitliche Verfügbarkeit des Kunden bzw. PO genannt. In einem Unternehmen findet die Retrospektive meistens nur dann statt, wenn ein Projekt überdurchschnittlich schlecht laufe. In den anderen 4 Fällen findet die Retrospektive normal mit dem PO statt.

Aus 5 Interviews geht hervor, dass in der Retrospektive keine konkrete Verbesserungsmaßnahmen erstellt werden. Außerdem gibt es bei manchen Teilnehmern ein paar sprachliche Verwechslungen, und zwar, dass die Retrospektive und das Review beide als Review bezeichnet werden.

Daily

Ein Unternehmen führt aufgrund der kleinen Teamgröße keine Dailys durch. Als Begründung wurde genannt, dass bereits ausreichend Kommunikation innerhalb des Teams passiere, so dass dieses Meeting nicht benötigt werde. Bei einem Unternehmen ist das Daily nicht täglich, da aufgrund der Teamgröße nicht so viele Dinge passieren würden, als das man ein tägliches Meeting bräuchte.

In 6 Unternehmen kommen im Daily die typischen drei Fragen zum Einsatz. Statt der drei Fragen kommen in 2 Teams eine Sprint Backlog-zentrierte Alternative zum Einsatz, wo kurz der Stand jedes Sprint Backlog Items besprochen wird. Dort werde diese Variante vorgezogen, da es manche Entwickler gibt, die nur in Teilzeit im Entwicklungsteam sind, und sich nicht immer selbstständig einen Überblick über den aktuellen Stand verschaffen.

Regelmäßige Probleme bei der Timebox des Dailys wird von 3 Teilnehmer berichtet, weshalb das Daily des Öfteren etwas überzogen wird, weil Diskussionen ausgiebig gehalten werden. In einem Team passiere dies aufgrund des Remote-Arbeitsumfelds und man deswegen lieber ein mal am Tag intensiv kommunizieren möchte. 4 der Teilnehmer haben explizit angesprochen, dass sie versuchen Diskussionsbedarf im Daily auszulagern, indem sie beispielsweise ein kurzfristiges Meeting nur mit den betroffenen Personen ansetzen, was allerdings nicht immer klappt und das Daily dann manchmal doch überzogen wird.

In 2 Teams nehmen externe Mitglieder des Scrum Teams am Daily teil. Bei einem Teilnehmer ist das der Kundensupport, der immer anwesend ist und Probleme bei laufenden Kundensystem direkt kommunizieren kann. In dem anderen Fall nimmt gelegentlich das Mitglied der externen Qualitätssicherung teil, um sich einen aktuellen Überblick zu verschaffen. Die Product Owner von 3 Teilnehmern nehmen als stille Teilnehmer für Rückfragen teil, was von den Entwicklungsteams als äußerst hilfreich empfunden wird.

Planning

Das Planning wird in einem Unternehmen nicht regelmäßig durchgeführt, da mit einer Spezifikation gearbeitet wird. In diesem Fall hat nur der Projektleiter, der allerdings Teil des

Scrum Teams ist, regelmäßig Kontakt mit dem Kunden bzw. PO. Bei einem weiteren Teilnehmer wird der Sprintumfang durch den PO festgelegt. In den meisten Fällen legt das Entwicklungsteam den Sprintumfang unter Rücksprache mit dem Product Owner selber fest, indem das Entwicklungsteam eigenständig so viele Items vom Product Backlog nimmt, wie es für realistisch hält, dass sie schaffen können.

Die Hälfte der Teams erstellt im Planning einen Umsetzungsplan und 4 der Teams fangen direkt mit der Umsetzung des Sprint Backlogs an, da die ausgewählten Product Backlog Items bereits verfeinert genug wären.

3 der Unternehmen formulieren kein konkretes Sprintziel, sondern orientieren sich lediglich an den ausgewählten Backlog Items.

5 Diskussion

In diesem Kapitel werden Einschränkungen der Validität aufgeführt, Antworten auf die Forschungsfragen gegeben und die Auswertungsergebnisse interpretiert.

5.1 Einschränkungen der Validität

Nach Wohlin et al. [19] beschreibt die Validität, wie zuverlässig die Ergebnisse sind und zu welchem Grad die Ergebnisse nicht durch Voreinstellungen oder subjektive Wahrnehmungen der Forscher erzielt werden [19]. In der qualitativen Forschung gibt es vier mögliche Einschränkungen der Validität [19]: Konstruktvalidität, Reliabilität sowie interne und externe Validität.

Die **Konstruktvalidität** beschreibt, inwiefern der untersuchte Aspekt (Fragen zu Scrum), wirklich dem entspricht, was die Forschung bezwecken sollte. Gefährdet wird sie, wenn die Teilnehmer Fragen des Leitfadens nicht versteht oder missinterpretiert. Missinterpretationen können natürlich nie ganz ausgeschlossen werden, jedoch wurde versucht die offiziellen Scrum Begriffe zu verwenden, um dem entgegenzuwirken. Falls die Teilnehmer eine Frage nicht verstehen, konnten sie jederzeit ohne negative Konsequenzen zur Klärung nachfragen, woraufhin die Frage in etwas paraphrasierter Form erneut gestellt werden kann.

Ein Grundverständnis von Scrum ist bei den Teilnehmern vorauszusetzen, da sie sich zum einen bereit erklärt haben, an einer Interviewstudie „zum Verständnis und zur Nutzung von Scrum“ teilzunehmen und zum anderen eine Teilnahme impliziert, dass Scrum in ihrem Arbeitsalltag verwendet wird, so dass Scrum eigentlich bekannt sein müsste. Sprachliche Probleme wurden versucht zu vermeiden, indem sich möglichst an gewählten Begriffen der Teilnehmer orientiert wurde.

Die suggestive Formulierung des Titels könnte die Teilnehmer negativ beeinflussen oder dazu führen, dass sich Teilnehmer vorher noch mal mit der Theorie auseinandersetzen, um einen besseren Eindruck zu hinterlassen. Auch, wenn dies keine Vorteile für die Teilnehmer hätte, könnte es sich trotzdem besser anfühlen. Die Interviews wurden vorher nicht getestet, aber an Scrum Aspekten orientiert, so dass sie sinnvoll konstruiert wurden und alle Bereiche abgedeckt sind.

Die **Reliabilität** beschreibt, wie reproduzierbar die Ergebnisse der Studie wären, wenn sie erneut durchgeführt würden [19]. Falls eine Studie nicht reliabel ist, besteht die Gefahr, dass die in ihr entstandenen Konklusionen falsch oder ungenau sind [19]. Vor allem bei der Auswertung, insbesondere beim Kodieren und der Kategorienbildung ist diese gefährdet [19]. Idealerweise kommen zwei verschiedene Forscher zu den sehr ähnlichen oder gleichen Codes und Kategorien. Diesem konnte nicht entgegengewirkt werden. Des Weiteren wird die Reliabilität durch einen Interviewleitfaden gefördert, da sich so die Ergebnisse leichter reproduzieren lassen. Außerdem wurde für die Auswertung ein etabliertes Verfahren in der qualitativen Inhaltsanalyse nach Mayring [14] angewendet, so dass Fehler, die durch inkorrekte Auswertung entstehen könnten, verringert werden.

Ein weiterer Aspekt ist die *externe Validität*, die festhält, wie verallgemeinerbar und auf andere Kontexte übertragbar die Ergebnisse sind [19]. Es gibt viele nicht abgedeckte Branchen, Unternehmensgrößen und es wurden nur deutsche Unternehmen interviewt. Daher lässt sich das nicht wirklich übertragen auf andere Kontexte. Außerdem ist die Stichprobe keinesfalls repräsentativ. Es wurde nämlich jeder Teilnehmer interviewt, der zugesagt hat. Wenn man sich überlegt, wer alles an einem Interview teilnehmen möchte, was die Teilnehmer zum Verständnis und zur Nutzung von Scrum ausfragt, ist es durchaus möglich, dass sich nur Leute bereit erklären, die sich damit etwas auskennen. Dadurch entsteht insgesamt bei der Auswertung kein repräsentatives Bild.

Die Stichprobengröße kann nicht ausgesucht werden, aber es wurde versucht eine nicht zu geringe Anzahl an Teilnehmern zu bekommen, indem genug Unternehmen kontaktiert wurden. Somit kann nicht zwangsläufig sichergestellt werden, dass man eine Sättigung in den qualitativen Daten erreicht.

