

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

Umfragestudie zur Klassifikation von Meetings in Softwareprojekten

Survey to Classify Meetings in Software Development
Projects

Bachelorarbeit

im Studiengang Informatik

von

Son Tung Nong

Prüfer: Prof. Dr. Kurt Schneider

Zweitprüfer: Dr. Jil Klünder

Betreuer: Jil Klünder

Hannover, 15.03.2021

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 15.03.2021



Son Tung Nong

Zusammenfassung

Meetings spielen in Softwareprojekten eine große Rolle und werden für verschiedene Zwecke eingesetzt. Diese Arbeit befasst sich mit den Fragen, welche Meetings in Softwareprojekten stattfinden und wie diese klassifiziert werden können.

Für die Beantwortung dieser Fragen wurde eine Umfragestudie durchgeführt, um Meetings in Softwareprojekten zu identifizieren und Eigenschaften bezüglich des Meetingzwecks, der Teilnehmer, der Frequenz und der Dauer, zu ermitteln. Dabei wurden insgesamt 21 Umfrageantworten gesammelt, in denen 66 Meetings angegeben wurden. Diese Meetings wurden zu den oben genannten Eigenschaften untersucht und ausgewertet.

Das Ergebnis liefert eine Klassifikation von 16 verschiedenen Meeting-Arten, die zusätzlich in acht Oberkategorien eingeordnet werden können. Diese Kategorien sind Kickoff, Planung, Review, Status Update, Retrospektive, Ad hoc, Workshop und Training. Die am häufigsten angegebenen Meeting-Arten waren *Meetings zum täglichen Update*, *Meetings zur Planung von Aufgaben für eine Zeitperiode*, *Meetings zur Reflexion einer Zeitperiode* und *Meetings zur Vorstellung eines neuen Projekts*. Grundsätzlich lassen sich viele Meetings wiedererkennen, die in der Literatur vorzufinden sind, wie beispielsweise die Scrum Events oder das Kickoff Meeting.

Die in dieser Arbeit vorgestellte Klassifikation gibt einen Überblick über Meetings, die in Softwareprojekten eingesetzt werden können und beschreibt Eigenschaften, die verschiedene Arten von Meetings haben.

Abstract

Survey to Classify Meetings in Software Development Projects

Meetings are a very important factor in software development projects, and are used for a variety of purposes. This study investigates which different types of meetings take place in software development projects, and how they can be classified.

For the purpose of this investigation, a survey was conducted, in which people were asked about their meetings in software development projects. Participants were asked to answer questions about the purpose of a meeting, the participants, the frequency as well as the duration of a meeting. In total, a sample of 21 answers was collected, in which participants have stated 66 meetings. The answers were evaluated with regard to the meeting properties stated above.

This results in a classification of 16 different types of meetings, that can be further categorized into 8 different categories. These categories are Kickoff, Planning, Review, Status Update, Retrospective, Ad hoc, Workshop and Training. The most common types of meetings were *Meetings for daily update*, *Meetings to plan tasks for a period of time*, *Meetings to reflect on a period of time* and *Meetings to present a new project*. In general a lot meetings from the literature can be found in the results of this study, like Scrum Events or the Kickoff Meeting.

The proposed classification in this study gives an overview of the different types of meetings, that take place in software development projects, and describes properties of these types of meetings.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Zielsetzung der Arbeit	1
1.2	Struktur der Arbeit	2
2	Grundlagen	3
2.1	Scrum	3
2.1.1	Überblick	3
2.1.2	Scrum Team	3
2.1.3	Scrum-Artefakte	4
2.1.4	Scrum Events	5
2.2	Weitere Meetings	7
2.2.1	Kick-off-Meeting	7
2.2.2	Defect Triage Meeting	7
3	Verwandte Arbeiten	9
4	Forschungsmethode	11
4.1	Ziel und Auswahl der Forschungsmethode	11
4.2	Erstellung und Aufbau der Umfrage	11
4.2.1	Fragen zu den Meetings	12
4.3	Teilnehmerakquise	14
4.4	Filtern der Datensätze	14
5	Ergebnisse	15
5.1	Demografische Daten	15
5.2	Klassifikation der Meetings	17
5.2.1	Ergebnis der Klassifikation	17
6	Interpretation und Diskussion	27
6.0.1	Welche Meetings finden in Softwareprojekten statt? . .	27
6.0.2	Klassifikation in Anlehnung an verwandte Arbeiten . .	29
6.1	Einschränkungen der Validität	31

7	Zusammenfassung und Ausblick	33
7.1	Zusammenfassung	33
7.2	Ausblick	34

Kapitel 1

Einleitung

Kommunikation ist in der Softwareentwicklung ein wesentlicher Aspekt für erfolgreiche Projekte [12]. Beispielsweise müssen Anforderungen an die Software richtig kommuniziert werden, damit keine falschen Funktionalitäten implementiert werden. Eine effektive Möglichkeit der Kommunikation ist der Austausch von “face-to-face” [1]. Meetings ermöglichen “face-to-face” Kommunikation mit mehreren Personen [1] und sind wesentlich verantwortlich für den Erfolg eines Softwareprojekts [12]. Deswegen nehmen Meetings einen großen Teil der Zeit in Softwareprojekten in Anspruch [12].

Meetings können für verschiedene Zwecke eingesetzt werden. Beispielsweise werden in Scrum, Meetings zur Planung von Aufgaben oder zur Überprüfung von Ergebnissen eingesetzt [21]. Doch dies sind nur einige wenige Beispiele, wofür Meetings eingesetzt werden können. In der Praxis können durchaus mehr Meetings stattfinden, als die Meetings, die in Scrum vorgeschrieben sind. Es stellt sich also die Frage, welche verschiedenen Meetings in Softwareprojekten noch stattfinden können und wofür diese eingesetzt werden.

1.1 Motivation und Zielsetzung der Arbeit

In dieser Arbeit soll untersucht werden, welche Meetings in Softwareprojekten stattfinden und wie diese klassifiziert werden können. Das Ziel ist eine Klassifikation zu entwickeln, in die alle Meetings eingeordnet werden können, die in Softwareprojekten stattfinden. Für die Entwicklung dieser Klassifikation, werden die Eigenschaften Meetingzweck, Meetingteilnehmer, Meetingfrequenz und Meetingdauer betrachtet.

In der Forschung lassen sich keine Arbeiten zur Klassifikation von Meetings in Softwareprojekten finden. Eine Klassifikation von Meetings in Softwareprojekten hilft Praktikern und Forschern zu verstehen, für welche Zwecke Meetings eingesetzt werden, oder auch nicht eingesetzt werden und welche Eigenschaften diese Meetings, beispielsweise in Bezug auf die

Teilnehmer, Frequenz und Dauer, haben.

Verschiedene Studien zeigen außerdem, dass Meetings häufig nicht effektiv sind [15]. Deswegen beschäftigen sich viele Forscher mit der Frage, wie Meetings effektiv oder “gut” gestaltet werden können. Eine Klassifikation von Meetings in Softwareprojekten bietet eine Grundlage für diese Art von Forschung, da Agendas und Richtlinien nur entwickelt werden können, wenn Eigenschaften von Meetings bekannt sind.

Um zu ermitteln, welche Meetings in Softwareprojekten stattfinden und welche Eigenschaften diese haben, soll eine Umfragestudie durchgeführt werden, die auf Personen abzielt, die in Softwareprojekten involviert sind. Dabei wird ermittelt, welche Meetings bei den Teilnehmern in Softwareprojekten stattfinden und Fragen zu den oben genannten Eigenschaften gestellt.

1.2 Struktur der Arbeit

Zunächst wird im zweiten Kapitel ein Überblick über die Meetings gegeben, die in der Literatur vorgesehen sind. Dabei wird vor allem auf Scrum und die Scrum Meetings eingegangen. Anschließend werden in Kapitel 3 Arbeiten vorgestellt, die Aufschluss über Meetings geben, die in der Softwareentwicklung stattfinden können. Außerdem werden verschiedene Arbeiten betrachtet, die Meetings klassifiziert haben. Danach wird in Kapitel 4 die Auswahl der Forschungsmethode erläutert, sowie der Aufbau der Umfrage, die Akquise der Teilnehmer und das Filtern der Datensätze. Im Anschluss werden die Ergebnisse der Auswertung vorgestellt. In Kapitel 6 werden die Ergebnisse interpretiert und diskutiert. Dabei werden auf Einschränkungen der Validität eingegangen und Ergebnisse werden in Anlehnung an verwandte Arbeiten eingeordnet. Im letzten Kapitel folgt eine Zusammenfassung und ein Ausblick für weiterführende Forschung.

Kapitel 2

Grundlagen

Das am häufigsten eingesetzte Framework in der Softwareentwicklung ist Scrum [13]. Da in Scrum bereits Meetings vorgeschrieben werden, soll in diesem Kapitel das Scrum Framework genauer betrachtet werden. Dies gibt einen grundlegenden Überblick über Meetings, die in der Softwareentwicklung stattfinden können und zeigt außerdem wofür diese eingesetzt werden. Auf dieser Grundlage kann außerdem untersucht werden, ob die Scrum Meetings auch in der Praxis eingesetzt werden.

2.1 Scrum

2.1.1 Überblick

Scrum ist ein Projektmanagement-Framework, das seinen Ursprung in der Softwareentwicklung hat [21]. Es definiert verschiedene Meetings, Rollen und Werkzeuge, die gemeinsam das Strukturieren und Managen von Teamarbeit unterstützen [3]. Die Idee ist eine iterative Vorgehensweise, wobei in wiederkehrenden, regelmäßig stattfindenden Zeitintervallen, den sogenannten Sprints, ein bestimmtes Arbeitskontingent erledigt wird [3].

