

Stakeholder-Awareness

Was ist hier los?

Alexander Kaack

Grundseminar - WiSe 2014/2015
Master Informatik
Department Informatik
Hochschule für Angewandte Wissenschaften Hamburg
Email: Alexander.Kaack@haw-hamburg.de

I. EINFÜHRUNG

Eine zu erkennende, zunehmende Marktdynamik sorgt für eine steigende Planungskomplexität in Projekten [1]. Innerhalb der Softwareentwicklung haben besonders agile Vorgehensmodelle zur Reduzierung dieser Komplexität beigetragen. Nach einer weltweiten Studie nutzen ca. 88 % der befragten Unternehmen agile Methoden. Dabei stehen *Scrum* und *Scrum* ähnliche Varianten an der Spitze mit 78 % [2]. *Scrum* versucht dabei der Komplexität, durch das Teilen von komplexen Sachverhalten in kleinere, beherrschbare Aufgaben, entgegen zu wirken [3].

Eine Möglichkeit, die sich durch das Zerteilen in kleinere Aufgaben ergibt, ist die Schätzung des zeitlichen Aufwands. Hierzu dienen unter anderem die Personenmonate (früher Mannmonate). Die Ableitung aus dieser Zuordnung ist die Anzahl an Monaten, die mit einer Person für das Projekt benötigt wird. Zum Beispiel werden für zwei Personenmonate zehn Personen veranschlagt. Eine problematische Betrachtungsweise, die hierdurch entsteht, ist der Schlüssel der teilnehmenden Personen. Es könnte assoziiert werden, dass eine Erhöhung des Personals um 100 % die errechneten Personenmonate um 50 % senken würde. Diese bedenkliche Sicht könnte in einem fortlaufenden Projekt, mit einem planungsabweichenden Ablauf, kritische Entscheidungen fällen lassen [4]. Eventuell könnte sogar eine entgegengesetzte Sichtweise eine erfolgversprechendere Auswirkung besitzen, indem eher Personal an den richtigen Stellen reduziert wird.

Dieses Entscheidungsmodell, welches vom Management gewählt wird, trägt einen besonderen Aspekt mit sich: Die vollständige Abschätzung des kompletten Projektverlaufs im Vorhinein. Diese Methode lässt sich wiederum auf das typische Vorgehen innerhalb des „Wasserfall-Modells“ abbilden. Man versucht die komplette Planung zum Anfang des Projektes zu tätigen.

Die Entscheidungen, die getroffen werden, fallen auf die sogenannten *Stakeholder* zurück. Das Hauptinteresse dieser lässt sich dabei auf drei Punkte reduzieren:

- 1) Das Projekt muss sich innerhalb des Budgets bewegen.
- 2) Das Projekt muss im Zeitplan bleiben.
- 3) Die geforderten Features werden umgesetzt (*Feature Complete*, sprich, wie viel der geforderten Features wurden umgesetzt).

Die Frage, die sich an dieser Stelle stellt ist:

„Wer sind die Stakeholder?“

Die Stakeholder sind zum Beispiel ein Endkunde, das Management oder andere Teams. Sie sind allgemein stark abhängig vom Softwareprojekt, denn grundsätzlich gilt, dass jeder, der ein berechtigtes Interesse am Verlauf des Projektes hat, ein Stakeholder in diesem Sinne ist. Bezogen auf die Art des Projekts ist allein die Definition des Kunden nicht immer trivial. Das Projekt könnte für einen Einzelkunden gelten, ein „Inhouse“ Projekt sein oder den Massenmarkt ansprechen. In allen diesen Fällen gilt dabei allerdings, dass das Management des Entwicklungsteams immer ein berechtigtes Interesse am Erfolg besitzt.

Besonders im Kontext der agilen Entwicklung mit *Scrum* steht die Schnittstelle zum Stakeholder im Fokus. Es wird versucht einen kontinuierlichen Kommunikations-Strom zu den Stakeholdern zu leisten. Dabei sind Qualität, beziehungsweise Feature-Stand, Zeit und Budget Fragen, auf die sich die meisten Anfragen reduzieren lassen. Es gilt dabei, ebenso die *Awareness* des gesamten Projekts, wie auch die des aktuell reduzierten Projektstatus zu vermitteln. Unter *Awareness* versteht man das Schaffen von Bewusstsein und über laufende Aktionen anderer Projektteilnehmer aufzuklären [5]. Es wird hiermit das Wahrnehmen und Verstehen angestrebt, um reagieren zu können.

Workspace *Awareness* [5] bietet zum Beispiel die Möglichkeit zu erfahren, wer sich im selben Arbeitsbereich (engl. *Workspace*) befindet, wer gerade woran arbeitet, was die anderen Personen gerade tun und was als Nächstes zu tun ist, beziehungsweise was welche Personen als nächstes tun werden. Das vorgenommene Ziel dabei ist eine erfolgreiche Zusammenarbeit. Zusätzlich soll der Koordinationsaufwand von Aufgaben reduziert und das Wechseln von Teilnehmern zwischen individueller und Gruppen-Arbeit erleichtert werden. Außerdem soll ein Kontext entstehen, innerhalb dessen Äußerungen anderer interpretiert und Aktivitäten anderer vorausgeahnt werden können.

Zwei Probleme, welche sich zudem aus dem agilen Kontext ergeben, sind die Wertschätzung der Entwicklungsteams und die Sichtbarkeit von Leistungen aus dem Team. Eine Aussage die hierzu getätigt werden kann ist, dass „gute Nerds unsichtbar sind“. Hinter dieser Aussage steckt die Erkenntnis, dass besonders Leistungsträger zum Teil unentdeckt bleiben können.