Jedoch lässt sich auch einiges an Gemeinsamkeiten zu ähnlichen Studien und Papern finden, so dass dies den Wahrheitsgehalt mancher Aussagen erhöht (mehr dazu in Kapitel 7) und vielleicht langfristig etwas mehr auf andere Kontexte übertragen lässt.

Zuletzt gibt es die *interne Validität*, welche beschreibt, wie sicher man sicher über Ursache-Wirkungsbeziehungen in den Ergebnissen sein kann und ob man alternative Erklärungen für die Ergebnisse ausschließen kann [19]. Durch die Teilnahme entstehen keine Vorteile für die Teilnehmer und die Auswertung ist anonymisiert, so dass es keinen Reiz gibt, Aussagen zu verzerren. Fast alle Interviews wurden bei gleichen Bedingungen durchgeführt, indem sie 9 mal per Webcam stattfinden konnten und nur ein mal ohne Webcam stattfinden mussten. Da dieses Interview mit technischen Schwierigkeiten behaftet war und nur teilweise in die Auswertung eingeflossen ist, ist dies keine große Validitätseinschränkung.

5.2 Beantwortung der Forschungsfragen

Im folgenden werden die beiden Forschungsfragen aus der Einleitung beantwortet. Eine der beiden zentralen Forschungsfragen lautete:

Wo und warum weicht man von den theoretischen Aspekten von Scrum in der Softwareentwicklung ab?

Diese Forschungsfrage lässt sich mithilfe der vier untergeordneten Forschungsfragen beantworten:

1 - Bei welchen Rollen weicht man in der praktischen Umsetzung der Scrum Teams von der Theorie ab und wodurch entstehen solche Abweichungen?

Ein wesentlicher Punkt für Abweichungen im Scrum Team sind die *Product Owner*. Eins der Hauptprobleme ist, dass ihnen oft die nötige Entscheidungsgewalt fehlt, wichtige Entscheidungen alleine treffen zu dürfen und stattdessen viel Rücksprache mit dem eigenen

Management geführt werden muss und er quasi nur als Mediator zwischen Entwicklungsteam und dem eigenen Management fungiert. Er ist sozusagen nicht der „echte“ PO und legt die Priorität nicht selber fest. Daneben gibt es den Punkt, dass die Rolle oft missverstanden wird, weil die zugehörige Person unerfahren oder im schlimmsten Fall inkompetent ist. Dies hat zur Folge, dass einzelne Aspekte der Rolle nicht optimal umgesetzt werden, wie beispielsweise die generelle Verwaltung der Anforderungen im Product Backlog, die Teilnahme an den jeweiligen Events, nicht willkürlich in den Sprint einzugreifen oder die Prioritäten wahllos zu verändern. Neben diesen Abweichungen gibt es in manchen Projekten Probleme mit der tatsächlichen Besetzung der PO Rolle, so dass es entweder mehrere oder gar keinen PO gibt.

Ein möglicher Grund für diese Probleme sind alte Unternehmensstrukturen in den Unternehmen der Kunden. Wenn die Organisation als Ganze noch nicht so viel Erfahrung in der agilen Softwareentwicklung sammeln konnte, ist es durchaus möglich, dass sich das auch auf einzelne Positionen im Unternehmen bemerkbar macht. Durch mangelnde Erfahrung wird im Unternehmen vielleicht der komplette Scrum-Prozess noch nicht genug verstanden, so dass man die PO Rolle nicht optimal besetzen kann oder sie überhaupt erst wertschätzt. Aufgrund der Tatsache, dass oft mit externen Kunden zusammengearbeitet wird, die dann auch den PO stellen für das jeweilige Projekt, wodurch sich keine wirklichen Synergien im Scrum Team bilden können, weil man mit immer unterschiedlichen Product Ownern zu tun hat.. Dies geschieht auch zum Großteil daraus, dass die Rolle unzureichend verstanden wird und der PO Anweisungen weiterleiten muss. Insgesamt entstehen die meisten Abweichungen bei den Product Owner aus externen Faktoren, die die Unternehmen nicht unmittelbar beeinflussen können und sich manchmal etwas anpassen müssen.

Abweichungen entstehen bei den *Scrum Mastern* nicht direkt durch einzelne Details der Rolle, sondern in den meisten Fällen aufgrund der Tatsache, dass sie sich um zu viele Aspekte gleichzeitig kümmern müssen und diese dann nicht ordentlich ausführen können. Dies wird begünstigt, wenn sie in zu vielen Projekten gleichzeitig eingesetzt sind oder eine Hybridrolle haben. Bei Hybridrollen müssen sie verschiedene Interessen vertreten und zwischen Entwickler und Scrum Master wechseln. Auch hier könnte der ständige Wechsel der Teams und Arbeitsumgebungen Probleme bereiten, so dass sich wieder keine Synergien bilden können und sich der Scrum Master jedes mal um eine völlig neue Organisation, Product Owner und Teile des Entwicklungsteams kümmern muss. Dadurch entstehen dann häufig Probleme mit dem Schutz der Sprintrahmen, wenn sie in zu vielen Teams eingesetzt sind und nicht das Kundenunternehmen nicht ausreichend coachen können.

Bei dem *Entwicklungsteam* können Abweichungen bei der Größe entstehen. Zu kleine Teams entstehen aufgrund kleiner Unternehmensgrößen und zu große können Teams durch Vorgaben vom Management entstehen. Ein weiterer Faktor für Probleme bei der Teamgröße ist das Zusammenarbeiten mit externen Kunden, da man dort in bereits bestehende Strukturen eingefügt wird und somit wirklich Kontrolle darüber hat, was einem begebenet und sich immer wieder neu anpassen muss.

Probleme bei der Crossfunktionalität können durch alte Unternehmensstrukturen mit den klar getrennten Rollen entstehen, so dass sich kleine Sub-Teams in den Entwicklungsteams bilden, die sich nie außerhalb ihres Spezialgebietes bewegen, weil sie es vorher nicht mussten.

2 - Bei welchen Events weicht man in der praktischen Umsetzung der einzelnen Scrum Events von der Theorie ab und wodurch entstehen solche Abweichungen?

Die größten Abweichungen in einem **Sprint** entstehen durch Änderungen an dessen Umfang, so dass der Sprint Backlog grundlegend verändert werden kann. Auch wenn man häufig flexibel auf den Kunden eingehen möchte, wie beispielsweise die Reaktion auf einen Fehler im Kundensystem, gibt es eine Vielzahl an nicht förderlichen Eingriffen. Diese Sprinteingriffe lassen sich möglicherweise auf ein mangelndes Verständnis von Scrum und Agilitätsprobleme der Kunden zurückführen, da dies direkt in Verbindung mit den o.g. PO Problemen steht. Weitere Abweichungen im Sprint sind, dass die Dauer nicht fest ist, also das Ende nach vorne oder hinten verschiebbar ist. Eine weitere Abweichung lässt sich in den zeitlich unbegrenzten Sprints finden. Dies passiert, da man die Kundenzufriedenheit maximieren möchte und sich deshalb dem Kunden möglichst entgegenkommt.

Abweichungen im **Daily** sind eher die Ausnahme, da es nur in zwei Fällen entweder nicht durchgeführt wird oder nicht täglich ist. Beide Fälle passieren aufgrund der kleinen Unternehmensgröße. Andere Abweichungen sind, dass die Timebox nicht eingehalten wird. Möglicherweise entsteht diese Abweichung dadurch, dass der Scrum Master die Timebox nicht strenger forciert und zu viele Diskussionen zulässt. Die letzte Abweichung im Daily entsteht dadurch, dass Mitglieder außerhalb des Scrum Teams im Daily aktiv sind. Zurückzuführen ist diese Abweichung einerseits auf das Arbeiten mit externen Kunden, wo bestimmte Teamstrukturen halbwegs vorgegeben sind und im anderen Fall will man immer auf dem neusten Stand von Kundenprobleme sein.

Auch im **Planning** passieren verhältnismäßig wenige Abweichungen, und zwar findet nur ein mal das Planning nicht regelmäßig statt und die andere Abweichung besteht darin, dass der Sprintumfang vom PO festgelegt wird. In beiden Fällen lässt sich dies auf alte Unternehmensstrukturen zurückführen, da ein mal mit einer Spezifikation gearbeitet wird und ein mal der PO dem Scrum Team nicht genug Selbstorganisation zutraut.

Die Hauptabweichung im **Review** ist, dass es ohne Stakeholder durchgeführt wird. Diese Abweichung lässt sich zu großen Teilen auf das Arbeiten mit externen Kunden zurückführen, so dass es nicht immer möglich ist einen Stakeholder aus seinem aktuellen Arbeitsumfeld herauszuholen, nur um Feedback zu geben. Vielleicht wird es ihnen von Kundenseite aus gar nicht ermöglicht sich als Stakeholder im Entwicklungsprozess einzubringen, da der Kunde diesen Teil vielleicht noch nicht genug wertschätzt.

