

Institut für Softwaretechnologie

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit

Optimierung von Sitzungen in Technischen Software-Reviews

Andreas Preikschat

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr. rer. nat. Stefan Wagner
Betreuer/in:	Dr. rer. nat. Ivan Bogicevic
Beginn am:	11. Mai 2017
Beendet am:	10. November 2017
CR-Nummer:	D.2.5

Zusammenfassung

Formale Software-Reviews sind effektive, effiziente und wichtige Bestandteile der Qualitätssicherung in Softwareprojekten. Im Technischen Review ist die Review-Sitzung ein großer Kostenfaktor, da viele Personen in diese involviert sind. Außerdem sind in einer Review-Sitzung oft nicht alle Teilnehmer aktiv und die Fähigkeiten des Moderators haben einen großen Einfluss auf den Erfolg eines Reviews. Als formal definierter und strukturierter Prozess kann dieser gut durch Werkzeuge unterstützt werden.

Diese Arbeit untersucht explorativ, wie diese Unterstützung gestaltet werden kann. Der geleistete Beitrag der Arbeit besteht in der Dokumentation von Ideen zur Verbesserung und der genaueren Evaluation ausgewählter Ideen. Eine Erkenntnis ist, dass die Verwendung von Gamecontrollern zur Klassifikation von Befunden dazu führt, dass eine Review-Sitzung strukturierter abläuft und die Arbeitslast des Moderators reduziert wird. Es wurden Hypothesen generiert, welche als Grundlage für weitere Untersuchungen dienen können.

Inhaltsverzeichnis

1. Einleitung	11
1.1. Motivation	11
1.2. Zielsetzung	11
1.3. Gliederung	12
2. Hintergrund	13
2.1. Geschichte und Definition des Review-Begriffs	13
2.2. Herausforderungen des Review-Prozesses	16
2.3. Notwendigkeit von Review-Sitzungen	16
2.4. Informelle Review-Techniken	17
2.5. Zusammenfassung	19
3. Verwandte Arbeiten	21
3.1. Werkzeugunterstützung im formalen Review-Prozess	21
3.2. Entscheidungsfindung in Gruppen	21
3.3. Soziale Faktoren	22
3.4. Zusammenfassung	23
4. Verbesserungsvorschläge	25
4.1. Vorschlag I – Befunde kollaborativ verfassen	25
4.2. Vorschlag II – Limitierung der Redezeit	27
4.3. Vorschlag III – Befunde im Vorfeld formulieren	28
4.4. Vorschlag IV – Interaktion mit Tablets	29
4.5. Vorschlag V – Interaktion mit Gamecontrollern	31
4.6. Vorschlag VI – Feedback zum Zeitverlauf geben	32
4.7. Vorschlag VII – Sitzungsdashboard	33
4.8. Vorschlag VIII – Präsentationsansicht	34
5. Umsetzung von Verbesserungen	37
5.1. Umsetzung von Vorschlag V: Interaktion mit Gamecontrollern	37
5.2. Umsetzung von Vorschlag VII und VIII: Präsentationsansicht mit Sitzungsdashboard	38
5.3. Verbesserungen an der Notaransicht	40
6. Technische Umsetzung	41
6.1. Beschreibung der Architektur und Quellcodequalität	41
6.2. Verwendete Bibliotheken	42
6.3. Erweiterung und Änderungen der Architektur	42
6.4. Probleme bei der Umsetzung	43

7. Evaluation	45
7.1. Planung der Evaluation	45
7.2. Erstes Review – unmodifizierte RevAger-Version	46
7.3. Zweites Review – verbesserte RevAger-Version	53
7.4. Gefahren für die Validität	59
7.5. Abgeleitete Hypothesen	60
7.6. Zusammenfassung	62
8. Zusammenfassung und Ausblick	63
Literaturverzeichnis	65
A. Anhänge zum ersten Review	69
A.1. Anonymisierter Auszug aus den Notizen der Beobachtung	69
A.2. Interviewfragen	70
B. Anhänge zum zweiten Review	71
B.1. Instruktionen und Hinweise für die Teilnehmer	71
B.2. Informationsblatt für die Teilnehmer	72
B.3. Anonymisierter Auszug aus den Notizen der Beobachtung	73
B.4. Interviewfragen	74

Abbildungsverzeichnis

4.1.	Entwurf: Gutachter verfasst Befund während der Sitzung	25
4.2.	Entwurf: Moderator wählt den nächsten Befund aus	26
4.3.	Entwurf: Zusammenführen von Befunden in der Webanwendung	28
4.4.	Entwurf: Weboberfläche zur interaktiven Klassifikation von Befunden	30
4.5.	Gutachter können mittels Gamecontroller interagieren	31
4.6.	Entwurf: Sitzungsdashboard mit Informationen zur Sitzung	33
4.7.	Istzustand der RevAger-Befundliste	34
4.8.	Entwurf: Präsentationsansicht zur besseren Übersicht	35
5.1.	Gamecontroller mit Beschriftung	37
5.2.	Präsentationsansicht zeigt Befunde und das Sitzungsdashboard an	38
5.3.	Präsentationsansicht zeigt allgemeine Informationen zur Sitzung an	39
5.4.	Ausschnitt aus der Befundliste der Ansicht für den Notar	40
7.1.	Diskussionszeit im Verlauf der ersten Review-Sitzung	47
7.2.	Vergleich der Diskussionsdauer unterschiedlicher Klassifikationen der ersten und zweiten Hälfte der ersten Review-Sitzung	49
7.3.	Diskussionszeit im Verlauf der zweiten Review-Sitzung	54
7.4.	Diskussionszeit mit unterschiedlich vielen interagierenden Personen	56

Tabellenverzeichnis

2.1.	Klassifikationen für Befunde nach Ludewig und Lichter [LL12]	14
2.2.	Empfehlung über die Annahme des Prüflings nach Ludewig und Lichter [LL12]	15
7.1.	Gesamtanzahl des Feedbacks pro Gutachter	56

1. Einleitung

1.1. Motivation

Formale Software-Reviews sind eine effiziente und effektive Maßnahme der Qualitätssicherung in Softwareprojekten. In Reviews begutachten mehrere Personen den Prüfling, dokumentieren Fehler und bewerten dessen Qualität. Dabei gibt es verschiedene Vorgehensweisen, um diese Prüfung durchzuführen. Das *Technische Review* ist solch eine Vorgehensweise. Im Technischen Review treffen sich alle beteiligten Personen in einer Review-Sitzung, um die im Vorfeld gefundenen Fehler zu diskutieren, zu bewerten und zu dokumentieren.

Diese sogenannte Review-Sitzung ist ein kostenintensiver Faktor im Technischen Review, da viele Personen daran beteiligt sind. So sind in einer Sitzung oft nur 4 Personen aktiv, während die anderen Personen 20 % bis 70 % der Zeit abgelenkt und nicht aktiv sind. Dies ist ein Grund, warum das Technische Review in vielen Unternehmen durch – weniger effektive – informelle Review-Techniken ersetzt wird. Konsequenterweise ist die Verbesserung der Effizienz von Sitzungen in Technischen Reviews eine aktuelle Herausforderung.

1.2. Zielsetzung

Forschungsziel: *Das Forschungsziel dieser Arbeit ist es, Technische Review-Sitzungen zu untersuchen und zu verstehen, wie diese effizienter gestaltet werden können. Dabei sollen eingesetzte Werkzeuge berücksichtigt werden.*

Die Vorgehensweise dieser Untersuchung ist *explorativ*. Im ersten Schritt werden Ideen generiert und es wird untersucht, welche dieser Ideen am erfolgversprechendsten sind und wie diese zielführend in ein Review-Werkzeug und den Prozess integriert werden können. Anschließend werden einige der Ideen in ein Review-Werkzeug implementiert und evaluiert. Aus den Ergebnissen werden Hypothesen abgeleitet.

Für die Umsetzung der Ideen wird das Review-Werkzeug *RevAger* verwendet, da es bereits einige sinnvolle Funktionen bietet. RevAger ist ein Programm zur Planung und Dokumentation von Review-Sitzungen. Es ist als studentisches Projekt an der Universität Stuttgart entstanden und als Open-Source-Software verfügbar. [Reva] Das Werkzeug bietet die Möglichkeit, neben geplanten Review-Sitzungen ein sogenanntes *Sofort-Review* durchzuführen. Es ermöglicht die Verwaltung von Rollen, Teilnehmern und Prüfaspekten. In der Review-Sitzung können mit dem Werkzeug Befunde erstellt und klassifiziert werden. Dabei können Referenzen, Dateien und betreffende Prüfaspekte den Befunden zugeordnet werden. Das Werkzeug erinnert standardmäßig nach zwei Stunden daran, das Review zu unterbrechen und später fortzuführen. Am Ende einer Review-Sitzung kann ein Protokoll der Sitzung in verschiedenen Dateiformaten exportiert werden. [Revb]

1.3. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Hintergrund Zu Beginn werden Grundlagen für diese Arbeit sowie Herausforderungen und Stärken von Review-Techniken zusammengefasst und beschrieben.

Kapitel 3 – Verwandte Arbeiten: Anschließend wird darauf eingegangen, wie Sitzungen in Technischen Reviews optimiert werden können und welche Ansätze dabei verfolgt werden können.

Kapitel 4 – Verbesserungsvorschläge: In diesem Kapitel werden Verbesserungsvorschläge vorgestellt und Vor- und Nachteile bewertet.

Kapitel 5 – Umsetzung von Verbesserungen: Nachfolgend werden aus den vorgestellten Verbesserungsvorschlägen einige Vorschläge für die Umsetzung ausgewählt und das Ergebnis der Umsetzung dokumentiert.

Kapitel 6 – Technische Umsetzung: Dieses Kapitel beschreibt die Architektur von RevAger, wie diese verändert wurde und welche Probleme es bei der Umsetzung der Verbesserungsvorschläge gab.

Kapitel 7 – Evaluation: Dieses Kapitel beschreibt die Forschungsfragen sowie die Planung, Durchführung und Ergebnisse der Evaluation. Ferner werden Gefahren für die Validität erläutert, sowie die abgeleiteten Hypothesen präsentiert.

Kapitel 8 – Zusammenfassung und Ausblick: Das letzte Kapitel fasst diese Arbeit zusammen und erläutert Möglichkeiten, weitere Untersuchungen durchzuführen.

2. Hintergrund

In diesem Kapitel werden Grundlagen über Review-Techniken zusammengefasst, sowie der geschichtliche Kontext berücksichtigt. Außerdem werden Herausforderungen und Stärken von Review-Techniken erläutert.

2.1. Geschichte und Definition des Review-Begriffs

Fagan [Fag76] hat in den 1970ern einen Prozess beschrieben, um Fehler in der Softwareentwicklung frühzeitig zu erkennen. Dieser Prozess wird *Fagan Inspektion* [Lig09] genannt. Die Fagan Inspektion ist eine formale, effiziente und wirtschaftliche Methode der Qualitätssicherung. Der Ablauf einer Fagan Inspektion ist definiert durch einen *Overview*, in dem gegebenenfalls abgegrenzt und definiert wird, welche Artefakte berücksichtigt werden sollen. In der darauffolgenden individuellen *Vorbereitung* bereiten sich die beteiligten Gutachter vor, indem sie versuchen, das Design und die Logik der Artefakte zu verstehen. In der *Inspektion* oder Inspektionssitzung werden die Artefakte vom sogenannten Leser vorgestellt. Dabei wird das Ziel verfolgt, Fehler in den Artefakten zu finden. Die Fehlerfindung soll also primär in der Inspektionssitzung geschehen. In der Phase der *Korrektur* werden Fehler korrigiert und anschließend wird in der sogenannten *Follow-up*-Phase überprüft, ob alle Fehler behoben wurden. Fagan konstatiert, dass die Kosten, einen Fehler durch eine Fagan Inspektion zu finden und zu beheben, um den Faktor 10 bis 100 kleiner seien, als diesen Fehler in einem späteren Schritt der Softwareentwicklung zu finden und zu beheben.

Ungefähr 30 Jahre später resümiert Fagan [Fag02] Vorteile von *Fagan Inspektionen*. Dabei nennt er – neben der Fehlerreduktion – beispielsweise Teambildung, Wissensweitergabe und Erfahrungsgewinn als weitere Nutzen von Inspektionen. Er betont, wie auch in [Fag76], dass die Fagan Inspektion als Qualitätssicherungsmaßnahme zwar viel Zeit benötige, aber insgesamt zu einer Zeitersparnis von 9–40 % führe. Außerdem schreibt er, dass die Fagan Inspektion, so wie er sie konzipiert hat, Fehler im *Entwicklungsprozess* aufdecken soll. Er betont, dass die Begriffe *Inspektion* und *Fagan Inspektion* für andere Techniken und Prozesse verwendet werden, welche nicht die notwendige Qualität aufweisen würden. Teilweise wird der Begriff *Inspektion* als Oberbegriff für *Review-Techniken* verwendet, so wie bei Parnas und Lawford [PL03]. Deswegen wird im Folgenden die Terminologie des *IEEE Standard for Software Reviews and Audits* [08] (IEEE 1028), mit Ergänzungen von Ludewig und Lichter [LL12], verwendet, um die verschiedenen Techniken klar voneinander abzugrenzen. Wird in der Literatur eine andere Terminologie verwendet, so wird eine äquivalente „Übersetzung“ angestrebt.

Der *IEEE Standard for Software Reviews and Audits* [08] wurde 1989 in einer ersten Version [IEE89] veröffentlicht und definiert verschiedene Rollen im Review-Prozess, sowie fünf Review-Techniken, welche im Weiteren als *formale* Review-Techniken bezeichnet werden. Um Verwechslungen zu

2. Hintergrund

vermeiden, werden die dortigen Terminologien verwendet. Alle definierten Techniken folgen einem ähnlichen Prozess, weshalb nachfolgend eine generische Prozessbeschreibung erläutert wird und Abweichungen einzelner Techniken später angebracht werden. Dieser Prozess ist folgender:

1. Damit ein Review durchgeführt werden kann, muss selbiges bei der *Projektplanung* berücksichtigt werden.
2. Wenn ein Review angestoßen wird, beginnt der Moderator mit der *Planung* des Reviews. Dazu gehört, das Review-Team auszuwählen und die Rollen im Review den entsprechenden Personen zuzuordnen. Weiterhin muss der Review-Termin angesetzt und der Prüfling verteilt werden.
3. In einem *Kick-off* wird ein Überblick über den Prüfling und den Review-Prozess gegeben. Abhängig von der Technik ist diese Sitzung in der Norm optional oder es ist erlaubt, diese Sitzung mit der Review-Sitzung zu vereinen.
4. Bevor die eigentliche Review-Sitzung stattfinden kann, müssen die Gutachter in der *individuellen Vorbereitung* den Prüfling inspizieren. Die Herangehensweise in der Vorbereitung unterscheidet sich teilweise je nach Technik.
5. In der *Review-Sitzung* werden Befunde gesammelt, diskutiert und klassifiziert. Die Klassifikation von Befunden ist im Standard vorgesehen, aber es sind keine festen Klassifikationen vorgeschrieben. Für die Inspektion sind mögliche Klassifikationen beispielhaft angegeben. Ludwig und Lichter klassifizieren Befunde in Technischen Reviews nach den in Tabelle 2.1 gezeigten Klassifikationen. Der *Notar* dokumentiert die Ergebnisse dieser Phase.