2.1.2 Scrum Team

Ein Scrum Team besteht aus drei Rollen: dem Product Owner, dem Scrum Master und dem Entwicklerteam. Scrum Teams sind interdisziplinär, d.h. die Mitglieder verfügen gemeinsam über alle notwendigen Fähigkeiten, um in jedem Sprint einen Wert zu schaffen. Ein Scrum Team besteht üblicherweise aus 10 oder weniger Personen und sollte damit so groß sein, dass es effektiv untereinander kommunizieren kann und trotzdem bedeutsame Arbeit fertigstellen kann. [21]

Product Owner

Der Product Owner ist dafür zuständig den Wert des Produktes zu maximieren. Um dies zu erreichen, muss er das Product Backlog effektiv verwalten. Konkret bedeutet dies, dass er die Product-Backlog-Einträge erstellt und klar kommuniziert. Insbesondere muss er die Priorisierung der Product-Backlog-Einträge vornehmen und damit die Reihenfolge der Arbeit festlegen. Er bestimmt also, woran in den Sprints gearbeitet wird. Der Product Owner steht im Kontakt mit den Stakeholdern und kann dessen Bedürfnisse berücksichtigen, allerdings ist er allein verantwortlich für das Product Backlog und trifft immer final die Entscheidung über Anpassungen am Product Backlog. [21]

Entwicklerteam

Zum Entwicklerteam gehören diejenigen Personen, die im Sprint die eigentliche Arbeit verrichten und das Inkrement erstellen. Dabei muss das Entwicklerteam nicht nur aus Programmierern bestehen, sondern kann, je nach Bedarf, auch andere Experten, wie zum Beispiel Designer, beinhalten. Das Entwicklerteam organisiert sich selbst und entscheidet unter sich, wie Aufgaben koordiniert werden. Auch bei Problemen innerhalb des Sprints trifft das Entwicklerteam intern Entscheidungen über mögliche Lösungen. [28]

Scrum Master

Der Scrum Master ist dafür verantwortlich, dass Scrum richtig umgesetzt wird und die Scrum-Theorie und -Praxis verstanden wird. Außerdem dient der Scrum Master dem gesamten Team, indem er den Mitgliedern in verschiedenen Bereichen Unterstützung leistet. Er dient dem Product Owner beispielsweise, indem er bei der Verwaltung des Product Backlogs und der Planung der Aufgaben für den nächsten Sprint hilft. Dem Entwicklerteam dient er, indem er das Team bei der Selbstorganisation unterstützt, Hindernisse im Sprint beseitigt und sicherstellt, dass alle Scrum-Events stattfinden und produktiv sind. Auch dem Unternehmen als Ganzes dient der Scrum Master, indem er die notwendigen Schulungen und Coachings zur Einführung von Scrum anbietet. [21]

2.1.3 Scrum-Artefakte

Die Artefakte von Scrum beschreiben, welche Arbeit zu erledigen ist und welche Arbeit bereits erledigt wurde [21]. Sie liefern Transparenz und geben Aufschluss über die Leistung eines Sprints, wodurch Überprüfung und Anpassung möglich sind. Man unterscheidet dabei drei wesentliche Artefakte [8].

Product-Backlog

Das Product Backlog ist eine Liste von Anforderungen, Features, Verbesserungen, Fehlerbehebungen und Aufgaben, die zur Erstellung eines Produkts notwendig sind [8] und dient als einzige Quelle von Arbeit, die durch das Scrum Team erledigt wird [21]. Das Product-Backlog ist dynamisch und kann bei Bedarf angepasst werden [8]. Es wird vom Product Owner verwaltet [21].

Sprint-Backlog

Das Sprint-Backlog besteht aus einer Reihe von Einträgen aus dem Product-Backlog und definiert die Aufgaben, die im Sprint umgesetzt werden sollen. Die Einträge aus dem Product-Backlog werden im Sprint-Backlog nochmal in kleinere, konkretere Aufgaben aufgeteilt, sodass Entwickler genau wissen, was zu tun ist. Das Sprint-Backlog dient als Plan für das Entwicklerteam und wird auch von diesem erstellt. [8]

Inkrement

Ein Inkrement ist ein potenziell auslieferbares Produkt und das Ergebnis der umgesetzten Product-Backlog Einträge nach einem Sprint [23]. Jedes Inkrement baut auf allen vorherigen Sprints auf, sodass die Funktionalitäten eines Inkrements mit jedem Sprint steigen [23]. Ein Inkrement sollte verwendbar sein, damit es einen Mehrwert erzielt [21].

2.1.4 Scrum Events

Zu den Scrum Events zählt der Sprint, das Sprint Planning, Daily Scrum, Sprint Review und die Sprint Retrospektive. Der Sprint ist dabei ein Container für alle anderen Events, was bedeutet, dass all diese Events innerhalb eines Sprints stattfinden. [21]

Sprint

Ein Sprint ist ein fest definierter Zeitraum in dem das Scrum Team Arbeit erledigt. Der Sprint ist ein zyklisches Event, in dem alle anderen Scrum Events stattfinden und hat einen fest definierten Zeitraum von einem Monat oder weniger. Im Sprint findet die eigentliche Wertschaffung statt, da hier ein verwendbares Produkt, das Inkrement, entwickelt wird. Die kurze Dauer und das ständige Wiederholen von Sprints, bietet die Möglichkeit sich häufig Feedback zu holen und den Fortschritt in Richtung des Produkt-Ziels besser überprüfen und anpassen zu können. [21]

Sprint Planning

Jeder Sprint beginnt mit einem Sprint Planning. Hierbei geht es darum, den anstehenden Sprint zu planen. Das gesamte Scrum Team ist dabei beteiligt, allerdings können bei Bedarf auch zusätzliche Personen eingeladen werden. Im Wesentlichen werden beim Sprint Planning folgende Themen behandelt:

1. Es wird gemeinsam ein Sprint-Ziel formuliert, das für die beteiligten Stakeholder einen Wert bringt.
2. Die Aufgaben des Product Backlogs werden besprochen und priorisiert. Dann wird abgeschätzt, welche Aufgaben im Sprint bearbeitet werden können.
3. Für die ausgewählten Aufgaben des Product Backlogs wird geplant, wie diese konkret umzusetzen sind und wie diese in kleinere Arbeitseinheiten zerlegt werden können.

Das Ergebnis des Sprint Planning ist das Sprint-Backlog. Das Sprint Planning beschränkt sich in der Regel auf acht Stunden bei einem vierwöchigen Sprint. [21]

Daily Scrum

Das Daily Scrum ist ein tägliches 15-minütiges Meeting für das Entwickler-team [21]. Es findet zur selben Zeit am selben Ort statt [21]. Der Zweck dieses Meetings besteht darin, den Fortschritt des Sprint-Ziels zu überprüfen und bei Bedarf Anpassungen vorzunehmen [21]. Jedes Teammitglied soll dafür ein kurzes Update geben, um sich mit dem Team zu synchronisieren [20]. Folgende drei Fragen werden dabei beantwortet:

1. Was wurde seit dem letzten Daily Scrum erledigt, um das Sprint-Ziel zu erreichen [20]?
2. Welche Probleme gibt es, die das Sprint-Ziel verhindern [20]?
3. Was soll bis zum nächsten Daily Scrum erledigt werden, um das Sprint-Ziel zu erreichen [20]?

Sprint Review

Beim Sprint Review wird das Inkrement des Sprints, dem Product Owner und anderen wichtigen Stakeholdern vorgestellt und der Fortschritt in Richtung des Produkt-Ziels diskutiert. Hier besteht für die Stakeholder und dem Product Owner die Möglichkeit, Feedback zu den Ergebnissen des Sprints zu geben, damit Anpassungen bezüglich des Product-Backlogs und zum weiteren Ablauf vorgenommen werden können. Das Sprint Review findet als vorletztes Event am Ende von jedem Sprint statt und beschränkt sich bei einem einmonatigen Sprint auf maximal vier Stunden. [21]

Sprint Retrospektive

Die Sprint Retrospektive findet als letztes Event eines jeden Sprints statt und ist ein Meeting innerhalb des Scrum Teams. Das Ziel ist es, Wege zu finden, um die Qualität und Effektivität der nächsten Sprints zu steigern. Dafür wird rückblickend auf den letzten Sprint geschaut und der Verlauf in Bezug auf Individuen, Interaktionen, Prozesse und Werkzeuge überprüft. Das Team bespricht, welche Aspekte gut liefen und wo Probleme aufgetaucht sind und diskutiert wirkungsvolle Verbesserungen für den nächsten Sprint. Die Sprint Retrospektive schließt einen Sprint ab und ist für einen einmonatigen Sprint auf maximal drei Stunden beschränkt. [21]

2.2 Weitere Meetings

Ausgehend von den Scrum Events, können Meetings bereits in vier Klassen eingeteilt werden. Meetings zur Planung, Meetings zur täglichen Fortschrittsüberprüfung, Review Meetings und Retrospektive Meetings. Im Folgenden werden kurz zwei weitere Meetings vorgestellt, die in Softwareprojekten stattfinden können.

2.2.1 Kick-off-Meeting

Ein Kick-off-Meeting ist ein Meeting zu Beginn des Projekts, bei dem sich Projektbeteiligte treffen, um ein neues Projekt vorzustellen. Das Ziel dieses Meetings ist den Projektbeteiligten, die Projektvision, die Projektziele und den Projektplan zu vermitteln und damit Motivation zu schaffen. Je nach Art des Projekts und Unternehmen nehmen bei Kick-off-Meetings das Projektteam teil, sowie Vertreter der Auftraggeberseite und andere Personen, die am Projekt beteiligt sind. [2]

2.2.2 Defect Triage Meeting

Bei Defect Triage Meetings geht es darum, Defekte oder Fehler der Software aufzuzeigen und zu diskutieren [9]. Die Defekte werden evaluiert und anhand ihrer Schwere priorisiert [9]. Defect Triage Meetings werden häufig in mittelgroßen und großen Softwareunternehmen eingesetzt [11].

Kapitel 3

Verwandte Arbeiten

Eine Studie von Stray und Moe [25] untersucht wie Koordination in globaler Softwareentwicklung stattfindet. Der Fokus der Studie bestand darin, zu untersuchen, wie viel Zeit in Meetings verbracht wird, welche Herausforderungen bei geplanten und ungeplanten Meetings entstehen und wie die Kollaborationssoftware Slack in globaler Softwareentwicklung eingesetzt wird. Für die Studie wurden die, in einem Unternehmen stattfindenden Meetings, observiert. Diese Meetings waren Stand-up meetings, Retrospektive Meetings, Demo Meetings, Scrum-of-Scrum Meetings, Seminare, Workshops, Task force Meetings und Bug triage Meetings. Die Studie gibt damit Einblicke in Meetings, die in Softwareprojekten stattfinden können. Der Fokus dieser Studie liegt allerdings nicht darin, Meetings zu klassifizieren oder herauszufinden, welche Arten von Meetings in Softwareprojekten stattfinden.