In der **Retrospektive** entstehen häufiger keine konkreten Verbesserungsmaßnahmen. Bei manchen Unternehmen hat es den Anschein, als ob es nur durchgeführt wird, weil man es machen muss und den tieferen Hintergrund, den kontinuierlichen Verbesserungsprozess vernachlässigt. Dies ist möglicherweise auf mangelndes Verständnis und Neuheit in Scrum zurückzuführen. Der externe PO ist oft nicht in der Retrospektive anwesend und wird auch nicht wirklich als Teil des Scrum Teams behandelt. Ähnlich wie bei den Reviews scheint sich der Kunde hier weitestgehend aus dem Entwicklungsprozess zurückziehen zu wollen, was auf alte Unternehmensstrukturen zurückführbar ist.

3 - An welchen Stellen weicht man bei der praktischen Umsetzung der einzelnen Scrum Artefakte von der Theorie ab und wodurch entstehen die Abweichungen?

Bei dem **Product Backlog** kommt es zu Abweichungen, wenn dieser nicht ordentlich geführt wird oder unpriorisiert ist. In den meisten Fällen ist dies auf die problematische PO Rolle zurückzuführen. Mangelnde Erfahrung und Kompetenz des POs kann dazu führen, dass die Product Backlog Items nicht klar ausgedrückt sind und nicht wertmaximierend verwendet werden.

Abweichungen im **Product Backlog Refinement** entstehen aus ähnlichen, wenn nicht denselben Gründen. In manchen Unternehmen finden Refinements gar nicht statt, was sich wieder möglicherweise durch das Problem beschreiben lässt, dass sich einige Kunden, also auch Product Owner vom Entwicklungsprozess zurückziehen wollen oder der Glaube, dass das Refinement überflüssig wäre und die komplette Verfeinerung immer im Planning stattfinden könne.

Abweichungen im **Sprint Backlog** passieren, da das Entwicklungsteam nicht vollständig für das Verwalten des Sprint Backlogs zuständig ist und daraus häufige Sprinteingriffe resultieren können. Das Scrum Team besteht also nicht unbedingt aus gleichberechtigten Mitgliedern, sondern der PO scheint etwas mehr Autorität zu haben. Eventuell lässt sich dies wieder auf Unerfahrenheit des PO zurückführen oder aber dem Bedürfnis eines Unternehmens einen externen Kunden zufriedenstellen zu wollen. Viele Unternehmen trauen sich nicht ihren Kunden zu widersprechen, weil sie ein Stück weit von ihnen abhängig sind, weil sie ohne Kunden nichts entwickeln können. Eine weitere Ursache dieser Probleme könnte durch mangelhaften Schutz des Sprintrahmens durch den Scrum Master passieren.

Bei den **Inkrementen** gibt es das Problem, dass die Software nicht jeden Sprint in einen für den Kunden einsetzbaren Zustand gebracht wird. Das Erstellen eines Inkrements wird durch traditionelle Denkweise und Herangehensweise behindert, wenn das Testen und die Entwicklung einen Sprint versetzt passieren. Ein weiterer Grund dafür ist, dass die Inkremente eine enorme Zeit brauchen, bevor sie vom Kunden in Produktion genommen werden, so dass ein einzelnes Inkrement an Wert verliert, da nur so selten releast wird. Aus selbigem Grund sind auch in einigen Unternehmen die mangelhaften oder nicht vorhandenen definitions of done zu erklären.

4 - Wissen die Scrum-Praktiker, welche Aspekte von Scrum in ihrem Entwicklungsprozess verändert oder weggelassen werden und warum man das macht?

Die meisten Teilnehmer haben ungefähr eine Vorstellung davon, wo sie im Entwicklungsprozess von der Theorie abweichen. Bei vielen Abweichungen konnten die Teilnehmer auch erklären, wieso bestimmte Abweichungen passieren oder entstehen. Häufige Abweichungen, wie es bei den Product Ownern oder Sprinteingriffen der Fall ist, sind meistens bekannt. Auch, dass die Stakeholder eigentlich im Review anwesend sein sollten, es aber nicht sind, ist bekannt. Für sehr viele Teilnehmer ist ebenfalls der zeitliche Rahmen eines Daily bekannt und dass diese trotzdem manchmal überzogen werden.

Auf der anderen Seite sind manche Abweichungen nicht unbedingt bekannt. Probleme bei den Scrum Mastern scheinen weitestgehend keine Beachtung zu finden, da sich meistens nur auf den Kunden als zentralen Problempunkt fokussiert wird. Weitere Abweichungen, die viele nicht bemerken, passieren bei den Retrospektiven, wenn diese ohne PO stattfinden oder keine konkrete Verbesserungsmaßnahmen erstellt werden. Feinere Abweichungen, wie eine transparente und einheitliche definition of done, sind einigen Teilnehmern ebenfalls nicht bekannt. Des Weiteren wird mangelnde Crossfunktionalität oder Selbstorganisation von manchen Entwicklungsteams nicht als problematisch betrachtet, so dass dies möglicherweise eine unbekannte Abweichung darstellt. Manchmal passieren auch kleinere sprachliche Mehrdeutigkeiten, wie beispielsweise der Verwechslung von Review und Retrospektive. Oftmals ist bei der Teamgröße nicht immer klar, ob das gesamte Scrum Team oder nur Entwicklungsteam gemeint ist. Außerdem wird häufig nicht zwischen dem Sprint und Product Backlog weiter unterschieden, sondern beides wird einfach als „Backlog“ bezeichnet.

Anhand der Abweichungen lässt sich nun die andere Forschungsfrage beantworten:

- Welche der theoretischen Aspekte von Scrum werden bei dessen Verwendung in der Softwareentwicklung umgesetzt?

Die Frage lässt sich pauschal nicht beantworten, allerdings gibt es einige Gemeinsamkeiten bei vielen Aspekten, die auf meisten zutreffen. Pro Scrum-Bereich gibt es eigentlich immer mehrere Unternehmen, die relativ nah, wenn nicht sogar in Übereinstimmung zur Theorie agieren. Insgesamt wird großer Wert auf das regelmäßige Einbeziehen von Kundenfeedback gelegt, was durch die kurzen Iterationszyklen ermöglicht wird. Die Sprintdauer befindet sich in fast allen Fällen innerhalb des empfohlenen Rahmens von 1 – 4 Wochen. Bei vielen Teams wird in jedem Sprint ein Inkrement erstellt. Anforderungen werden in fast allen Fällen dynamisch im Product Backlog verwaltet, so dass es Freiheiten und Flexibilität bei der Umsetzung gibt. Außerdem wird versucht, halbwegs crossfunktionales Team mit verschiedenen Expertisen zu bilden, so dass die Aufgaben im Normalfall nicht outgesourct werden.

Generell wird viel mit dem PO kommuniziert und Rücksprache gehalten, um letztendlich das zu entwickeln, was der PO auch möchte. Manche der Teams sind vollständig crossfunktional und in fast allen Fällen arbeiten die Teams selbstorganisiert, so dass sie sich ihre Aufgaben im Sprint selber aussuchen und einteilen können. Bei den Events gibt es, abgesehen von ein paar Kleinigkeiten, nicht sonderlich gravierende Abweichungen. Die Events finden bei fast allen Unternehmen jeden Sprint statt. Bei dem Daily Scrum wird immer anstehende Tag geplant und sich im Entwicklungsteam abgesprochen und manchmal wird sogar der ausgelagert, um die Timebox einzuhalten. Das Planning wird von fast allen wie vorgesehen durchgeführt. Der Kern der Reviews und Retrospektiven wird auch nach Theorie umgesetzt.

5.3 Interpretation

Die Auswertung und Beantwortung der Forschungsfragen haben gezeigt, dass oft von der

Theorie abgewichen wird. Jedes Unternehmen hat eigentlich mindestens eine, wenn nicht sogar mehrere Aspekte, bei denen Abweichungen oder Anpassungen in irgendeiner Art und Weise vorgenommen wurden. Es gibt kein Unternehmen, das immer zu 100% die Theorie immer eingehalten hat, jedoch sind drei dem relativ nahe gekommen. Viele Abweichungen entstehen nicht unbedingt als Standard, sondern meistens als Resultat eines als unveränderbar wahrgenommenen Umstandes, wie beispielsweise Probleme mit externen Product Ownern. Außerdem passieren Abweichungen häufig aufgrund alter Unternehmensstrukturen oder genereller Unerfahrenheit mit Agilität und Scrum. Jedoch ist in vielen Aspekten leider nicht bekannt, welche Konsequenzen die bestimmten Abweichungen haben und ob diese nicht auch – je nach Kontext – nicht manchmal vorteilhaft sein können (mehr dazu in Kapitel 6).

