

Methodische Entwicklung zur konzeptionellen Erfassung einer Softwarelösung für Scrum Master

Alexander Kaack

Department Informatik
Hochschule für Angewandte Wissenschaften Hamburg
Email: Alexander.Kaack@haw-hamburg.de

I. EINLEITUNG

Durch den Einfluss der stetig anwachsenden Marktdynamik ist eine ebenso steigende Planungskomplexität zu verzeichnen [1]. Im Bereich der Softwareentwicklung tragen besonders agile Vorgehensmodelle zur Reduzierung der vorhandenen Komplexität bei. Das Ziel dabei ist die Kontrolle des Projekts durch die Unterteilung in beherrschbare Teilaufgaben und Zeitabschnitte.

Nach einer weltweiten Studie ist die agile Softwareentwicklung bei einem Großteil der Organisationen vorzufinden. Die häufigsten Gründe hierfür sind das Managen von wechselnden Prioritäten, die verbesserte Produktivität und die verbesserte Sichtbarkeit des Projekts. Die eindeutige Mehrheit der Befragten verwendet dabei *Scrum* als agiles Vorgehensmodell [2].

Doch weiterhin scheitern die meisten Softwareprojekte. Nach dem aktuellen *Chaos Report* werden nur 9 % der Projekte in Großunternehmen, 16,2 % in mittleren Unternehmen und 28 % in Kleinunternehmen als erfolgreich ausgezeichnet. Definiert wurden die Unternehmensgrößen anhand ihres Umsatzes. Diese Kategorisierung kann der Tabelle I entnommen werden.

| Unternehmensgröße | Kriterium |
|-------------------|--|
| Groß | Umsatz < \$500 Millionen Dollar |
| Mittel | \$200 Millionen Dollar < Umsatz < \$500 Millionen Dollar |
| Klein | \$100 Millionen Dollar < Umsatz < \$200 Millionen Dollar |

Tabelle I. Kriterium der Unternehmensgrößen nach [3] S. 5

Als größte Herausforderungen während der Entwicklung wurden dabei der mangelnde Input von User, unvollständige Anforderungen & Spezifikationen sowie wechselnde Anforderungen & Spezifikationen angegeben. Als Faktoren, die den Prozess unmittelbar beeinträchtigt und letztendlich zum Abbruch der Softwareentwicklung geführt haben, wurden unvollständige Anforderungen, das fehlende Einbeziehen von Usern und die mangelnden *Ressourcen* (Personal) verantwortlich gemacht.

Aus den Ergebnissen lässt sich vor allem ableiten, dass kleinere Unternehmen erfolgreicher Software entwickeln und viele geschilderte Probleme nicht mit der eigentlichen Kultur von agiler Softwareentwicklung übereinstimmen. So kann man daraus schließen, dass generell keine agile Softwareentwicklung verfolgt wird oder die vollständige Adaption der agilen Entwicklung an das Unternehmen gescheitert ist.

So werden in der Studie *9th Annual State of Agile Survey* [2] auch im Kontext der agilen Entwicklung Gründe angegeben, warum agile Projekte scheitern. Hierzu zählen vor allem fehlende Erfahrung mit agilen Methoden, die Firmenphilosophie oder Kultur, die sich nicht mit den Kernwerten der agilen Softwareentwicklung verstehen und der fehlende Support des Managements.

Demnach schlägt die Adaption der agilen Methoden fehl, da sich die Organisationskultur nicht ändern lässt. Ebenso stellen der Mangel an Fachpersonal mit ausreichend agiler Expertise, wie auch die generelle Resistenz der Organisation zur Integration des agilen Gedankens, die größten Herausforderungen dar [2].

Innerhalb des Scrum Frameworks beinhaltet die Rolle des *Scrum Masters* zum einen die Einführung von Scrum im Unternehmen, wie auch die Beseitigung von Hindernissen (engl. *Impediments*), die eine erfolgreiche Softwareentwicklung nach Scrum verhindern. Der Scrum Master hat somit eine der wichtigsten Aufgaben, bezogen auf die aktuellen Probleme der Softwareentwicklung. Dieser ist dabei allerdings nicht als Hauptverantwortlicher zu sehen, sondern als essentieller Bestandteil der Softwareentwicklung nach Scrum, wonach das gesamte Team die durchzuführenden Projekte zu vertreten hat.

Ziel dieser Ausarbeitung ist es, dass zum einen der aktuelle Wissenstand im Bezug auf Scrum und an Anlehnung der Herausforderungen und Aktivitäten vom Scrum Master reflektiert wird. Hierdurch soll vor allem auch die Relevanz des Scrum Masters verdeutlicht werden. Bei einer nicht vollständig vollzogenen Integration des agilen Gedankens in die Unternehmenskultur durch den Scrum Master, steht dabei ein sich kontinuierlich verbessernder Entwicklungsprozess auf dem Spiel. Dieser steht zusätzlich den verschiedensten Herausforderungen und Problemstellungen gegenüber, die es zu erfassen gilt. Demnach gilt es, besonders Hindernisse innerhalb des Entwicklungsprozesses optimal aufzunehmen und abzuarbeiten. Mögliche bestehende Informationssysteme können bereits Faktoren während der Entwicklungsphase aufnehmen und einordnen, bzw. lassen sich durch geeignete Software unterstützen.

Um ein praxisrelevantes Ergebnis zu den aktuellen Möglichkeiten der Hinderniserfassung und den Bedarf von Unterstützungsmöglichkeiten zu bieten, gilt es, Informationen in aktiven Szenarien zu erfassen. Hierzu wurde ein Interviewleitfaden entwickelt, der im Zuge der weiteren Forschungsarbeiten durchgeführt werden soll.

Das Fazit wird die Ergebnisse dieser Arbeit zusammenfassen und einen Ausblick auf das weitere Vorgehen aufzeigen.