Die Komplexität dieser Probleme lässt sich zudem durch den Kontext der verteilten Entwicklung um ein Vielfaches steigern.

II. VERWANDTE ARBEITEN

Bezogen auf die vorgestellten Problematiken und Herausforderungen bei der Bewältigung des kontinuierlichen Informationsflusses und der Awareness zu den Stakeholdern, werden nachfolgend Arbeiten beschrieben, die sich mit angrenzenden Herausforderungen beschäftigt haben.

Wie bereits in der „Einführung“ erwähnt, ist die Thematik der Wertschätzung innerhalb der agilen Softwareentwicklung eine existierende Problemstellung. Die Arbeit „Hin zu mehr Sichtbarkeit und Wertschätzung in der Softwareentwicklung“ [6] hat sich mit dieser Problematik befasst. Flache Hierarchien und flexible projekt-orientierte Teams sind Faktoren die die Reduzierung der Wahrnehmung vor Folge haben. Die daraus resultierende Ableitung ist die Entstehung der „Awareness gaps“, welche insbesondere den Transfer von relevantem Wissen zwischen den Teams und die Bildung eines Gemeinschaftsgefühls beeinflussen. Letztendlich führen diese Punkte zu einer reduzierten Anerkennung von Wertschätzung der Leistungen in agilen Teams.

Zur Untersuchung und Verbesserung dieser Awareness ist die Applikation „Ambient Surface-System“ [6] entstanden, welche in der Abbildung 1 zu sehen ist. Diese Applikation ag-



Abbildung 1. Ambient Surface-System

gregiert Informationen aus dem bestehenden System *Jenkins*, welches für die kontinuierliche Integration (engl. continuous integration abgekürzt CI) von Code zur Verfügung steht, sowie dem *JIRA*-System¹, mit *JIRA Agile*², welches als Ticket- und agiles Projektmanagement-Tool verwendet wird. Durch dieses System ist es somit einfacher, wichtige Neuigkeiten und auch den Projektstatus aus den vorhandenen Systemen zu visualisieren und zu übermitteln. Die Applikation läuft auf einem Multi-Touch-Screen und steht sämtlichen Teammitgliedern zur Verfügung. Das Ergebnis dieses Feldversuches zeigt, dass eine tägliche Nutzung des Systems nachgewiesen ist. Insbesondere zählen hierzu die typischen Zeiten zum Arbeitsbeginn, zur Mittagspause und zum Feierabend. Es wurde insgesamt festgestellt, dass das System durch seine

häufige Nutzung akzeptiert wurde und auch die gewünschte Transparenz in den Projekten förderte.

Eine darauf aufbauende Debatte ist die eigentliche Informations-Visualisierung für agile Entwicklungs-Teams [9]. Das Ziel einer Arbeit [9] war es zu erkunden und zusammenzufassen, welche Visualisierungstechniken zur Darstellung von Informationen während der Design- und Entwicklungsphase im Softwarezyklus geeignet sind. Zudem sollte die Bedeutsamkeit vom Schaffen informativer Arbeitsplätze zum effektiven Wissenstransfer und der Kommunikation agiler Teams hervorgehoben werden.

Vorteile, die sich aus den Untersuchungen ergeben, sind die verbesserte Koordination der Aktivitäten, worüber ein direktes Feedback und Reagieren ermöglicht wird. Zudem bietet es die Möglichkeit, den Projektstatus besser gegenüber dem Kunden zu visualisieren und somit auch einen optimierten Überblick zu gewährleisten.

Wichtige Informations-Visualisierungen für agile Softwareteams bieten das aus Scrum resultierende Burndown-Chart, das zum Beispiel auch im *JIRA*-System (inklusive *JIRA Agile*) mit zur Verfügung steht. Es ermöglicht dem Team eine Darstellung des aktuellen Projektverlaufs zum geplanten Projektverlauf. In der Abbildung 2 ist einfach zu erkennen, wenn es eine Abweichung von der Planung gibt. Diese äußert sich über den Verlauf des grün eingezeichneten Graphen gegenüber der rot eingezeichneten Linie. Befindet dieser sich unter der roten Linie ist man unter dem geplanten Aufwand und somit schneller als geplant. Ist dieser über der roten Linie, so ist man hinter dem geplanten Aufwand. Die Herausforderung besteht somit insgesamt darin, den geplanten Aufwand exakt zu entsprechen, beziehungsweise sich an dieser Linie zu orientieren und eine möglichst geringe Abweichung unter der Linie zu besitzen. Man versucht sich somit immer dicht an dem

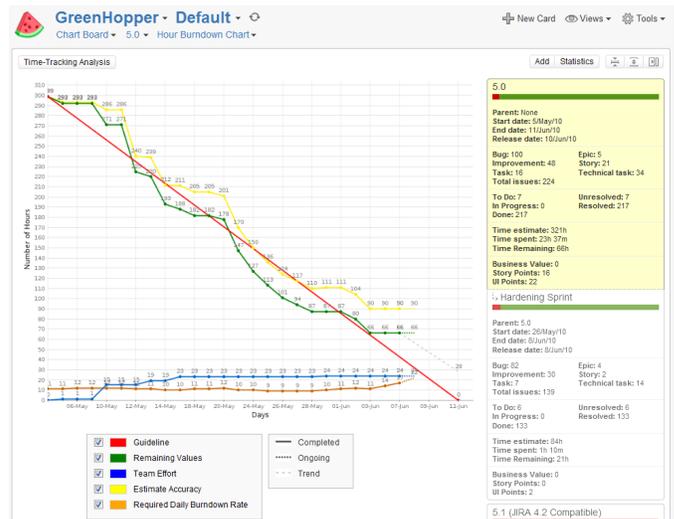


Abbildung 2. Viewing the Hour Burndown Chart [10].

geplanten Aufwand zu bewegen, um ein optimales Verhältnis zwischen „Soll“ und „Ist“ zu erreichen.