Ein zentraler Punkt bei den Abweichungen sind die Probleme mit den Product Ownern, die man gerade bei externen Kunden nur schwer beeinflussen kann und oft von Projekt zu Projekt verschieden sind. Bei Problemen mit der Kundenagilität ist dies auch nicht einfach veränderbar bzw. man kann niemanden dazu zwingen agil zu werden. Selbst, wenn ein Unternehmen bereits dabei ist, agiler zu werden, passiert dieser Prozess nicht von heute auf morgen. Es lässt sich vermuten, dass durch Unerfahrenheit entstehende Probleme mit zunehmender Erfahrung in der Praxis sich verbessern könnten. Ebenfalls befindet sich das Problem mit den alten und statischen Unternehmensstrukturen bei vielen schon im Wandel, so dass sich viele Unternehmen mehr in Richtung Agilität und weg vom klassischen Projektmanagement bewegen möchten.

Manche Abweichungen erscheinen problematisch, wie die mangelnde Entscheidungsgewalt eines Product Owners und sollten sich wahrscheinlich nicht zum Standard etablieren. Man könnte darauf schließen, dass die Vorteile der agilen Softwareentwicklung spürbarer zum tragen kommen, wenn an dieser Stelle der Prozess reibungsloser ablaufen würde, wenn der PO auch wirklich seine Rolle voll und ganz ausüben darf und kann, so dass die Entwicklung effektiver funktionieren kann. Manchen Unternehmen fällt es schwer, den externen Product Owner in seinen Sprinteingriffen einzuschränken, weil man die Kundenzufriedenheit durch Konfliktvermeidung nicht schaden möchte und so vielleicht ungünstige Probleme unbehandelt lässt. Durch die externen PO erscheint es so, als ob der kontinuierliche Verbesserungsprozess der Scrum Teams nur begrenzt stattfinden kann, da von Projekt zu Projekt sich mit einem neuen PO eingearbeitet werden muss und somit nie wirklich Teil des Teams werden kann. Eigentlich möchte man sich mit der Zeit immer mehr verbessern, was dadurch schwieriger wird.

Bei den häufig beschriebenen Agilitätsproblemen der Kunden nehmen Scrum Master eine essentielle Rolle ein, was von vielen bereits passiert. Gleichzeitig wäre es dafür vorteilhaft, wenn die Scrum Master von ihren eigenen Unternehmen mehr wertgeschätzt werden und für nicht zu viele Dinge gleichzeitig eingesetzt werden. Falls ein Scrum Master in zu vielen Projekten ist, kann es schnell dazu kommen, dass er sich nicht richtig um seine Aufgaben innerhalb der verschiedenen Projekte kümmern kann. Eine Hybridrolle hingegen birgt Interessenkonflikte, weil ständig zwischen den Rollen gewechselt werden muss und verschiedene Interessen der beiden Rollen vertreten werden.

Eine interessante Beobachtung ist, dass Probleme mit Kunden und Product Ownern häufiger auftreten, wenn die Scrum Master wenig Erfahrung haben oder sich nur sehr kurzfristig mit der Thematik befasst haben. Dies könnte dazu führen, dass essentiellen Teile der Rolle nicht

ausreichend verstanden werden und somit gewisse Coachingaspekte nicht richtig umgesetzt werden können.

Immerhin wird sich schon bemüht, dass das Entwicklungsteam in der Lage ist ein Inkrement selbstständig zu erstellen. Um die Prioritäten allerdings ideal einhalten zu können und Leerlauf zu vermeiden, erscheint es sinnvoll, wenn sich Unternehmen noch mehr in die Crossfunktionalität wagen und Experten auch außerhalb ihres Gebietes Aufgaben erledigen lassen oder ihnen die Chance geben sich weiterzubilden bzw. andere zu unterstützen. Dies kann durch regelmäßige Wissenstransfers oder Pair Programming passieren, wie es in manchen Unternehmen gemacht wird. Bei unzureichend crossfunktionalen Teams besteht die Gefahr, dass der PO nach Teambzusammensetzung anstelle von Wertmaximierung priorisieren muss, nur damit alle ausgelastet werden. Natürlich kann man nicht von heute auf morgen crossfunktional werden und die durch Spezialisierung entstehende Expertise ist nicht zu vernachlässigen. Damit jedoch nicht zu viel Leerlauf entsteht, könnte ein Mittelweg zwischen Experten und „jeder macht alles“ die beiden Aspekte gut vereinen.

Da viele der Entwicklungsteams im Planning dazu neigen sich zu viel vorzunehmen, könnte es hilfreich sein, wenn der Scrum Master dem Entwicklungsteam hierfür mehr Unterstützung anbietet oder man das Sprintziel extra etwas geringer ansetzt. Dies kann auch einen positiven Eindruck bei den Stakeholdern und dem PO hinterlassen, was letztendlich die Atmosphäre im Entwicklungsprozess harmonischer werden lässt. Falls man sich regelmäßig zu viel vornimmt, kann dies ergebnisorientierte Stakeholder enttäuschen, die irgendwann das Team und das Projekt infrage stellen, weil doch nie etwas geschafft werde. Hierfür könnten Puffer auch sinnvoll sein, damit das Team nicht maximal ausgelastet ist, sondern man ein wenig Spielraum für Probleme hat.

Bei dem Daily kann man sich überlegen, wie schlimm es ist, wenn ab und zu die Timebox nicht eingehalten wird. Die Timebox soll für jedes Team, Sprintlänge und Projekt gleich groß sein, was nicht sonderlich sinnvoll erscheint. Ein Entwicklungsteam bestehend aus 3 Entwicklern in einem 2-wöchigen Sprint hat wahrscheinlich deutlich weniger Kommunikationsbedarf, als ein Entwicklungsteam mit 8 Entwicklern in einem 3-wöchigen Sprint.

Viele Unternehmen veranstalten am Ende des Sprints eine Art Meetingtag, wo quasi alle Events nacheinander abgearbeitet werden. Dies ist eine Reaktion auf die Probleme des Kunden, die nicht an zu vielen Terminen Zeit für das Scrum Team haben oder Zeit nehmen wollen. Der Kontrapunkt zu einem Meetingtag jedoch ist, dass es als lästig empfunden werden kann, da man alle X Wochen einen Tag hat, der nur mit Meetings gefüllt ist. Dadurch kann die Produktivität und das Engagement der Teilnehmer sinken, da recht schnell die Konzentration und Motivation sinkt, wodurch die Kernkomponenten des Verbesserungsprozess verloren gehen, wenn manche Teilnehmer nicht voll und ganz bei der Sache sind.

Viele Unternehmen wollen sehr flexibel auf kritische Fehler reagieren und erlauben deshalb Eingriffe in den Sprint, die dessen Umfang tiefgreifend verändern können. Hier kann man sich natürlich die Frage stellen, wie viel Sinn es macht, einen Kunden bis zum Sprintende warten zu lassen oder ob es in solchen Fällen nicht sinnvoll ist, wenn man den Sprint abbricht und das Problem behebt oder die Umfangsänderung einfach zulässt. Wenn man wichtige Systemfehler bei den Kunden schnell beheben möchte, ist es natürlich wichtig, dass die

Priorität im Sprint stets einzuhalten ist, damit bis zu diesem Zeitpunkt immerhin das wichtigste schon bearbeitet wurde. Außerdem kommt die Frage auf, warum so viele Teams mit diesen Problemen zu kämpfen haben und was die Ursache von solchen Fehlern ist.

6 Verwandte Arbeiten

In der Einleitung wurde bereits beschrieben, dass sich Scrum immer weiter verbreitet hat und mittlerweile zum Mittel der Wahl in der Softwareentwicklung geworden ist [2]. Es wurde allerdings schon häufiger festgestellt, dass in agilen Vorgehensweisen, also auch bei der Verwendung von Scrum, von der Theorie abgewichen wird [4], und, dass das Abweichen Nachteile nach sich ziehen kann [4].