Tabelle 2.1.: Klassifikationen für Befunde nach [LL12].

Kritischer Fehler	Der Fehler ist so gravierend, dass wesentliche Aspekte des Prüflings nicht dem Einsatzzweck gerecht werden. Dieser Fehler muss unbedingt vor Freigabe behoben werden.
Hauptfehler	Ein Hauptfehler sollte vor der Freigabe des Prüflings behoben werden, da dies ein grundlegender Fehler ist, welcher sich deutlich auf den Einsatzzweck auswirkt.
Nebenfehler	Nebenfehler müssen nicht vor der Freigabe behoben werden, sollten aber bei Gelegenheit behoben werden.
Gut	Der begutachtete Bestandteil enthält keine Fehler.

6. Nach der Sitzung behebt der Autor die Befunde. Der Moderator überprüft, ob in der *Nacharbeit* oder auch *Korrektur* alle Befunde behoben wurden. Eventuell wird eine weitere Review-Sitzung angesetzt.

In der Norm werden folgende Techniken definiert:

Inspektion Die Inspektion ist die formalste beschriebene Technik in IEEE 1028. [08] In einer Inspektion wird der gesamte Inspektionsprozess – also jeder Teilschritt dieses Prozesses – umfangreich dokumentiert, um Anhaltspunkte für Prozessverbesserungen zu sammeln. In der Vorbereitungsphase sollen gefundene Fehler an den Moderator gemeldet werden, damit er die Inspektionssitzung gegebenenfalls verschieben kann. Diese Möglichkeit sollte

berücksichtigt werden, wenn in der Vorbereitung zu viele Fehler gefunden werden. Die Norm stellt sogenannte Inspektionsraten zur Verfügung, welche beschreiben, wie viele Seiten bzw. Programmzeilen ein Gutachter pro Stunde begutachten sollte, abhängig von der Art des Prüflings.

Die Inspektion hat das Primärziel, möglichst viele Fehler in der Vorbereitung zu finden. Im Gegensatz dazu verfolgt die *Fagan Inspektion* in der Vorbereitungsphase das Ziel, den Prüfling zu verstehen. [JT98]

Technisches Review Während die Inspektion das Primärziel hat, möglichst viele Fehler zu finden, hat das Technische Review das Primärziel, zu überprüfen, ob der Prüfling den Anforderungen und Vorgaben entspricht. Dazu suchen die Gutachter in der Vorbereitungsphase nach Fehlern, welche dann in der Review-Sitzung dokumentiert und klassifiziert werden. Aufgrund der Befundliste wird am Ende der Sitzung entschieden, ob der Prüfling den Anforderungen entspricht. Ludewig und Lichter [LL12] nennen für das Technische Review konkrete Vorschläge, welche Empfehlungen ausgesprochen werden können (siehe Tabelle 2.2). Diese Empfehlungen sind in *IEEE Standard for Software Reviews and Audits* [08] nicht explizit für das Technische Review – wohl aber für die Inspektion – vorgeschrieben.

Tabelle 2.2.: Empfehlung über die Annahme des Prüflings nach [LL12].

Akzeptiert ohne Änderungen	Es sind keine Änderungen erforderlich und der Prüfling ist angenommen.
Akzeptiert mit Änderungen	Es sind Änderungen erforderlich, der Prüfling ist aber angenommen und kein weiteres Review erforderlich.
Nicht akzeptiert	Der Prüfling muss korrigiert werden und ein weiteres Review muss durchgeführt werden.

Ludewig und Lichter nennen die *Dritte Stunde* als weiteren Bestandteil von Technischen Reviews. In der Dritten Stunde wird, ohne Protokoll und ohne Regeln, über die Dinge gesprochen, für die in der Review-Sitzung kein Platz oder keine Zeit war.

Walkthrough Das Walkthrough ist eine Review-Technik, die sich besonders zur Wissensweitergabe eignet. In der Review-Sitzung wird der Prüfling vom Autor schrittweise beschrieben und erklärt. Fragen können währenddessen gestellt werden. Dabei gefundene Fehler werden, wie in den anderen Techniken, dokumentiert und in der Nacharbeit behoben.

Management Review Das Management Review hilft bei der Überwachung des Projektfortschritts sowie der Einhaltung von Zeit- und Terminplänen. Außerdem soll es dabei unterstützen, Entscheidungen zu treffen.

Audit Der Fokus eines Audits ist es, ein unabhängiges Urteil über einen Prüfling zu erhalten. Der Auditleiter organisiert, analog zum Moderator in anderen Review-Techniken, das Review. Anders als bei anderen Techniken werden die Befunde außerhalb der Sitzung in einem Auditbericht dokumentiert, welcher nach der Abschlusssitzung finalisiert wird. Die Abschlusssitzung selbst dient dazu, vorläufige Beobachtungen, Ergebnisse und Empfehlungen der Auditoren zu präsentieren.

2.2. Herausforderungen des Review-Prozesses

Fagan sagt über die Review-Sitzung einer Fagan Inspektion:

„Successful execution is an obvious, but frequently overlooked, determinant in effective inspections“ [Fag02]

Wenngleich er dies über die Fagan Inspektion sagt, gilt dies auch für andere Review-Techniken, wie nachfolgend erläutert wird.

Ludewig und Lichter [LL12] beschreiben häufige Probleme und Gründe für das Scheitern von Technischen Reviews. Fehlende bzw. nicht vorgesehene Zeit in der Projektplanung für die Vorbereitung und Durchführung von Reviews ist einer dieser Gründe. Außerdem ist es wichtig, dass die beteiligten Personen ausreichend geschult sind (siehe [08]), dies trifft besonders auf den Moderator zu. Das Fehlen eines guten Moderators ist ein häufiges Problem in Reviews. Es muss beachtet werden, dass ein Review keine rein technische Angelegenheit ist, sondern auch soziale Komponenten beinhaltet. Entwickler haben teilweise Angst an Reviews teilzunehmen, weil sie sich vor Kritik fürchten. Ein *guter* Moderator weiß damit umzugehen.

Auch Liggesmeyer [Lig09] betont die Wichtigkeit des Moderators. Außerdem führt er aus, „dass Inspektionen als ein rein fachliches Instrument verstanden werden [müssen]“ [Lig09] und nicht als Bewertungsverfahren von persönlicher Leistung, da der Autor sonst dazu verleitet ist, Fehler abzustreiten, was die Effizienz der Inspektion mindern könnte. Deshalb sollen Vorgesetzte an solchen Sitzungen nicht teilnehmen. Es besteht ebenfalls die Gefahr, dass sich die Inspektion „im Kreis dreht“ [Lig09], wenn keine geeigneten Maßnahmen ergriffen werden. Des Weiteren sollen in Reviews nur Fehler dokumentiert werden und wenn nötig müssen diese auch diskutiert werden, nicht aber sollen *Lösungsansätze* erörtert werden.

2.3. Notwendigkeit von Review-Sitzungen

Votta Jr [Vot93] legt in seiner Arbeit aus dem Jahr 1993 den Fokus auf die Review-Sitzung von Technischen Reviews¹ und beschreibt Probleme dieser. Er untersucht – mit Verweis auf Eick et al. [ELL+92] – den *Synergieeffekt* in Review-Sitzungen. Dazu hat er in der individuellen Vorbereitungsphase gefundene Befunde vor der Sitzung gelabelt. Alle Befunde, die nach der Sitzung nicht gelabelt waren, sind durch Synergie, also durch die gemeinsame Diskussion gefunden worden: Im Mittel 4 % der Befunde. Votta Jr konstatiert, dass *Synergie nicht aufträte*. Er beschreibt, dass es drei wichtige Fähigkeiten gebe, die i. d. R. in einem Review-Team vorhanden sein müssen, damit Synergie geschieht, welche da sind:

1. Interpersonale Fähigkeiten, wie zwischenmenschliche Kommunikation.
2. Die Auffassungsgabe, die Sachlage rational zu erfassen sowie
3. die Fähigkeit, Aufgaben sinnvoll aufzuteilen.

¹Ursprünglich wird das Wort *inspection* verwendet. Es ist erkennbar, dass das *Technische Review* gemeint ist.

Sind alle diese Fähigkeiten vorhanden, können persönliche Schwächen der Teilnehmer durch das Review-Team kompensiert werden und so Synergieeffekte entstehen. Außerdem stellt er fest, dass die Review-Sitzung keinen Einfluss darauf habe, wie viele *falsch positive* Befunde in der Korrektur als solche identifiziert werden. In anderen Worten: Das Fehlen einer Review-Sitzung erhöhe nicht den Korrekturaufwand des Autors durch mehr falsch positive Befunde. Andererseits erläutert er, dass die Kosten für die spätere Behebung eines *falsch negativen* Befunds wesentlich höher seien als die Kosten um einen *falsch positiven* Befund in der Korrekturphase zu identifizieren. Weiterhin hat er ermittelt, dass es 4 % falsch negative Befunde gebe. Er schlussfolgert, dass unbedingt mögliche falsch positive Befunde erfasst werden sollen, da es kostengünstiger sei, diese Befunde in der Korrekturphase erneut zu überprüfen, als Gefahr zu laufen, Fehler zu übersehen und später beheben zu müssen. Als weiteren Aspekt beschreibt er, dass in Review-Sitzungen im Durchschnitt nur vier der Anwesenden aktiv sind. Die restlichen $n - 4$ Beteiligten folgen nur 30 % bis 80 % der Zeit dem Geschehen. Bei einer angenommenen Sitzungsdauer von zwei Stunden gehen $\frac{2}{5} \cdot (n - 4)$ bis $\frac{7}{5} \cdot (n - 4)$ Entwicklerstunden verloren. Votta Jr beschreibt, dass die *Wartezeit* auf die Review-Sitzung, verursacht durch Terminkonflikte, die Kosten des gesamten Entwicklungsprozesses um 13 % erhöhen würde.

Johnson und Tjahjono [JT98] vermuten, dass der Grund für die unterschiedlichen Ergebnisse bezüglich dem Synergieeffekt von Fagan und Votta Jr darin begründet sind, dass die Inspektion und das Technische Review das Ziel der Review-Sitzung unterschiedlich definieren; nämlich Befund-*Erkennung* versus Befund-*Dokumentation*. Sie vergleichen – da dies Votta Jr [Vot93] nicht untersucht – ob Inspektionen mit Sitzung oder Technische Reviews ohne Sitzung effizienter sind. Das Ergebnis ihres kontrollierten Experiments ist, dass sitzungsbasierte Inspektionen signifikant teurer sind, aber signifikant weniger falsch positive Befunde hervorbringen. Die Anzahl der richtig positiven Befunde hingegen unterscheidet sich nicht signifikant. Allerdings sehen Johnson und Tjahjono die Möglichkeit, dass die Effizienz der Fehlerfindung einzelner Personen in Technischen Reviews ohne Sitzung steigen könne. Im Gegensatz zu Votta Jr, der Technische Reviews untersucht, finden sie bei Inspektionen einen Synergieeffekt: 30 % der Befunde würden in der Sitzung identifiziert.

2.4. Informelle Review-Techniken

Die in *IEEE Standard for Software Reviews and Audits* definierten Review-Techniken sind ein Teil der möglichen Techniken, um ein Review durchzuführen. Mit der Zeit haben sich, neben diesen *formalen* Techniken, diverse Techniken für verschiedene Einsatzgebiete und mit unterschiedlichem Vorgehen etabliert. Parnas und Lawford [PL03] bezeichnen Reviews, welche mit weniger Aufwand, verglichen mit der Fagan Inspektion oder Techniken aus IEEE 1028, umgesetzt werden können, als *informelle Review-Techniken*. Als ein – vermutlich extremes – Beispiel ist hier Pair-Programming zu nennen, was nach Hoffmann [Hof13] die Grundgedanken von Reviews übernehme und sie schon früher, nämlich bei der Entstehung des Programmcodes, anwende. Nach Williams et al. [WKCJ00] kann hier im weiteren Sinne von einem informellen Review gesprochen werden.

Bacchelli und Bird [BB13] definieren darüber hinaus, dass eine Technik ein *modernes Code Review* ist, wenn sie a) informell, b) werkzeugunterstützt und c) in der Praxis verbreitet ist. Ihre Arbeit untersucht die Motivationen sowie Auswirkungen von modernen Code Reviews im Softwareunternehmen Microsoft. Um die Motivation für die Durchführung von Reviews zu erfassen, führen

2. Hintergrund

sie Umfragen mit Entwicklern² durch. Dabei finden sie heraus, dass die größte Motivation für Reviews das Finden von Fehlern ist, gefolgt von der Verbesserung des Programmcodes, Finden von alternativen Lösungen, Förderung von Bewusstsein und Transparenz im Team und der Wissensweitergabe. Weiterhin untersuchen sie, ob die Erwartungen und Motivationen sich in den Auswirkungen wiederfinden. Dazu analysieren sie aufgezeichnete Kommentare des eingesetzten Review-Werkzeugs *CodeFlow*. Dabei stellen sie fest, dass Reviews mehr zur Verbesserung von Programmcode beitragen, als sie helfen Fehler zu finden. Sie erörtern, dass dieses Ergebnis nicht auf ihr Sample zurückzuführen sei. Als einen Grund nennen sie, dass sie in Interviews festgestellt haben, dass Entwickler teilweise die niedrige Qualität von Reviews bemängeln; unter anderem, dass auf Formatierungsfehler geachtet wird und schwerwiegendere Fehler oft nicht gefunden werden. Obwohl Bacchelli und Bird nicht damit gerechnet haben, Belege dafür zu finden, dass Wissensweitergabe durch Reviews geschieht, haben sie Kommentare gefunden, die z. B. der Dokumentation und damit der Wissensweitergabe galten.