Eine weitere Möglichkeit zur Unterstützung der Softwareentwicklung ist die Visualisierung der Build-Prozesse für CI, wie sie bereits beim Ambient Surface-System vorgestellt wurde. Die folgende vorgestellte Arbeit hat diese Art der

¹<https://www.atlassian.com/software/jira> - abgerufen am 28.02.2015

²<https://de.atlassian.com/software/jira/agile> - abgerufen am 28.02.2015

Visualisierung auf eine einfache Ampel transformiert. Die Abbildung 3 zeigt dieses Beispiel. Diese Art der Visualisierung



Abbildung 3. Traffic light visualization for CI. [9]

und die daraus entstehende Assoziation, ist in dieser Arbeit besonders positiv bei den Probanden aufgenommen worden.

Bekanntere Maßnahmen der Status-Vermittlung sind die sogenannten Taskboards, wie sie auf nicht digitalisierten Boards vorkommen. Diese Darstellungsmöglichkeit wird zusammen mit dem Burndown-Chart im JIRA-System angeboten und bietet einen guten Überblick über alle Tickets innerhalb des Sprints. Es ist inzwischen eine gängige Methode, bei Meetings oder beim Projektmonitoring, das Taskboard zu verwenden.

Ergebnisse aus dieser Arbeit sind, dass besonders die Entwickler ein Interesse an der Awareness-Pflege besitzen. Dies bezieht sich dabei auf das Interesse zu den Arbeiten der Kollegen im Allgemeinen, die besitzenden Abhängigkeiten zu diesen Arbeiten und die relevanten Informationen zu den Aufgaben, welche diese Arbeit beschreiben. Die vorgestellten Visualisierungsdarstellungen ermöglichten den Teams eine Hilfe beim Einhalten von knappen Deadlines und der Auslieferung von hoch priorisierten *User-Stories*. Besonders die Taskboards bieten einen ständigen Blick auf das Projekt, mit allen darin enthaltenen, beziehungsweise identifizierten, Aufgaben. Insgesamt ist eine Verbesserung des Wissenstransfers und der Awareness von Artefakten festzustellen.

Eine Problemstellung, mit der sich eine Arbeit [11] beschäftigt, ist die Theoriegenerierung zum besseren Verständnis des fehlenden Vertrauens in verteilten agilen Teams. Diese lässt sich mit dem Thema Stakeholder-Awareness verbinden und stellt zudem Auswirkungen von mangelnder Transparenz dar, falls diese nicht genügend unterstützt wird. In dieser Studie waren 45 Praktiker aus dem agilen Umfeld von 28 verschiedenen Firmen in den USA, Indien und Australien Teilnehmer. Die Ergebnisse wurden über *Face-to-Face, One-on-One* Interviews mit offenen Fragen gewonnen. Ein Ergebnis dieser Studie ist die Ableitung der Problemstellung auf das Modell „Causal-Consequences Theoretical Family for the Lack of Trust“ [11], welches in Abbildung 4 zu sehen ist. Es stellt die Gründe und die Konsequenzen des Verlusts von Vertrauen dar. Einige dieser sind, dass kein Empfinden für Zugehörigkeit existiert. Ebenso gibt es Teammitglieder, die sich nicht als Teil der Gruppe und nicht mit der Aufgabe

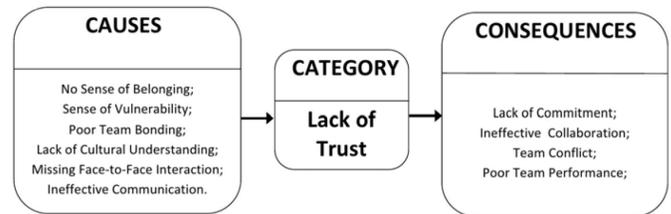


Abbildung 4. Causal-Consequences Theoretical Family for the Lack of Trust. [11]

identifizieren können. Auch ist eine fehlende Beziehung zum Kunden selbst dargestellt. Personen im Team weisen zum Teil eine Verwundbarkeit im Zusammenhang der Kommunikation zum Kunden auf. Zudem vermeiden diese Personen Situationen, welche ihre Unwissenheit gegenüber dem Kunden bloßstellen könnten, was sich besonders auf das Domänen-Fachwissen der Kunden bezieht und die daraus resultierende, mögliche falsche Umsetzung ihrerseits. Konsequenzen, die sich aus diesen Gründen generieren lassen sind zum Beispiel ein zu beobachtender Mangel an Engagement. In schlechten Fällen ist keinerlei Interesse an der Auslieferung von einfachen *Business-Values* (Geschäftswert einer Anforderung gegenüber dem Kunden) zum Kunden zu erkennen. Als weiter Punkt ist die ineffektive Zusammenarbeit zu betonen. Besonders in der Kollaboration zwischen Kunden und Managern mit dem Team wird sich auf einzelne Teammitglieder, anstatt auf das ganze Team, begrenzt. Dies hatte auch zur Folge, dass das nicht vorhandene Vertrauen und unrealistische Erwartungen zwischen Kunden und den Entwicklungsteams entstehen. Dies verletzt das Team an sich und erzeugt sogar unter den Teammitgliedern Konflikte. Insgesamt wirkt sich dies auch auf die Team-Effektivität aus [11]. Zu erkennen ist insbesondere der Verlust des Fokus bezüglich der Aufgaben, *Deadlines* und der Auslieferung der Software. Diese Studie zeigt, dass mehrere Faktoren zur frühzeitigen Erkennung eines Verlusts von Vertrauen existieren. Bezüglich dieser Faktoren kann ein höherer Grad der Transparenz bei der Unterstützung zur Bewältigung beitragen. Parallel ist die Informationsversorgung zum Endkunden, wie auch der Austausch zwischen den Teammitgliedern ein gravierendes Problem. Besonders die Komplexitätssteigerung, durch den Kontext der verteilten Entwicklung, zeigt eine drastischere Ausprägung der Gründe und Konsequenzen.