Dieses Kapitel beschäftigt sich mit der bisherigen Forschung, die sich mit Theorieabweichungen von Scrum und Agilität in der Softwareentwicklung beschäftigt hat. Dafür werden verschiedene Paper betrachtet und dessen Resultate in Verbindung mit denen dieser Ausarbeitung gebracht. Allerdings handelt es sich dabei nicht um eine vollständige Auswertung der existierenden Literatur, sondern dient als kleiner Einblick in bereits durchgeführte Forschung zu diesem Thema. Durch eine systematische Literatursauswertung ließe sich vermutlich einiges mehr an Erkenntnissen und bekannten Forschungsergebnissen finden, allerdings war dies nicht zentraler Bestandteil der Ausarbeitung.

Der 14. jährliche „state of agile report“ [3] hat durch über 40000 Teilnehmer aus der agilen Unternehmenswelt mittels Fragebögen Daten zur Agilität in den Unternehmen gesammelt. Auch hier wird zeigt sich, dass Scrum die weitverbreitetste agile Vorgehensweise ist und ca. 3 von 4 der teilnehmenden Unternehmen Scrum bzw. Scrum mit Ergänzungen, wie z.B. Kanban oder XP verwenden [3]. Bei dem Prozess sein Unternehmen agiler zu machen, hat sich als eine der Hauptfaktoren herausgestellt, dass die Unternehmenskultur und -struktur eine bei traditionellen Strukturen und statische Hierarchien den Wandel verlangsamten können [3]. Viele Unternehmen sind noch nicht bereit, tiefgründige Änderungen vorzunehmen, so dass bestimmte Ebenen Probleme mit der Einführung von Agilität bekommen können [3]. Außerdem wird betont, dass die Reife der Agilität mit steigender Erfahrung zunimmt, so dass die Vorteile von Agilität mehr zum Vorschein kommen können [3]. Allerdings haben ca. ein Drittel der Teilnehmer erst weniger als 3 Jahre Erfahrung mit Agilität [3], was u.U. dazu beitragen kann, dass noch nicht alles optimal läuft und man an der einen oder anderen Stelle Anpassungen gemacht werden müssen, ohne dessen Auswirkungen voll und ganz zu verstehen.

Neben dem „state of agile report“ zeigen die HELENA Daten [2] ein ähnliches Bild: Scrum oder eine Variante davon kommt auch hier in ungefähr drei Viertel der betrachteten Softwareprojekte zum Einsatz. Der Datensatz beinhaltet fast 1500 Projekte und es wurde untersucht, welche Methoden, Praktiken sowie welche Kombinationen davon in der Softwareentwicklung Anwendung finden [2]. Hybride Ansätze sollten durch kontinuierliche Verbesserung durch Verwendung der eigenen Erfahrungswerte und nicht durch wahllose Abweichungen passieren, da sonst die Gefahr besteht die angestrebten Projektziele nicht zu erreichen [2]. Außerdem wird berichtet, dass 40% der Unternehmen Schwierigkeiten haben Agilität einzuführen [2], was wiederum zu unerwünschten Abweichungen führen kann.

2018 untersuchten Hohl et al. [4] mithilfe von Interviews, wie 14 der ursprünglichen 17 Erfinder des agilen Manifests [5] zu den aktuellen Entwicklungen in der agilen Softwareentwicklung stehen. Von den 17 Erfindern konnten 14 interviewt werden [4]. Viele waren der Meinung, dass das rapide Wachstum der Agilität schnell eine Ursache für Probleme geworden ist und jetzt viele Unternehmen plötzlich die Dringlichkeit verspüren, Agilität

introduzieren zu müssen, weil es so verbreitet ist [4]. Außerdem wird Agilität häufig auch falsch in den Augen der ursprünglichen Verfasser [5] interpretiert oder verstanden, so dass viele willkürliche Abweichungen entstehen [4]. Ein weiteres Problem besteht darin, dass die Agilität sehr kommerzialisiert wurde und dadurch falsche Experten zum Vorschein kommen, die eigentlich kein tiefes Verständnis von der Materie haben [4]. Häufig scheitert das Einführen von Agilität an den veralteten bürokratischen Strukturen der Unternehmen und höheren Management-Ebenen [4], ähnliche wie es der „state of agile report“ [3] zeigt. Zudem wird betont, dass viele der agilen Vorteile ausbleiben, wenn man essentielle Aussagen des agilen Manifests, wie beispielsweise „working software over comprehensive documentation“ [5], vernachlässigt [4]. Auch dieses Paper berichtet davon, dass Scrum die weit verbreitetste Vorgehensweise in der Softwareentwicklung ist [4]. Diese Aspekte der Studien [3, 4, 5] lassen sich auch in der Auswertung der Interviews wiederfinden, da viele Unternehmen von Problemen mit unerfahrenen Kunden und alten Unternehmensstrukturen berichtet haben.

Ein aktuelles Paper von Tell et al. [7], was sich mit der Zusammensetzung und dem systematischen Aufbau von hybriden Entwicklungsansätzen befasst [7]. Wie in den anderen Papern auch, wird in diesem Paper wieder erwähnt, dass bei Entwicklungsmethoden häufig von der intendierten bzw. ursprünglich gedachten Anwendung abgewichen wird [7]. Viele der Abweichungen von der eigentlichen Theorie führen zu den hybriden Methoden [7]. Allerdings existiert kein wirklich gemeinsames Fundament, wie man hybride Ansätze am besten systematisch erstellen kann, um sie dann letztendlich optimal auf die jeweiligen Unternehmensumstände zuschneiden zu können, so dass ungünstige Abweichungen vermieden werden können [7]. Bei der Analyse wurde quantitativ vorgegangen und für eine Kombination von Scrum mit Lean Development und iterativer Entwicklung ergeben sich Möglichkeiten für 643 verschiedene Variationen [7]. Durch die Kombinierung von Scrum mit Kanban und XP bzw. Scrum mit nur einem von beidem ergeben sich viele weitere mögliche Varianten [7].

Diebold et. al. [6] untersuchten 2015, wie Scrum in der Softwareentwicklung umgesetzt wird und wollten eventuelle Abweichungen untersuchen, sowie dessen Hintergründe nachvollziehen [6]. Somit weist dieses Paper erhebliche Ähnlichkeiten in dem Forschungsaspekt mit dieser Ausarbeitung auf und es konnten einige Ähnlichkeiten, aber auch Unterschiede in den Ergebnissen festgestellt werden. Um die Daten zu erheben, wurden Interviews 10 Teilnehmern aus deutschen Softwareunternehmen durchgeführt. Anders als die Interviews dieser Ausarbeitung, wurden sie nicht aufgenommen, sondern die Notizen wurden nur während des Interviews angefertigt [6].

Neben den Scrum Aspekten wurde etwas ausführlicher über Qualitätssicherung und Requirements Engineering geforscht, als es in dieser Ausarbeitung geschehen ist. Nachdem die Interviews qualitativ ausgewertet wurden, stellte sich heraus, dass lediglich eins der interviewten Unternehmen nah an Empfehlungen des Scrum Guides ist [6]. Alle anderen Teilnehmer sind in mindestens einem Scrum-Aspekt auf irgendeine Art und Weise abgewichen [6]. Am häufigsten werden Abweichungen und Anpassungen aus pragmatischen Gründen gemacht, wodurch man sich mehr Effizienz im Entwicklungsprozess verspricht [6]. Allerdings entstehen viele Abweichungen auch aus dem Grund heraus, dass die traditionelle Denkweise und alte Strukturen noch in vielen Unternehmen verankert ist und so die einwandfreie Umsetzung von Scrum erschwert wird [6]. In vielen Teams tauchte gar kein Product Owner auf, so dass die Rolle von manchen Entwicklungsteams selber übernommen

wurde [6]. Manche Teams haben es gar nicht als nötig betrachtet, einen PO zu haben [6]. Teams, die einen PO hatten, haben nie wirklich Probleme mit nicht-förderlichen Sprinteingriffen bekommen [6], anders als es bei der Auswertung in Kapitel 4 der Fall war.

Meistens gab es einen Scrum Master als Hybridrolle, also neben der Scrum Master Tätigkeit war diese Person zusätzlich Entwickler oder im Management tätig, weil die Rolle an sich nicht genügend Arbeit für eine Vollzeitposition bietet [6]. Bei vielen Unternehmen gab es keine wirkliche definition of done, was ggf. zu Kommunikationsproblemen führen kann [6]. Die Sprintdauer stimmte mit den Empfehlungen des Scrum Guides [5] überein und die am häufigsten vorkommende Länge war 4 Wochen [6]. Das Daily wurde bei fast allen normal durchgeführt, allerdings gab es bei manchen Problemen mit der vorgesehenen Timebox und manche haben es nicht täglich gehalten [6]. Auf die Durchführung von Reviews wurde großer Wert gelegt, jedoch ist dies meistens ohne Stakeholder passiert [6]. Manchmal wurde die Retrospektive vernachlässigt, so dass sie nicht in jedem Unternehmen regelmäßig stattgefunden hat [6]. Somit bestehen bei Diebold et al. [6] viele Gemeinsamkeiten in den Abweichungen, die in dieser Ausarbeitung festgestellt worden sind.