Rigby und Bird [RB13] untersuchen die Unterschiede zwischen formalen und modernen Reviews, dafür analysieren sie Review-Daten von insgesamt 13 Projekten. Dazu gehören Daten von Microsoft, *Gerrit*³-basierte Reviews bei Google, Reviews durchgeführt mit dem Tool *CodeCollaborator*⁴ bei AMD sowie einige Open-Source-Projekte, welche Reviews per Mailinglisten oder mit Hilfe von webbasierten Tools durchführen. Sie kommen zu dem Schluss, dass sich der Review-Prozess in vielen Unternehmen und Projekten ähnlich entwickelt habe. Als Gemeinsamkeiten stellen sie folgende Punkte vor: Moderne Reviews seien leichtgewichtig und flexibel. Diese würden früh, schnell und oft durchgeführt. Der Umfang der Änderungen sei tendenziell klein und meistens seien nur zwei Gutachter in das Review involviert, bei komplexeren Änderungen eventuell mehr. Moderne Reviews dienen, in den untersuchten Unternehmen und Projekten, nicht nur der Fehlerfindung, sondern sind ein Mittel der gemeinsamen Lösungsfindung.

McIntosh et al. [MKAH16] behandeln in ihrer Arbeit den Einfluss von modernen Reviews auf die Softwarequalität. Dafür extrahieren und analysieren sie Review-Daten aus den Open-Source-Projekten Qt⁵, VTK⁶ und ITK⁷. Um die Qualität dieser Projekte zu bewerten, wurde die Anzahl der Fehler ermittelt, welche innerhalb der ersten sechs Monate nach einem Release entdeckt wurden. Diese Fehler bezeichnen sie als *post-release defects*. McIntosh et al. untersuchen weiter die Zusammenhänge zwischen der Anzahl von *post-release defects* und einigen Eigenschaften von durchgeführten Reviews. Zu diesen Eigenschaften gehört der Grad an Mitwirkung, beispielsweise die aufgewendete Zeit oder die Anzahl an Kommentaren, der Gutachter in einem Review. Als weitere Eigenschaft betrachten sie die Expertise der Gutachter, welche durch die Anzahl vorheriger Änderungen an einer Datei bestimmt wird. Bei beiden dieser Eigenschaften beobachteten McIntosh et al., dass sie einen positiven Einfluss auf die Anzahl von *post-release defects* haben.

² Außerdem befragen sie Manager über deren Motivation Reviews durchführen zu lassen. Diese Ergebnisse werden hier nicht berücksichtigt.

³<https://www.gerritcodereview.com/>

⁴<https://smartbear.com/product/collaborator/overview/>

⁵<https://www.qt.io/>

⁶<http://www.vtk.org/>

⁷<https://itk.org/>

2.5. Zusammenfassung

In diesem Kapitel wurden *formale* Review-Techniken beschrieben und definiert. [08; Fag76] Es wurden Nutzen [Fag02], Probleme und Herausforderungen von formalen Techniken erläutert; beispielsweise hohe Anforderungen an den Moderator. [Lig09; LL12] Ferner wurde in [JT98; Vot93] der Aufwand von Review-Sitzungen deutlich. Weiterführend wurden Merkmale von *informellen* und *modernen* Reviews erläutert [BB13; RB13] und der Einfluss dieser auf die Softwarequalität untersucht [MKAH16].

3. Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt und zusammengefasst, welche beschreiben wie Prozesse verbessert werden können. Dabei werden die Aspekte der Werkzeugunterstützung, Entscheidungsfindung in Gruppen sowie soziale Faktoren in Prozessen berücksichtigt.

3.1. Werkzeugunterstützung im formalen Review-Prozess

Dieser Abschnitt geht auf Werkzeuge im formalen Review-Prozess ein: Leider existieren kaum aktuelle Veröffentlichungen dazu, deshalb sei auf das Alter der Veröffentlichungen hingewiesen.

Macdonald et al. [MMB+95] vergleichen unterschiedliche Review-Werkzeuge¹ und beschreiben, dass Review-Werkzeuge eine gewisse Sorgfältigkeit erzwingen können. Als Beispiel nennen sie die Zuordnung und Beachtung von Checklisten durch Gutachter in der Vorbereitungsphase. Außerdem schreiben sie, dass diese Eigenschaft bei Inspektionen vorteilhaft sei, da so Daten erhoben werden können, die bei der Prozessverbesserung helfen. Weiterhin können Review-Werkzeuge dafür verwendet werden, um zu erfassen, ob die investierte Vorbereitungszeit der Gutachter ausreichend ist oder eine Teilnahme an der Sitzung nicht sinnvoll ist.

Macdonald und Miller [MM98] vergleichen die Effizienz von papierbasierten Review-Sitzungen mit werkzeugunterstützten Sitzungen. Neben Problemen und Herausforderungen beschreiben sie Vorteile und Anforderungen an die Werkzeuge. Ein Vorteil ist, dass Dokumente elektronisch vorliegen und so das Durchsuchen und Referenzieren dieser möglich ist. Daraus ergibt sich der Vorteil, dass beim Dokumentieren von Befunden die Position des Fehlers eindeutig gekennzeichnet werden kann. Macdonald und Miller schlagen vor, einen Abstimmmechanismus zu verwenden, um die Diskussionen zu beschleunigen. Sie warnen, dass ein Werkzeug nicht zu komplex sein soll. Außerdem weisen sie darauf hin, dass eine überwiegend mausbasierte Bedienung eine Verlangsamung beim Dokumentieren verursachen könne. Abschließend ermitteln sie in einem kontrollierten Experiment, dass papierbasierte Review-Sitzungen nicht effizienter seien als werkzeugunterstützte Sitzungen.

3.2. Entscheidungsfindung in Gruppen

Dalkey und Helmer [DH63] beschreiben das Delphi-Verfahren, mittels dem systematisch und zuverlässig eine *Meinung* bzw. *Konsens* einer Gruppe eingeholt werden kann. Dabei wird versucht, den Einfluss der Beteiligten aufeinander zu minimieren. Um dies zu erreichen, werden die einzelnen Personen in mehreren *Runden* zur zentralen Frage in Interviews oder durch Fragebögen befragt

¹Auf den Vergleich und die Werkzeuge wird hier nicht weiter eingegangen.

3. Verwandte Arbeiten

und so der direkte Austausch der Gruppe vermieden. Nach den einzelnen Befragungsrunden werden der Gruppe Informationen bereitgestellt, die von einzelnen Personen in der Befragung als relevant eingestuft wurden. Die letzte Befragungsrunde stellt die endgültige Einschätzung der Gruppe fest.

Grenning [Gre02] schreibt, dass es bei der Aufwandsschätzung von Stories im Planning Game zwei wesentliche Probleme gebe: Das Schätzen dauere lange und nicht alle Teilnehmer würden sich daran beteiligen. Grenning beschreibt ein Vorgehen, um diese Probleme zu lösen: Als Erstes wird die Story vom Kunden vorgestellt und eventuelle Unklarheiten geklärt. Jeder Entwickler gibt nun verdeckt seine Aufwandsschätzung ab. Dazu legt jeder eine Spielkarte mit der Aufwandsschätzung verdeckt vor sich auf den Tisch. Wenn alle beteiligten Personen dies getan haben, drehen alle gleichzeitig ihre Karte um. Wird so ein Konsens gefunden, wird mit der nächsten Story fortgefahren. Wird hier kein Konsens gefunden, werden die Gründe für die unterschiedlichen Schätzungen diskutiert. Kann auch so kein Konsens gefunden werden, kann die Story verschoben oder aufgeteilt werden oder es wird die niedrigste Schätzung gewählt. Dieses Vorgehen habe den Vorteil, dass alle Entwickler sich beteiligen *müssen* und dass die Schätzung anderer die eigene nicht beeinflusst. Grenning beschreibt weiter, dass die Diskussion so effizienter sei: Die Diskussionen seien kürzer und zielgerichteter. Wichtige Diskussionen fänden aber trotzdem statt.

3.3. Soziale Faktoren

Singer und Schneider [SS12a] beschreiben, dass viele Methoden der Softwaretechnik nicht oder nicht optimal genutzt werden. Sie schlagen in ihrer Arbeit eine Methode vor, um die Akzeptanz solcher Methoden durch „soziale Software“ zu verbessern. Als Beispiele für soziale Software nennen sie soziale Netzwerke. Als Element von sozialer Software identifizieren sie den *Nutzer*, welcher *Beziehungen* aufbauen und *Inhalte* erstellen kann. Dieser Inhalt kann *geteilt* und durch *Annotationen* erweitert werden. Singer und Schneider dokumentieren, dass soziale Software den Nutzer zur Verhaltensänderung anregen kann, indem er mit Aktivitäten anderer Nutzer konfrontiert wird. Außerdem sei Gamifikation, welche die Motivation steigern könne, ein weiterer Einflussfaktor. In Singer und Schneider [SS12b] wurde diese Methode erfolgreich erprobt.

Den Aspekt von Gamifikation in informellen Reviews betrachten Khandelwal, Sripada und Reddy [KSR17] in ihrer Arbeit aus dem Jahr 2017. Dort untersuchen sie, basierend auf ihrer früheren Arbeit [SRK16], wie hilfreich Kommentare in informellen Reviews sind. Dazu haben sie ein Experiment mit 183 Studierenden, die den Programmcode anderer Studierenden inspizierten, durchgeführt. Als Plattform für diese informellen Reviews kamen zwei gamifizierte Plattformen (Github CC² und Codebrag³), die beispielsweise das „Liken“ von Codeänderungen ermöglichen, sowie zwei nicht gamifizierte Plattformen (Bitbucket⁴ und Phabricator⁵), zum Einsatz. Die Autoren konnten die Nullhypothese nicht ablehnen, was – ihrer Aussage nach – einigen nicht kontrollierten Variablen geschuldet sei. Sie schlagen vor, diese Probleme zu beheben und ein weiteres Experiment durchzuführen, denn ihre Umfrage habe ergeben, dass fast 54 % der Probanden Gamifikation im Review-Prozess sehr begrüßen würden.

²<https://github.com/tzachz/github-comment-counter>

³<http://codebrag.com/>

⁴<https://bitbucket.org>

⁵<https://phacility.com>

3.4. Zusammenfassung

In Anschluss an Kapitel 2 wurde in diesem Kapitel beschrieben wie Prozesse, beispielsweise Sitzungen in Technischen Reviews, verbessert werden können. Dabei wurde auf wichtige Eigenschaften von Review-Werkzeugen eingegangen. [MM98; MMB+95] Anschließend wurden soziale Faktoren wie Gamifikation [SRK16], soziale Software [SS12a] und Beteiligung an der Entscheidungsfindung [DH63; Gre02] thematisiert.

4. Verbesserungsvorschläge

In diesem Kapitel werden Verbesserungsvorschläge für RevAger gesammelt, beschrieben und bewertet. Inspiriert sind diese Verbesserungsvorschläge durch eigene Erfahrungen mit Sitzungen verschiedener Art. Durch Brainstorming und Diskussionen mit interessierten Personen sind aus diesen Ideen die nachfolgenden Verbesserungsvorschläge entstanden. Die Auswahl der Verbesserungsvorschläge für die Umsetzung ist in Kapitel 5 beschrieben.

4.1. Vorschlag I – Befunde kollaborativ verfassen

In RevAger existiert momentan keine Möglichkeit, um Befundtexte vorzubereiten. Das bedeutet, dass der Notar jeden Befund in der Review-Sitzung formulieren muss. Die anderen Beteiligten können dazu beitragen, indem sie Befunde klar formulieren. Es ist aber auch denkbar, in RevAger Möglichkeiten des kollaborativen Schreibens zu schaffen.

Dazu verbindet sich jeder Gutachter mit einem Tablet zu einem, in RevAger eingebetteten, Webserver. Auf seinem Tablet kann jeder Gutachter eigenständig genau einen Befund-Vorschlag formulieren (siehe Abbildung 4.1).

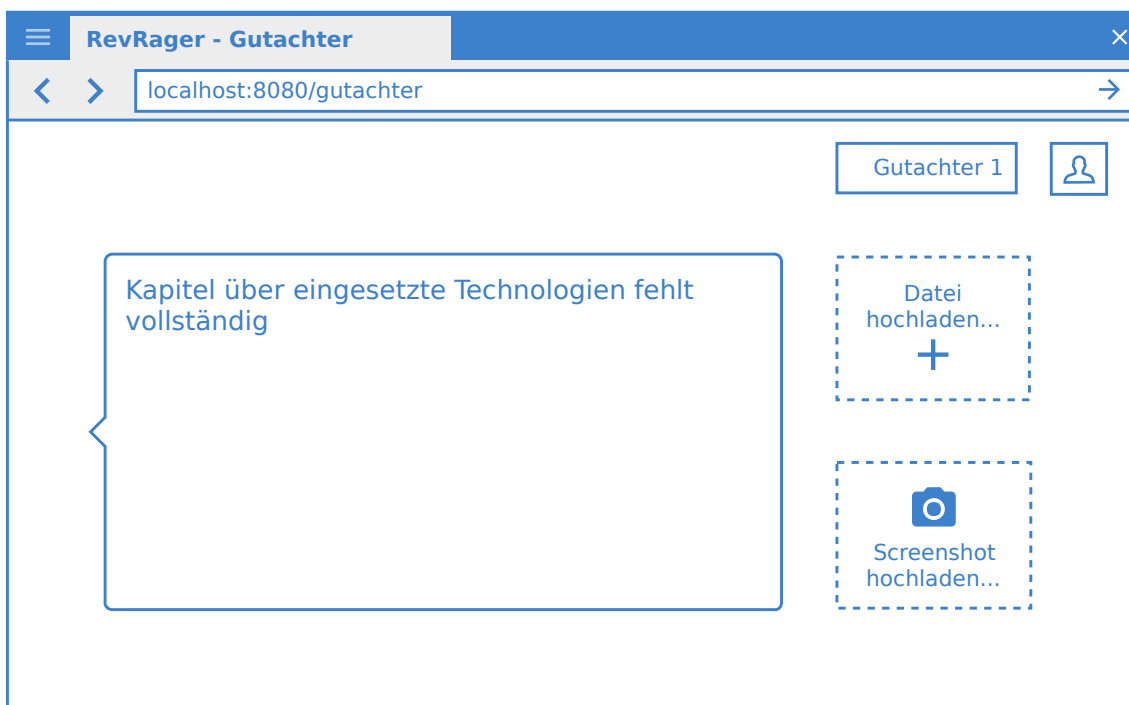


Abbildung 4.1.: Gutachter verfasst während der Sitzung seinen nächsten Befund.

4. Verbesserungsvorschläge

Der Moderator sieht auf seinem Tablet die vorformulierten Befund-Vorschläge aller Gutachter und bewertet diese mit 1–5 Sternen (siehe Abbildung 4.2). Nachdem der aktuelle Befund dokumentiert ist und der Notar einen neuen Befund anlegt, wird automatisch der Befund mit der höchsten Bewertung in RevAger übernommen. Sowohl der Befund und die Bewertung auf dem Tablet des Moderators als auch der Befundtext auf dem Tablet des Verfassers werden zurückgesetzt. Ab diesem Zeitpunkt kann nun nur der Notar den Befund bearbeiten und der Befund kann wie gewohnt diskutiert werden. Der Gutachter, der diesen Befund eingebracht hat, kann nun einen neuen Befund auf seinem Tablet formulieren. Die Befunde der anderen Gutachtern bleiben erhalten.