Geeignete Visualisierungsmittel bieten eine gute Möglichkeit bei der Unterstützung und Verbesserung der Projekt-Awareness gegenüber Stakeholdern und dem Entwicklungsteam. Darüber hinaus müssen diese Mittel eine geeignete Aggregation und Transformation der benötigten Information zu den entsprechenden Fachdomänen bieten. Auch hier stellt sich die Frage, wer sind die Stakeholder und wie können diese identifiziert werden.

Wie bereits im Kapitel I („Einführung“) beschrieben, ist die Fragestellung zur Definition von Stakeholdern nicht immer trivial. Zwei Arbeiten [12] [13] beschäftigen sich zudem mit der Problemstellung der Identifikation von Stakeholdern an sich und der Verbindung von Stakeholdern zur Team Agilität innerhalb der Softwareentwicklung.

Bei der ersten Arbeit [12] geht es um die Identifikation der Stakeholder und die Forschung zur Entwicklung eines Modells.

Dieses Model hat das Ziel, ein Stakeholder Konzept zu entwerfen, um Managern beim Identifizieren und der Beteiligung von Stakeholdern zu unterstützen. Dabei wird zur Kenntnis genommen, dass diese Stakeholder den Prozess beeinflussen und durch den Prozess beeinflusst werden. Im Kontext der Softwareentwicklung wird bei dieser Identifikation oft nur auf Kunden, Endbenutzern oder Projektsponsoren eingegangen. Dabei wird der Fokus auf die Produktentwicklungsteams vernachlässigt, wie auch eine Vielzahl von Stakeholder-Gruppen, welche wiederum die unterschiedlichsten Interessen am Projekt besitzen. Die Forschungsumgebung bezieht sich im Kontext dieser Arbeit auf eine Produktentwicklungsorganisation, welche über mehrere Orte und Zeitzonen hinweg verteilt ist und dabei Standorte in den USA, Europa und Asien besitzt. Diese Organisation ist dabei verantwortlich für mehrere Portfolios von Geschäftskunden und Serverprodukten. Die Produkte sind dabei für den generellen Verkauf am Massenmarkt, im Unterschied zu individuellen Produkten für Endkunden, bestimmt. Das Problem, welches sich in dieser Konstellation ergibt, ist das Nichtvorhandensein eines einzelnen Kunden. Hierdurch entsteht die Herausforderung und der Konflikt beim erfolgreichen Einführen von agilen Entwicklungsprozessen. Der Zeitraum der Untersuchung erstreckt sich insgesamt auf einen Zeitraum von zwei Jahren, wo Beobachtungen, Interviews und Retrospektiven durchgeführt wurden.

Viele Team-Transformationen, zum agilen Prozess, schlagen durch die fehlende Aufmerksamkeit aller benötigten Stakeholder fehl. Dies wird begründet, durch den Mangel von Awareness und der teils mangelnden Möglichkeiten.

Die Tabelle I zeigt eine Übersicht von möglichen Stakeholdern, welche sich innerhalb einer Produktentwicklungsorganisation befinden können. Diese Übersicht ist dabei keinesfalls vollständig. Vielmehr soll sie die Vielfalt der Stakeholder deutlich machen und die Schwierigkeit der Filterung, bei einer hohen Anzahl von Interessensgruppen, verdeutlichen. Der Vorschlag, innerhalb dieser Arbeit, ist die Grup-

Software Developer	User Experience	Account Manager
QA Engineer	Interaction Designer	Technical Support
Test Automation	Universities, Colleges	Investors
Feature Test	Research Institutes	Accountant
System Test	Technical Writer	Sales Manager
Performance Test	Development Manager	Solution Architect
Alpha Test	QA Manager	Lab Administrator
Early Field Trials	Product Manager	Scrum Master
Marketing	Program Manager	Agile Coach
Enterprise Architect	System Architect	Software Architect
Security Engineer	Release Manager	Trainer
HR	Lawyer	Media
Customers	Recruiter	Support Engineer
Channel Partner	Partners	Technical Lead
Engineering Director	Distributor	Environmental Groups
Vice President	Product Management Director	Sales Engineer
Senior Vice President	Resellers	Technical Marketing Engineer
President	Integrators	Analysts
CEO	General Manager	CIO
Open Source community	CTO	Stock holders
Standards Bodies	3 rd Party Developers	Special Interest Groups
Operations Manager	Regulatory Bodies	Stock Exchange