II. RELATED WORK

Die aktuellen Forschungsarbeiten zum Thema agile Methoden reichen von der Vorteilsdarstellung und Präsenz der Frameworks wie Scrum, Kanban und Co. [4], wie auch der verbesserten Kommunikation der Teammitglieder [5], bis hin zur möglichen Komplexitätsreduzierung für global verteilte agile Teams [6]. Darüber hinaus ist die generelle Einführung von Scrum [7], wie auch die sich daraus ergebene Problematik der Unternehmensschnittstellen, von besonderer Bedeutung [8]. In der Rolle des Scrum Masters ist die Erfassung von Hindernissen [9] eine der Aktivitäten, die bezogen auf mögliche Auswirkungen und Beseitigungskosten nicht trivial ist [10].

Die Relevanz bzgl. agiler Methoden ist wie in der Einleitung beschrieben weiterhin steigend, wodurch ebenfalls die Bedeutung ansteigt. Studien zeigen zudem auf, dass Scrum innerhalb der Softwareentwicklung immer häufiger verwendet wird. Auch ist zu erkennen, dass ein Wechsel von anderen agilen Vorgehensweisen zu Scrum stattfindet [4]. Vorteile, die sich unter anderem besonders bei Scrum wiederfinden, sind Planbarkeit, häufige Auslieferungen, verbesserte Kundenzufriedenheit und die gesamte Transparenz des Entwicklungsprozesses. Zudem ist die Flexibilität, besonders bezogen auf das Reagieren von Veränderungen im Projektverlauf und vor allem die insgesamt verbesserte Qualität von Software, hervorzuheben. Agile Prozesse ermöglichen eine bessere und kontinuierliche Interaktion zwischen Entwicklern und Kunden. Auch wird ein verbessertes Return on Investment (ROI) geboten. Grund hierfür ist auch die verbesserte Produktivität der Teams [4].

Die Produktivität der Teams lässt sich zudem auch durch die Verbreitung von Wissen und den Austausch von Kompetenzen steigern. Besonders bei der kontinuierlichen Weiterentwicklung des Team spielen diese Punkte eine besondere Rolle. Vergleicht man hier die hierarchische und klassische Struktur mit der agilen Herangehensweise, ist ein eindeutiger Vorteil messbar [5]. Die direkten und kurzen Wege ermöglichen auch Teams untereinander eine ideale Austauschbasis. Darüber hinaus bietet die Möglichkeit eines direkten Wissenstransfers mit dem Kunden eine Reduzierung des Informationsverlustes. Die klassischen Hierarchien sorgen für eine Informationsreduzierung durch die pyramidenförmige Unternehmensstruktur. Hierdurch wird auch die Adaptionsgeschwindigkeit von Änderungen deutlich verringert [5].

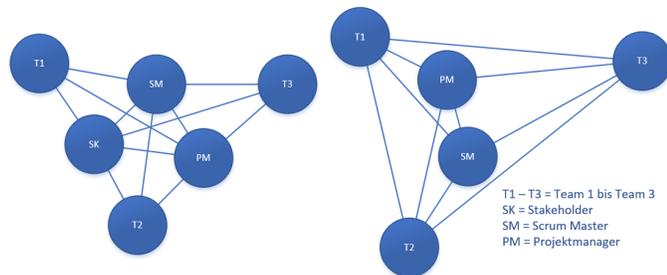


Abbildung 1. SCRUM Netzwerk der Wissensverbreitung (links) - SCRUM Netzwerk des Kompetenzaustausches (rechts) nach [5] S. 3

Eine Möglichkeit zur Visualisierung der Wissensverbreitung und des Kompetenzaustausches bietet die *Organisational Network Analysis* (kurz ONA). Die Abbildung 1 stellt ein solches Netzwerk für das Scrum Framework dar. Um nicht jedes Teammitglied einzeln zu visualisieren werden stattdessen Rollen verwendet. Es ist zu erkennen, dass die kurzen Wege eine direkte Kommunikation ermöglichen, die bei einer Visualisierung von hierarchischen Strukturen mit den entsprechenden Rollen nicht möglich ist.

Im Kontext der global verteilten Entwicklung sind selbst die Wege mit Scrum nicht mehr optimal. Die körperliche Entfernung verhindert die sonst so vorteilhaften Austauschmöglichkeiten. Hierdurch ergibt sich eine direkte Komplexitätssteigerung der Projektdurchführung. Eine Reduzierung dieser Komplexität versucht die Standardisierung und Routinierung zur Koordinierung von Abhängigkeiten zu erreichen. Hierzu zählen die Orts-, Personen-, Zusammenarbeits- und Artifizielle-Abhängigkeiten. Die größte Herausforderung stellt die Standardisierung von artifizialen Abhängigkeiten dar [6]. Hierunter wurde im konkreten Fall die technische Infrastruktur verstanden. Diese beinhaltet unter anderem die Code-, Versionskontroll-, *Continuous Integration*-, Test- und Tracking-Systeme.

Die erfolgreiche Integration von agilen Praktiken und Methoden innerhalb der Softwareentwicklung hängt unabhängig von der Verteilung des Teams zudem stark von Unternehmensstrukturen ab. Insbesondere stellen sich Unternehmen mit einer klassisch hierarchischen Unternehmensstruktur besonderen Herausforderungen gegenüber. Diese beinhaltet die traditionelle IT-Manager-Sicht, in der die Produktivität, Effizienz und das Erreichen der Ziele zum einen sichergestellt werden soll und dabei in Reihenfolge und routiniert stattfindet. Das strenge Einbeziehen der Unternehmenskultur soll zu einem verbesserten Verständnis einer erfolgreichen Integration von agilen Praktiken innerhalb der Softwareentwicklung dienen. Ebenso sollen hierdurch auch Hauptindizes in der Organisation bestimmt werden können [7].