Eine weiteres Paper, was einer ähnlichen Fragestellung nachgegangen ist, war eine empirische Studie zu Scrum „anti-patterns“ und „Scrum-But“ [8]. „Scrum-But“ beschreibt das Phänomen, wenn Unternehmen sagen „Wir benutzen Scrum, aber machen irgendwas anders“ [20] und ein anti-pattern ist eine Abweichung, die auf den ersten Blick attraktiv erscheint, aber langfristig schädlich sein kann [8]. Hier wurden wieder die Hintergründe der Abweichungen untersucht und anschließend Empfehlungen für einen besseren Umgang mit den Abweichungen gegeben [8]. Anti-patterns entstehen oft aus einer guten Intention heraus, jedoch kann ein mangelndes Verständnis über die Konsequenzen der Abweichung dazu führen, dass sich die Abweichung anders auswirkt, als erwartet [8].

Eloranta et al. [8] haben für ihr Vorhaben 18 Scrum Teams aus 11 finnischen Softwareunternehmen interviewt. Eine der größten festgestellten Problembereiche waren die Product Owner und die Sprints. Die häufig vom Kunden stammenden Product Owner sind oftmals nicht berechtigt eigenständig eine Entscheidung über den weiteren Verlauf des Projekts treffen zu dürfen, so dass ständige Rücksprache mit dem überstehenden Management erforderlich ist [8]. Die traditionellere Herangehensweise mancher Kunden erfordert Kompromisse, so dass man sich anpassen muss. Häufig legen diese Unternehmen noch Wert die Anforderungen detailliert im Vorfeld mittels einer Spezifikation festzuhalten [8]. Insgesamt scheinen manche Kunden also noch große Probleme zu haben in die agile Welt einzutreten [8]. Die Unerfahrenheit könne außerdem dazu führen, dass die Entwicklungsteams des Öfteren durch nicht-hilfreiche Sprinteingriffe vom Product Owner oder anderen Kundenvertretern gestört wurden [8]. Viele Scrum-Praktiker äußerten die Befürchtung ihre Kundenbeziehungen zu schädigen, wenn sie in einen Konflikt über Agilität geraten würden und deshalb möchte man ihnen oft nicht widersprechen, auch wenn der Kunde die PO Rolle falsch besetzt, woraus dann ggf. Nachteile in der Entwicklung entstehen [8]. Außerdem durften viele Teams nicht selbstorganisiert arbeiten, da Aufgaben durch den PO oder einen Projektmanager zugeteilt worden sind [8]. Zudem berichteten einige Teilnehmer von einem nicht priorisiertem Backlog oder Projekten ohne PO [8], ähnliche wie es bei Diebold et al. [6] der Fall ist.

Hingegen kommt es bei den Sprints häufiger zu Problemen, wenn das Testen von der

Entwicklung getrennt wurde, so dass pro Sprint kein releasefähiges Inkrement entsteht [8]. In vielen Fällen lag das an mangelnder Crossfunktionalität bzw. festgelegter Rollen im Entwicklungsteam [8]. Auch in dieser Untersuchung stellte sich heraus, dass die Retrospektive gelegentlich vernachlässigt wird [8] und die häufigste Sprintlänge 2 Wochen ist [8]. Auf der anderen Seite wurde das Daily Scrum als nicht problematisch eingestuft, da es von allen Teams durchgeführt wurde und für niemanden Probleme bereitet hatte [8].

Man muss jedoch beachten, dass nicht alle Abweichungen automatisch schlecht sind. Manche Abweichungen entstehen aus einem guten Grund, weil es die Situation so erfordert hat oder es noch nicht anders geht [8]. Größere Unternehmen neigen vermehrt dazu von der Theorie abzuweichen, weil dort viele der hierarchischen Strukturen noch sehr fest sitzen und dann das Bedürfnis, schnell agil zu werden, nicht von heute auf morgen passieren kann [8]. Mit steigender Scrum Erfahrung eines Unternehmens stieg auch die Neigung vom Standardprozess abzuweichen [8]. Dies geschieht meistens im Kontext eines kontinuierlichen Verbesserungsprozesses, weil bestimmte Dinge ausprobiert wurden und Potential für Verbesserungen gesehen wird [8]. Vergleicht man die Untersuchung von Diebold et al. [6] und Eloranta et al. [8] mit dieser Ausarbeitung, lassen sich viele Überschneidungen finden, insbesondere bei den alten Unternehmensstrukturen und PO Problemen.

Eine Untersuchung aus dem Jahr 2017 [21] stellt vor, wie man die Scrum-Hülle sinnvoll mit Engineering-Praktiken ergänzen kann. Scrum fokussiert sich auf das Projekt- und Anforderungsmanagement und sagt wenig bis gar nichts über Engineering-Praktiken, wie dem eigentlichen Entwickeln oder Testen, aus [21]. Deshalb wurde durch einzelne Verbesserungen und Ergänzungen an Scrum die Methode „Scrum++“ erstellt, was ein verbessertes Scrum sein soll [21]. Bei der Erstellung von „Scrum++“ wurde zunächst ein Abhängigkeitsnetz agiler Praktiken erstellt, welches Harmonien und Hindernisse einzelner Praktiken sichtbar macht, was anschließend geclustert wurde [21]. Insgesamt umfasst die neu entstandene Methode 30 agile Praktiken und besteht aus 4 Kernaspekten [21]. Die Kernaspekte beinhalten Sachen zur technischen Umsetzung, zur Fortschrittmessung und zur Anforderungsverwaltung [21]. Einige der angewendeten Praktiken sind beispielsweise User Stories, Burndown Charts, Pair Programming, Continuous Integration, Continuous Delivery und ein Product Vision Board [21]. Gerade ein Product Vision Board dient für die mittel- bis langfristige Projektplanung [21] und scheint gerade bei den häufigen Kundenproblemen mit Agilität sinnvoll.

Vergleicht man die untersuchten Aspekte insgesamt mit den Ergebnissen aus Kapitel 4, lassen sich einige Ähnlichkeiten und Unterschiede feststellen. Beispielsweise sind Gemeinsamkeiten, dass es häufig Probleme mit den alten Strukturen in den Unternehmen oder mit den Product Ownern gibt [6, 8], wo das Verständnis von Agilität noch nicht sehr unreif ist. Jeweils wird viel Wert auf kurze Iterationszyklen mit den 2-4 wöchigen Sprints gelegt [6]. Eine weitere Gemeinsamkeit lässt sich in den durchgeführten Reviews ohne Stakeholdern finden [8].

Viele der Forschungen betonen [4, 6, 8], dass die Auswirkungen von Abweichungen nicht bekannt sind und sie vielleicht negative Konsequenzen haben, die man zur Zeit gar nicht bedenkt. Öfter stellt sich heraus, dass viele Abweichungen aus Notwendigkeit heraus entstehen, weil man beispielsweise noch in alten Hierarchien feststeckt und sich nicht sofort etwas verändern kann [6, 8], da dieser Prozess meistens ein bisschen Zeit braucht.

Ausgenommen von der Retrospektive legen die Unternehmen auch großen Wert auf die Durchführung der anderen Events [6, 8]. Die Retrospektive, auch wenn dessen Umsetzung nicht perfekt passiert, wurde in den Interviews dieser Ausarbeitung nur in einem Fall unregelmäßig durchgeführt, wohingegen Diebold et al. [6] und Eloranta et al. [8] von unregelmäßig durchgeführten Retrospektiven in mehreren Fällen berichten.

Weitere Unterschiede in den Ergebnissen sind beispielsweise nicht selbstorganisierte Teams [8]. Im Vergleich zur Untersuchung mit Diebold et al. [6] wurden dort deutlich weniger Probleme mit Sprinteingriffen berichtet. Jedoch berichteten Eloranta et al. [8] auch von häufigen Sprinteingriffen, wie es in dieser Ausarbeitung der Fall ist. Ein weiterer Unterschied in den Ergebnisse zu Eloranta et al. [8] ist zu sehen, wo Anforderungen häufig noch sehr detailliert festgehalten wurden, was in dieser Interviewstudie nur ein mal vorgekommen ist.