Abbildung 4.2.: Moderator wählt den nächsten Befund aus.

Diese „Warteschlange“ könnte folgende Vorteile haben:

- Gutachter können den nächsten Befund vorbereiten. Dies kann unter Umständen eine Zeitersparnis bedeuten und die Inaktivität der Gutachter (siehe Votta Jr [Vot93]) verringern. Außerdem ist sofort klar, was zur Diskussion steht; Gutachter müssen sich nicht erst einigen, was Gegenstand des aktuellen Befunds ist.
- Da jeder Gutachter nur einen Befund in die Warteschlange einreihen kann, ist der Moderator in der Lage, potenziell kritische Befunde vorzuziehen. Außerdem kann es sein, dass Gutachter wichtigere Befunde zuerst vorschlagen.

Durch die Verwendung von Tablets können folgende Probleme entstehen:

- Gutachter verwenden eventuell ihre Tablets dafür, annotierte PDFs oder Notizen anzuzeigen. Durch diese Doppelbenutzung kann es passieren, dass der Interaktion oder den eigenen Notizen zu wenig Aufmerksamkeit geschenkt wird.

- Das Vorhandensein von Tablets kann dazu führen, dass Gutachter sich durch andere Aktivitäten ablenken lassen, besonders da die Bewertung über eine Webseite läuft; weitere Tabs mit ablenkendem Inhalt sind in Reichweite.
- Die Verwendung eines eigenen Webservers in RevAger kann in manchen Umfeldern zu Problemen führen. Zum Beispiel müssen sich alle beteiligten Personen in das richtige WLAN einwählen.

4.2. Vorschlag II – Limitierung der Redezeit

Der Moderator einer Review-Sitzung hat einen großen Einfluss auf den Erfolg eines Reviews. Seine Aufgabe ist es, Sorge zu tragen, dass alle Gutachter angemessen an der Diskussion teilnehmen und kein Teilnehmer dominiert. Um die Aufgabe zu erfüllen, ist der Vorschlag, die Redezeiten aller Teilnehmer zu erfassen und zu visualisieren. Damit soll der Moderator bei der Moderation der Diskussion unterstützt werden. Die Umsetzung soll technisch gestützt sein, sodass die Erfassung und Visualisierung automatisch erfolgen kann. Dazu kann in RevAger ein Statusbereich eingeführt werden, in welchem der prozentuale Anteil der Redezeit aller Teilnehmer visualisiert wird.

Die Identifikation der sprechenden Person kann durch mehrere Mikrofone im Raum, beispielsweise an jedem Sitzplatz, realisiert werden. Jeder Person wird ein Mikrofon zugeordnet. Aus den Pegeln der Mikrofone kann abgeleitet werden, welche Person wie lange spricht. Eine weitere Lösung ist die Verwendung einer Bibliothek zur Sprachidentifikation, wie zum Beispiel *Recognito*¹.

Mögliche Vorteile der Visualisierung von Redezeiten:

- Das Bewusstsein für den Umfang von Diskussionen und die Verteilung von Redezeiten könnte helfen, Probleme diesbezüglich zu erkennen.
- Der Moderator kann *Zahlen* heranziehen, wenn eine Person gebeten wird, sich kurz zu fassen. Es könnte für Gutachter sogar ein Anreiz sein, sich in der Diskussion kurz zu fassen.

Nachteile und Herausforderungen:

- Beide vorgeschlagenen technischen Lösungsansätze haben Schwächen. Die ausschließliche Betrachtung der Pegel der Mikrofone, welche an jedem Platz platziert sind, ist fehleranfällig. Das Ändern der Sitzposition oder der Lautstärke des Sprechens einer Person wird eine Änderung des Pegels verursachen.

Die Verwendung der *Recognito* Bibliothek ist nicht sinnvoll, da sie in einem kurzen Versuch nicht zuverlässig funktioniert hat und, laut Dokumentation, nur unter guten Bedingungen funktioniert.

- Die Aufzeichnung von Stimmen ist datenschutzrechtlich relevant und kann in der Praxis problematisch sein.

¹<https://github.com/amaurycrickx/recognito>

4.3. Vorschlag III – Befunde im Vorfeld formulieren

In der Review-Sitzung werden die im Vorfeld gesammelten Befunde zusammengetragen, diskutiert, bewertet und dokumentiert. Um diesen Prozess zu beschleunigen, können Befunde schon vor der Sitzung von den Gutachtern formuliert werden. Der Moderator kann Duplikate identifizieren und die Sitzung besser strukturieren. Ziel dieses Vorschlags ist es, die benötigte Zeit zur Dokumentation eines Befunds zu verkürzen. Außerdem werden so mögliche falsch positive Befunde erfasst, so wie Votta Jr [Vot93] dies empfiehlt.

Um diesen Vorschlag umzusetzen, soll RevAger als Webanwendung neu implementiert werden. Der Review-Prozess kann dadurch insgesamt besser von RevAger unterstützt werden: Der Moderator erstellt das Review und setzt einen Sitzungstermin an, dabei kann der Kalender der Teilnehmer eingebunden werden oder eine Umfragefunktion implementiert, um Terminkonflikte zu vermeiden. Anschließend lädt der Autor den Prüfling hoch und gibt externe Referenzen an. In der individuellen Vorbereitung geben die Gutachter ihre fertig formulierten Befunde in die Webanwendung ein. Der Moderator kann das Review aufgrund der schon formulierten Befunde strukturieren, indem Befunde sortiert und Duplikate identifiziert werden. Abbildung 4.3 zeigt einen Entwurf für solch eine Oberfläche.



Abbildung 4.3.: Links die Befunde für die Sitzung und rechts die Befunde der Gutachter.

Auf der rechten Seite sind die Befunde der Gutachter zu sehen. Der Moderator kann von dort per Drag-and-drop Befunde in die Befundliste für die Sitzung (links) übernehmen. Übernommene Befunde werden in der Befundliste des Gutachters grau hinterlegt. Hat ein weiterer Gutachter diesen Befund ebenfalls dokumentiert, kann der Moderator diesen Befund, ebenso per Drag-and-drop, mit dem Duplikat in der Befundliste für die Sitzung verknüpfen. In der Befundliste

für die Sitzung ist beispielsweise zu sehen, dass zwei Gutachter Rechtschreibfehler gefunden haben; anstatt mehrere Befunde zu haben, werden alle Befunde bezüglich Rechtschreibung zu einem Befund zusammengefasst. In der Review-Sitzung werden die zusammengefassten und deduplizierten Befunde noch kurz diskutiert und anschließend klassifiziert.

Daraus ergeben sich, mit Verweis auf [Vot93], möglicherweise folgende Vorteile:

- Eine Zeitersparnis während der Sitzung, da Befunde bereits formuliert sind.
- Der Moderator kann Duplikate identifizieren und die Sitzung besser strukturieren.
- Die Webanwendung kann die Vorbereitungszeit automatisch ermitteln und Gutachter können zeitnah daran erinnert werden, sich vorzubereiten. Bei Bedarf können sie sogar aus einer Sitzung ausgeschlossen werden, wenn abzusehen ist, dass eine Teilnahme nicht zielführend ist.
- Alle im Vorfeld dokumentierten Befunde werden, bei ordnungsgemäßem Ablauf, in der Review-Sitzung thematisiert. So werden auch potenziell falsch positive Befunde dokumentiert, so wie dies Votta Jr [Vot93] empfiehlt.
- Solch eine Webanwendung kann der Grundstein für eine soziale Anwendung sein, so wie sie Singer und Schneider [SS12a] vorschlagen.

Als mögliche Nachteile wurden identifiziert:

- Dieses Vorgehen bringt einige Veränderungen mit sich gegenüber dem üblichen Ablauf eines Technischen Reviews. So werden die Befunde nicht mehr in der Sitzung formuliert und die benötigte Zeit zum Erfassen eines Befunds kann kürzer werden. Dies könnte dazu führen, dass andere Gutachter den Befund nicht nachvollziehen können oder eine wichtige Diskussion über den Befund nicht mehr stattfindet.
- Insgesamt würde die Umsetzung dieses Vorschlags eine große Änderung des Vorgehens bedeuten. Dabei bleibt unklar, welche negativen Effekte dadurch entstehen können.
- Eine Neuimplementierung von RevAger als Webanwendung einschließlich aller bereits existierenden Funktionen ist im Rahmen dieser Bachelorarbeit nicht umsetzbar.

4.4. Vorschlag IV – Interaktion mit Tablets

In der Review-Sitzung soll über die Befunde und die Klassifikationen dieser ein Konsens gefunden werden. [LL12] Das bedeutet, dass ein Moderator bei jedem Befund sicherstellen muss, dass ein Konsens vorliegt. In einer Pilotbeobachtung an der Dualen Hochschule Baden-Württemberg in Stuttgart ist aufgefallen, dass die Konsensfindung oft problematisch ist. Ziel dieser Verbesserung ist es, den Moderator zu entlasten und zu unterstützen, indem die beteiligten Personen Feedback geben können.

Dazu soll in RevAger ein Webserver implementiert werden, sodass Sitzungsteilnehmer den Text des aktuellen Befunds sehen und diesen selbstständig klassifizieren können, so wie Macdonald und Miller [MM98] dies empfehlen. Weiterhin sollen Gutachter in der Review-Sitzung die Möglichkeit haben, Feedback zu geben. Die erste Form eines Feedbacks soll ein *Pausenwunsch* sein. Als Zweites soll darauf hingewiesen werden können, dass der *Fokus der Diskussion* verloren geht.

4. Verbesserungsvorschläge

In RevAger soll dabei nicht gezeigt werden, welche Person welche Klassifikation oder welches Feedback gibt. Es soll nur angezeigt werden, wie viele Personen eine Bewertung oder Feedback gegeben haben. Abbildung 4.4 zeigt einen Entwurf der den Gutachtern angezeigten Webseite mit den möglichen Feedback- und Abstimmmöglichkeiten.

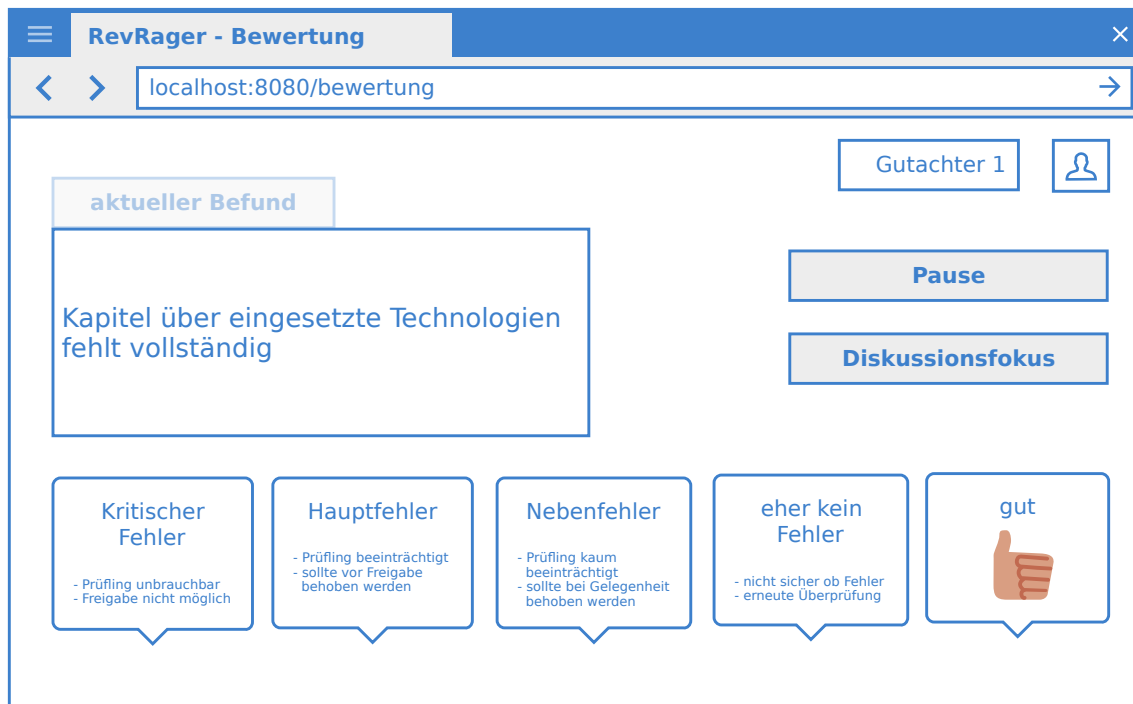


Abbildung 4.4.: Gutachter können interaktiv Befunde klassifizieren und Feedback geben.

Vorteile dieses Vorschlags sind:

- Das Feedback kann dem Moderator helfen, die Diskussion effizienter zu gestalten.
- Ein Abstimmmechanismus kann die Diskussion beschleunigen [MM98] und Gutachter beeinflussen sich nicht gegenseitig [DH63]. Außerdem werden die Gutachter expliziter in die Entscheidungsfindung eingebunden, was die Motivation steigern kann. [BBH+08]
- Basierend auf Votta Jr [Vot93] wird vorgeschlagen, die in der Literatur üblichen Klassifikationen für Technische Reviews (siehe Tabelle 2.1 auf Seite 14) um die Klassifikation „eher kein Fehler“ zu erweitern. Diese Klassifikation soll bedeuten, dass nicht sicher ist, ob es ein Fehler ist. Durch eine erneute Überprüfung durch den Autor in der Korrekturphase können eventuelle Fehler identifiziert werden. Zudem können höhere Kosten, die entstehen würden, wenn dieser Fehler erst später behoben wird, vermieden werden.
- Anwesende können Feedback geben, ohne eine Person unterbrechen zu müssen oder sich überwinden zu müssen. Eventuell kann die Hürde gesenkt werden, Gedanken der Art „Wir drehen uns im Kreis!“ oder „Wir sollten weitermachen!“ zu äußern. Solche Gedanken sind mir aus eigener Erfahrung bekannt. Analog zur „Pause“-Karte, wie sie im Planning Game eingesetzt wird, kann der Wunsch nach einer Pause angebracht werden. Khandelwal, Sripada

und Reddy [KSR17] haben – allerdings nicht in formalen Reviews – ähnliche Aspekte aus dem Bereich Gamifikation untersucht, beispielsweise das „Liken“ von Programmcodezeilen.

Das Verwenden von Tablets hat einige inhärente Probleme, wie beim Vorschlag über das kollaborative Verfassen von Befunden (Vorschlag I) beschrieben wurde.