Besonders die Schnittstelle vom agilen Vorgehen, zu nicht agilen Methoden des Unternehmens, schafft Frustrationen innerhalb des Teams. Darunter versteht sich zum einen die Wertschätzung von Leistungen der Softwareentwicklung, wie aber auch der Wechsel von agilem Vorgehen, in nicht agile Vorgehensweisen. Die Motivation der Teammitglieder beeinflusst stark die Effizienz des Teams. Im Kontext von nicht agilen Unternehmensstrukturen führt die Adaption des Entwicklungsprozesses zu einem Motivationsabbau [8]. Hiermit ist gemeint, dass die von der Unternehmensführung vorgegebenen Planungselemente für Projekte zu verwenden sind, die z.B. eine Kosten- und Leistungsdefinition zu Projektbeginn beinhalten. Der Motivationsabbau lässt sich zudem durch den Austausch der Arbeitsumgebung von agil geführten Projekten zu herkömmlich strukturierten Projekten noch weiter steigern [8].

Die Betrachtung der Motivation innerhalb des Teams sollte somit auch von Scrum Mastern überwacht werden, da sich die Motivation als Hindernis für den erfolgreichen Projektverlauf entwickeln kann. Innerhalb des Scrum Frameworks ist eine der wichtigsten Aufgaben des Scrum Masters die Beseitigung solcher Impediments. Im Allgemeinen dient der Scrum Master aber vor allem als Vermittler, von und zur Softwareentwick-

lung. Der Grad der Komplexität solcher Aufgaben ist dabei nicht zu unterschätzen. Kategorisieren lassen sich die Aufgaben im Kontext der global verteilten Entwicklung mit Scrum nach einer Studie in sechs Aktivitäten [9]. Die Tabelle II stellt diese Aktivitäten dar.

| Aktivität | Beschreibung |
|---------------------------|--|
| Prozess-Moderator | Einhaltung der agilen Methoden |
| Standup-Vermittler | Die Einhaltung, dass Teammitglieder den Status teilen und Impediments melden |
| Sprint Planer | Unterstützung bei der Vorselektierung der User Stories und der Arbeitslast-Planung |
| Scrum of Scrum Vermittler | Die koordinierte Zusammenarbeit mit anderen Scrum Mastern |
| Integrations-Moderator | Erleichterung der Programmcode-Zusammenführung für parallel arbeitende Entwicklungsteams |

Tabelle II. Scrum Master Aktivitäten nach [9] S. 14

In der Funktion als Prozess-Moderator ist der Scrum Master vor allem auch als Schnittstellenkoordinator zu verstehen. Dabei fällt auch die Koordination der parallel zur den agilen Methoden geführten plan-basierten Methoden in großen Unternehmen in den Aufgabenbereich. Insbesondere sind *Business-Stakeholder* entsprechend des agilen Frameworks zu schulen.

Teams und Organisationen im agilen Kontext haben Schwierigkeiten, wie Impediments zu managen und messen sind. Darüber hinaus stellt die Priorisierung und Darstellung der Auswirkungen eine besondere Herausforderung dar. Eine Möglichkeit, die Impediments besser zu untersuchen, bietet die Technik der *Impediment Impact Digrams* [10]. Inhalt dieses Verfahrens ist zum einen die Priorisierung von Impediments. Welches Impediment sollte im Backlog als Erstes beseitigt werden und wer muss dabei mit einbezogen werden. Zum anderen kann diese Technik dabei behilflich sein, die relativen Kosten und die relative Dauer zu bestimmen, die für eine Beseitigung der Impediments nötig wäre. Insgesamt kann somit ein verbessertes Kapazitätsmanagement erfolgen und die Kapazität an den effizientesten Stellen eingeplant werden. Ein solches Werkzeug kann einem Scrum Master einen wesentlichen Mehrwert bieten.

III. DIE RELEVANZ DER SCRUM MASTER ROLLE

Der Scrum Master hat eine Vielzahl von Aufgaben zu bewältigen. Eine der Hauptaufgaben ist dabei die Beseitigung von auftauchenden Hindernissen [11]. Als Impediment wird dabei jedes Hindernis gewertet, welches das Team bei der Arbeit behindert oder blockiert [12]. Somit gilt unter anderem, dass der Scrum Master optimale Arbeitsbedingungen für das gesamte Entwicklungsteam anstrebt[13]. Dabei ist es besonders wichtig die Transparenz vorhandener Probleme zu gewährleisten[13], wodurch Probleme erst sichtbar werden. Im Zuge dessen ist die kontinuierliche Abarbeitung bzw. Beseitigung der Hindernisse in dem Team ebenso deutlich zu machen. Die Wichtigste Aufgabe des Scrum Masters ist somit die Auflösung der Impediments. Die Aufgabe des Teams ist dabei an diesen mit den entsprechenden Problemen heranzutreten[13]. Diese werden dann in einem *Impediment Backlog* gepflegt und darüber auch abgearbeitet[13][12]. Inhalt

dieses Backlogs können z.B sein:

- 1) Es steht kein *Product Owner* (PO) zur Verfügung.
- 2) Das Team benötigt mehr Tester.
- 3) Der Arbeitsraum ist zu laut.
- 4) Es werden bessere Pair-Programming-Arbeitsplätze benötigt.
- 5) Der PO hat keine Entscheidungskompetenz.
- 6) Es bestehen persönliche Konflikte zwischen den Teammitgliedern.
- 7) Das Füllen von Entscheidungen als Impediment.

Der letzte Punkt in der Aufzählung stellt eine Besonderheit dar. Generell trifft der Scrum Master keine Entscheidungen für das Team. Das Team entscheidet selbst über das ein oder andere Problem. Die einzige Ausnahme ist, falls das Team nicht in der Lage ist eine Entscheidung zu fällen. Dann gilt die Entscheidung selbst als Hindernis für den Entwicklungsprozess [13]. In diesem Fall gilt, dass der Scrum Master eine Diskussion zur Problemlösung einleitet, damit ggf. keine weiteren unnötigen Diskussionen geführt werden. Im äußersten Fall muss dieser sogar die Entscheidung an eine weitere Person delegieren oder zu allerletzt selbst fällen.