Einige dieser Unterschiede können darauf zurückzuführen sein, dass die beiden Untersuchungen zu Scrum in 2015 [6, 8] veröffentlicht wurden und sich seitdem das Agilitätsbewusstsein in manchen Organisationen bereits verbessert hat, so dass z.B. die Wichtigkeit die Anforderungen nicht exakt mit einer Spezifikation in Stein zu meißeln, gestiegen ist. Außerdem handelt es sich meistens um kleine Datenmengen und vor allem bei quantitativen Auswertungsaspekten der Interviews können schnell Diskrepanzen auftreten.

7 Fazit und Ausblick

Das letzte Kapitel gibt einen zusammenfassenden Überblick über die Ausarbeitung und beschreibt basierend auf den Ergebnissen Möglichkeiten und Anhaltspunkte für weitere Forschung.

Die durchgeführte Interviewstudie hat das Verständnis und die Nutzung von Scrum in der Softwareentwicklung untersucht. Sie ist aufgrund des rapiden Anstiegs in der Benutzung von Scrum und aufgrund der Tatsache, dass häufig von der Theorie abgewichen wird, entstanden. Insgesamt wurden 10 Interviews mit unterschiedlichen Scrum Teams durchgeführt. Mithilfe eines Leitfadens wurden die Interviews via Videotelefonie als semi-strukturiertes Interview durchgeführt und im Anschluss mithilfe der qualitativen Inhaltsanalyse nach Mayring ausgewertet.

Bei der Auswertung stellt sich heraus, dass viele Unternehmen Probleme haben sich an die Theorie zu halten und an vielen Stellen Abweichungen entstehen. Keins der interviewten Teilnehmer hält sich immer zu 100% an die Theorie. In irgendeinem Bereich kommt immer mindestens eine, wenn nicht mehrere, Abweichungen zum Vorschein. Oft entstehen sie gar nicht aus eigenem Willen heraus, sondern manchmal spielen externe Faktoren die entscheidende Rolle, welche von den Unternehmen nicht unmittelbar beeinflussbar sind. Vor allem bei Kunden kam es zu Problemen, auf die man auch in irgendeiner Art und Weise angewiesen ist und ihnen kein Verhalten explizit vorschreiben kann.

Allerdings sind Abweichungen nicht auf jede Branche pauschalisierbar, und manche Abweichungen entstehen eher als andere, je nach Branche. So kann es sein, dass in regulierten Branchen durch gesetzliche Vorgaben bestimmte Fristen zwingend eingehalten werden müssen und dadurch die Sprints oder Teams im Notfall auch mal etwas angepasst werden müssen. Hingegen haben Unternehmen, die sich um das Herstellen von Individualsoftware kümmern, viel Kontakt zu ihren Kunden und das ist für sie eine zentrale Quelle der Probleme. Man ist auf die enge Zusammenarbeit angewiesen und meistens gibt es Probleme bei den von den Kunden gestellten Product Ownern. Hier trifft ein häufiges Problem aus alten Unternehmensstrukturen auf, wodurch die Product Owner entweder zu wenig Entscheidungsgewalt haben oder noch sehr wenig Erfahrung in der agilen Softwareentwicklung sammeln konnten. Bei den Unternehmen, die Projekt immer in ein neues Team reinkommen, gibt es ähnliche Abweichungen. Dort kann sich der kontinuierliche Verbesserungsprozess eines Scrum Teams nicht entfalten, da man sich immer neuen Umständen anpassen muss. Dadurch entstehen dann eben solche Abweichungen, die man als Unternehmen nicht direkt beeinflussen kann.

Jedoch gibt es nicht nur Abweichungen, sondern manche Aspekte werden aus der Theorie auch häufig in der Praxis umgesetzt. Alle Unternehmen der Interviewstudie haben Wert auf das Einbeziehen von Kundenfeedback gelegt und somit dem klassischen Problem der statischen Anforderungen im Wasserfallmodell entgegengewirkt. Auch wird großer Wert auf die ständige Kommunikation und Berücksichtigung der Kundenbedürfnisse gelegt. Des Weiteren kommen die kurzen Iterationszyklen zum Einsatz und die Entwicklungsteams sind meistens selbstorganisiert und halbwegs crossfunktional.

Anders als bei den vorgestellten verwandten Arbeiten, haben sich in dieser Interviewstudie

weniger Probleme mit den Retrospektiven und der Anforderungsverwaltung gefunden. In 9 von 10 Fällen wurden Anforderungen ohne klassischer Spezifikation verwaltet und die Retrospektiven regelmäßiger durchgeführt. Jedoch gibt es auch Gemeinsamkeiten zu den verwandten Arbeiten, und zwar, dass die meisten Abweichungen bei Product Ownern passieren. Viele Probleme entstehen auch dadurch, dass sich die Agilität vielleicht etwas zu schnell verbreitet hat und auf ein mal sehr viele eine Dringlichkeit verspüren, agil werden zu müssen. Das birgt natürlich Risiken und Gefahren, da Agilität ein tiefes Verständnis benötigt und man seine Unternehmensstruktur nicht innerhalb einer Tages grundlegend verändern kann. Allerdings lassen sich die Resultate dieser Interviewstudie nicht unbedingt verallgemeinern oder auf andere Branchen ohne weiteres übertragen. Dies ist auch eine der Schwachstellen dieser Interviewstudie, da sich die Teilnehmer nur bedingt ausgesucht werden konnten und bestimmte Branchen, Unternehmensgrößen, vielleicht auch Unternehmenskulturen unter- bzw. gar nicht repräsentiert werden.

Obwohl zwar vielen bekannt ist, dass Abweichungen gemacht werden, ist nicht viel über dessen Auswirkungen bekannt. Zwar hat die HELENA-Studie schon einige Zahlen bzgl. der Erfolgsaussichten eines Projektes bei Abweichungen gegeben, jedoch braucht man tieferegehende Analysen, um einzelne Abweichungen fundiert beurteilen zu können. Dabei könnte es hilfreich sein, wenn die Auswirkungen einzelner Aspekte systematisch erforscht werden, auch wenn sich solche Studien in der Softwareentwicklung nicht immer einfach realisieren lassen.

Daneben könnten Meta-Analysen über die bereits existierende Forschung zu Scrum und seinen Abweichungen vorgenommen werden, wodurch sich vielleicht einige Sachen schon etwas mehr generalisieren lassen. Wenn man tieferegehendes Wissen zu bestimmten Abweichungen und seine Hintergründe hat, kann man anfangen Strategien zu entwickeln, wie man besser mit gewissen Abweichungen umgehen kann und welche Abweichungen für welche Kontexte sinnvoll sind. Falls sich manche Abweichungen als sinnvoll herausstellen, lassen sich langfristig gesehen manche idealerweise so zuschneiden, so dass man sie je nach Unternehmensbedürfnissen, Projekt, Team, dem Kunden und der eigenen Branche optimal einsetzen kann.

Insbesondere ist eine ungeklärte Fragestellung, wie problematisch Abweichungen an der PO Rolle sind, weil es dabei scheinbar am häufigsten zu Problemen kommt. Zu der Problematik mit der Kundenagilität könnten Untersuchungen angestellt werden, inwiefern sich das denn im Laufe der Zeit verändert und wie man solche Probleme eventuell besser angehen kann. Gleichzeitig wäre es sinnvoll zu wissen, wie ein Unternehmen mit alten Strukturen schneller zur Agilität wechseln kann, wenn dies vom Unternehmen erwünscht ist. Außerdem stellt sich die Frage, wie man die Crossfunktionalität der Teams erhöhen kann, so dass die Sub-Teams innerhalb des Entwicklungsteams vermieden werden können und der kontinuierliche Verbesserungsprozess im Entwicklungsteam angeregt werden kann.

In einigen Interviews hat sich gezeigt, dass häufig am Ende des Sprints alle Events am selben Tag nacheinander stattfinden, um so der knappen Kundenverfügbarkeit etwas entgegenzukommen. Nun kommt die Frage auf, ob dies nicht negative Auswirkungen auf den Scrum Prozess haben kann, wie beispielsweise Konzentrationsprobleme oder eine genervte Einstellung zu diesem Tag, so dass sich die Effektivität der einzelnen Events an diesem Tag

verringert oder ob dies keine Auswirkungen hat und vielleicht sogar eine gute Anpassung an die mangelnde Flexibilität des Kunden ist.

Abgesehen von den PO Eingriffen in den Sprint, wird der Sprintumfang häufig aufgrund von kritischen Fehler in Kundensystemen verändert. Zunächst erscheint es sehr flexibel und tolerant von den Scrum Teams zu sein, dies zu erlauben. Allerdings stellt sich die Frage, wieso diese Fehler so häufig passieren und wenn sie so häufig vorkommen, warum man dann Scrum verwendet.