Des Weiteren gibt es in der Forschung verschiedene Studien zu den eingesetzten Praktiken in der Softwareentwicklung. Die HELENA-Studie von Kuhrmann et al. [13] ist eine groß angelegte, internationale Studie, in der untersucht wird, welche Methoden und Praktiken in der Praxis eingesetzt werden. Die Studie zeigt, dass Scrum und Iterative Development die am häufigsten eingesetzten Frameworks sind und das Code Reviews die am häufigsten eingesetzte Praktik ist [13]. Außerdem zeigt die Studie den Einsatz von weiteren Praktiken wie Daily Stand Ups, Retrospektiven oder Sprint Reviews.

Auch die “State of Agile” [6] ist eine groß angelegte, internationale Studie, die jährlich den Einsatz von agilen Praktiken in der Softwareentwicklung untersucht. Die Ergebnisse aus dem Jahre 2019 zeigen, dass 85% der Teilnehmer Daily Stand Ups einsetzen und Daily Stand Ups damit die am häufigsten eingesetzte agile Praktik ist. Darauf folgen Retrospektiven mit 81%, Sprint/Iteration Planning mit 79% und Sprint/Iteration Reviews mit 77%. Auch diese Umfrage bietet damit Einblicke in mögliche Meetings, die in Softwareprojekten stattfinden.

Die Ergebnisse beider Studien geben also Aufschluss darüber, welche Meetings in Softwareprojekten vorkommen können.

Meetings können nach vielen verschiedenen Faktoren und Eigenschaften klassifiziert werden. Anfänge zur Klassifikation von Meetings in Unternehmen macht Schwartzman [22] im Jahre 1986. Dieser klassifiziert Meetings nach dem Grad der Spontanität und unterscheidet dabei zwischen geplanten und ungeplanten Meetings. Ähnlich dazu klassifiziert Boden 1994 [5] formale und informale Meetings. Formal bedeutet dabei, dass das Meeting an einem festgelegten Termin stattfindet und eine Leitung, als auch eine Agenda enthält [5]. Informale Meetings sind Meetings in einer kleineren Gruppe, ohne Leitung und ohne Agenda [5].

In einer weiteren Studie von Monge et al. [16] wurden Meetings in amerikanischen Unternehmen untersucht. Insgesamt wurden dabei 903 Umfrageantworten gesammelt. Die Ergebnisse der Studie zeigen, wofür Meetings am häufigsten eingesetzt wurden. Die meisten Meetings wurden zur Lösung von Konflikten, zum Treffen von Gruppenentscheidungen und zur Lösung von Problemen eingesetzt. Des Weiteren wurden in der Studie fünf verschiedene Arten von Meetings vorgestellt. Diese umfassen Staff, Task Force, Information Sharing, Brainstorming und Ceremonial Meetings.

Volkema [27] klassifiziert sechs verschiedene Arten von Meetings basierend darauf, ob diese einen einzelnen Zweck verfolgen (“single-focused”) oder mehrere Zwecke (“multifocused”) und basierend auf das Format des Meetings. Das Format des Meetings beschreibt, ob Informationen primär von einer Person kommen (“hierarchical”) oder gleichmäßig von verschiedenen Personen des Meetings (“organic”).

Eine weitere Möglichkeit Meetings zu klassifizieren entwickelten Allen et al. [10]. In ihrer Arbeit wurde eine Taxonomie für Meetings auf Basis des Meetingzwecks entwickelt. Auch in dieser Studie wurde eine qualitative Umfrage durchgeführt, in der Teilnehmer nach dem Zweck des letzten Meetings gefragt wurden. Aus der Studie resultierte eine Taxonomie mit 16 Kategorien auf Basis des Meetingzwecks.

All diese Arbeiten betrachten allerdings allgemeine Meetings in Unternehmen und keine Meetings in Softwareprojekten. In dieser Arbeit wird sich auf Meetings innerhalb von Softwareprojekten beschränkt.

Kapitel 4

Forschungsmethode

4.1 Ziel und Auswahl der Forschungsmethode

In dieser Arbeit sollen zwei Fragen beantwortet werden:

1. Welche Meetings finden in Softwareprojekten statt?
2. Wie lassen sich Meetings in Softwareprojekten klassifizieren?

Für die Beantwortung der ersten Frage müssen Personen befragt werden, die in Softwareprojekten involviert sind. Um dabei möglichst viele Meetings abzudecken, sollten so viele unterschiedliche Personen wie möglich, aus verschiedenen Branchen, befragt werden. Eine Umfragestudie bietet sich dafür gut an, da damit eine größere Breite abgedeckt werden kann, als beispielsweise eine Interviewstudie.

Für die Entwicklung einer Klassifikation müssen Faktoren ermittelt werden, die einzelne Klassen voneinander abgrenzen. In diesem Falle wurde sich für die Faktoren Zweck des Meetings, Teilnehmer, Frequenz/Häufigkeit, und Dauer. Um die genannten Eigenschaften für jedes Meeting zu ermitteln, wurden offene als auch geschlossene Fragen gestellt. Für die Ermittlung des Meetingzwecks, sowie der Meetingfrequenz wurde ein Freitextfeld zur Verfügung gestellt, um qualitative Daten zu erheben. Die Meetingdauer, sowie die Meetingteilnehmer wurden über geschlossene Frage ermittelt.

4.2 Erstellung und Aufbau der Umfrage

Die Umfrage wurde mit dem Umfragetool LimeSurvey erstellt. Um möglichst viele Teilnehmer anzusprechen, wurde die Umfrage in den Sprachen Englisch und Deutsch zur Verfügung gestellt. Es wurde außerdem die Anonymität der Umfrageteilnehmer gewährleistet, da keine persönlichen Daten erhoben werden mussten.

Zu Anfang der Umfrage wird der Teilnehmer mit einer Willkommensnachricht begrüßt, in der das Ziel der Umfrage, sowie der Kontext kurz erläutert wird.

Die Umfrage besteht im Wesentlichen aus zwei Teilen. Im ersten Teil geht es um allgemeine Fragen zum Teilnehmer und zum Unternehmen. Hier wurden Fragen gestellt zur Unternehmensgröße, Unternehmensbranche und Rolle des Teilnehmers im Softwareprojekt. Des Weiteren wurde als Ausschlusskriterium gefragt, ob der Teilnehmer überhaupt in Softwareprojekten involviert ist.

Im zweiten Teil der Umfrage wurde der Teilnehmende gebeten, an sein letztes Softwareprojekt zurückzudenken und zu überlegen, welche verschiedenen Arten von Meetings in diesem Projekt stattfanden. Um die Aktualität der Antworten zu gewährleisten und die Beantwortung der Fragen ein wenig zu erleichtern, wurde explizit nach dem letzten Softwareprojekt gefragt, anstatt nach generellen Softwareprojekten. Zu jeder Meeting-Art wurde eine Gruppe von Fragen (siehe 4.2.1) gestellt. Die gleichen Fragen wurden für jede Art von Meeting, die vom Teilnehmer angegeben wurde, wiederholt. Ganz am Ende der Umfrage, bestand noch die Möglichkeit, einen Kommentar zu hinterlassen.

4.2.1 Fragen zu den Meetings

Die folgenden Fragen wurden zu jedem Meeting gestellt, das vom Teilnehmer angegeben wurde.

1. *Welcher der folgenden Kategorien würden Sie dieser Meeting-Art zuordnen? Falls sich keine Kategorie eignet, können Sie auch eine Neue angeben.*

Hier sollte aus einer der folgenden Kategorien ausgewählt werden.

- Kick-Off Meeting
- Planning Meeting
- Daily/Stand-Up Meeting
- Review Meeting
- Retrospective Meeting
- Unscheduled/Ad hoc Meeting
- Defect/Bug Triage Meeting
- Handover Meeting
- Workshop
- Sonstiges

Die hier genannten Kategorien leiten sich teilweise aus den in Kapitel 2 genannten Meetings ab. Die Einteilung der Meetings in einer dieser Kategorien, dient zum einen als Denkanstoß für den Teilnehmer, da hier mögliche Meetings aufgelistet sind, die der Teilnehmer haben könnte. Zum anderen erleichtert eine Einordnung vom Teilnehmer die spätere Auswertung. Sollte der Teilnehmer Meetings haben, die zu keiner der genannten Kategorien passen, konnte bei Auswahl der Antwortmöglichkeit "Sonstiges" eine weitere Kategorie spezifiziert werden.

2. *Was ist der Zweck des Meetings? Bitte geben Sie eine kurze Beschreibung.*

Hier sollte in einem Freitextfeld der Meetingzweck beschrieben werden.

3. *Bitte wählen Sie aus welche Personen standardmäßig an dem Meeting teilnehmen.*

Hier konnten mehrere Antwortmöglichkeiten ausgewählt werden.

- Project/Team Manager
- Architect
- Tester
- Product Manager/Owner
- Quality Manager
- Developer
- Analyst/Req. Engineer
- Trainer/Coach
- Scrum Master
- Sonstige

4. *Wie häufig findet dieses Meeting statt und/oder während welcher Phase des Projektes?*

Die Frequenz des Meetings sollte in einem Freitextfeld beschrieben werden.

5. *Wie lange dauert ein Meeting dieser Art in der Regel?*

Hier sollte aus einer der folgenden Antwortmöglichkeiten gewählt werden.

- < 30 Minuten
- 30-60 Minuten
- 1-2 Stunden
- 2-4 Stunden
- > 4 Stunden

4.3 Teilnehmerakquise

Die Umfrage wurde am 12.01.2021 aktiviert. Um Personen zu erreichen, die in Softwareprojekten involviert sind, wurden Unternehmen im Bereich der Softwareentwicklung angeschrieben. Im Verlauf von einigen Wochen wurden dabei etwa 200 Unternehmen kontaktiert. Dies erfolgte über Email oder ein Kontaktformular. Weiterhin wurde die Umfrage an private Kontakte aus dem Bekannten- und Freundeskreis verschickt. Zusätzlich wurde die Umfrage auf Plattformen wie Reddit, XING, LinkedIn und Facebook geteilt. Dabei wurden vor allem Gruppen ausgesucht, die für Personen aus der agilen Softwareentwicklung oder aus dem Projektmanagement gedacht sind. Auch verschiedene Vereine, Organisationen und Veranstalter für Softwareentwickler, wie zum Beispiel die Java User Group oder agile wednesday, wurden kontaktiert. Die Verteilung der Umfrage über Unternehmen stellte sich als mäßig heraus. Die meisten Teilnehmer wurden über private Kontakte aus dem Bekannten- und Freundeskreis, Xing-, LinkedIn- und Facebook-Gruppen erreicht. Für diese Arbeit wurden die Datensätze bis zum 10.02.2021 verwendet. Zu diesem Zeitpunkt wurde die Umfrage 106-mal gestartet.