Eine weitere wichtige Aufgabe ist das Umsetzen und Prüfen der erforderlichen Rahmenbedingungen zum optimalen Ablauf des Entwicklungsprozesses nach Scrum. Hierzu gehört von Anfang an die generelle Einführung von Scrum im Team über beispielsweise Workshops [13], um die notwendigen Regeln, Prinzipien, Abläufe und die Kultur zu vermitteln. Insbesondere gehört dazu auch, über das Scrum Team hinaus, ein Verständnis für Scrum zu schaffen [11]. Innerhalb dieses Prozesses können die verschiedensten Stakeholder involviert sein, die sich innerhalb der Organisation nicht immer in einer direkten Relation zum Team befinden müssen. Es ist ebenso wichtig den Kunden das Scrum Framework näher zu bringen, wie aber auch das Coachen des PO (früher Projektleiter) [13]. Weiterführend muss eine kontinuierliche Prüfung durchgeführt werden, ob Scrum mit allen Regeln eingehalten wird [11].

Die Rolle beinhaltet auch die Identifizierung von Adaptionproblematiken bei Betrachtung der Schnittstelle vom Scrum Framework zum Unternehmen. Der Scrum Master stellt die Differenzen innerhalb der Projektführung fest und muss dem Unternehmen diese mitteilen. Abhängig vom Unternehmen und der Entwicklung ist die Auflösung dieser Problematik nicht zwingend möglich, sollte aber trotzdem erfasst werden [8].

Ebenfalls gehört zum Aufgabenbereich das Vorbereiten und moderieren aller Meetings [13]. Hierzu zählen, das *Sprint Planning*, das *Daily Stand Up* (ob sich beispielsweise die Tickets im korrekten Status befinden [13]), das *Sprint Review* und die *Sprint Retrospective*.

Gegenüber dem Product Owner muss der Scrum Master das Team schützen [13]. Darunter versteht man, dass die Anforderungen vom Product Owner in einer geregelten Form in den Entwicklungsprozess überführt werden, sodass die Tickets nicht in einen laufenden Sprint integriert werden, oder eine Überlastung des Teams durch zu viele in einem Sprint umzusetzenden Anforderungen stattfindet. Er trägt somit auch eine Mitverantwortung für den Erfolg und die Produktivität des Entwicklungsteams [13]. Entsprechend muss der Scrum Master das Team abschirmen und ggf. komplett isolieren, sodass der

Erfolg und eine angemessene Produktivität gewährleistet werden kann. Die Abbildung 2 skizziert diese Tätigkeit gegenüber den verschiedenen möglichen Stakeholdern.

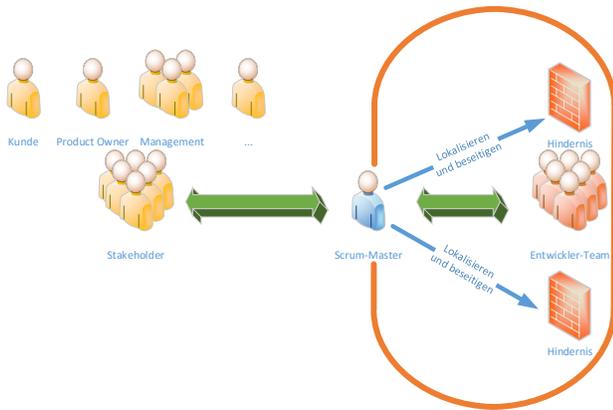


Abbildung 2. Abstrakte Darstellung eines Scrum Masters im Team

Im Fall einer mehrstufigen Scrum-Hierarchie muss er ebenfalls die Rolle des Teamsprechers, Coaches und der Kontaktperson zu verschiedenen anderen Teildaten besetzen, welche auch die dafür stattfindenden Meetingteilnahmen koordiniert [14].

Im klassischen Berufsbild lässt sich der Scrum Master deshalb auch als Rolle des IT-Koordinators generalisieren. Als Aufgaben dieser Rolle verstehen sich die Filterung des Informationsflusses mit entsprechender Aufgabenverteilung an die IT-Abteilungen, wie auch die Priorität und das Interesse, IT-Abteilung gegenüber höher gestellten Abteilungen zu vertreten [15].

Als eine nicht zu unterschätzende Aufgabe, bei der Informationskoordination, gilt die *Awareness* des Projektes. Insbesondere zählt hierzu die Kommunikation mit dem mittleren und oberen Management [13]. Der Scrum Master muss dafür sorgen, dass diesen Ebenen ein gefilterter, kontinuierlicher Projektstatus bereitgestellt wird. Er bildet somit eine Brücke vom Softwareprojekt ins Management. Insbesondere ist dabei eine geeignete Form der Informationen notwendig, die sich z.B. grafisch in Form eines *Gantt-Charts* darstellen lässt.

Die Kommunikationstransparenz ist insgesamt für das komplette Projekt notwendig. Als Aufgabe hierzu dient die Beseitigung der Defizite von indirekten Kommunikationsmitteln wie E-Mails, Blogs und *Wikis* [14]. So ist das regelmäßig stattfindende Treffen der Gemeinschaft aller Scrum Master relevant [14], um auch ein gemeinsames Verständnis der Probleme zu gewährleisten und eine gemeinsame Projektkultur zu entwickeln [14]. In Bezug auf die indirekte Kommunikation sind aber ebenfalls, unter Berücksichtigung der Vor- und Nachteile, die Einführung solcher Systeme wie Wikis und E-Mail Listen notwendig [14]. Im Projektverlauf steht der Scrum Master somit den verschiedensten Problemstellungen gegenüber, die sich nicht nur auf das Entwicklungsteam an sich beziehen und einen erhöhten Koordinationsaufwand verlangen.

IV. HERAUSFORDERUNGEN UND PROBLEMSTELLUNGEN

Als besondere Problemstellung stellt sich das Verständnis von Scrum zwischen der Entwicklung und dem IT-Management heraus [7][9]. Dabei wird die Agilität oft als Haupteigenschaft verstanden, wodurch weitere Fehlinterpretationen stattfinden können. Bei der Einführung und Etablierung agiler Entwicklungsmethoden ist somit besonders die Integration und Entwicklung der Unternehmenskultur zu berücksichtigen. Ebenfalls ist auf die Konfrontation von planbasierten Methoden in den Unternehmen zu agilen Methoden zu achten [9]. Demnach gibt es besonders bei der Betrachtung von Zeit, Kosten und Qualität innerhalb der Projektabwicklung deutliche Unterschiede zum traditionellen Projektmanagement. Hierzu gehören z.B. die Bestimmung von vordefinierten Zielen zu Milestones und die Budgetierung des Projektes, die sich im klassischen Projektmanagement wiederfinden [8].