4.5. Vorschlag V – Interaktion mit Gamecontrollern

Alternativ zur Interaktion mit Tablets (Vorschlag IV) können Gamecontroller verwendet werden, welche über ein USB-Hub an den Laptop des Notars angeschlossen werden. Die Tasten auf dem Gamecontroller ermöglichen, analog zu Vorschlag IV, die Interaktion: das Klassifizieren eines Befunds oder das Geben von Feedback. Außerdem kann – je nach Gamecontroller-Modell – haptisches Feedback genutzt werden, um eine Eingabe zu signalisieren: Wenn ein Gutachter eine Bewertung abgibt, können die anderen Gutachter, durch ein Vibrieren des Gamecontrollers daran erinnert werden, ebenfalls abzustimmen. Abbildung 4.5 zeigt einen handelsüblichen Gamecontroller, wie er eingesetzt werden kann.



Abbildung 4.5.: Gutachter können mittels Gamecontroller interagieren.

Dieser Vorschlag hat dieselben Vorteile wie Vorschlag IV, aber vermeidet dessen Nachteile. Im Folgenden sind nur abweichende Nachteile beschrieben:

- Für die Verwendung von Gamecontrollern muss eine spezielle Bibliothek verwendet werden. Die Unterstützung von verschiedenen Betriebssystemen und verschiedenen Gamecontrollern ist von dieser Bibliothek abhängig. Das bedeutet, dass es zu unvorhersehbaren Problemen bei der Umsetzung oder Evaluation kommen kann.

4.6. Vorschlag VI – Feedback zum Zeitverlauf geben

In der Review-Sitzung werden mehrere Befunde nacheinander diskutiert und dokumentiert. Die Idee ist, dass gemessen wird, wie lange dies dauert. Das Ziel ist es, *problematische Situationen*, wie beispielsweise lange Diskussionen, zu erkennen und die Teilnehmer, besonders den Moderator, darauf hinzuweisen. Mögliche Ereignisse, die auf einen problematischen Verlauf der Sitzung hinweisen könnten, sind:

- Eine *Höchstdauer* für die Erfassung eines Befunds wird überschritten.
Beispiel: Es wird die Lösung eines Problems diskutiert oder die Diskussion „dreht sich im Kreis“.
- Die Dauer seit der *letzten Änderung* des aktuellen Befunds überschreitet einen bestimmten Wert.
Beispiel: Ein Befund wurde dokumentiert, die Beschreibung ist also vollständig. Gutachter könnten nun anfangen, Lösungsansätze zu diskutieren. Dies sollte unterbunden werden. [LL12]
- *Unterschreitung* einer minimalen Bearbeitungszeit eines Befunds.
Beispiel: Es wird für jeden Abschnitt in einer Spezifikation ein Befund *Rechtschreibung überprüfen* und *Falsche Schriftart* erstellt. Für Fehler dieser Art sollte ein einzelner Befund für den ganzen Prüfling erstellt werden. [LL12]
- Darüber hinaus können noch weitere Informationen, beispielsweise Feedback aus dem *Vorschlag IV* und *Vorschlag V*, berücksichtigt werden.
Beispiel: Drei von vier Gutachtern haben den aktuellen Befund klassifiziert, das eigentliche Problem ist also von der Mehrheit der Gutachter verstanden worden. Die Sinnhaftigkeit der Diskussion sollte vom Moderator überprüft werden.

Die Visualisierung dieser Ereignisse kann durch Farben geschehen: Die Hintergrundfarbe des Textfeldes für die Beschreibung des Befunds kann sich je nach Situation von Weiß, über Gelb zu Orange ändern. Aber andere Umsetzungen sind hier denkbar.

Argumente, die dieses Vorgehen unterstützen:

- Der Moderator wird bei seinen Entscheidungen bezüglich Diskussionsführung unterstützt.
- Die Einhaltung bzw. effiziente Nutzung der Zweistundengrenze für die Sitzung [Fag76; LL12] kann kontinuierlich unterstützt werden.

Dieser Vorschlag hat einige ungeklärte Punkte bzw. Nachteile:

- Es ist nicht klar, wie lange ein gutes Review-Team durchschnittlich für die Dokumentation eines Befunds benötigt und wie groß die Varianz um diesen Durchschnitt ist. Deshalb besteht das Risiko, dass zu kleine, aber auch zu große, Werte und Grenzen gewählt werden.
- Es ist offen, ob ein Feedback in Form von Fortschrittsanzeigen und Einfärben von Textfeldern verstanden und korrekt interpretiert wird.
- Übermäßige „Warnungen“ können Stress verursachen. [SHP+14]

4.7. Vorschlag VII – Sitzungsdashboard

Durch einige Vorschläge² entstand die Idee, ein Sitzungsdashboard zu erstellen, das verschiedene Informationen visualisiert. Abbildung 4.6 zeigt einen Entwurf dieses Sitzungsdashboards.

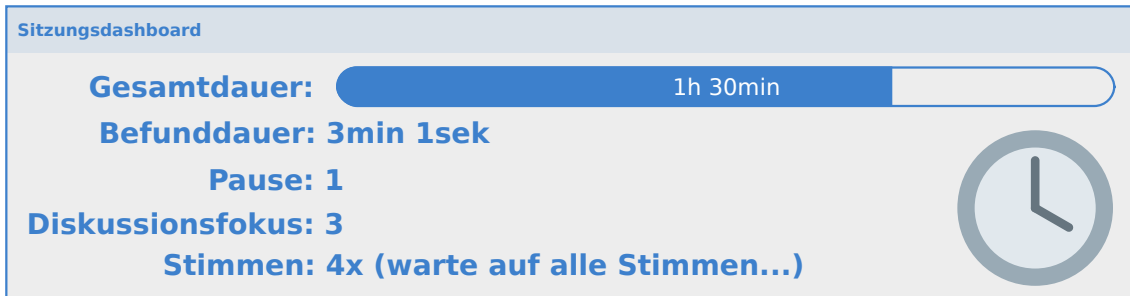


Abbildung 4.6.: Sitzungsdashboard mit Informationen zur Sitzung.

Die Informationen aus Abbildung 4.6 sind wie folgt zu interpretieren:

- Eine Fortschrittsanzeige der Gesamtsitzungsdauer, welche in RevAger konfigurierbar ist.
- Die gesamte Diskussionszeit des aktuellen Befunds.
- Der Wunsch nach einer Pause (siehe Vorschlag IV) wird angezeigt. Gutachter können diesen Wunsch mehrfach äußern. Wird das Review in RevAger pausiert, wird dieser Zähler zurückgesetzt.
- In Vorschlag IV/Vorschlag V wurde vorgeschlagen, dass Gutachter über ein Tablet oder Gamecontroller dem Moderator anonym das Feedback geben können, dass die Diskussion momentan nicht auf das Wesentliche fokussiert ist. Auch dieses Feedback kann pro Gutachter mehrfach gegeben werden.
- Die Anzahl der abgegebenen Klassifikationen eines Prüflings sowie die Klassifikationen selbst. Die Klassifikationen werden erst angezeigt, wenn alle Gutachter ihre Stimme abgegeben haben, damit sich Gutachter nicht gegenseitig beeinflussen. [DH63] Außerdem wird nicht angezeigt, welche Klassifikation von welchem Gutachter vergeben wurde.

Damit eine Änderung im Sitzungsdashboard nicht einem Gutachter, der sich gerade bewegt hat, um den Gamecontroller zu bedienen, zugeordnet werden kann, sollen bestimmte Werte nicht sofort, sondern erst nach einer zufälligen Verzögerung, zum Beispiel zwischen einer und fünf Sekunden, aktualisiert werden.

Ein Uhr-Piktogramm (siehe rechts unten in Abbildung 4.6) wird langsam eingeblendet, wenn bestimmte Schwellwerte erreicht sind, damit die Anwesenden dies wahrnehmen. Außerdem wird bei einer Änderung der Zähler für die *Pause* und den *Diskussionsfokus* die Zeile für circa fünf Sekunden fett dargestellt, damit diese Änderung bemerkt wird. Bei der Anzeige des Endergebnisses der Stimmen wird die Klassifikation mit den meisten Stimmen fett hervorgehoben.

²Limitierung der Redezeit (Vorschlag II), Interaktion mit Tablets (Vorschlag IV) bzw. Gamecontroller (Vorschlag V) und Feedback zum Zeitverlauf geben (Vorschlag VI)

4.8. Vorschlag VIII – Präsentationsansicht

In der aktuellen Version von RevAger dupliziert der Notar während einer Review-Sitzung seinen Bildschirm auf den Beamer, da RevAger über keine separate Präsentationsansicht verfügt. Das bedeutet, dass bei unterschiedlicher Beamer- und Bildschirmauflösung die kleinere gewählt werden muss und so der nutzbare Bildschirm- oder Beamerplatz verringert wird. Insgesamt ist der verfügbare Platz in der Anwendung bereits jetzt limitiert, wie Abbildung 4.7 zeigt. Dabei enthält die Befundliste einige Elemente, die nur für den Notar relevant sind:

- 1) Die Toolbar mit verschiedenen Buttons zum Speichern und Exportieren des Protokolls sowie Buttons zur Steuerung des Reviews.
- 2) Die Anzeige *aller* erfassten Befunde.
- 3) Verschiedene Buttons, zum Beispiel zum Verschieben, Löschen und Erstellen von Befunden, Referenzen, Dateien und Prüfaspekten.

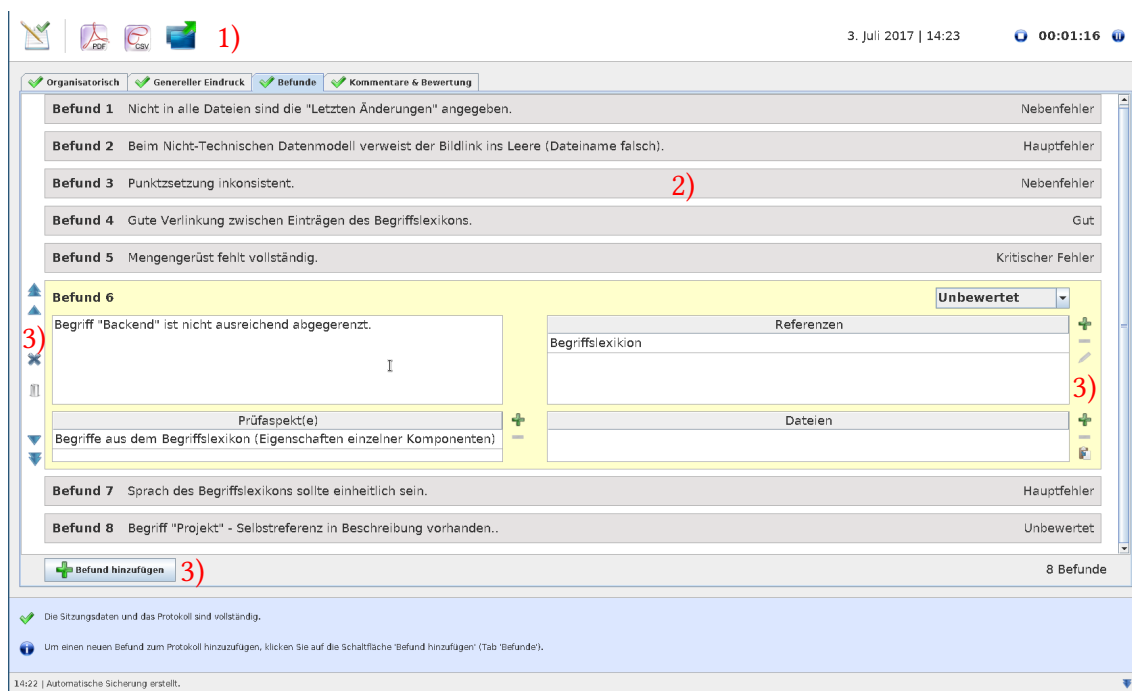


Abbildung 4.7.: Die Befundliste enthält Elemente, welche für Gutachter nicht relevant sind:
1) Die Toolbar, 2) *alle* vergangenen Befunde und 3) verschiedene Buttons.

Ziel dieses Vorschlags ist es, eine Präsentationsansicht zu konzipieren, die unabhängige Auflösungen und Seitenverhältnisse für Beamer und Bildschirm erlaubt und auf die Bedürfnisse des Moderators und der Gutachter optimiert ist. In dieser Ansicht sollen nicht benötigte Schaltflächen entfernt werden (siehe Abbildung 4.7) und größere Schriften verwendet werden. Weiterhin soll die Anzahl der angezeigten Befunde auf zwei begrenzt werden. Es sollen immer nur der zuletzt bearbeitete und der aktuelle Befund angezeigt werden. Im unteren Bereich soll das in Vorschlag VII vorgeschlagene Sitzungsdashboard integriert werden. Abbildung 4.8 zeigt einen Entwurf dieser Ansicht.


Vorheriger Befund: Befund 5		Kritischer Fehler
Mengengerüst fehlt vollständig.		
Aktueller Befund: Befund 6		Unbewertet
Begriff "Backend" ist nicht ausreichend abgegrenzt.	Referenzen	
	- Referenz 1	
	Dateien	
	- Datei 1	
	Prüfaspekte	
	- Prüfaspekt 1	
Sitzungsdashboard		
Gesamtdauer:	1h 30min	
Befunddauer: 3min	Isek	
Pause: 1		
Diskussionsfokus: 3		
Stimmen: 4x (warte auf alle Stimmen...)		
		

Abbildung 4.8.: Präsentationsansicht zur besseren Übersicht.

Dieser Vorschlag hat folgende Vorteile:

- Die RevAger-Entwickler haben selbst schon den Wunsch einer Präsentationsansicht dokumentiert³.
- Es ist möglich, Informationen größer und angemessener darzustellen und den verfügbaren Platz vollständig zu nutzen.
- Es besteht die Möglichkeit, Feedback zum Zeitverlauf (Vorschlag VII) angemessen in die Oberfläche zu integrieren.
- Das Sitzungsdashboard (Vorschlag VII) kann hier sinnvoll integriert werden.

³<https://blueprints.launchpad.net/revager/+spec/spec-separate-beamermode>

5. Umsetzung von Verbesserungen

In diesem Kapitel werden die zur Umsetzung ausgewählten Vorschläge genannt und das Ergebnis der Umsetzung gezeigt. Technische Details zur Umsetzung sind in Kapitel 6 zusammengefasst. Für die Umsetzung wurden folgende Vorschläge gewählt:

- Vorschlag V – Interaktion mit Gamecontrollern
- Vorschlag VII – Sitzungsdashboard
- Vorschlag VIII – Präsentationsansicht

Diese Vorschläge lassen sich gut kombinieren und sind in der verfügbaren Zeit realisierbar. Ferner sind bei diesen Vorschlägen keine Veränderungen des Review-Prozesses nach [08] notwendig.