4.4 Filtern der Datensätze

Ein Großteil der Teilnehmer haben die Umfrage bereits bei oder nach den allgemeinen Fragen abgebrochen und keine Meetings angegeben. Einige Teilnehmer haben die Umfrage mit bedeutungslosen Antworten gefüllt. Die daraus resultierenden Datensätze lieferten keinen Mehrwert und wurden daher bei der Auswertung nicht berücksichtigt. Einige Teilnehmer haben bei der ersten Frage "Sind Sie in Softwareprojekten involviert?", als Antwort "Nein" angegeben aber trotzdem Meetings angegeben.

Es wurden somit nur Antworten berücksichtigt, bei denen die Teilnehmer mindestens ein Meeting mit Beschreibung des Meetingzwecks angegeben haben und bei der Frage "Sind Sie in Softwareprojekten involviert?", als Antwort "Ja" angegeben haben. Insgesamt wurden 85 Datensätze herausgefiltert. Somit ergibt sich eine Anzahl von 21 verwendbaren Datensätzen. Davon sind 20 Antworten komplett vollständig und eine Antwort teilweise vollständig beziehungsweise nicht beendet. Nicht beendete Umfrageantworten sind trotzdem wertvoll, wenn mindestens ein Meeting angegeben wurde.

Kapitel 5

Ergebnisse

5.1 Demografische Daten

In Abbildung 5.1 ist zu sehen, dass Teilnehmer aus Unternehmen aller fünf Größenordnungen teilgenommen haben. Die Verteilung ist eher ausgeglichen, allerdings sind die meisten Teilnehmer aus Unternehmen mit 251-2499 Mitarbeitern. Am wenigsten vertreten waren Teilnehmer aus Unternehmen mit weniger als 11 Personen.

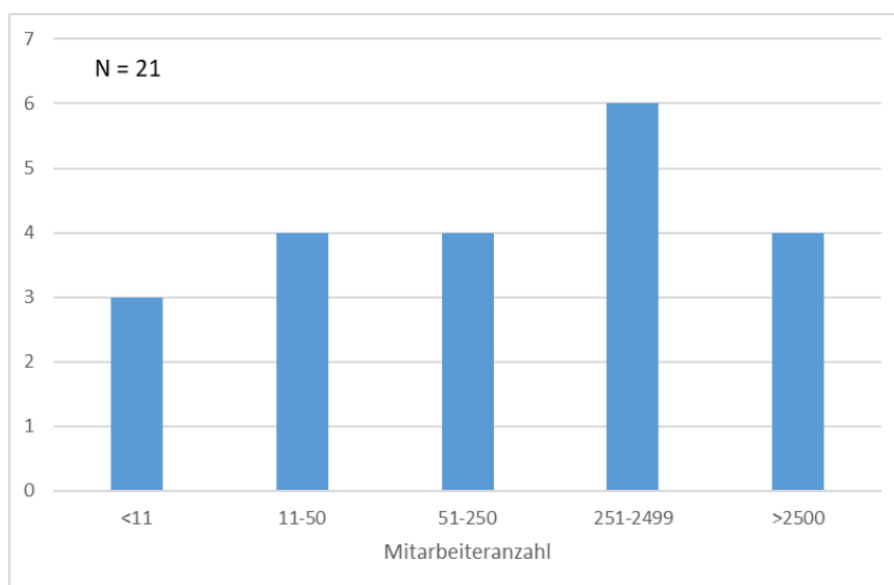


Abbildung 5.1: Mitarbeiteranzahl der teilnehmenden Unternehmen

Bezüglich der Anwendungsdomäne des Produkts/Projekts war eine Mehrfachnennung möglich. Hier ist zu sehen, dass Webanwendungen die am häufigsten vertretende Dömäne ist. Auch Mobile Anwendungen ist eine sehr

häufig angegebene Domäne. Gar nicht vorkommend sind Anwendungen aus der Luftfahrt, Raumfahrt, Spielen, sowie Medizinprodukte und Gesundheitswesen (siehe Abbildung 5.2).

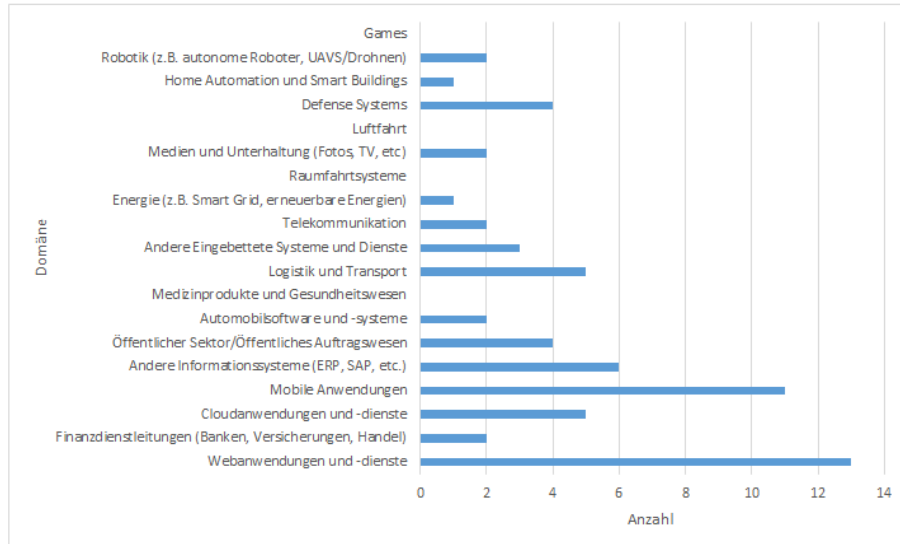


Abbildung 5.2: Anwendungsdomäne der teilnehmenden Unternehmen

Teilnehmer wurden außerdem gefragt, welche Rolle sie am häufigsten zugewiesen werden. In Abbildung 5.3 wird deutlich, dass der Großteil der Teilnehmer die Rolle eines Developers haben. Mit 19% folgt die Rolle des Projekt-/Teammanagers. Gar nicht vertreten waren Trainer/Coaches, Qualitätsmanager und Architekten. Ein Teilnehmer gab seine Rolle als “CEO” an.

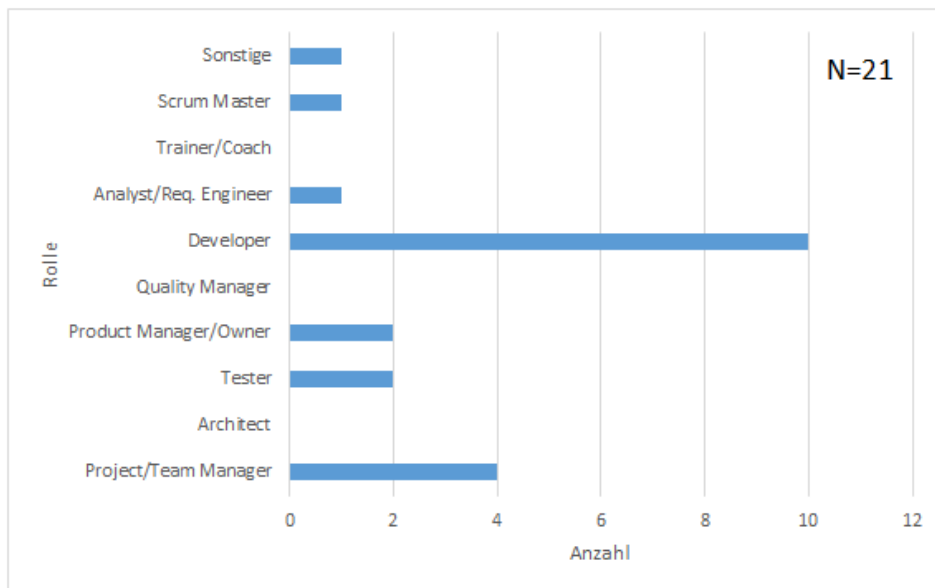


Abbildung 5.3: Rollenverteilung der Teilnehmer

5.2 Klassifikation der Meetings

Insgesamt haben 21 Teilnehmer 66 Meetings angegeben. Diese wurden zu Angaben bezüglich des Meetingzwecks, den Teilnehmern, der Frequenz und der Dauer untersucht. Entscheidend bei der Klassifikation der Meetings waren Angaben zum Meetingzweck. Meetings, die ähnliche Zwecke verfolgen wurden gruppiert. Anschließend konnten auf Basis der anderen Faktoren wie Teilnehmer, Frequenz und Dauer weitere Unterteilungen durchgeführt werden.

Jede Art von Meeting wurde zusätzlich einer Oberkategorie zugeordnet. Beispielsweise wurden Meetings, die ad hoc und nur nach Bedarf stattfinden, einer eigenen Oberkategorie zugewiesen, während Retrospektive Meetings einer anderen zugewiesen wurden. Eine gute Grundlage für diese Oberkategorien waren bereits die Kategorien aus der Umfrage (siehe 4.2.1). Teilweise wurde allerdings auch neu evaluiert.

5.2.1 Ergebnis der Klassifikation

Insgesamt wurden auf Basis von 66 angegebenen Meetings, 16 Arten von Meetings identifiziert, die in den Eigenschaften Meetingzweck, Meetingteilnehmer, Meetingfrequenz, und Meetingdauer Unterschiede aufweisen. Das Ergebnis ist in Tabelle 5.1 zusammengefasst.

Im folgenden wird für jede Art von Meeting eine Definition gegeben, in denen auf Aspekte wie Meetingzweck, Meetingteilnehmer, Meetingfrequenz

und Meetingdauer eingegangen wird. Außerdem werden Beispiele aus der Umfrage angegeben, aus denen die Definitionen hervorgehen. Des Weiteren werden Oberkategorien definiert und deren Eigenschaften beschrieben. Es ist anzumerken, dass die Definition der Meetings für mehrere Antworten verallgemeinert sind und nur soweit reichen, wie die Antworten der Umfrage hergeben.

Kickoff

Kickoff Meetings markieren den Start eines Projektes und finden einmalig am Anfang statt. Hierbei geht es um die Vorstellung eines neuen Projekts.