Tabelle I. Stakeholder in einer Produktentwicklungsorganisation [12]

perierung der Stakeholder in sechs Gruppen: „Product Owner Team“, „Product Delivery Team“, „Program Sponsor Team“, „Product Consumers“, „Product Council“ und „Program Core Team“. Diese Art der Gruppierung ermöglicht es, dass einer Vielzahl von Stakeholdern strukturiert Führungsrichtlinien,

einem agilen Kontext und Verantwortlichkeiten zugeteilt werden können. Die Gruppen „Product Owner Team“, „Product Delivery Team“, „Product Council“ und „Program Core Team“ermöglichten zudem ein Mapping auf ein Scrum of Scrum Modell und darüber hinaus ein Mapping von traditionellen Rollen der Organisation auf agile Rollen. Die Vorgehensweise orientierte sich dabei an einer rein statischen Struktur des Unternehmens und grenzt sich von möglichen dynamischen Strukturen ab, welche sich auf die Interaktion von verschiedenen Stakeholdern untereinander fokussiert und wie agile Prozesse, die fortlaufenden Möglichkeiten von Stakeholder Engagements unterstützt. Die agilen Praktiken, wie das Sprint Planning, Sprint Review, Daily Scrum, Informationsartefakte, das Scrum of Scrum, die Retrospektive und die kontinuierliche frühe Förderung der Stakeholder-Identifikation begünstigen die Integration der Stakeholder sowie deren Bedürfniserfüllung in dem Entwicklungsprozess. Weiterhin verhelfen agile Praktiken wie „CI“, *Unit Testing*, *Test Driven Development* und hoch frequentierte Releases, das Feedback über den Zustand des Produkts zu verstärken und das Erreichen der Stakeholder-Vorstellung zu ermöglichen [12].

Die zweite Arbeit [13], bezüglich der Verbindung von Stakeholdern zu Team Agilität innerhalb der Softwareentwicklung, beschäftigt sich ebenfalls mit der Thematik der Einbeziehung von Stakeholdern im Entwicklungsprozess. Dabei wird versucht, das Einbeziehen mit dem Grad der Agilität von Teams zu verbinden. Auch wird auf die Bedeutsamkeit der Einbindung von Stakeholdern eingegangen, wie auch auf die bisher unterschiedlichen Ergebnisse zur Auswirkung der Einbeziehung von Stakeholdern innerhalb der Produkt- und Serviceentwicklung hingewiesen. Daraus begründet sich die Bedeutsamkeit der Bindung von Stakeholdern in der neuen Entwicklung von Softwaresystemen, welche agile Methoden nutzen.

Zur Untersuchung dieser Debatte dienen zwei Softwareentwicklungsprojekte im Finanzsektor. Jedes Projekt besteht dabei aus einem Team, welches jeweils mehr als drei Jahre Erfahrungen im agilen Entwicklungsumfeld besitzt. Die Teams besitzen dabei eine vergleichbare Teamstruktur und entwickeln verschiedene Softwaresystemlösungen. Team A entwickelt ein Softwareprodukt und Team B ein Produkt zur Unterstützung zur Service-Auslieferung.

Wie auch in der vorherigen Arbeit, wird auf die häufig verbreiteten Limitierungen bei der Identifikation der Stakeholder eingegangen. Dabei beziehen sich diese auf Kunden, Geschäftsleute, Entwickler, Projektsponsoren und Endbenutzer. Die Begründung hinter der häufigen Limitierung der Stakeholder auf einzelne Rollen beruht dabei auf der Reflexion aus dem *Agile Manifesto* [14] [13]. In extremeren Fällen fällt die Limitierung dabei sogar auf einzelne Stakeholder, wie den Kunden oder das Entwicklungsteam zurück. Hierdurch wird häufig auch in publizierten Studien oft auf einen einzelnen Bereich des gesamten Stakeholder-Portfolios verwiesen.

Das Problem der Stakeholder Identifikation liegt daran, dass die Theorien und Techniken dem strategischen Level der Unternehmen zu Grunde liegen. Der Vorschlag hierbei ist, dass die Identifikation auf dem operativen Levels liegt, in dem die neue Softwareentwicklung stattfindet. Hierzu gehört, dass die Stakeholder zum einen identifiziert werden und die Quellen des Einflusses verstanden werden. Zusätzlich sollten auch die Interessen der Stakeholder in die Unternehmensplanung

aufgenommen werden. Als dritter Punkt wird das Wahrnehmen der dynamischen Theorien vorgeschlagen, die die Identifikation und Salienz (Auffälligkeiten) von Stakeholdern hervorheben sollen. Zur letztendlichen Identifikation wird ein Vier-Level Framework verwendet. Dies ermöglicht es, dass die Stakeholder in vier Level unterteilt werden können. Die Abbildung 5 zeigt die Einordnung dieser Level und setzt diese in Relation zueinander. Unter dem Level „Business



Abbildung 5. Vier Levels von Stakeholder [13]

Environment“ versteht sich die industrielle Interessensgruppe, beziehungsweise die Domänenexperten. Das „Offering“ beinhaltet diejenigen, die in der Organisation in der Entwicklung eingebunden sind, wie zum Beispiel die Ingenieurs- und Technikentwicklung, wie auch das Marketing oder der Verkaufs-Bereich. Das Level „Product“ beinhaltet Stakeholder, die das Produkt direkt verwalten oder nutzen. Hierzu gehört der Endbenutzer, wie auch die Servicebereitstellung, die das Produkt benutzen. Im letzten und obersten Level liegt das „Service Delivery“ wo sich Servicebereitsteller aus dem Unternehmen, die Endkunden und auch die Endbenutzer, die Leistungsempfänger vom Produkt/Service selbst sind, befinden. Des Weiteren sind jedem Level verschiedene Stakeholder-Gruppen zugeordnet. Ein Beispiel hierfür ist wie bereits genannt die Gruppe der „Servicebereitsteller“. Dieses Framework mit Leveln und Gruppen wurde innerhalb von Workshops mit den jeweiligen Teams genutzt, um die Kooperations-Intensivität zu messen. Jeder Teilnehmer eines Teams konnte jeweils zur Gruppe eine generelle Irrelevanz oder Relevanz zuordnen und darauf basierend eine Annäherung zur Relevanz von 1-10 angeben, wobei 10 der höchsten Annäherung entspricht.