5.1. Umsetzung von Vorschlag V: Interaktion mit Gamecontrollern

Abbildung 5.1 zeigt einen beschrifteten Gamecontroller, wie er in der Umsetzung eingesetzt wurde. Die Tasten sind mit Abkürzungen der möglichen Klassifikationen und Feedbackmöglichkeiten



Abbildung 5.1.: Gamecontroller mit Beschriftung.

5. Umsetzung von Verbesserungen

beschriftet. Da die Joysticks die gleiche Tastenbelegung wie einige Tasten haben, wurden diese abmontiert. Bei der Umsetzung wurde, aufgrund von Problemen mit den Gamecontrollern (siehe Abschnitt 6.4.3 auf Seite 44), kein Gebrauch vom haptischen Feedback gemacht. Der Effekt des Drückens einer Taste tritt, je nach Taste, zeitverzögert ein: Wird ein Befund klassifiziert, ist diese Klassifikation „sofort“ im System verfügbar. Bei dem Feedback „Diskussionsfokus“ oder „Pause“ hingegen wird zwischen einer und fünf Sekunden gewartet, bevor dies angezeigt wird.

5.2. Umsetzung von Vorschlag VII und VIII: Präsentationsansicht mit Sitzungsdashboard

Wegen der Ähnlichkeit dieser beiden Vorschläge wird das Ergebnis der Umsetzung in einem Abschnitt präsentiert. Wie bei der Vorstellung dieser Verbesserungen vorgeschlagen, wurde eine Präsentationsansicht für die Teilnehmer implementiert, wie in Abbildung 5.2 gezeigt.

The screenshot displays a presentation interface with the following components:

- Vorheriger Befund: Befund 5** (Previous Finding: Finding 5) - **Kritischer Fehler** (Critical Error). Description: Mengengerüst fehlt vollständig.
- Aktueller Befund: Befund 6** (Current Finding: Finding 6) - **Unbewertet** (Unrated). Description: Begriff "Backend" ist nicht ausreichend abgegrenzt.
- Referenzen** (References) - Begriffslexikon
- Dateien** (Files)
- Prüfaspekt(e)** (Assessment Aspect(s)) - Begriffe aus dem Begriffslexikon (Eigenschaften einzelner Komponenten)
- Sitzungsdashboard** (Session Dashboard) at the bottom showing:
 - Gesamtdauer: 90min 15sek
 - Befunddauer: 3min 9sek
 - Pause: 8
 - Diskussionsfokus: **12**
 - Stimmen: 1xKritischer Fehler; **2xHauptfehler;**
- A clock icon is located in the bottom right corner of the dashboard area.

Abbildung 5.2.: Präsentationsansicht zeigt Befunde und das Sitzungsdashboard an.

Diese Ansicht besteht aus drei Komponenten:

Vorheriger Befund In diesem Teil der Ansicht wird stets der vorher bearbeitete Befund angezeigt. Außer der Beschreibung des Befunds und der gewählten Klassifikation wird in diesem Teil nichts weiter angezeigt.

Aktueller Befund Analog zum oberen Teil der Ansicht wird an dieser Stelle der Befund angezeigt, welcher vom Notar angeschaut bzw. bearbeitet wird. Im Gegensatz zum *vorherigen Befund* werden in dieser Komponente weitere Informationen zum Befund angezeigt: Referenzen, Dateien und zugeordnete Prüfaspekte.

5.2. Umsetzung von Vorschlag VII und VIII: Präsentationsansicht mit Sitzungsdashboard

Sitzungsdashboard Im Sitzungsdashboard wird die Gesamt- und Bearbeitungsdauer der Befunde visualisiert. Außerdem wird das Feedback der Gutachter angezeigt. Wird der *Pause-* oder *Diskussionsfokus*-Wert geändert, wird dieser für kurze Zeit hervorgehoben; in Abbildung 5.2 ist dies beim *Diskussionsfokus* der Fall. Weiterhin werden die Klassifikationen der Gutachter angezeigt; dies geschieht erst, wenn alle Gutachter ihre Klassifikation abgegeben haben. Nachdem dies eingetreten ist, wird in der letzten Zeile des Sitzungsdashboards das Ergebnis aufgeschlüsselt. Dabei werden nur vergebene Klassifikationen aufgeführt und nach der Schwere sortiert: *Kritischer Fehler, Hauptfehler, Nebenfehler, eher kein Fehler, Gut*. Sind alle vergebenen Klassifikationen gleich häufig gewählt, werden diese mit normaler Schriftgröße dargestellt. Ist dies nicht der Fall, werden die Klassifikationen, mit den meisten Stimmen fett und die anderen Klassifikationen 20 % kleiner dargestellt.

Zusätzlich dazu wurde eine Ansicht mit allgemeinen Informationen zur Sitzung erstellt. Diese ist in Abbildung 5.3 zu sehen. Bestandteil dieser Ansicht sind die Freitextfelder, welche im Rahmen

Genereller Eindruck des Prüflings
Der Prüfling macht, bis auf einige Ausnahmen, einen guten Eindruck.

Abschließende Bewertung des Prüflings:
Akzeptiert, Akzeptiert mit Änderungen, Nicht akzeptiert, Nicht beendet, Nicht beendet (Änderungen nötig), Nicht beendet (komplette Überarbeitung nötig)

Kommentare zur Sitzung:
Die Gamecontroller haben die Sitzung enorm beschleunigt.

Kommentare zum Protokoll:
Aufgrund der kritischen Fehler kann der Prüfling nicht freigegeben werden.

Abbildung 5.3.: Präsentationsansicht zeigt allgemeine Informationen zur Sitzung an.

einer Review-Sitzung ausgefüllt werden. Dazu gehört der *Generelle[r] Eindruck des Prüflings*, die *Abschließende Bewertung des Prüflings*, *Kommentare zur Sitzung* sowie *Kommentare zum Protokoll*. Diese Ansicht wurde erstellt, damit die Review-Teilnehmer, analog zu den Befunden, verfolgen können, was der Notar protokolliert. Sind die Felder vom Notar noch nicht ausgefüllt, wird ein hellblauer Platzhalter angezeigt, mit einer knappen Erklärung bzw. einem Hinweis bezüglich des Ausfüllens. In Abbildung 5.3 werden beispielsweise Vorschläge für die abschließende Bewertung des Prüflings aufgelistet. Ändert der Notar eines dieser Freitextfelder, wird das entsprechende Feld in der Ansicht gelb hinterlegt.

5.3. Verbesserungen an der Notaransicht

Zusätzlich zu den in Kapitel 4 vorgeschlagenen Verbesserungen wurden einige kleinere Funktionen für den Notar implementiert. Abbildung 5.4 zeigt einen Ausschnitt aus der Befundliste der Ansicht für den Notar. Der Notar sieht, neben der Befundnummer, das *aktuelle Meinungsbild* bezüglich

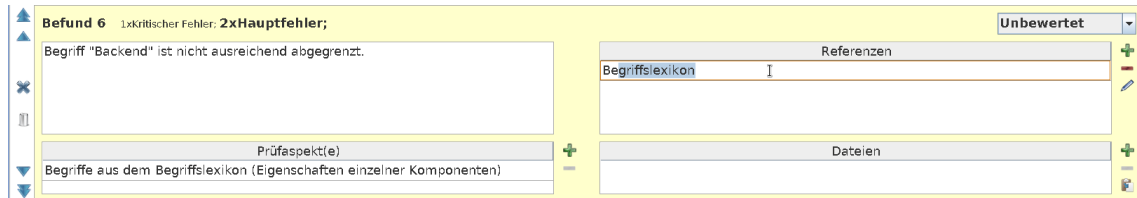


Abbildung 5.4.: Ausschnitt aus der Befundliste der Ansicht für den Notar.

der Klassifikation des Befunds. Im Gegensatz zur Präsentationsansicht sieht der Notar auch dann das vorläufige Meinungsbild, wenn noch nicht alle Gutachter ihre Einschätzung abgegeben haben. Eine weitere hinzugefügte Funktion ist die befundübergreifende *Autovervollständigung von Referenzen*. In der Oberfläche nicht sichtbar ist die neue Tastenkombination `Strg+N`, welche einen neuen Befund erstellt.

6. Technische Umsetzung

Dieses Kapitel beschreibt verwendete Bibliotheken, die Architektur von RevAger, Änderungen an der Architektur und Probleme bei der Umsetzung.

6.1. Beschreibung der Architektur und Quellcodequalität

Singletons und Gottklassen Die Architektur von RevAger ist durch eine extreme Verwendung des Singleton-Patterns gekennzeichnet. Dabei sind die Singletons nicht korrekt implementiert: Konstruktoren sind nicht als privat deklariert, sodass mehrere Instanzen erzeugt werden können. In RevAger existieren zusätzlich Gottklassen, die diese – als Singleton-Klassen kopierten – Klassen instanziiieren und notwendige *Getter* für diese Singletons zur Verfügung stellen.

Starke Kopplung zwischen Entitäten und Managementklassen Für viele Entitäten existiert eine sogenannte *Management*-Klasse. Diese übernehmen oft Funktionen, die in den dazugehörigen Entitäten angesiedelt sein sollten. Für Befunde bzw. die *Finding*-Klasse existiert beispielsweise das *FindingManagement*. Jeder Befund kann beliebig viele Referenzen haben. Problematisch ist, dass die Listen der Referenzen über einen *Getter* nach außen sichtbar sind. Das *FindingManagement* fügt selbstständig neue Referenzen zu den Listen hinzu. Hier wäre es besser, wenn die *Finding*-Klasse geeignete Methoden zum Hinzufügen einer Referenz anbieten würde.

Warnungen und Codesmells Generische Programmierung, welche seit Java 1.5 verfügbar ist, wird in dem Projekt an vielen Stellen nicht verwendet. Der Quellcode ist weiterhin durch hunderte Compiler-Warnungen und Codesmells (SonarLint) gekennzeichnet, dazu gehören z. B. nicht geschlossene Dateien und zu hohe Methodenkomplexität. Vor und während der Implementierung wurden einige dieser Probleme, in eigenen Branches, behoben und in die master-Branch des Projekts übernommen.

Boilerplatecode In dem RevAger-Projekt sind in vielen Klassen relativ viele Javadocs und Kommentare vorhanden. Leider erklären diese Kommentare nicht das *Warum*, sondern das offensichtliche *Was*. So sind die Javadocs von Konstanten in der Regel von der Form `The Constant <name of the constant>`.

6.2. Verwendete Bibliotheken

Vor der Umsetzung der Verbesserungsvorschläge war Java 1.6 die Standardlaufzeitumgebung. Dies wurde auf *Java 1.8* erhöht, um neuere Funktionen, wie Lambdas, verwenden zu können. Für die Implementierung der Gamecontroller wurden die *Jinput*¹ und *Jutils*² Bibliotheken verwendet. Beide stehen unter der BSD-Lizenz. Um Operationen mit Strings zu vereinfachen, wurde die *Commons Lang*³ Bibliothek des Apache Commons Projekts verwendet.

6.3. Erweiterung und Änderungen der Architektur

Im Folgenden ist erläutert, wie die Architektur von RevAger erweitert und geändert wurde, um die Verbesserungsvorschläge zu implementieren.

6.3.1. Ansteuerung der Gamecontroller

Für die Einbindung der Gamecontroller in RevAger wurde die Bibliothek *Jinput* verwendet. Folgende Klassen wurden von mir erstellt:

FindingStatus Der sogenannte FindingStatus ist eine Entität, welche zum Finding gehört; ein FindingStatus hat ohne ein Finding keine Semantik. Diese Klasse repräsentiert und kapselt die befundspezifischen Informationen: So werden dort die über die Gamecontroller abgegebenen Klassifikationen, die Diskussionszeit des Befunds sowie die Anzahl des *Diskussionsfokus* Feedbacks hinterlegt.

Dashboard Die Dashboard-Klasse ist dem FindingStatus übergeordnet. Sie verwaltet die Anzahl des *Pause*-Feedbacks und stellt die angezeigten Informationen in der Präsentationsansicht und Notaransicht zur Verfügung. Das Dashboard ist unabhängig von der Implementierung der Präsentationsansicht, da auch ohne diese die Gamecontroller verwendet werden können.⁴ Die Klasse ist als Singleton-Klasse konzipiert, da es logisch betrachtet per Review-Sitzung genau eine Instanz eines Dashboards geben darf.

ControllerManager Der ControllerManager kapselt die Funktionen der *Jinput*-Bibliothek und löst nach einem Tastendruck am Gamecontroller das gewünschte Ereignis aus. Das heißt, wenn an einem Gamecontroller die Klassifikation *Hauptfehler* gewählt wird, teilt er dies dem Dashboard mit.

DashboardEvent Die Kommunikation von ControllerManager zu Dashboard geschieht mittels Events und Time-outs. Der Grund dafür ist, dass einige Ereignisse ihren Effekt zeitverzögert zeigen. Der ControllerManager erstellt bei einem Ereignis ein neues Event, das auf den aktuellen Befund verweist. Ist das Time-out des Events abgelaufen, wird die gewünschte Methode – unter Berücksichtigung des Befunds, dem das Ereignis galt – aufgerufen. Dabei

¹<https://github.com/jinput/jinput>

²<https://web.archive.org/web/20170410015343/https://java.net/projects/jutils>

³<https://commons.apache.org/proper/commons-lang/>

⁴Das Ergebnis der Abstimmung über die Klassifikation wird dem Notar – auch ohne Präsentationsansicht – angezeigt.

ist das `DashboardEvent` eine abstrakte Klasse. Für die verschiedenen Ereignisse gibt es entsprechende Ableitungen: `ClassificationEvent`, `FocusEvent` und `BreakEvent`.

6.3.2. Präsentationsansicht mit Sitzungsdashboard

Um keine Abhängigkeiten zwischen der Notar- und Präsentationsansicht zu schaffen, wurde das Observer-Designpattern verwendet. Wenn der Notar einen Befund bearbeitet, wird der entsprechende `Setter` aufgerufen. Dieser markiert das Objekt als „geändert“ und alle Observer werden benachrichtigt. Beim Öffnen der Präsentationsansicht registriert diese auf allen Befunden einen Listener und wird über Änderung dieser informiert.

Aufgrund der Projektarchitektur und der Verwendung des Observer-Designpatterns musste an einigen Stellen die Kopplung zwischen Entitäten und deren Management-Klassen verringert werden bzw. der Zusammenhalt der Entitäts-Klassen erhöht werden (siehe Abschnitt 6.1 auf Seite 41). So hat beispielsweise das `FindingManagement` eine neue Referenz in das `Finding`-Objekt eingefügt, indem die neue Referenz eigenmächtig in die Liste der Referenzen eingefügt wurde. Damit das Observer-Designpattern hier richtig angewendet werden kann, muss der Observable wissen, wenn und was sich geändert hat. In anderen Worten, der `Finding`-Klasse wurden Methoden zum Hinzufügen und Löschen von Referenzen hinzugefügt.