- **Meeting zur Vorstellung eines neuen Projekts**

Insgesamt haben sieben Teilnehmer diese Art von Meeting. Bei dieser Meeting-Art geht es um die Vorstellung des Projektes und des Projektziels, als auch der Klärung von Rahmenbedingungen und der zeitlichen Planung. Die Projektstruktur wird erläutert und der Umfang des Projektes geklärt. Je nach Projekt werden hier außerdem Stakeholder vorgestellt und Rollen verteilt.

Jeder dieser Teilnehmer gab an, dass diese Art von Meeting einmalig am Anfang des Projektes stattfindet. Bei der Dauer des Meetings haben Teilnehmer gleichermaßen 30-60 Minuten und 1-2 Stunden angegeben (siehe Tabelle 5.2. Eine Person gab eine Dauer von über vier Stunden an. Im folgenden werden Beispiele von Angaben aus der Umfrage gezeigt.

“Rahmenbedingungen klären. Timeline, also wann geht das Projekt los und wieviel Zeit steht für welche Phasen zur Verfügung. Stakeholder, also welche Parteien sind im Projektteam und sind relevante Teilnehmer in der Kommunikation von Status und Co. Scope, was ist in Scope und out of Scope für das Projekt. Meilensteine, also das übergeordnete Ziel in unter Ziele zerbrechen, um motivierter zu arbeiten, aber auch um flexibler in der Planung zu sein.”

“Vorstellung Projektziel / erste Projektstruktur + Zeitliche Planung; Abholen aller Stakeholder ”

Oberkategorie	Meeting-Art	Anzahl (N=66)
Kickoff	Meetings zur Vorstellung eines neuen Projekts	7
Planung	Meetings zur Planung von Aufgaben für eine Zeitperiode	10
Status Update	Meetings zum täglichen Update	13
	Meetings zum wöchentlichen Update	1
Review	Meetings zur Überprüfung von Ergebnissen einer Zeitperiode	6
	Meetings zur Übergabe des Projekts	4
	Meetings zur Überprüfung von Code und Technologien	1
Retrospektive	Meetings zur Reflexion des Projekts	4
	Meetings zur Reflexion eines Zeitraums	9
Ad hoc	Meetings zum Austausch von Informationen	1
	Meetings zur Klärung von Fragen	1
	Meetings zur Besprechung von Problemen und Bugs	2
	Meetings zur Planung eines Workshops	1
Workshop	Workshops zur Erhebung von Anforderungen	3
	Workshops zur Integration	1
Training	Meetings zur Einarbeitung der Anwender in das neue Produkt	2

Tabelle 5.1: Klassifikation der Meetings

Meeting-Art	Meetingdauer				
	<30 Min	30-60 Min	1-2 Std	2-4 Std	>4 Std
Meetings zur Vorstellung eines neuen Projekts (N=7)		3	3		1
Meetings zur Planung von Aufgaben für eine Zeitperiode (N=10)		7	2	1	
Meetings zum täglichen Update (N=13)	12	1			
Meetings zum wöchentlichen Update (N=1)		1			
Meetings zur Überprüfung von Ergebnissen einer Zeitperiode (N=6)	1	3	2		
Meetings zur Übergabe des Projekts (N=4)			2	2	
Meetings zur Überprüfung von Code und Technologien (N=1)		1			
Meetings zur Reflexion des Projekts (N=4)		2		2	
Meetings zur Reflexion eines Zeitraums (N=9)		5	1	3	
Meetings zum Austausch von Informationen (N=1)			1		
Meetings zur Klärung von Fragen (N=1)		1			
Meetings zur Besprechung von Problemen und Bugs (N=2)			1	1	
Meetings zur Planung eines Workshops (N=1)			1		
Workshops zur Erhebung von Anforderungen (N=3)		1		1	1
Workshops zur Integration (N=1)			1		
Meetings zur Einarbeitung der Anwender in das neue Produkt (N=2)			1	1	

Tabelle 5.2: Von den Teilnehmern angegebene Meetingdauer für jede Art von Meeting

Planung

Meetings dieser Oberkategorie dienen dem Zweck der Planung. Des Weiteren finden diese Meetings wiederkehrend und nach bestimmten Zeitintervallen statt.

- **Meeting zur Planung von Aufgaben für eine Zeitperiode**

Insgesamt gab es hierzu Angaben von 10 Personen. Das sind 47,6% der Teilnehmer und ist hiermit die am zweithäufigsten angegebene Meeting-Art. In diesem Meeting geht es um die Planung der Aufgaben für einen festgelegten Zeitraum, wie zum Beispiel einen Sprint oder eine Iteration. Bei sechs Teilnehmern findet diese Art von Meeting alle zwei Wochen statt, während vier Personen sogar wöchentlich Meetings zur Planung von Aufgaben einsetzen.

Die Dauer dieser Art von Meeting beträgt bei den meisten Teilnehmern 30-60 Minuten (siehe Tabelle 5.2).

“Besprechen der Aufgaben für die nächsten 2 Wochen.”

“Im Planning Meeting werden die Sprints für die Entwicklungsteams geplant. Product Owner stellt dem Dev Team die nächsten notwendigen Stories vor, die zu Erreichung eines Increments benötigt werden. Das Dev Team schätzt und diskutiert technische Feinheiten. Das Ergebnis des Meetings ist ein geplanter Sprintinhalt, der auf die Erreichung der gesetzten Meilensteine innerhalb des Projektes einzahlt.”

Review

In die Kategorie Review fallen Meetings, die dem Zweck der Überprüfung dienen. Ergebnisse, die sich auf das Produkt des Projekts beziehen, werden vorgestellt, überprüft und bewertet.

- **Meetings zur Überprüfung von Ergebnissen einer Zeitperiode**

Bei dieser Art von Meeting werden Ergebnisse einer Zeitperiode vorgestellt und es wird überprüft, ob die vorher definierten Ziele erreicht wurden. Insgesamt wurden sechs Angaben zu dieser Art von Meeting eingeordnet. Diese Meeting-Art findet nach Abschluss einer Zeitperiode, also zum Beispiel nach einer Iteration oder nach einem Sprint, statt. Konkret bedeutet das für drei Teilnehmer eine Meetingfrequenz von zwei Wochen und für zwei Teilnehmer eine Meetingfrequenz von einer Woche. Bei einer weiteren Person findet diese Art von Meeting nur alle drei Monate statt.

Bezüglich der Dauer haben die meisten Teilnehmer 30-60 minütige Meetings (siehe Tabelle 5.2).

“Dev Team stellt den abgeschlossenen Sprint vor. I.d.R. erfolgt die Präsentation durch das Dev Team selbst. Ziel ist es, das Increment

des Sprints zu präsentieren.”

“Prüfen, was umgesetzt wurde; wurde das Sprintziel erreicht?”

“review everything that was done, giving feedback, to see what everyone was doing”

- **Meeting zur Überprüfung von Code und Technologien**

Bei einer Person finden Meetings für Code Reviews und zur Erklärung von Technologien und Architekturmustern statt. Ein Code Review ist eine Methode zur Sicherung der Softwarequalität, bei der Quellcode, von anderen Personen als der Autor selber, gelesen und geprüft wird [4]. Diese Art von Meeting findet kurz vor Projektende statt oder wenn es von einem Entwickler gefordert wird. Bei der Dauer des Meetings gab der Teilnehmer 1-2 Stunden an.

“Code Reviews, Erklärung von Architekturmustern/Technologien etc.”

- **Meeting zur Übergabe des Projektes**

Hierbei geht es um die Überprüfung der Endergebnisse des Projektes und die Übergabe des Projekts an den Kunden, an die Leitung oder an eine andere Abteilung. Vier Personen haben Angaben zu dieser Art von Meeting gemacht. Bei der Frequenz gab es unterschiedliche Antworten. Ein Teilnehmer gab eine Häufigkeit von einem Mal pro Projekt an. Andere Teilnehmer haben Angaben von “alle drei Monate” oder “alle sechs Monate” gemacht. Eventuell ist dies so zu interpretieren, dass Projekte bei diesen Teilnehmern alle drei Monate bzw. alle sechs Monate stattfinden. Bei der Meetingdauer haben Teilnehmer gleichermaßen Angaben von 1-2 Stunden und 2-4 Stunden gemacht.

“Übergabe des Projektes”

“Handover of project to customer service”

“Abnahme”

Status Update

Bei diesen Meetings geht es um die Besprechung von Ergebnissen des letzten Tages oder der letzten Woche, sowie die Planung der Aufgaben bis zum nächsten Meeting. Außerdem können Probleme/Auffälligkeiten besprochen werden, die seit dem letzten Meeting aufgetreten sind.

- **Meeting zum täglichen Update**

Diese Art von Meeting wurde am häufigsten angegeben. 62% der Teilnehmer haben tägliche Meetings. Bei dieser Meeting-Art werden Ergebnisse des letzten Tages besprochen, Aufgaben für den Tag geplant und aufgetretene Probleme kommuniziert. Bei 12 von 13 Teilnehmern gehen diese Meetings

nicht länger als 30 Minuten. Lediglich eine Person gab eine Dauer von 30-60 Minuten an.

“Tägliches Update zu Inhalten des aktuellen Sprints. I.d.R. nach dem Schema ‘Was habe ich gestern erreicht?’, ‘Was habe ich mir für heute vorgenommen?’ und ‘Wo brauche ich Hilfe?’.”

“Tägliche Fortschrittsüberwachung; Jeder Core-Team-Member trägt vor, was gestern erledigt wurde, was heute ansteht und welche Hindernisse bestehen, damit diese schnellst möglich beseitigt werden oder Aufgaben ggf. geschoben werden”

- **Meeting zum wöchentlichen Update**

Ein Teilnehmer hat wöchentliche Status Update Meetings, in denen Ergebnisse der letzten Woche vorgestellt werden und besprochen wird, was für die nächste Woche getan werden soll. Hierfür wurde eine Dauer von 30-60 Minuten angegeben.

“Reviewing the work that has been done in the last week that is relevant to everyone and discussing about what’s currently important for all of the employees.”

Retrospektive

Bei Retrospektiven geht es darum, auf bestimmte Phasen des Projekts oder auf das Projekt selber zurückzublicken und diese in Bezug auf die Teamarbeit, eingesetzte Methoden oder Techniken zu untersuchen und zu schauen, was gut und was schlecht gelaufen ist. Das Ziel von Retrospektiven Meetings ist es, Wege zur Verbesserung zukünftiger Phasen oder Projekte zu finden.