Personen in strukturellen Positionen, die mit der agilen Welt in einer direkten Verbindung stehen, beeinflussen die Entscheidungsfindung oft direkt. Hierdurch können sich kritische Architektur und Design Entscheidungen ergeben, die wiederum den Erfolg des Projektes gefährden [7].

Vor der Akzeptanz von agiler Softwareentwicklung, wurde z.B. die Softwareentwicklung innerhalb einer Studie wie folgt charakterisiert:

- 1) Design Prozess
 - a) bedacht und formal
 - b) Hintereinanderreihung von Schritten
- 2) Problem-Lösung
 - a) Auswahl des besten Weges durch gut geplante und formalisierte Aktivitäten
- 3) Lernvoraussetzung
 - a) Einzelne Iteration/Adaption
- 4) Theoretische Wurzeln
 - a) Logischer Positivismus & wissenschaftliche Methode
- 5) Merkmale
 - a) Kontrolle & Leitung
 - b) Der Manager ist der Koordinator
 - c) Design-geprägtes Implementationsvorgehen
 - d) weniger dialektisch

Charakterisierung nach [7] S. 44

Aber auch nach der generellen Akzeptanz von agiler Softwareentwicklung muss weiterhin die *Awareness* gegenüber den Stakeholdern verbessert werden und kontinuierlich stattfinden.

Beispielsweise erwarten Organisationen auch nach der erfolgreichen Einführung von agilen Methoden, dass das Projektmanagement genügend Druck gegenüber der Entwicklung aufbringen muss, da nach ihrem Verständnis sonst keine kontinuierliche Softwareentwicklung möglich ist. Es hat sich in Studien allerdings gezeigt, dass dieses Vorgehen und dem daraus resultierenden Scheitern der Projekte zudem das agile Vorgehen bereits im Entwicklungsprozess der Scrum-Einführung beeinflusst [7].

Ebenfalls beeinflusst die Team-Motivation den produktiven Projektverlauf maßgeblich. Insbesondere ist die Adaption von Standard-IT-Prozessen in Großunternehmen zu agilen Prozessen ein Faktor. Milestones, die vor dem Projektstart

definiert werden, können nur sehr schlecht mit dem jeweiligen Sprint Ende und den daraus resultierenden Produktprototypen synchronisiert werden. Hierdurch ist das Feedback beim Erreichen eines Milestones vorhanden, wo nur der letzte Prototyp präsentiert werden kann, welcher aber nicht die aktuell bewältigten Herausforderungen widerspiegelt. Wiederum findet ein wirkliches Sprintende kein Gehör [8].

Bezüglich der Teammitglieder ist das Feedback zum Stakeholder essentiell, welches durch hierarchische Strukturen verhindert wird. Ein Wissenstransfer zum Kunden, wie aber auch ein Kompetenzaustausch zwischen den Teams, wird hierdurch deutlich erschwert. Diese beiden Punkte sind aber besonders bei Scrum einer der Produktivitäts-Träger, die sich langfristig auf das Team auswirken [5].

Jeder der genannten Herausforderungen stellt Impediments innerhalb des Projektverlaufs dar, welche es zu erfassen gilt. Direkt durch das Team sichtbare Impediments werden dabei häufig in Standupmeetings genannt. Allerdings fehlt in vielen Fällen die kontinuierliche Abfrage vom Status des Impediments und unter anderem auch die Anforderungen die Impediments zu schließen, sofern eines beseitigt wurde [7]. Impediments werden somit häufig nur auf Zuruf erfasst, sind aber generell nicht unbedingt ein fester Bestandteil der Standupmeetings.

Sind die Hindernisse erstmalig erfasst, so gilt es diese zu priorisieren. Hierzu ist es notwendig, dass die möglichen Auswirkungen betrachtet werden. Die Messung dieser Auswirkung stellt dabei keine triviale Aufgabe dar. Ebenso ist es wichtig darzustellen, wie viel Aufwand bezüglich Kosten und Dauer investiert werden muss, um die entsprechenden Impediments zu beseitigen [10].

V. ERFASSUNG VON HINDERNISSEN ÜBER BESTEHENDE INFORMATIONSSYSTEME

Eine Möglichkeit der automatischen Erfassung von Impediments für den Scrum Master sind bestehende Informationssysteme. Folgend werden Vorschläge skizziert, die einem Scrum Master als Unterstützung dienen kann, um eine Verbesserung des Entwicklungsprozesses nach Scrum verfolgen zu bieten. Diese Ideen wurde in Anlehnung an die typischen Aufgaben und ermittelten Impediments von Scrum Mastern aus den genutzten Quellen entwickelt.

Die hierfür benötigten Informationen lassen sich z.B. aus einem zur Verfügung stehenden Ticket und Projektmonitoring-System entnehmen. Hierzu wären, abhängig vom zur Verfügung stehenden System, die Entwicklung einer Erweiterung notwendig. Dabei kann es sich entweder um eine separate Softwarelösung (eigener Dienst) oder um eine Erweiterung der bestehenden Lösung handeln. Ebenso können eventuell zur Verfügung stehende Funktionen innerhalb der betriebenen Anwendungen diese Informationen bereits zur Verfügung stellen.