Das Ergebnis dieses Workshops bringt hervor, dass einige Stakeholder im Vergleich zu anderen offensichtlich mehr Relevanz für die Entwicklungsprojekte besitzen und das einige Stakeholder in diesen Projekten irrelevant sind. Dabei wurden keine Stakeholder identifiziert, die nicht einer der vorhandenen Gruppen zugeordnet werden konnten. Im Ergebnis fielen Gruppen auf, die von der Entwicklung wahrgenommen werden müssen. Das Team hat wiederum allerdings keinen Einfluss auf diese. Ebenso wurden Gruppen entdeckt, welche involviert werden sollten, bei denen diese Einbindung bisher erfolglos war.

Wenn den vorher genannten Gruppen eine Relevanz zugeordnet wird, ergibt sich das Bild, dass beim Entwicklungsteam etwa die Hälfte eine Relevanz dem Level „Environment“

zuordnen. Das „Offering“ wird bei beiden Teams des Workshops annähernd mit 100 % bewertet. Beim „Product-Level“ sind es 100 % für das Team A und 75 % für Team B. Beim Level „Service Delivery“ wiederum sind nur ein Viertel als relevant markiert.

Die höchste Intensität ist dem Team A zugeordnet für:

- Company’s engineering/technical development
- Partner (external & internal partners)
- Company’s product maintenance

Für das Team B:

- Company’s sales
- Supplier
- Company’s engineering/technical development
- Company’s product maintenance
- Company’s service delivery(delivering service by using this product)

Wenn man sich nun die Zuordnung der Intensitäten gegenüber den Leveln ansieht, fokussiert sich Team B eher am „Product-Level“, gefolgt vom „Offering-Level. Team A orientiert sich am „Offering“ und „Environment“, dicht gefolgt vom „Product-Level“.

Team B besitzt insgesamt einen größeren Score für agile Veranlagung und für die Stakeholder Kooperations-Intensität. Hieraus kann abgeleitet werden, um so agiler das Team ist, um so höher ist auch die Stakeholder Kooperations-Intensität, was auch aus dem Agilen Manifesto hervorgeht. Die Arbeit geht dabei sogar soweit, dass alternativ die Intensität der Kooperation mit Stakeholdern das agile Verhalten fördert. Der einzige Weg zum Anpassen an die kontinuierlich wechselnden Bedürfnissen der Stakeholder in komplexen Geschäftsumgebungen, ist das Anpassen an agile Entwicklungsmethoden.

Agile Teams kommunizieren intensiver mit den Stakeholdern in den Leveln „Offering“und „Service Delivery“, was diese Studie allerdings nicht zeigen konnte. Die Kooperations-Intensität mit Stakeholdern in diesen beiden Leveln ist mehr eine organisatorische Charakteristik, als das Angehören eines variablen Levels der Agilität des Entwicklungsteams. Die herausstehende Ausnahme ist, dass das Team B als einziges Team den Endkunden mit einbezieht und dieser Gruppe dabei eine sehr geringen Intensität zuordnet. Team A identifiziert den Endkunden sogar gar nicht. Dies wird laut Aussage der Arbeit als häufige, in der Industrie als spezifizierte Charakteristik, angesehen. Die Zuordnung der Teams zu einem agilen Entwicklungsmodell sind dabei wie folgt: Das Team A geht nach Scrum vor und das Team B nach Kanban.

Das Ergebnis dieser Untersuchung ist zudem, dass die Kooperations-Intensität den technischen Stakeholdern und Endkunden besser zugeordnet werden kann, als den Geschäftspersonen.

III. LÖSUNGSANSATZ

In diesem Abschnitt soll näher auf den Lösungsansatz zu der vorgestellten Awareness zu Stakeholdern eingegangen werden und einen Fokus der Arbeit, mit den daraus resultierenden Herausforderungen, darstellen. Der Lösungsansatz bezieht sich vor allem auf die Möglichkeit eines *Toolchanges* (Einführung einer neuen Software-Lösung) zur Komplexitätsreduzierung in

der Thematik der verbesserten Awareness zwischen Stakeholdern und dem Entwicklungsteam. Dieser Toolchange soll dabei ein Hilfsmittel darstellen, um Probleme innerhalb des Projekts im Diskurs der kontinuierlichen Informationsversorgung zu den Stakeholdern zu ermöglichen. Dieser Ansatz basiert dabei auf dem Prinzip des Eingreifens im Prozess des Entstehens, innerhalb der Softwareentwicklung. Hierdurch können Problematiken frühzeitig erkannt werden und die Probleme auf Faktoren zurückgeführt werden. Somit soll es den Stakeholdern ermöglicht werden, dass das Softwareprojekt insgesamt transparenter wird und das gegen eine frühzeitige Abweichung besser gegengesteuert werden kann.