- **Meeting zur Reflexion eines Zeitraums**

Neun Personen haben Meetings zur Reflexion eines Zeitraums. Hierbei blickt das Team auf einen bestimmten Zeitraum zurück und schaut, was gut verlief, und wo optimiert werden kann. Vor allem wird dabei auf die Zusammenarbeit im Team geschaut. Ziel dieser Meetings ist es, Wege zu finden, um zukünftige Prozesse zu optimieren. Diese Art von Meeting kann nach jeder Iteration oder jedem Sprint stattfinden, kann aber auch nach Erreichung gewisser Meilensteine abgehalten werden oder bei Bedarf, wenn festgestellt wird, dass die Projektperformance ungenügend ist. Drei Teilnehmer haben bei der Frequenz konkrete Angaben von zwei Wochen gemacht. Auch wurden einzelne Angaben von einem Mal pro Woche oder einem Mal alle paar Monate gemacht. Die Meetingdauer kann der Tabelle 5.2 entnommen werden. Die meisten gaben eine Dauer von 30-60 Minuten an.

“Einschätzung jedes Core-Team-Mitgliedes, wie der Sprint gelaufen ist; was war gut, was war nicht so gut; was können wir verbessern; die Retrospektive bezieht sich mehr auf die Zusammenarbeit im Team; hier ist entscheidend, dass sich jeder ehrlich äußert und dies auch darf, ohne dass sich andere angegriffen fühlen.”

“Inspect the way the team worked in the previous sprint and identify experiments to try to improve the way the team worked.”

- **Meeting zur Reflexion des Projekts**

Vier Teilnehmer setzen Retrospektive Meetings einmalig am Ende von Projekten ein, um auf das gesamte Projekt zurückzublicken und Wege zur Optimierung für zukünftige Projekte zu finden. Meetings dieser Art können 30-60 Minuten oder 2-4 Stunden lang sein.

“Rückblick auf das gesamte Projekt”

“Besprechung der guten und auch der verbesserungswürdigen Dinge.”

“full review, controlling, evaluation, definition of learnings”

Ad hoc

Während die Meetings in den anderen Oberkategorien geplante Meetings sind, unterscheiden sich die Meetings in dieser Oberkategorie dadurch, dass sie unregelmäßig, ungeplant und nur nach Bedarf stattfinden.

- **Meeting zum Austausch von Informationen**

Bei einem Teilnehmer finden unregelmäßig Meetings zum Austausch von Informationen statt, die in der Regel eine Dauer von 1-2 Stunden haben.

“Wissenstransfer”

- **Meeting zur Klärung von Fragen**

Ein weiterer Teilnehmer hat nach Bedarf 30-60 minütige Meetings, zur Klärung von offenen Fragen.

“Offene Fragen klären.”

- **Meeting zur Besprechung von Problemen und Bugs**

Hierzu zählen Meetings aus der Kategorie Defect/Bug Triage, die als eigene Kategorie in der Umfrage aufgelistet wurde. Teilnehmer, die Meetings in diese Kategorie eingeordnet haben, gaben alle an, dass Defect/Bug Triage Meetings lediglich bei Bedarf stattfinden. Daher ist diese Kategorie als eigene Oberkategorie weggefallen.

Bei dieser Art von Meeting geht es um die Besprechung kritischer Bugs und Problemen. Sie können 1-2 Stunden oder 2-4 Stunden dauern.

“Triage critical and difficult issues”

“Wenn ein Bug aufgefallen ist und die Aufmerksamkeit mehrerer Entwickler erfordert oder eine Änderung zu Sideeffects führen kann”

- **Meeting zur Planung eines Workshops**

Bei einem Teilnehmer finden nach Bedarf Meetings statt, um kurzfristige Team-Workshops zu planen. Diese Meetings dauern in der Regel 1-2 Stunden.

“Kurzfristige Planung eines Team-Workshops”

Workshop

Nach Lipp [14] ist ein Workshop ein Arbeitstreffen, in der eine Gruppe von Betroffenen an der Lösung einer Aufgabe arbeiten. Sie finden unregelmäßig statt und sind meist von längerer Dauer. In Workshops geht es nicht um die Vermittlung von Wissen oder der Übung des Gelernten. Sie grenzen sich daher von Seminaren und Trainings ab. Des Weiteren werden in Workshops Aufgaben bearbeitet und erfordern die aktive Teilnahme von allen Mitgliedern. Auch finden Workshops unregelmäßig statt und sind daher keine Routinemeetings. Die Definition nach Lipp [14] wird für die hier angegebenen Workshops verwendet.

- **Workshops zur Erhebung von Anforderungen**

Insgesamt haben drei Teilnehmer Workshops zur Erhebung von Anforderungen. Hierbei trifft sich das Team mit Stakeholdern, um gemeinsam Anforderungen zu erarbeiten, zu priorisieren und zu besprechen, wie diese umgesetzt werden sollen. Workshops dieser Art finden 3-4 Mal zu Beginn des Projektes statt, können aber auch im Laufe des Projektes, bei Bedarf, angesetzt werden. Die Dauer dieser Workshops kann von zwei bis über vier Stunden gehen, allerdings gibt auch eine Person eine Dauer von 30-60 Minuten an.

“Erarbeitung der Stories: Welche Epics gibt es, welche Stories gehören dazu; wie soll was wann umgesetzt werden; ‘Einschwören’ des Core-Teams; Einbeziehen weiterer Stakeholder (bspw. spätere Nutzergruppe) in die Workshops, um Akzeptanz zu sichern; Entwicklung von Prozessen; Hinterfragen bisheriger Prozesse; Bedürfnisse und Nutzen erfassen”

“Anforderungsaufnahme, Priorisierung mit Kunde und ggfls. Präsentation Umsetzungsideen, die besprochen werden sollten”

- **Workshop zur Integration**

Bei einem Teilnehmer findet relativ zu Anfang des Projektes ein einmaliges Integrationsworkshop statt, bei dem sich das Entwicklerteam, Architekt und

Projektmanager treffen, um technische Feinheiten der Software zu klären. Weitere Angaben zum Zweck wurden nicht gemacht. Diese Workshops sollen 1-2 Stunden dauern.

“Im Verlauf des Projektes werden versch. Arten von Workshops mit versch. Teilnehmern angesetzt. Bevor die Software auf Produktion ausgerollt wird, wird ein Integrationsworkshop mit Dev Team, Architekt und Projekt Managern aufgesetzt, um alle Technischen Feinheiten zu klären...”

Training

Trainings sind ähnlich wie Workshops, mit dem Unterschied, das es hierbei um die Übung und das Vermitteln von Wissen geht. Trainings sind meist auch von längerer Dauer und finden unregelmäßig statt.

- **Meeting zur Einarbeitung der Anwender in das neue Produkt**

Wird Software für Anwender entwickelt, können Trainings aufgesetzt werden, die dazu dienen, Nutzer in das neue Produkt einzuarbeiten. Dies geschieht in der Regel gegen Ende des Projektes, wenn das Produkt in einem testbaren Zustand ist. Insgesamt gaben zwei Teilnehmer diese Art von Meeting an. Die Dauer wird mit 1-2 Stunden und 2-4 Stunden angegeben.

“Einarbeitung der Anwender in das neue Projekt”

“...Ist die Software eine Anwendung für User, wird ein User Training aufgesetzt, sobald die Software in testbaren Zustand ist.”

Kapitel 6

Interpretation und Diskussion

In diesem Abschnitt werden die Ergebnisse interpretiert und diskutiert. Dabei wird Bezug genommen auf Erwartungen, Ursachen und Folgen der Ergebnisse, sowie eine Einordnung in den Forschungsstand gegeben. Des Weiteren werden mögliche Einschränkungen der Validität genannt.

6.0.1 Welche Meetings finden in Softwareprojekten statt?

Eine Klassifikation der in Softwareprojekten stattfindenden Meetings wurde in Tabelle 5.1 dargestellt.

Die Ergebnisse zeigen, dass die in der Literatur vorgeschriebenen Meetings zu Scrum, in ähnlicher oder identischer Art, auch in der Praxis eingesetzt werden. In den Meeting-Arten *Meetings zum täglichen Update*, *Meetings zur Planung von Aufgaben für eine Zeitperiode*, *Meetings zur Überprüfung von Ergebnissen einer Zeitperiode* und *Meetings zur Reflexion eines Zeitraums* lassen sich die Scrum Events, Daily Scrum, Sprint Planning, Sprint Review und Sprint Retrospektive, wiedererkennen. Zusätzlich zu den Sprint Retrospektiven, finden generell auch Retrospektiven zu ganzen Projekten statt, wenn Meilensteine erreicht werden oder auch bei Bedarf, wenn die Teamperformance schlecht ist.

Insgesamt war die am häufigsten angegebene Art von Meeting *Meetings zum täglichen Update*, die von 62% der Teilnehmer angegeben wurde. Darauf folgen *Meetings zur Planung von Aufgaben für eine Zeitperiode* mit 47,6% und *Meetings zur Reflexion eines Zeitraums* mit 42,9%. Bei einigen Teilnehmer finden Retrospektiven sowohl zu Zeiträumen innerhalb des Projektes, als auch zu ganzen Projekten statt. Insgesamt setzen 52,4% (11) der Teilnehmer Retrospektiven ein.

Allerdings muss man anmerken, dass aus Gründen von fehlender Motivation oder Lust, wahrscheinlich nicht jeder Teilnehmer alle Meetings spezifiziert hat, die auch bei ihm stattfinden.

Die "14th Annual State of Agile Report" [6] ist eine Umfrage aus dem Jahre 2019, in der untersucht wird, welche agilen Praktiken in Unternehmen

der Softwareentwicklung eingesetzt werden. Die Studie ergab, dass 85% der Teilnehmer in ihrem Unternehmen Daily Standups einsetzen. 81% der Teilnehmer praktizieren Retrospektiven, 79% Sprint/Iteration Planning und 77% Sprint/Iteration Reviews. Hier lassen sich also Übereinstimmungen mit den Ergebnissen der Arbeit erkennen.

In der Umfrage wurde nicht explizit nach eingesetzten Praktiken gefragt, allerdings haben sieben Teilnehmer explizit erwähnt, dass Sprints in ihren Softwareprojekten stattfinden. Das heißt bei mindestens sieben der 21 Teilnehmer wird Scrum praktiziert. Die Helena Studie zeigt, dass die am häufigsten eingesetzten Frameworks in der Softwareentwicklung Scrum und Iterative Softwareentwicklung sind [13].