Status-Wechsel - Der häufige Wechsel von Tickets kann ein Indiz dafür sein, dass ein nicht optimaler Workflow innerhalb des Ticket-Systems verfolgt wird. Dies hat zur Folge, dass zur Verfügung stehende Berichte ggf. verfälscht werden können. Ebenso kann dieser Punkt angeben, dass ein Ticket häufig bearbeitet wurde, aber nicht zum Abschluss kommt. Die Realisierung eines solchen Indiz kann z.B. wie folgt aussehen:

- 1) Das Zählen, wie häufig Tickets insgesamt einen alten Status passieren.
- 2) Das Aufnehmen der Verweildauer, wodurch sich ergibt, wie lange sich ein Ticket in einem Status aufhält. Hierdurch kann dann ausgesagt werden, dass es einen Status gibt, in dem ein Ticket niemals länger als z.B. 1 Minute verweilt. Dieses Ticket würde somit in gewisser Weise immer einen Status überspringen.

Ticket Zuweisung - Ein Ticket wird generell einer Person im System zugewiesen und kann je nach Workflow oder Entwicklungsstand an andere Personen im Team zugewiesen werden. Durch das Zählen der Zuweisungen (wie oft wird das Ticket jemanden zugewiesen?) kann herausgefunden werden, ob z.B. ein Ticket generell von einzelnen Personen nie oder selten abgeschlossen wird, ob einzelne Personen generell Tickets immer abschließen oder ob ein Ticket zwischen Personen hin und her gereicht wird. Der errechnete Wert ergibt dabei nur im Verhältnis zu einem vorher definierten Standardwert einen wirklichen Indizwert auf eine mögliche Störung oder ein mögliches Risiko bei Teammitgliedern.

Push und Pull - Hinter diesem Punkt verbirgt sich die angestrebte Kultur, wie mit Aufgaben im Team generell umgegangen wird. Nach Scrum strebt das Team eine Pull-Mentalität an. Das bedeutet, dass Aufgaben niemanden direkt aufgeladen werden, sondern dass sich das Team selbstständig mit Aufgaben versorgt. Mit dem Messen der Zuweisungen an Teammitgliedern kann ein Verstoß dieser angestrebten Kultur identifiziert werden.

Zuweisungs-Verteilung - Mit der Zählung von aktiven Tickets pro Teammitglied lässt sich aussagen, ob Personen einen Großteil der Entwicklungsaufgaben abarbeiten müssen. Hierdurch leitet sich ab, ob eine eventuelle Überlastung einzelner Personen vorliegt.

Abarbeitungen der Tickets - Die Abarbeitung der Tickets lässt sich bezogen auf einen Tag bereits über das Burndown-Chart darstellen, sofern dieses verwendet wird. Ebenso kann die generelle Tages-Performance hierdurch statistisch gemessen werden. Der Vorteil, welcher sich hieraus ergibt, ist z.B., dass sich der Erfolg einer Impediments-Beseitigung in ein Leistungsverhältnis setzen lässt.

Abarbeitungen gegen das Burndown Chart - Hiermit ist gemeint, dass die Abweichung der abgearbeiteten Tickets besonders auch in der Historie auf eine Verschlechterung und Verbesserung der Abschätzung von Tickets schließen lässt. Ebenso können Tickets herausgefiltert werden, die generell immer wieder geschoben werden, weil Storypoint Werte zu niedrig sind. Diese haben im Burndown Chart keine oder nur eine sehr geringe positive oder negative Auswirkung und sollten ggf. im Business-Value neu bewertet werden, um die Relevanz im Bezug auf das Ziel neu zu definieren.

Kommentar Faktor - Viele Ticket-Systeme bieten im Rahmen ihrer Existenz die Möglichkeit von Kommentaren zu. Die Entwickler haben somit die Chance, eine Diskussion über ein Ticket zu führen, um die Abarbeitung voranzutreiben. Die Messung von Kommentaren bietet die Filterung von Tickets, über die eventuell endlos diskutiert wird. Diese Tickets können somit ein Indiz dafür sein, dass es ein Problem im Team gibt oder, dass ein größeres Hindernis innerhalb der Entwicklung selbst vorliegt.

Abarbeitungsdauer von Tickets - Über die Messung der Verweildauer von Tickets können verschiedene Rückschlüsse gezogen werden, die bereits auch in anderen Punkte skizziert wurden. Das Verweilen in einem Status zeigt insbesondere *Zombie-Tickets* auf, die entsprechend diskutiert werden müssen.

Um die aktuelle Verwendung, Relevanz und Realisierungsmöglichkeiten in der Praxis zu betrachten, müssen weitere Informationen aus bestehenden, aktiven Scrum Teams bezogen werden.

VI. EMPIRISCHE ERHEBUNG

Zur Erfassung vor praxisnahen Problemstellungen bieten sich besonders Untersuchungen in Unternehmen an. Hierzu soll im späteren Projektverlauf eine Befragung über Einzelinterviews von Scrum Mastern erfolgen.

Folgend sollen zunächst Probleme dargestellt werden, die aus der aktuellen und klassischen Literatur bezogen wurden.

- 1) Verständnis für agile Softwareentwicklung gegenüber dem Unternehmen.
- 2) Kontinuierliche Awareness für agile Softwareentwicklung im Unternehmen.
- 3) Das klassische IT-Management an der Schnittstelle zur Softwareentwicklung [7][8].
- 4) Impediments Verfolgung - Status und Abarbeitung [7].
- 5) Die Priorisierung gemäß der möglichen Auswirkungen und die gezielte Abarbeitung von Impediments [10]
- 6) Meeting-Effizienz [9].
- 7) Wissensverbreitung [5].
- 8) Kompetenzaustausch [5].