Eine Tool-Lösung sollte somit Informationen aus verschiedenen, bestehenden Systemen aggregieren, die zur Ableitung für die Punkte Qualität (Feature Complete), innerhalb des Zeitplans und innerhalb des Budget eine Hilfestellung bieten. Hierzu müssen nicht nur die Daten beschafft und zusammengetragen werden, sondern vielmehr müssen die Ableitungen dieser Daten die Fachdomänen tangieren. Zu Bestimmung der Fachdomänen müssen zudem die Stakeholder selbst identifiziert werden.

A. Fokus

Der Fokus dieser Arbeit zielt auf die Awareness zwischen unteren und mittleren Managements zum Entwicklungsteam ab. Hierdurch soll eine verbesserte Transparenz des Projektstatus, zu den Stakeholdern und vor allem ein verbessertes Projektmonitoring insgesamt ermöglicht werden. Das untere und mittlere Management wird dabei definiert als erste Entscheidungsinstanz, die nicht direkt dem Entwicklungsteam zugeordnet wird, sondern als Projektleitung oder nächste Leitungsinstanz zur Entwicklung steht. Fokus der Informationen sind dabei die Punkte Feature-Complete, beziehungsweise Qualität, Budget und Zeitplan. Diese Informationen sollen zum einen eine verbesserte Wahrnehmung des Projektverlaufs ermöglichen und zum anderem einen fachlichen Übergang ermöglichen. Die direkte Schnittstelle wird dabei die Anbindung an die Awareness des Entwicklungsteams an sich sein, die sich zum Beispiel im Rahmen der Retrospektive näher mit vorhanden Problemquellen beschäftigt. Die Schnittstellen zu anderen Projekten werden unter dem Abschnitt V näher beschrieben.

B. Herausforderungen

Bei den oben genannten Punkten ergeben sich Herausforderungen, die es zu lösen gilt. Es muss definiert werden, inwieweit die Aggregation von Informationen durchgeführt werden kann. Hierzu müssen die entsprechenden Systeme und die zu Verfügung stehenden Informationen identifiziert und gefiltert werden. Dabei ist die Abstraktion dieser resultierenden Daten wiederum auch eine weiterführende Problematik. Ebenso muss untersucht werden, wie ein kontinuierlicher Informationsfluss digital vereint unterstützt werden kann und sich ein Softwaresystem an die konsumierenden Endpunkte anbinden könnte.

IV. AUSBLICK

Der Ausblick der hier gegeben werden soll, stellt zum einen die Zusammenfassung dieses Dokuments dar und zum anderen die weitere Vorgehensweise im weiterführenden Projekt. Im Anschluss an diesen Hauptabschnitt werden die koexistierenden

Projekte vorgestellt, die eine Schnittstelle zu diesem Projekt besitzen.

A. Zusammenfassung

Diese Ausarbeitung hat sich in der „Einführung“ mit der generellen Herleitung der Motivation der agilen Softwareentwicklung mit Scrum und auf die immer noch existierende Problematik der Schätzung von Projektaufwendungen befasst. Hieraus wurde die Seite für die tragenden Entscheidungen klar gezeigt und auf die benötigten Unterstützungen hingewiesen. Die Entscheidungsträger, in vielen Fällen auch die Stakeholder, wurden nachfolgend erläutert und darauf folgend auf die problematische Wertschätzung von Teammitgliedern hingewiesen. Die weiterführende Debatte der Awareness wurde anschließend näher betrachtet und erläutert. Im Kapitel II „Verwandte Arbeiten“ wurden aufbauend Arbeiten vorgestellt, die sich mit den angrenzenden Forschungsfeldern beschäftigen. Zu Beginn wurde weiter auf den Punkt der Wertschätzung von Teammitgliedern eingegangen und der Versuch zur verbesserten Awareness innerhalb eines Feldversuches vorgestellt. Die folgende Forschung basierte auf dem Punkt der Visualisierung für agile Softwareentwicklung und zeigte zum einen die möglichen Informationsschnittstellen, wie auch die resultierenden positiven Effekte, die im Team sichtbar waren. Die dritte Arbeit fokussierte sich auf die Problematik des Vertrauens. Hierbei wurden Gründe und Konsequenzen dargestellt, die auch die Sensibilität innerhalb der Kommunikation vom Team zum Stakeholder verdeutlichen. Die abschließenden Arbeiten innerhalb dieses Kapitels beschäftigten sich mit der problematische Definition der Stakeholder an sich und der Intensität der Verbindung zu Stakeholdern innerhalb der agilen Softwareentwicklung. Dabei wird die Problematik der Identifikation und Fokussierung von diesen deutlich. Die letzte Arbeit in diesem Kapitel stellte zudem eine Verbindung und Darstellung der Kooperations-Intensivität zu Stakeholdern von agilen Teams dar. Im Abschnitt III „Lösungsansatz“ ist näher auf einen möglichen Lösungsansatz zur Verbesserung der Stakeholder-Awareness eingegangen worden. Der Abschnitt „Fokus“ grenzte die anstehende Forschung weiter ab und wurde über den nachfolgenden Punkt „Herausforderungen“ innerhalb dessen auf die Risiken des Projektverlaufs hingewiesen wurde, abgerundet.

Nachfolgend soll der Arbeitsplan im Projekt und abschließend die Schnittstellen zu anderen Projekten verdeutlicht werden.

B. Arbeitsplan im Projekt

Innerhalb des Projekts soll sich die Arbeit zum einen auf den Aufbau einer Testinfrastruktur und zum anderen auf die erste Modellierung eines Informationspools zur Transparenz des Projektstatus verteilen. Das grundsätzliche Ziel ist somit auch den Übergang zu den anderen Schnittstellen klarer zu definieren und einen möglichen Prototypen für eine Softwaresuite zu skizzieren. Es soll zudem ein erweiterter Aufwand in die Recherche vorhandener Softwareprodukte investiert werden, um im optimalen Fall die regulären Standardanwendungen als Datenquelle nutzen zu können und die Implementierungsaufwand nicht unnötig zu erhöhen.