Des Weiteren zeigt sich in den Ergebnissen, dass “Meetings zur Planung von Aufgaben für eine Zeitperiode” oder “Meetings zur Überprüfung von Ergebnissen einer Zeitperiode” bei den meisten Teilnehmern alle zwei Wochen und bei einigen wöchentlich stattfinden. Kein Teilnehmer hat eine längere Frequenz angegeben. Dies könnte darauf hindeuten, dass bei den meisten Teilnehmern in Iterationen von zwei Wochen oder sogar einer Woche entwickelt wird. Der Scrum Guide schreibt eine Sprintdauer von vier Wochen oder weniger vor [21].

Außerdem lässt sich festhalten, dass durchaus Meetings zu Code Reviews stattfinden, allerdings nur eine Person Angaben dazu gemacht hat. Code Reviews sind eine der am häufigsten eingesetzten Praktiken in der Softwareentwicklung [13]. Sie können auf verschiedene Arten praktiziert werden. Zum Beispiel als einfaches “über die Schulter schauen”, mit der Hilfe von Software oder auch als Meeting [24]. Eine Umfrage aus dem Jahre 2019 von SmartBear [24] zeigt, dass 52% der Teilnehmer, die Code Reviews praktizieren, Meetings für Code Reviews einsetzen. Aus der Umfrage resultiert allerdings, dass lediglich eine Person Meetings zu Code Reviews hat.

Des Weiteren bestätigen die Ergebnisse, dass Meetings auch ungeplant oder ad hoc stattfinden können. Verschiedene Studien zeigen bereits, dass in der Softwareentwicklung häufig ungeplante Meetings vorkommen [15] und Meetings von Forschern als geplant und ungeplant klassifiziert werden [22].

Eine beachtliche Anzahl an Teilnehmern setzen außerdem Kick-Off-Meetings ein, die in Kapitel 2 vorgestellt wurden. Defect Triage Meetings wurden hingegen nur zweimal angegeben.

In dieser Arbeit wurden zwei Arten von Workshops identifiziert. Zum einen wurden *Workshops zur Erhebung von Anforderungen* identifiziert, welche sich in der Literatur auch als Requirements Workshops wiederfinden lassen [7]. Zum anderen wurden Integrationsworkshops genannt. Allerdings bleibt unklar, wofür diese eingesetzt werden und was in diesen Workshops getan wird. Es wurde außerdem keine Literatur zu Integrationsworkshops gefunden. Hier bieten sich also Möglichkeiten weiterer Forschung an, beispielsweise in Form einer Observation oder Interviewstudie.

Abschließend ist anzumerken, dass sich die hier aufgeführten Meetings

auf Softwareprojekte beschränken. Ein Teilnehmer wies darauf hin, dass neben den projektbezogenen Meetings, auch unternehmensbezogene Meetings stattfinden können. Das bedeutet, dass im Alltag eines Praktikers durchaus mehr Meetings stattfinden können, als die hier angegebenen.

6.0.2 Klassifikation in Anlehnung an verwandte Arbeiten

In der Literatur finden sich häufig Arbeiten, die Meetings nach Funktionen wie Information Sharing, Problem Solving oder Decision Making unterscheiden [19]. Nachfolgend soll, die in dieser Arbeit resultierte Klassifikation, weitergeführt werden und in die vorhandene Literatur eingeordnet werden, indem die resultierten Meeting-Arten, nach den Funktionen “information sharing” [17], “brainstorming/problem solving” [27], “decision making” [17], “coordination and monitoring” [17], “planning” [17] und “training” [26] klassifiziert werden. Eine mögliche Klassifikation nach diesen Funktionen ist in Tabelle 6.1 dargestellt. Es ist allerdings anzumerken, dass einige Arten von Meetings mehrere Funktionen haben können, weswegen eine 100 prozentige Übereinstimmung nicht gegeben werden kann.

Zu “information sharing” Meetings gehören alle Meetings, wo Produkte, Projekte oder Ideen erklärt und präsentiert werden [27]. Dies ist der Fall bei *Meetings zur Vorstellung eines neuen Projekts*, *Meetings zur Überprüfung von Ergebnissen einer Zeitperiode* und *Meetings zur Übergabe des Projektes*, da hier ein Produkt oder ein Projekt präsentiert wird. Des Weiteren können hier *Meetings zum Austausch von Informationen* und *Meetings zur Klärung von Fragen* eingeordnet werden, da auch hier Informationen ausgetauscht werden.

Bei “brainstorming/problem-solving” Meetings geht es darum, Probleme zu analysieren, neue Ideen oder Konzepte zu generieren oder Probleme zu lösen [27]. Hierzu können Retrospektiven gezählt werden, da es darum geht, Wege zur Verbesserung von Prozessen zu finden, oder Lösungswege für bestehende schlechte Performance zu finden. Ähnlich dazu sind *Meetings zur Überprüfung von Code und Technologien*, da auch hier Wege gesucht werden, um Code zu optimieren. Des Weiteren können *Workshops zur Erarbeitung von Anforderungen* in diese Kategorie eingeordnet werden, da hier aktiv Anforderungen erarbeitet werden und Prozesse entwickelt werden.

“Coordination and monitoring” ist bei Status Update Meetings der Fall, da hier Fortschritt überwacht wird, indem Ergebnisse überprüft werden und Probleme kommuniziert werden. Außerdem koordiniert sich das Team in Bezug auf anzugehende Aufgaben.

“Decision Making” wird bei Meetings zur Besprechung von Problemen und Bugs gemacht, da es darum geht, kritische Probleme zu besprechen und letztendlich zu entscheiden, wie mit diesen Problemen umgegangen werden soll. Allerdings haben diese Meetings auch eine “information sharing” Funktion, da sich hier über vorhandene Fehler und Bugs ausgetauscht wird.

Planning wird bei *Meetings zur Planung eines Workshops* und *Meetings zur Planung einer Zeitperiode* gemacht, da es darum geht Pläne zu entwickeln.

Meetings zur Einarbeitung von Anwendern können auch hier, wieder in die Kategorie “training” eingeordnet werden.

Zur Einordnung von *Workshops zur Integration* fehlen die notwendigen Informationen und wurde deswegen ausgelassen.

Funktion	Meeting-Art
“information sharing” [17]	Meetings zur Vorstellung eines neuen Projekts Meetings zur Überprüfung von Ergebnissen einer Zeitperiode Meetings zur Übergabe des Projekts Meetings zum Austausch von Informationen Meetings zur Klärung von Fragen
“brainstorming/problem solving” [27]	Meetings zur Reflexion eines Zeitraums Meetings zur Reflexion des Projekts Meetings zur Überprüfung von Code und Technologien Workshops zur Erarbeitung von Anforderungen
“decision making” [17]	Meetings zur Besprechung von Problemen und Bugs
“coordination and monitoring” [17]	Meetings zum täglichen Update Meetings zum wöchentlichen Update
“planning” [17]	Meetings zur Planung von Aufgaben einer Zeitperiode Meetings zur Planung eines Workshops
“training” [26]	Meetings zur Einarbeitung der Anwender in das neue Produkt

Tabelle 6.1: Einordnung der Meeting-Arten nach Funktion (in Anlehnung an verwandte Arbeiten)

6.1 Einschränkungen der Validität

Im folgenden Abschnitt soll die Validität der Ergebnisse überprüft werden. Nach Wohlin et al. [29] beschreibt die Validität die Zuverlässigkeit der Ergebnisse. Demnach gibt es vier mögliche Aspekte der Validität.

Die Konstruktvalidität beschreibt zu welchem Grad die gestellten Fragen, wirklich beschreiben, was untersucht werden soll [29]. Sie wird gefährdet, wenn Teilnehmer Fragen missinterpretieren oder falsch verstehen [29]. In der Umfrage wurden zu jedem Meeting, das beim Teilnehmer stattfindet, die gleichen Fragen nach folgendem Schema gestellt: Was ist der Zweck des Meetings? Welche Personen nehmen am Meeting teil? Wie häufig findet diese Art von Meeting statt? Wie lange dauert ein Meeting dieser Art in der Regel? Die Fragen waren also sehr allgemeinverständlich und Teilnehmer konnten zusätzlich die Umfrage auf Deutsch oder Englisch beantworten. Des Weiteren sollte eine Einteilung der Meetings, in eine der angegebenen Kategorien, vorgenommen werden. Hier können einzelne Kategorien missinterpretiert oder nicht verstanden werden. Da die Klassifikation der Meetings allerdings in der Auswertung neu evaluiert wurde, hat dies keine Auswirkung auf die Ergebnisse. Eine Einschränkung der Konstruktvalidität ist daher unwahrscheinlich.

Die interne Validität beschreibt, ob Ursache-Wirkungsbeziehungen durch einen unbekanntem Faktor gefährdet werden [29]. Da sich die Teilnehmerakquise zunächst als schwierig feststellte, wurden zur Akquise der Teilnehmer jeweils fünf Amazon Gutscheine im Wert von 10€ verlost. Die Teilnahme an der Auslosung war freiwillig. Nach dem Filtern der Datensätze, haben letztendlich sechs der 21 Teilnehmer an der Auslosung teilgenommen. Eine Einschränkung der internen Validität ist also nicht ausgeschlossen.

Die Reliabilität beschreibt, inwiefern die Ergebnisse vom Forscher selbst abhängig sind [29]. Die Ergebnisse sollten reproduzierbar sein, wenn die gleiche Studie von einem anderen Forscher durchgeführt werden würde [29]. Es sollte die Frage beantwortet werden, welche Meetings in Softwareprojekten stattfinden. Daher wurde jede genannte Art von Meeting in die Ergebnisse aufgenommen, egal wie häufig sie angegeben wurde. Würde man die Umfrage erneut mit mehr Datensätzen durchführen, könnte es zu mehr Meetings kommen, als die hier angegebenen. In der Arbeit wurde eine Klassifikation entwickelt, die Faktoren wie Zweck, Häufigkeit, Teilnehmer und Dauer des Meetings betrachten. Teilweise wurden allerdings Angaben zum Meetingzweck ungenau oder nur knapp von Teilnehmern beschrieben. Daher könnten Antworten bei der Auswertung, durchaus auch missinterpretiert worden sein. Bei erneuter Durchführung der Studie durch eine andere Person könnte es eventuell zu einer anderen Klassifikation kommen. Leider gibt es keine Studien, die ähnliche Forschung betrieben haben und daher auch keinen Vergleich. Eine erneute Durchführung mit mehreren Forschern zur Kodierung der Klassen wäre sinnvoll.