6.4. Probleme bei der Umsetzung

Wie in Abschnitt 4.5 auf Seite 31 geschrieben, war bei der Implementierung von Vorschlag V damit zu rechnen, dass es plattformabhängige Probleme geben wird. Tatsächlich sind einige Probleme aufgetreten, welche nachfolgend erläutert werden.

6.4.1. Laden nativer Bibliotheken

`JInput` greift zur Ansteuerung der Gamecontroller, je nach Betriebssystem, auf `dll`- bzw. `so`-Dateien zurück, um die Hardware ansprechen zu können. Damit `JInput` diese Bibliotheken verwenden kann, müssen diese Dateien in Javas Bibliothekspfad vorhanden sein. Dazu müssen diese mittels `-Djava.library.path` der *Java Virtual Machine* zugänglich gemacht werden. Dabei müssen die Dateien auf der Festplatte abgelegt werden; sie können nicht in einer `jar`-Datei liegen. Das bedeutet, dass für die korrekte Funktionsweise nicht nur die fertige `exe`-Datei heruntergeladen und vorhanden sein muss, sondern auch die passenden Systembibliotheken.

Bei der Umsetzung habe ich mich bemüht, dies zu beheben, aber die Lösung dieses Problems nicht umgesetzt, da es für die Evaluation nicht relevant ist. Dieses Problem kann umgangen werden, indem die nativen Bibliotheken in der `jar`-Datei mitgeliefert werden und beim Starten in einen temporären Ordner kopiert werden. Mittels `System.load("/tmp/.../...")` kann beim Starten der Anwendung programmatisch die richtige Bibliothek zum Bibliothekspfad hinzugefügt werden.

6.4.2. Unterschiedliches Verhalten der Gamecontroller

Die Tasten der Gamecontroller haben auf Windows und Linux unterschiedliche Namen. Leider sind die Tastenbelegungen auf verschiedenen Windows-Versionen unterschiedlich. Dies ist dem Umstand geschuldet, dass *Input*, je nach Betriebssystem, auf verschiedene native Bibliotheken zurückgreift. Im Rahmen dieser Arbeit wurden die Namen der Tasten im Quellcode hartcodiert; dies ist für den praktischen Einsatz von Nachteil. Später sollte ein Assistent implementiert werden, über den die Tastenbelegung frei gewählt werden kann; dies könnte einmalig in den Einstellungen getan werden oder in jeder Sitzung festgelegt werden.

6.4.3. Haptisches Feedback

Bei der Implementierung des haptischen Feedbacks ist es regelmäßig zu Abstürzen der Gamecontroller gekommen. Um einen abgestürzten Gamecontroller verwenden zu können, muss RevAger neu gestartet werden, da die Bibliothek dies nicht anders unterstützt.⁵ Nach mehreren Anläufen, dieses Problem zu beheben wurde von der Verwendung von haptischem Feedback abgesehen. Ich vermute, dass dieses Verhalten auf den hohen Einschaltstrom zurückzuführen ist, welchen der USB-Hub nicht zur Verfügung stellen konnte. Möglicherweise kann hier ein besserer USB-Hub Abhilfe schaffen.

6.4.4. Referenzfeld verliert Fokus

Vom Notar wurde berichtet, dass das Referenzfeld eines Befunds in RevAger den Fokus verliere, wenn eine Taste auf dem Gamecontroller gedrückt wird. Diesen Fehler konnte ich nicht reproduzieren. Der Vollständigkeit halber sei dies hier erwähnt.

⁵Die Bibliothek stellt eine Singleton-Klasse zur Verfügung, mittels derer auf die, zum Zeitpunkt der Initialisierung verfügbaren, Gamecontroller zugegriffen werden kann. Wird ein Gamecontroller nach dieser Initialisierung entfernt und erneut angeschlossen, kann auf diesen Gamecontroller nicht mehr zugegriffen werden.

7. Evaluation

In diesem Kapitel werden die Planung und Durchführung der Evaluation erläutert und deren Ergebnisse präsentiert. Die Evaluation wurde mit dem Ziel durchgeführt, die folgende Forschungsfrage zu beantworten:

RQ1 Wie können Review-Sitzungen effizienter gestaltet werden?

Dabei soll in einem ersten Review untersucht werden, wie die Verbesserungsvorschläge sinnvoll in RevAger integriert werden können und was bei der Umsetzung berücksichtigt werden muss. Weiterhin soll dieses erste Review Daten und Hinweise für weitere Forschungsfragen liefern. In einem zweiten Review wird versucht, diese Forschungsfragen zu beantworten und die Auswirkungen der Veränderungen (siehe Kapitel 5 auf Seite 37) zu bewerten. Als Ergebnis sollen Hypothesen formuliert werden.

7.1. Planung der Evaluation

Es gab drei Möglichkeiten, die Verbesserungsvorschläge zu evaluieren:

- a) Den Einsatz der unmodifizierten sowie der modifizierten RevAger-Version in der Review-Übung *Einführung der Softwaretechnik*.
- b) Das Designen eines Experiments und die Akquirierung von Studierenden der Vorlesung *Einführung der Softwaretechnik*.
- c) Evaluation in einem Studienprojekt.

Für die Evaluation habe ich mich für Möglichkeit c) entschieden, da die Teilnehmer des Studienprojekts mindestens im fünften Semester sind und durch die Review-Übung in der Lehrveranstaltung *Einführung der Softwaretechnik* bereits Erfahrungen mit Technischen Reviews haben. Außerdem ist mit einem höheren Eigeninteresse der Teilnehmer zu rechnen, da die Prüflinge von ihnen erstellt und gefundene Fehler auch korrigiert werden. Zusätzlich dürfen die Teilnehmer die Zeitaufwände in ihrer Zeiterfassung eintragen. Die Nachteile und Gefahren der Möglichkeit c) sind in Abschnitt 7.4 auf Seite 59 nachzulesen.

Um die zwei RevAger-Versionen zu evaluieren, habe ich an zwei Technischen Review-Sitzungen dieses Studienprojekts teilgenommen und eine *Beobachtung* durchgeführt. Um den Verlauf der Sitzung nicht zu beeinflussen, war ich als passiver Teilnehmer anwesend und habe mich auf das Erstellen von Notizen beschränkt; Fragen oder Anmerkungen zum Verlauf des Reviews wurden nicht gestellt.¹ Insgesamt habe ich während der Beobachtungen neun Seiten an Notizen erstellt.

¹In der zweiten Review-Sitzung habe ich beim Erklären der Klassifikationen (siehe Unterabschnitt 7.3.1) versäumt, die eingeführte Klassifikation „eher kein Fehler“ zu erläutern. Dies wurde kurz nach Beginn der Sitzung nachgeholt.

Nach der Review-Sitzung wurden alle Teilnehmer von mir interviewt. Als Interviewform wurde das *unstrukturierte Interview* mit offenen Fragen² gewählt. Je nach Verlauf des Interviews wurde die Reihenfolge variiert und zusätzliche Fragen gestellt, um weitere Details zu erhalten. Die Interviews wurden, mit expliziter Erlaubnis der interviewten Personen, aufgezeichnet. Dabei wurde betont, dass die Aufzeichnungen nur auf meinem Computer gespeichert werden und keiner anderen Person zugänglich gemacht werden. Es wurden insgesamt über 280 Minuten aufgezeichnet, welche später von mir transkribiert wurden. Daraus entstanden über 30 Seiten an Interview-Notizen.

Außerdem wurden beide RevAger-Versionen geringfügig modifiziert, um bestimmte Ereignisse in eine *Log-Datei* zu protokollieren, damit grundlegende Daten zur Verfügung stehen. Geloggt wurde, wenn die Bearbeitung eines Befunds begonnen wurde³ und wenn die Befundliste verlassen wurde, um beispielsweise die abschließende Bewertung zu verfassen. Im zweiten Review wurden zusätzlich die Eingaben an den Gamecontrollern geloggt. Den Teilnehmern wurde erklärt, dass die Log-Dateien keine personenbezogenen Daten enthalten.

7.2. Erstes Review – unmodifizierte RevAger-Version

In der ersten Evaluation wurde nach Anhaltspunkten für sinnvolle Verbesserungen gesucht. Dabei wurden bestehende Verbesserungsvorschläge auf ihre Sinnhaftigkeit geprüft und es wurde untersucht, was bei der Umsetzung berücksichtigt werden muss. Im Rahmen der Forschungsfrage wurden für diese Evaluation folgende Unterfragen formuliert:

RQ1.1 Wie verändert sich die Diskussionsdauer über den Verlauf der Review-Sitzung?

RQ1.2 Welche Faktoren beeinflussen den zeitlichen Verlauf der Review-Sitzung?

RQ1.3 Welchen Einfluss hat die fehlende Anonymität bei der Klassifikation der Befunde auf die Review-Sitzung?

RQ1.4 Welche Interaktionsmöglichkeiten können geschaffen werden, um die Effizienz von Review-Sitzungen zu verbessern?

RQ1.5 Welche Faktoren haben einen Einfluss auf die Motivation?

7.2.1. Vorbereitungen für die Evaluation

Vor dem Review-Termin wurde dem Notar die RevAger-Version mit aktivierter Protokollierung zur Verfügung gestellt. Ebenso wurden die wichtigsten Funktionen von RevAger mit dem Notar besprochen, damit eine flüssige Bedienung während des Reviews gewährleistet war. Ein Auszug der Notizen der Beobachtung ist in Anhang A.1 gezeigt. Die Fragen des Interviews befinden sich in Anhang A.2.

²Im Interview zum ersten Review wurde bei einigen Fragen zusätzlich nach einer Bewertung auf einer Skala gefragt. Dabei ist der offene Charakter der Fragen erhalten geblieben.

³Die Diskussionsdauer von Befunden wurde aus der Startzeit des Befunds und dem nachfolgenden Ereignis berechnet.

7.2.2. Erkenntnisse bezüglich RQ1.1

Forschungsfrage RQ1.1: Wie verändert sich die Diskussionsdauer über den Verlauf der Review-Sitzung?

In der Review-Sitzung konnte beobachtet werden, dass in den letzten 30 Minuten die Review-Geschwindigkeit erhöht wurde. Teilweise wurde so schnell vorgegangen, dass Gutachter nicht die Zeit hatten zu überprüfen, ob für die aktuelle Passage des Dokuments ein Befund vorliegt bzw. sie mussten ihre Befunde nachtragen. Eine inhaltliche – nicht wörtliche – Wiedergabe einer solchen Situation (siehe Anhang A.1 auf Seite 69):

Moderator: Hat jemand was zu Anforderung 15? 16? 17?

Gutachter: Bei 16 hab ich etwas.

Diese Beobachtung konnte in den Interviews mittels Frage 10 (siehe Anhang A.2 auf Seite 70) und der Log-Datei bestätigt werden.

Weiterhin hat die Log-Datei gezeigt, dass zu Beginn der Review-Sitzung deutlich länger für die Dokumentation von Befunden benötigt wurde, als am Ende. Abbildung 7.1 zeigt die Diskussionszeit pro Befund in verschiedenen Phasen des Reviews. Dabei ist zu erkennen, dass im Verlauf der Sitzung die Diskussionszeit abgenommen hat. In den ersten 30 Minuten wurde im Mittel (Median)

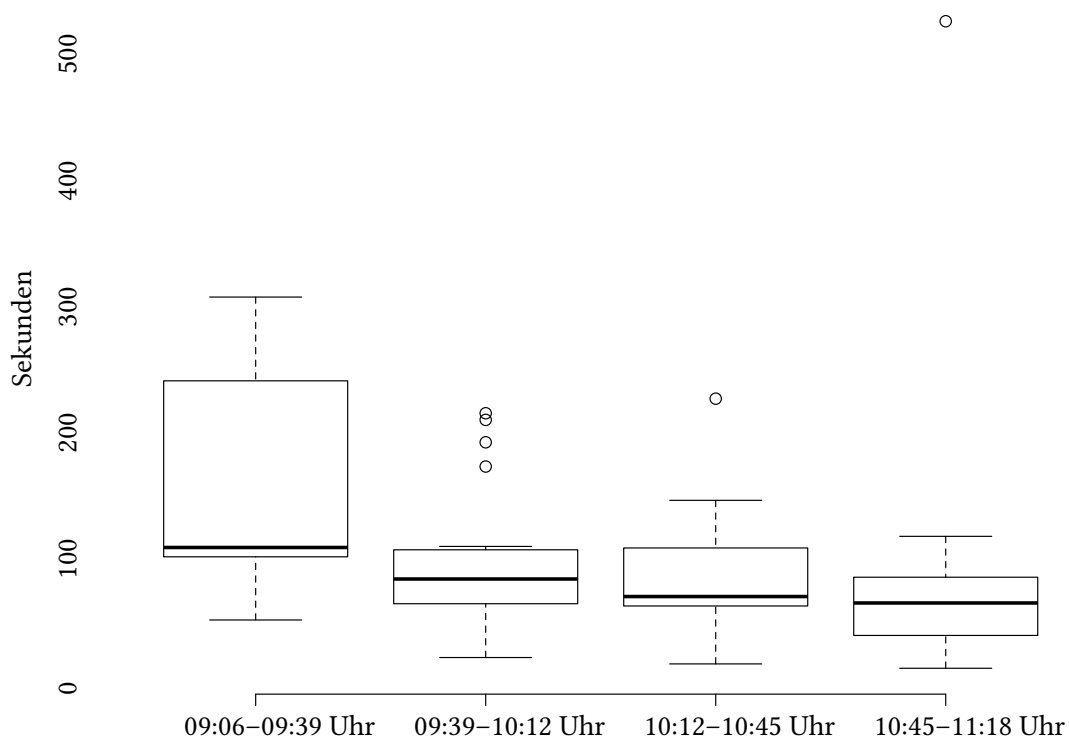


Abbildung 7.1.: Diskussionszeit im Verlauf der ersten Review-Sitzung.

am längsten diskutiert und in den letzten 30 Minuten am wenigsten. In der ersten halben Stunde wurden fünf Befunde dokumentiert und in den letzten 30 Minuten 23 Befunde. Die Differenz zwischen kleinstem und größtem Median beträgt 38,81s. In Abbildung 7.1 auf Seite 47 sind nur Befunde enthalten, die in der jeweiligen Zeitspanne erstellt wurden. Zwei gelöschte Befunde wurden, wegen Bearbeitungszeiten von drei bzw. vier Sekunden, vollständig ausgeschlossen.

Antwort zur Forschungsfrage RQ1.1: In dem beobachteten Studienprojekt konnte beobachtet werden, dass die Diskussionsdauer pro Befund zu Beginn des Reviews im Mittel höher ist; im weiteren Verlauf der Sitzung nimmt diese Dauer immer weiter ab.