V. SCHNITTSTELLEN

Die dargestellte Projektarbeit wird mit den Kommilitonen Leon Fausten und Erwin Lang durchgeführt. Der gemeinsame

Kontext ist die agile Softwareentwicklung mit Scrum und die verbesserte Unterstützung dieser Teams durch digitale Mittel. Gegliedert ist dieser Bereich dabei in drei Teile.

Der erste Teil beschäftigt sich dabei mit der direkten kollaborativen Zusammenarbeit in der aktiven Planung einer Softwarearchitektur. Hier ergibt sich die Problemstellung beim angestrebten Erreichen der nicht funktionalen Zielen, wie Skalierbarkeit, ohne starke Anpassungen an der Architektur vornehmen zu müssen. Innerhalb von Scrum ist das Sprint-Planning als Phase des Softwarezyklus anvisiert. Dieser Konsens wird von Leon Fausten angegangen.

Der zweite Teil konzentriert sich innerhalb von Scrum auf die Retrospektive und wird von Erwin Lang bearbeitet. Dabei wird insbesondere der Verlauf des abgearbeiteten Sprints näher untersucht. Ziel der Arbeit ist vor allem die optimierte Transparenz der Aktivitäten innerhalb des Sprints. Hierdurch soll es anschließend besser möglich sein, dass die Störungsfaktoren innerhalb des Projekts entdeckt werden und darauf basierend Entscheidungen besser gefällt werden können.

Der dritte Teil, welcher von meiner Person bearbeitet wird, bezieht sich auf die beschriebene Schnittstelle vom Team zu den Stakeholdern und wird sich der Untersuchung und Optimierung der Awareness, zu diesen Schnittstellen, stellen.

REFERENZEN

- [1] I. Barkalov, J. Martin, S. Mume, M. Metz, P. Wagner, and K. Jiang, *Forward Visibility Studie (2. Auflage)*. Capgemini Consulting, 2013, abruf: 2015-02-28. [Online]. Available: <http://www.de.capgemini-consulting.com/resource-file-access/resource/pdf/forward-visibility-edition-2.pdf>
- [2] VersionOne, *8th Annual State of Agile Survey*. VersionOne, 2013, abruf: 2015-02-28. [Online]. Available: <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>
- [3] K. Schwaber and J. Sutherland, *The Scrum Guide*. scrumguides.org, 2011, abruf: 2014-12-14. [Online]. Available: <http://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>
- [4] F. Brooks, *The Mythical Man-Month, Anniversary Edition: Essays On Software Engineering*. Pearson Education, 1995.
- [5] A. Gerlicher, "Computer-supported cooperative work (cscw) — kollaborative systeme und anwendungen," in *Kompendium Medieninformatik*, ser. X.media.press, R. Schmitz, Ed. Springer Berlin Heidelberg, 2007, pp. 143–195, abruf: 2015-02-28. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36630-0_3
- [6] J. Schwarzer, L. Barnkow, P. Kastner, and K. von Luck, "Hin zu mehr sichtbarkeit und wertschätzung in der softwareentwicklung," in *Mensch & Computer 2013: Interaktive Vielfalt*, S. Boll, S. Maaß, and R. Malaka, Eds. München: Oldenbourg Verlag, 2013, pp. 341–344, abruf: 2015-02-28. [Online]. Available: <https://users.informatik.haw-hamburg.de/ubicomp/arbeiten/papers/MuC2013.pdf>
- [7] Atlassian, *Atlassians JIRA Website*. Atlassian, 2015, abruf: 2015-02-28. [Online]. Available: <https://www.atlassian.com/software/jira>
- [8] —, *Atlassians Jira Agile Website*. Atlassian, 2015, abruf: 2015-02-28. [Online]. Available: <https://de.atlassian.com/software/jira/agile>
- [9] J. Paredes, C. Anslow, and F. Maurer, "Information visualization for agile software development," in *Software Visualization (VISOFT), 2014 Second IEEE Working Conference on*, Sept 2014, pp. 157–166, abruf: 2015-02-28. [Online]. Available: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6980227>
- [10] Atlassian, *Greenhopper BurnDownChart*. Atlassian, 2013, abruf: 2015-02-28. [Online]. Available: <https://confluence.atlassian.com/download/attachments/391087285/gh-chartboard-burndownhour.png>
- [11] S. Dorairaj, J. Noble, and P. Malik, "Understanding lack of trust in distributed agile teams: A grounded theory study," in *Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on*, May 2012, pp. 81–90, abruf: 2015-02-28. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6272500>
- [12] K. Power, "Stakeholder identification in agile software product development organizations: A model for understanding who and what really counts," in *Agile Conference (AGILE), 2010*, Aug 2010, pp. 87–94, abruf: 2015-02-28. [Online]. Available: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5562805>
- [13] T. Juhola, M. Yip, S. Hyrynsalmi, T. Makila, and V. Leppanen, "The connection of the stakeholder cooperation intensity and team agility in software development," in *Management of Innovation and Technology (ICMIT), 2014 IEEE International Conference on*, Sept 2014, pp. 199–204, abruf: 2015-02-28. [Online]. Available: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6942425>
- [14] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," 2001, abruf: 2015-02-28. [Online]. Available: <http://www.agilemanifesto.org/>