Die externe Validität beschreibt, inwiefern sich die Ergebnisse generalisieren lassen [29]. Ein Großteil der Umfrageantworten wurden gesammelt, nachdem die Umfrage in Social Media Gruppen zu agiler Softwareentwicklung und Projektmanagement gepostet wurde. Die Herkunft der Antworten ist nicht bekannt, allerdings lässt sich feststellen, dass die meisten Antworten auf Deutsch verfasst wurden. 16 von 21 Personen haben die Umfrage auf Deutsch beantwortet. Der Rest der Antworten war auf Englisch und eine sogar auf Spanisch. Es ist also davon auszugehen, dass die meisten Antworten aus deutschen Unternehmen stammen und die meisten agile Praktiken einsetzen. Des Weiteren sind keine Unternehmen der Branche Games, Luftfahrt, Raumfahrtsysteme und Medizinprodukte in der Umfrage vertreten. Der Anteil an Unternehmen, die Webanwendungen entwickeln, ist mit 20% die am häufigsten vorkommende Anwendungsdömäne, gefolgt von Mobilien Applikationen mit 17,5%.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Das Ziel der Arbeit war die Beantwortung der Fragen, welche Meetings in Softwareprojekten stattfinden und wie sich diese klassifizieren lassen. Für die Beantwortung der Fragen wurde eine Umfragestudie durchgeführt. In der Umfrage wurde ermittelt, welche Meetings bei den Teilnehmer während Softwareprojekten stattfinden und zu welchem Zweck. Des Weiteren wurden Eigenschaften zu den Teilnehmern des Meetings, Frequenz des Meetings und Dauer des Meetings ermittelt. Anhand dessen wurde eine Klassifikation von 16 verschiedenen Meeting-Arten entwickelt, die zusätzlich in acht Oberkategorien eingeteilt werden können (siehe 5.1).

Die am häufigsten stattfindenden Meetings in Softwareprojekten sind *Meetings zum täglichen Update*, *Meetings zur Planung von Aufgaben einer Zeitperiode* und *Meetings zur Reflexion eines Zeitraums*. Auch *Meetings zur Überprüfung von Ergebnissen einer Zeitperiode* werden häufig eingesetzt. In diesen Meetings sind die Scrum Events wiederzuerkennen.

Bei Retrospektiven finden zusätzlich zu Sprint Retrospektiven, auch Projekt Retrospektiven, Retrospektiven nach Erreichung von Meilensteinen oder Retrospektiven bei schlechter Teamperformance statt. Zusätzlich zu Daily Scrums können auch Weekly Scrums eingesetzt werden.

Am Anfang von Projekten finden häufig Kick-Off-Meetings statt, in denen das neue Projekt vorgestellt wird. Auch können am Anfang eines Projektes mehrere Requirements Workshops eingesetzt werden, um Anforderungen zu erheben. Ein weiteres Workshop das eingesetzt wird, ist das Integrationsworkshop. Allerdings sind die vorhandenen Informationen zu dieser Art von Workshop sehr begrenzt. Zu verschiedenen Zwecken können während des Projektes Ad Hoc Meetings stattfinden, um offene Fragen zu klären, sich generell auszutauschen oder Probleme und Bugs zu diskutieren.

Am Ende eines Projektes können Code Review Meetings eingesetzt werden, um die Qualität der Software zu optimieren. Auch User Trainings werden am Ende eingesetzt, um Nutzer in das neue Produkt einzuarbeiten. Zum Schluss finden Meetings zur Abnahme oder Übergabe des Projektes statt in denen das Endprodukt präsentiert und abgenommen wird.

Generell lässt sich feststellen, dass viele Meetings, die in der Literatur vorzufinden sind, auch in der Praxis, zumindest für die gleichen Zwecke, eingesetzt werden. Beispiele hierfür sind die Scrum Events, das Kickoff Meeting oder Requirements Workshops.

7.2 Ausblick

Die Arbeit gibt einen Überblick über mögliche Arten von Meetings, die in Softwareprojekten stattfinden können. Diese Meeting-Arten beschreiben allgemein zu welchem Zweck Meetings stattfinden, wer daran teilnimmt, wie häufig diese Meetings gehalten werden und wie lange diese Meetings dauern. Da einige Meeting-Arten sehr knapp definiert wurden, wäre es sinnvoll, nochmal Interviewstudien oder Observationen durchzuführen, um Eigenschaften im Detail zu untersuchen. Auch wäre eine erneute Durchführung der Studie mit einer repräsentativeren Anzahl an Teilnehmern, sowie mehreren Forschern zur Kodierung der Klassen sinnvoll, um mehr Meetings zu identifizieren und die Ergebnisse der Arbeit zu erweitern, zu bestätigen oder zu widerlegen.

Des Weiteren wurde zwar untersucht welche Meetings in Softwareprojekten stattfinden, allerdings bleibt unklar, ob der Einsatz dieser Meetings überhaupt sinnvoll ist. Beispielsweise könnte man untersuchen, wie sinnvoll der Einsatz von wöchentlichen Status Update Meetings ist. Eine weitere wichtige Forschungsfrage ist außerdem, wie Meetings effektiv oder generell "gut" gestaltet werden können. Studien zeigen, dass Meetings häufig nicht effektiv und effizient sind [15], was viele Teilnehmer frustriert [18].

Auf der Grundlage dieser Klassifikation kann weiterführende Forschung daran anknüpfen, für diese Arten von Meetings, Agendas oder Richtlinien zu entwickeln, um Meetings dieser Art effektiver oder besser zu gestalten.

Literaturverzeichnis

- [1] S. Ambler. Agile modeling: effective practices for extreme programming and the unified process. 01 2002.
- [2] G. Angermeier. Kick-off-veranstaltung. <https://www.projektmagazin.de/glossarterm/kick-off-veranstaltung>. zuletzt aufgerufen am: 15.03.2021.
- [3] Atlassian. 14th annual state of agile report. <https://www.atlassian.com/de/agile/scrum>.
- [4] T. Baum, O. Liskin, K. Niklas, and K. Schneider. A faceted classification scheme for change-based industrial code review processes. In *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pages 74–85, 2016.
- [5] D. Boden. The business of talk. organizations in action. *Cambridge: Polity Press*, 1994.
- [6] Digital.ai. 14th annual state of agile report, May 2020.
- [7] E. Gottesdiener. *Requirements by Collaboration: Workshops for Defining Needs: Workshops for Defining Needs*. 2002.
- [8] C. Harris. Agile scrum-artefakte. <https://www.atlassian.com/de/agile/scrum/artifacts>. zuletzt aufgerufen am: 13.03.2021.
- [9] M. Irshad. Defect triage meetings: Challenges and lessons learned: Extended abstract. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE'17*, page 74–77, New York, NY, USA, 2017. Association for Computing Machinery.
- [10] A. A. Joseph, B. Tammy, W. S. Cliff, and G. R. Steven. Understanding workplace meetings: A qualitative taxonomy of meeting purposes. *Management Research Review*, 37(9):791–814, Jan 2014.

- [11] N. Kaushik, M. Amoui, L. Tahvildari, W. Liu, and S. Li. Defect prioritization in the software industry: Challenges and opportunities. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, pages 70–73, 2013.
- [12] J. Klünder, K. Schneider, F. Kortum, J. Straube, L. Handke, and S. Kauffeld. Communication in teams - an expression of social conflicts. pages 111–129, 01 2016.
- [13] M. Kuhrmann, P. Tell, J. Klünder, R. Hebig, S. Licorish, and S. MacDonell. Helena stage 2 results, Nov 2018.
- [14] U. Lipp and H. Will. Das große workshop-buch: Konzeption, inszenierung und moderation von klausuren, besprechungen und seminaren. pages 13–14, 2008.
- [15] A. Meyer, E. T. Barr, C. Bird, and T. Zimmermann. Today was a good day: The daily life of software developers. *IEEE Transactions on Software Engineering*, pages 1–1, 2019.
- [16] C. M. Monge, Peter R. and J. Wyer. A profile of meetings in corporate america: results of the 3m meeting effectiveness study. 1989.
- [17] R. W. Napier and M. K. Gershenfeld. *Groups: Theory and Experience*. Boston: Houghton Mifflin, 1989.
- [18] N. Prenner, J. Klünder, and K. Schneider. Making meeting success measurable by participants' feedback. *3rd International Workshop on Emotion Awareness in Software Engineering*, 2018.
- [19] N. Romano and J. Nunamaker. Meeting analysis: Findings from research and practice. page 13 pp., 02 2001.
- [20] S. Roock. Scrum - auf dem bierdeckel erklärt. *dpunkt.verlag*, 2016.
- [21] K. Schwaber and J. Sutherland. Der scrum guide: Der gültige leitfaden für scrum: Die spielregeln, November 2020.
- [22] H. B. Schwartzman. The meeting as a neglected social form in organizational studies. *Research in Organizational Behavior*, Vol. 8:233–258, 1986.
- [23] Scrum-Events. Scrum artefakte. <https://www.scrum-events.de/scrum-artefakte.html>. zuletzt aufgerufen am: 13.03.2021.
- [24] SmartBear. The 2019 state of code review: Trends and insights into collaborative software development, 2019.

- [25] V. Stray and N. B. Moe. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and slack. *Journal of Systems and Software*, 170:110717, 2020.
- [26] The 3M Meeting Management Team. *How to Run Better Business Meetings: A Reference Guide for Managers*. New York: McGraw-Hill, 1987.
- [27] R. J. Volkema and F. Niederman. Organizational meetings: Formats and information requirements. *Small Group Research*, 26(1):3–24, 1995.
- [28] D. West. Scrum-rollen und die wahrheit über positionsbezeichnungen in scrum. <https://www.atlassian.com/de/agile/scrum/roles>. zuletzt aufgerufen am: 13.03.2021.
- [29] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

Abbildungsverzeichnis

5.1	Mitarbeiteranzahl der teilnehmenden Unternehmen	15
5.2	Anwendungsdomäne der teilnehmenden Unternehmen	16
5.3	Rollenverteilung der Teilnehmer	17

Tabellenverzeichnis

5.1	Klassifikation der Meetings	19
5.2	Von den Teilnehmern angegebene Meetingdauer für jede Art von Meeting	20
6.1	Einordnung der Meeting-Arten nach Funktion (in Anlehnung an verwandte Arbeiten)	30