Anhang 1: Interviewleitfaden

Begrüßung

- Rahmenbedingungen des Interviews nennen, für Teilnahme bedanken

Hauptteil

Randinformationen zum Unternehmen/Person

- Können Sie ein mal kurz den Tätigkeitsbereich des Unternehmens beschreiben?
- Wofür sind Sie als X im Unternehmen zuständig?
- Wie lange haben Sie schon mit Scrum zu tun?

Entwicklungsprozess allgemein

- Wie sieht der Softwareentwicklungsprozess normalerweise bei Ihnen aus?
- Welche Aspekte von Scrum verwenden Sie bei der Softwareentwicklung?
- Welche Abweichungen zur Scrum-Theorie sind Ihnen bekannt?
- Wissen Sie vielleicht, was der Hintergedanke dieser Abweichung ist, warum man das gemacht hat?

Scrum-Team

- Wie groß ist das Scrum Team bei Ihnen?
- **Entwicklungsteam**
 - Um welche Aufgaben kümmert sich das Entwicklungsteam?
 - Arbeiten die Teams selbstorganisiert oder erfolgt eine Aufgabenzuweisung?
 - Sind die Teams crossfunktional?
 - Wie viele Entwicklungsteams gibt es? (falls etwas zu Skalierung erwähnt wurde)
 -
- **Product Owner**
 - Haben Sie einen Product Owner?
 - Was macht der Product Owner? Wie oft haben Sie mit ihm Kontakt?
 - Wie streng wird sich an die Prioritäten bzw. den Product Backlog gehalten?
 - (falls externer PO) Haben Sie Probleme mit der Agilität des Kundenunternehmens?
 - Haben die PO genügend Entscheidungsgewalt?
- **Scrum Master**
 - Haben Sie einen Scrum Master?
 - Was macht der Scrum master? Wie oft haben Sie mit ihm Kontakt?

Scrum-Events

- Welche (Scrum-)Meetings gibt es bei Ihnen?
- Daily Scrum
 - Worum geht es normalerweise bei einem Daily Scrum? Was wird gesagt und thematisiert?
- **Planning**
 - Wie sieht ein Planning normalerweise aus?
 - Wer nimmt an einem Planning teil?
- **Review**

- Wie sieht ein Review normalerweise aus?
- Wer nimmt an einem Review teil?
- **Retrospektive**
 - Wie sieht eine Retrospektive normalerweise aus?
 - Wer nimmt an einer Retrospektive teil?

Sprints

- Wie lange geht ein Sprint? Ist die Dauer immer gleich?
- Wie werden Anforderungen für einen Sprint erhoben?
- Wie wird zu Beginn eines Sprints der bevorstehende Sprint geplant?
- Woher wissen Sie, wie viel Sie in einem Sprint schaffen?
- Können Änderungen am Sprint während eines Sprints vorgenommen werden und wie sieht so etwas aus? Wie oft passiert das?
- Wie wird festgelegt, welche Aufgaben man im Sprint als nächstes bearbeitet?
- Wie tracken Sie den Fortschritt innerhalb eines Sprints? Benutzen Sie Software dafür?
- Was passiert, wenn die Ziele eines Sprints nicht erreicht wurden?
- Was passiert, wenn die Ziele eines Sprints frühzeitig erreicht wurden?
- Entsteht am Ende des Sprints immer ein Inkrement?

Ende

- Haben Sie noch ergänzende Anmerkungen zum Interview? Gibt es noch relevante Dinge, über die wir gar nicht gesprochen haben?
- Für die Teilnahme bedanken und Aufnahme beenden

Literaturverzeichnis

- [1] Ken Schwaber. SCRUM development process. in: Proceedings of 10th Annual Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), 1995, S. 117–134
- [2] Marco Kuhrmann, Paolo Tell, Jil Klünder, Regina Hebig, Sherlock Licorish, Stephen MacDonell (Eds.): Complementing Materials for the HELENA Study (Stage 2). [online] (November 2018)
- [3] Digital.ai. 14th annual state of agile report. VersionOne Inc (2019). URL: <https://stateofagile.com/>. (zuletzt aufgerufen am 01.09.2020)
- [4] Philipp Hohl, Jil Klünder, Arie van Bennekum, Ryan Lockard, James Gifford, Jürgen Münch, Michael Stupperich und Kurt Schneider. Back to the future: origins and directions of the “Agile Manifesto” – views of the originators. Journal of Software Engineering Research and Development. (2018)
- [5] Beck et al. Manifesto for Agile Software Development. (2001). URL: <https://agilemanifesto.org/> (zuletzt aufgerufen am 01.09.2020)
- [6] Diebold, Philipp & Ostberg, Jan-Peter & Wagner, Stefan & Zendler, Ulrich. (2015). What Do Practitioners Vary in Using Scrum?. Lecture Notes in Business Information Processing.
- [7] Tell, Paolo & Klünder, Jil & Küpper, Steffen & Raffo, David & MacDonell, Stephen & Münch, Jürgen & Pfahl, Dietmar & Linssen, Oliver & Kuhrmann, Marco. Towards the Statistical Construction of Hybrid Development Methods. (2020)
- [8] Eloranta, Veli-Pekka & Koskimies, Kai & Mikkonen, Tommi. Exploring ScrumBut – An Empirical Study of Scrum Anti-Patterns. Information and Software Technology. (2015).
- [9] Ken Schwaber, Jeff Sutherland. The Scrum Guide. Scrum.org (2017). URL: <https://www.scrumguides.org/scrum-guide.html> (zuletzt aufgerufen am 31.08.2020)
- [10] Michael C. Layton. Scrum for Dummies. Wiley. (2015)
- [11] Stefan Roock. Scrum - auf dem Bierdeckel erklärt. It-agile, dpunkt.verlag (März 2016)
- [12] Chris Spanner. Scrum of Scrums – so wird skaliert. Atlassian. URL: <https://www.atlassian.com/de/agile/scrum/scrum-of-scrums> (zuletzt abgerufen am 13.09.2020)
- [13] Ken Schwaber. Nexus Guide. Scrum.org (Januar 2018). URL: https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-English_0.pdf (zuletzt abgerufen am 13.09.2020)

- [14] Craig Larman, Bas Vodde. Large-Scale Scrum: More with LeSS (Mai 2014)
- [15] Lea Genau, Ein semistrukturiertes Interview führen. Scribbr (August 2020) URL: <https://www.scribbr.de/methodik/semistrukturiertes-interview/> (zuletzt abgerufen am: 12.09.2020)
- [16] Cornelia Endres. Leitfadeninterview für die Bachelorarbeit – Beispiele & Ablauf. BachelorPrint <https://www.bachelorprint.de/forschung/leitfadeninterview/> (zuletzt aufgerufen am 12.09.2020)
- [17] Philipp Mayring. Qualitative Inhaltsanalyse. In: Günter Mey, Katja Mruck. (eds) Handbuch Qualitative Forschung in der Psychologie. VS Verlag für Sozialwissenschaften (2010).
- [18] Isabel Steinhardt, Induktives und deduktives Codieren. Hypothesos. URL: <https://sozmethode.hypothesos.org/842> (zuletzt aufgerufen am 13.09.2020)
- [19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell und A. Wesslén. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell, MA, USA. 2000. S. 68 ff.
- [20] Unbekannt. What is ScrumBut? Scrum.org. URL: <https://www.scrum.org/resources/what-scrumbut> (zuletzt aufgerufen am 01.09.2020)
- [21] Philipp Diebold, Sven Theobald, Anna Schmitt und Cordula Schmidt. Weg vom unvollständigen Scrum! Hin zum vollständigeren „Scrum++“. Projektmanagement & Vorgehensmodelle 2017

Tabellen- und Abbildungsverzeichnis

Abbildung 1: Vereinfachte Scrum Übersicht.....	4
Abbildung 2: Die häufigsten Abweichungskategorien.....	20
Tabelle 1: Tätigkeiten der Interviewteilnehmer.....	17
Tabelle 2: Tätigkeitsbereiche der Unternehmen.....	18
Tabelle 3: Unternehmensgrößen der Interviewteilnehmer.....	18
Tabelle 4: Vorgehensweisen der Unternehmen.....	19
Tabelle 5: Kategorien der Artefakte.....	19
Tabelle 6: Kategorien der Scrum Teams.....	22
Tabelle 7: Kategorien der Scrum Events.....	25
Tabelle 8: Gründe von Sprintumfangsänderungen.....	25