Durch eine mögliche Kategorisierung kann dargestellt werden, wie man Probleme gruppieren kann, um besondere Charakteristiken der Probleme deutlich zu machen. Hierzu dient die Tabelle III.

| Kategorie | Definition |
|-------------------------------|--|
| Extra Features | Extra Features sind Features, welche hinzugekommen sind, ohne dass ein nachweislicher Bedarf oder eine valide Hypothese existiert. |
| Verspätung | Eine Verspätung ist eine Situation, in der etwas später passiert, als es geplant war. Dies impliziert ein Zurückhalten, beeinflusst durch Behinderungen oder Einsätze. |
| Übergaben | Tritt immer auf, wenn eine nicht fertiggestellte Arbeit von einer anderen Gruppe oder Person übernommen werden muss. |
| Fehleranfrage | Fehleranfragen beziehen sich auf Anfragen bezüglich des Systems (inkl. Team und Organisation) und sind Anfragen, aufgrund eines Fehlers etwas zu tun oder etwas richtig für den Kunden zu tun. |
| Aktive Arbeitsdurchführung | Analogien im Softwareentwicklungsbestand. Arbeit, die bisher nicht abgeschlossen ist und hierdurch keinen Wert gegenüber dem Business oder Kunden darstellt. |
| Kontextwechsel | Ein Kontextwechsel findet statt, wenn bei Personen oder Teams ein Wechsel ihrer Aufmerksamkeit zwischen Aktivitäten gleichzeitig stattfindet. |
| Unnötige Bewegung | Eine unnötige Bewegung findet bei Personen, Arbeit oder Wissen statt, welche unnötig ist, den optimalen Arbeitsfluss behindert und eine zusätzliche Ineffizienz schafft. |
| Extra Prozesse | Extra Prozesse generieren extra Arbeit, welche wiederum Zeit und Aufwand generieren, ohne das ein Mehrwert geschaffen wird. |
| Ungenutztes Personalpotential | Darunter versteht sich das ungenutzte Potential von Personen oder zu fördernde Fähigkeiten von Personen, um das volle Potential ihrer Begabungen nutzen zu können. |

Tabelle III. Hindernis Kategorisierung nach [10] S. 41

A. Ziel

Das Ziel der Interviews wird wie folgt beschrieben. Zum einen soll es möglich sein, dass Probleme und Herausforderungen eingestuft werden können. Darunter ist zu verstehen, welche Kategorie die häufigsten Probleme bzw. Hindernisse hält. Zum anderen soll es möglich sein, dass eine Generalisierung der Probleme ermöglicht wird, die sich unabhängig von der Unternehmens- und Projektgröße darstellen lässt. Ggf. lassen sich bestimmte Probleme auch auf die Unternehmenskategorisierung referenzieren. Die Erfassung des Interviewleitfadens wird entsprechend die Fragen, wie auch die Essenz der Befragungen kennzeichnen und darüber hinaus auch den generellen Ablauf widerspiegeln.

B. Entwicklung eines Interviewleitfadens

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

Der Ablauf des Interviews ist wie folgt gegliedert:

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

Diese Gliederung soll es ermöglichen, die Antworten besser zu kategorisieren und auswerten zu können. Zudem soll hierdurch ein verbesserter Ablauf des Interviews ermöglicht werden.

1) Befragung zur Kontextbildung des Scrum Masters:

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

- Wie viele Projekte haben Sie bisher als Scrum Master betreut?
- Wie viele Jahre haben Sie als Scrum Master Erfahrungen sammeln können?
- Besitzen oder besitzen Sie während Ihrer Tätigkeit als Scrum Master parallel eine weitere Rolle im Unternehmen? Wenn ja welche?
- Besitzen Sie Programmiererfahrung? Wenn ja, wie viele Jahre?
- Über welchen Zeitraum erstreckten sich Ihre letzten drei Projekte, indem Sie als Scrum Master tätig waren?
- Betreuen Sie generell mehrere Projekte parallel, wenn ja wie viele im Durchschnitt?
- Besitzen Sie Erfahrungen bei der Teilnahme und Führung von Scrum of Scrum Meetings?
- Wie ist die durchschnittliche Anzahl an Personen (gruppiert nach Rolle) im Projekt?
- Wie sind die Art der Projekte Ihrer Meinung nach zu beschreiben?
- In welche Größenklasse lässt sich Ihr Unternehmen eingliedern?

- Würden Sie Ihr Unternehmen generell als "hierarchiegetriebenes" Unternehmen bezeichnen?
- Um was für eine Art von Unternehmen handelt es sich?

2) *Befragung bzgl. Hindernissen in Projekten:* Um die häufigsten Hindernisse in einem Unternehmen zu erfassen, müssen die folgenden Fragen gestellt werden. Hierdurch wird es auch möglich sein, dass die Hindernisse bzgl. Auswirkungen (engl. *Impact*), Beseitigungsaufwand und Priorisierung gruppiert werden können.

- Was sind Ihre häufigsten Hindernisse mit einem großen Impact in Projekten, bezogen auf das Team?
- Was sind Ihre häufigsten Hindernisse, bezogen auf das Team, die einen großen Beseitigungsaufwand besitzen?
- Was sind Ihre häufigsten Hindernisse, bezogen auf das Team, die Sie besonders hoch priorisieren?
- Was sind Ihre häufigsten Hindernisse mit einem großen Impact in Projekten, bezogen auf ihre Tätigkeit als Scrum Master?
- Was sind Ihre häufigsten Hindernisse, bezogen auf ihre Tätigkeit als Scrum Master, die einen großen Beseitigungsaufwand besitzen?
- Was sind Ihre häufigsten Hindernisse, bezogen auf ihre Tätigkeit als Scrum Master, die Sie besonders hoch priorisieren?
- Wann werden Hindernisse bei Ihnen erfasst?
- Wie werden Hindernisse bei Ihnen erfasst?
- Wie werden Hindernisse bei Ihnen verfolgt und archiviert?

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

- Welche Tools unterstützen Sie in Ihrer Tätigkeit als Scrum Master?
- Wie würden Sie die eingesetzten Tools im Verhältnis priorisieren?
- Dient eines der verwendeten Tools zur Erhebung der Impediments? Wenn ja, welche Art von Hindernissen werden dort erfasst?
- Was würde Ihrer Meinung nach eine zusätzliche Hilfestellung für die Arbeit des Scrum Masters bieten?

VII. FAZIT

Diese Arbeit hat die aktuelle Wissensbasis für die Rolle des Scrum Masters sowie den zugehörigen Kontext dargestellt, wodurch ersichtlich ist, dass der Scrum Master mehreren Herausforderungen gegenüber steht. Die Relevanz der Rolle, innerhalb der agilen Softwareentwicklung nach Scrum, wurde zudem schriftlich festgehalten. Auch wurden darauf aufbauend die besonderen Herausforderungen und Problemstellungen verdeutlicht, nachdem die Beseitigung, wie aber auch die Erfassung der Hindernisse, sich als nicht triviale Herausforderung herausstellt. Bestehende Informationssysteme

können beim Erfassungs- und Evaluierungsprozess von Hindernissen hinzugezogen werden. Um die Relevanz dieser erfassten Hilfestellungen, wie auch die generelle Lage von Scrum Master im aktiven Entwicklungsprozess zu bieten, wurde der Ablauf eines Interviews definiert. In diesem wurden mehrere Fragen entwickelt, wodurch eine Erfassung und Eingliederung der Ergebnisse ermöglicht wird.

Der nächste Schritt wird sein, dass die beschriebene Empirische Erhebung bei einer Vielzahl von Unternehmen durchgeführt wird. Die Eingliederung der Ergebnisse wird erwartete Prognosen bestätigen oder widerlegen. Ebenso werden neue Erkenntnisse gewonnen, wodurch sich neue Aussagen von aktuellen Herausforderungen und Problemstellungen der Scrum Master treffen lassen. Darauf aufbauend lässt sich dann aussagen, inwieweit eine Unterstützungsmaßnahme für die Arbeiten des Scrum Master aussehen kann.

Die Aktivitäten der nächsten Arbeit wären somit die direkte Erhebung von Anforderungen, die sich in einem möglichen Prototypen widerspiegeln können. Hierzu kann dann ein System skizziert werden, wodurch eine Verbesserung der Scrum Master Aktivitäten gegeben wäre.

Als weitere Unterstützungsmaßnahme kann die im Zuge dieser Arbeit vorgestellte Organisational Network Analysis (ONA) [5] dienen. Hierdurch wird es ermöglicht, dass bestehende Strukturen der Entwicklungsteams besser analysiert werden können. Ebenso wäre es interessant, ob ein Impediment Impact Diagram [10] einen deutlichen Mehrgewinn innerhalb eines Workshops bieten kann. Eine Fortsetzung würde sich nach einem positiven Verlauf des Workshops ergeben. In dieser kann die Fragestellung geklärt werden, inwieweit sich das Impediment Impact Diagram in den aktiven Scrumablauf integrieren lässt und ob dies sinnvoll erscheint.

Über die Evaluierung der weiteren Unterstützungsmaßnahmen, wie auch der skizzierten Softwarelösung die sich aus den erfassten Interviewergebnissen darstellen lässt, ist es somit möglich, dass essentielle Informationen bezogen werden, die zum Erfolg der agilen Softwareentwicklung und somit auch von Softwareprojekten beitragen.

REFERENZEN

- [1] I. Barkalov, J. Martin, S. Mume, M. Metz, P. Wagner, and K. Jiang, *Forward Visibility Studie (2. Auflage)*. Capgemini Consulting, 2013, abruf: 2015-11-16. [Online]. Available: <https://www.de.capgemini-consulting.com/resource-file-access/resource/pdf/forward-visibility-edition-2.pdf>
- [2] VersionOne, *9th Annual State of Agile Survey*. VersionOne, 2015, abruf: 2015-11-16. [Online]. Available: <https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>
- [3] TheStandishGroup, *The Standish Group Report CHAOS*. The Standish Group, 2014, abruf: 2015-11-16. [Online]. Available: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
- [4] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, "Empirical study of agile software development methodologies: A comparative analysis," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 1, pp. 1–6, Feb. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2693208.2693233>
- [5] A. Becheru, "Agile development methods through the eyes of organisational network analysis," in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, ser. WIMS '14. New York, NY, USA: ACM, 2014, pp. 53:1–53:8. [Online]. Available: <http://doi.acm.org/10.1145/2611040.2611101>
- [6] M. Esbensen and P. Bjørn, "Routine and standardization in global software development," in *Proceedings of the 18th International Conference on Supporting Group Work*, ser. GROUP '14. New

- York, NY, USA: ACM, 2014, pp. 12–23. [Online]. Available: <http://doi.acm.org/10.1145/2660398.2660413>
- [7] L. Kompella, “Agile methods, organizational culture and agility: Some insights,” in *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE 2014. New York, NY, USA: ACM, 2014, pp. 40–47. [Online]. Available: <http://doi.acm.org/10.1145/2593702.2593708>
 - [8] L. Gren, R. Torkar, and R. Feldt, “Work motivational challenges regarding the interface between agile teams and a non-agile surrounding organization: A case study,” in *Agile Conference (AGILE), 2014*, July 2014, pp. 11–15.
 - [9] J. Bass, “Scrum master activities: Process tailoring in large enterprise projects,” in *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*, Aug 2014, pp. 6–15.
 - [10] K. Power, “Impediment impact diagrams: Understanding the impact of impediments in agile teams and organizations,” in *Agile Conference (AGILE), 2014*, July 2014, pp. 41–51.
 - [11] *Requirements-Engineering und -Management : aus der Praxis von klassisch bis agil*. München: Hanser, 2014.
 - [12] J. Goll and D. Hommel, “Das agile rahmenwerk scrum,” in *Mit Scrum zum gewünschten System*. Springer Fachmedien Wiesbaden, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-658-10721-5_4
 - [13] R. Wirdemann, *Scrum mit User Stories*. München [u.a.: Hanser, 2011, iD: 846279372.
 - [14] J. Eckstein, *Agile Softwareentwicklung in großen Projekten Teams, Prozesse und Technologien - Strategien für den Wandel im Unternehmen*, 2nd ed. Heidelberg: dpunkt.verlag GmbH, 2012.
 - [15] L. Heinrich, A. Heinzl, and F. Roithmayr, *Wirtschaftsinformatik-Lexikon*. De Gruyter, 2004. [Online]. Available: <https://books.google.de/books?id=Uq4FCgAAQBAJ>