7.2.3. Erkenntnisse bezüglich RQ1.2

Forschungsfrage RQ1.2: Welche Faktoren beeinflussen den zeitlichen Verlauf der Review-Sitzung?

Aus den Notizen der Beobachtung und Informationen aus den Interviews konnten einige Faktoren identifiziert werden, welche den zeitlichen Verlauf des Reviews möglicherweise beeinflusst haben:

Faktor I – Dokumentation zu langsam Als Einflussfaktor wurde in den Interviews und der Beobachtung identifiziert, dass die Teilnehmer des Reviews gelegentlich auf den Notar warten mussten, da dieser die Dokumentation des Befunds noch nicht abgeschlossen hatte. Dies wurde von zwei Gutachtern sowie vom Notar berichtet:

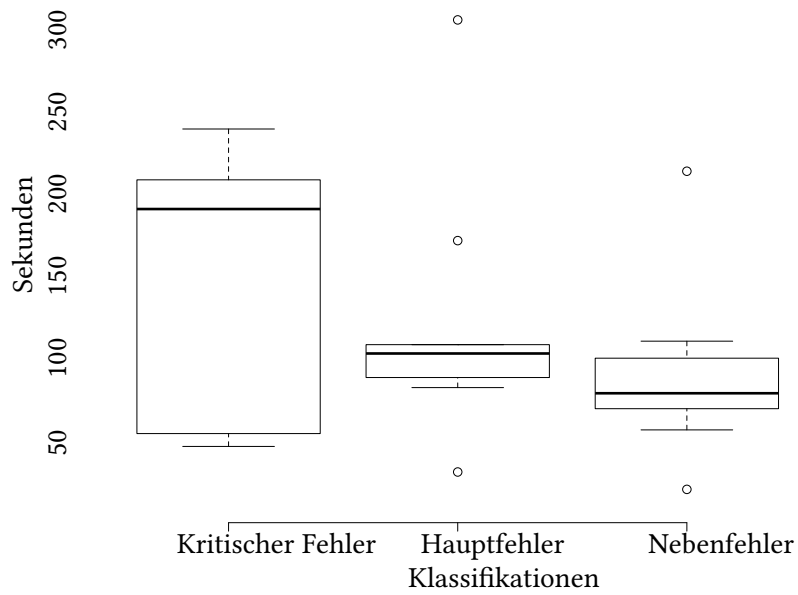
„Man hinkt [...] halt als Protokollant doch hinterher.“

Hier könnte eine Rolle spielen, dass der Notar erst noch erkennen muss, was Gegenstand des Befunds ist und wie dies formuliert werden muss bzw. wie ein Befund zu klassifizieren ist. In der Review-Sitzung konnte beobachtet werden, dass der Notar teilweise explizit um Feedback gebeten hat oder danach gefragt hat, was zu dokumentieren sei.

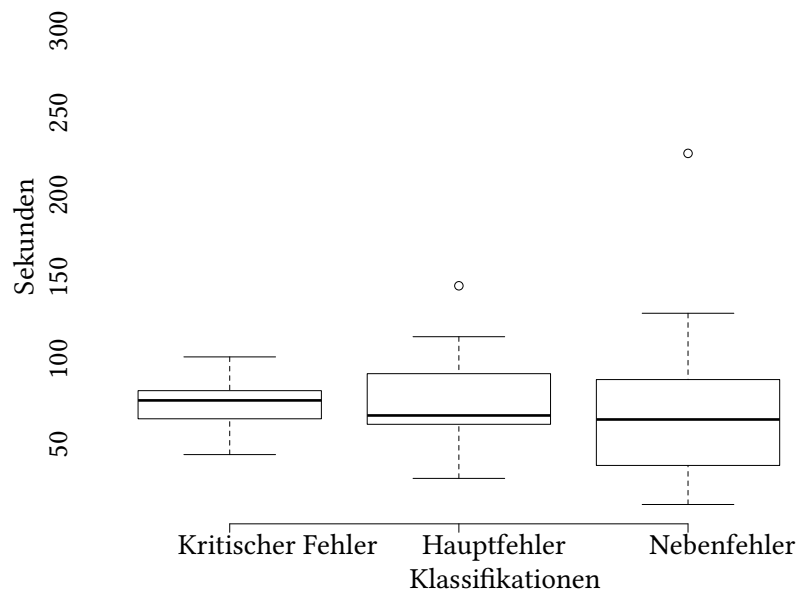
Faktor II – Unklare Definitionen der Klassifikationen Nach Aussagen des Review-Teams seien die Bedeutungen der Klassifikationen (siehe Tabelle 2.1 auf Seite 14) nicht klar gewesen und es habe verschiedene Auffassungen dieser gegeben. Einige Teilnehmer erklärten, dass sie die Definitionen der Klassifikationen nicht oder nur grob kennen. In der Sitzung gab es Diskussionen darüber, welche Bedeutung die Klassifikationen haben und wie Befunde einzuordnen seien. Ein Teammitglied hat das Vorgehen beim Klassifizieren wie folgt bewertet:

„Zum einen schwierig einzuschätzen, [welche Klassifikation zu vergeben ist] und [...], dass da jeder so ein bisschen was anderes drunter versteht.“

In der Review-Sitzung habe ich beobachtet, wie über die Klassifikation „Kritischer Fehler“ diskutiert wurde. Dies ist auch in der Log-Datei zu sehen: Abbildung 7.2a auf der nächsten Seite zeigt, dass in der ersten Hälfte der Sitzung, besonders bei Befunden mit der Klassifizierung „Kritischer Fehler“, deutlich länger benötigt wurde. In der zweiten Hälfte der Sitzung ist dieser Unterschied, wie in Abbildung 7.2b zu sehen ist, zurückgegangen. In den Boxplots in Abbildung 7.2 wurden insgesamt drei Befunde mit den Klassifikationen „Gut“, „Unbewertet“, sowie zwei gelöschte Befunde ausgeschlossen, da diese Klassifikationen nur in einer Sitzungshälfte vorkamen.



(a) Diskussionszeit für Befunde in der ersten Hälfte der ersten Review-Sitzung.



(b) Diskussionszeit für Befunde in der zweiten Hälfte der ersten Review-Sitzung.

Abbildung 7.2.: Vergleich der Diskussionsdauer unterschiedlicher Klassifikationen der ersten und zweiten Hälfte der Review-Sitzung.

Faktor III – Konsensfindung bei der Klassifizierung In der Beobachtung wurde kein *einheitliches* Vorgehen beim Klassifizieren der Befunde vorgefunden, sondern drei mögliche Variationen eines Vorgehens dokumentiert. Dabei war das Vorgehen jeweils davon geprägt, welche Rolle die Person innehatte:

Gutachter Die Befunde werden von den Gutachtern eingebracht und eine Klassifizierung selbstständig vorgeschlagen. Teilweise wurden die Vorschläge vom Notar übernommen, gelegentlich noch diskutiert. Dabei waren selten alle Gutachter in die Meinungsfindung involviert. Von Gutachtern wurde teilweise berichtet, dass später im Review die Lust, sich einzubringen, gesunken sei.

Moderator Zusätzlich zum unaufgeforderten Vorschlag einer Klassifikation durch Gutachter hat der Moderator teilweise explizit gefragt, wie ein Befund klassifiziert werden soll.

Notar Es konnte beobachtet werden, dass der Notar selbstständig eine Klassifikation ausgewählt hat. Gelegentlich hat er explizit um Feedback gebeten. Im Interview hat der Notar berichtet, dass er oft einfach eine Klassifikation festlegt und auf Feedback gewartet habe. Ein Schweigen habe der Notar als Zustimmung gewertet.

Antwort zur Forschungsfrage RQ1.2: Im Rahmen der Evaluation konnten drei Faktoren identifiziert werden, die möglicherweise einen Einfluss auf den zeitlichen Verlauf einer Review-Sitzung haben. Dazu gehört die Dokumentationsgeschwindigkeit des Notars (Faktor I) und Diskussionen über die Definition der möglichen Klassifikationen (Faktor II). Weiterhin ist aufgefallen, dass es kein eindeutiges und festgelegtes Vorgehen bei der Konsensfindung für Klassifikationen (Faktor III) gibt.

7.2.4. Erkenntnisse bezüglich RQ1.3

Forschungsfrage RQ1.3: Welchen Einfluss hat die fehlende Anonymität bei der Klassifikation der Befunde auf die Review-Sitzung?

In den Interviews wurde in Frage 8 und 9 (siehe Anhang A.2 auf Seite 70) versucht, Informationen über den Einfluss der fehlenden Anonymität bei der Klassifikation der Befunde zu erhalten. Dabei hat Frage 8 ergeben, dass drei Gutachter bei 0–20 % aller Befunde eine andere Meinung bezüglich der endgültigen Klassifikation von Befunden hatten. Zwei Gutachter haben angegeben, dass sie bei 20–40 % aller Klassifikationen anderer Meinung waren. Die Frage, wie oft eine abweichende Meinung geäußert wurde (Frage 9) hat keine Anzeichen geliefert, dass die fehlende Anonymität einen Einfluss auf die Klassifikation hat.

Darüber hinaus hat keine der befragten Personen die Atmosphäre des Reviews (siehe Frage 12) negativ bewertet; einzig eine Person hat geäußert, dass sich eine andere Person eventuell durch einige Befunde gekränkt gefühlt haben könnte. Dem gegenüber stehen fünf positive Bewertungen und Äußerungen über die Sitzung. So wurde die Atmosphäre beispielsweise als „locker“ bezeichnet.

Antwort zur Forschungsfrage RQ1.3: In der gegebenen Konstellation konnten keine Anzeichen gefunden werden, dass die fehlende Anonymität bei der Klassifikation problematisch ist.

7.2.5. Erkenntnisse bezüglich RQ1.4

Forschungsfrage RQ1.4: Welche Interaktionsmöglichkeiten können geschaffen werden, um die Effizienz von Review-Sitzungen zu verbessern?

Die Beobachtung und das Interview mit dem Notar haben gezeigt, dass dem Notar explizites Feedback geholfen hätte:

„Dass einfach die Leute [...] Handzeichen [geben] oder kurz sagen 'passt'“

Dabei hat er gesagt, dass das Feedback von mehr als einer Person gegeben werden sollte. In der Sitzung konnte beobachtet werden, dass der Notar explizit nachgefragt hat, wenn nicht klar war, was er zu dokumentieren hat. Auch wurde durch einige Anmerkungen in der Sitzung zum Ausdruck gebracht, dass mehr Feedback gewünscht wird. So hat der Notar gesagt:

„Ich kann's ja auch ändern, aber ich hör' ja nichts.“

Auch konnte beobachtet werden, wie der Notar in der Dropdownliste der Klassifikationen abwechselnd zwei verschiedene Klassifikationen ausgewählt hat und darauf gewartet hat, welche es sein soll. Andererseits hat der Notar im Interview erklärt, dass er ein Schweigen als Zustimmung gewertet habe. Ähnlich beschreibt der Moderator dies, in Anbetracht des Zeitdrucks, als angemessene Lösung:

„Solang' sich keiner beschwert, sind wohl alle einverstanden.“

Antwort zur Forschungsfrage RQ1.4: In dem beobachteten Review haben Moderator und Notar auf explizites Feedback seitens der Gutachter gewartet. Ist dies ausgeblieben, wurde dieser Umstand als Zustimmung gewertet (siehe auch Faktor III in Abschnitt 7.2.3 auf Seite 48). Folgende Interaktionsmöglichkeiten wären denkbar, um dieses Feedback zu geben:

- Feedback über die Vollständigkeit und Korrektheit der Befundbeschreibung.
- Bestätigung bzw. Einspruch über die gewählte Klassifikation eines Befunds.

7.2.6. Erkenntnisse bezüglich RQ1.5

Forschungsfrage RQ1.5: Welche Faktoren haben einen Einfluss auf die Motivation?

Hinweis: Die meisten der hier genannten Aspekte wurden von ein bis zwei Personen geäußert. Deshalb wurde weitestgehend darauf verzichtet, die Angaben zu quantifizieren. Die Aussagen dieses Abschnitts sind nicht als Verallgemeinerung zu verstehen.

Insgesamt wurde die Atmosphäre in der Review-Sitzung als *locker* und *witzig* bezeichnet. In dieser Atmosphäre könne man, nach Aussage einer Person, gut diskutieren. Unter anderem deshalb, weil sich niemand durch Befunde angegriffen oder gekränkt gefühlt habe (vgl. Abschnitt 7.2.4 auf Seite 50). Es wurde geäußert, dass sinnvolle Diskussionen mit anderen Spaß machen würden. Motivierend sei die Tatsache, dass der Prüfling durch das Review besser werde. Eine Person hat konkretisiert, dass es schön sei, ein Ziel zu haben. Abschließend ein Kommentar zum Review:

„Es war auf jeden Fall das beste [Review] was ich bisher gemacht hab, weil es einfach nicht zu zähflüssig war.“

Der Notar hat gesagt, dass es teilweise demotivierend gewesen sei, dass kein Feedback gegeben wurde (vgl. Abschnitt 7.2.5 auf Seite 51). Weiterhin wurde der Zeitdruck am Ende der Sitzung als demotivierend genannt:

„Der Zeitdruck am Ende ist nicht gerade motivierend [...], man hat keine Chance sich zu äußern.“

Antwort zur Forschungsfrage RQ1.5: Motivierende Aspekte sind, auch wenn die verfügbaren Daten nicht sehr groß sind: Eine angenehme Atmosphäre, sinnvolle Diskussionen und das Ziel, den Prüfling zu verbessern. Negativ auf die Motivation können sich Zeitdruck und mangelndes Feedback zum Notar auswirken.

7.2.7. Bedeutung der Erkenntnisse

Anhand der Erkenntnisse in diesem Abschnitt habe ich einige Forschungsfragen ausgewählt und diese, unter Berücksichtigung der umgesetzten Verbesserungsvorschläge, konkretisiert bzw. neue Forschungsfragen entwickelt. Dabei wurde der Aspekt der Anonymität nicht weiter untersucht. Die Forschungsfragen sind in Abschnitt 7.3 auf der nächsten Seite genannt.

Ausblick auf RQ1.6 In Unterabschnitt 7.2.2 (RQ1.1) ist aufgefallen, dass in dem untersuchten Review anfangs mehr Zeit benötigt wurde als am Ende und in Unterabschnitt 7.2.3 (RQ1.2) wurden Faktoren identifiziert, die möglicherweise einen Einfluss darauf haben. Außerdem ist bei der Untersuchung von RQ1.4 (siehe Unterabschnitt 7.2.5) aufgefallen, dass besseres Feedback für den Notar seitens der Gutachter hilfreich sein könnte.

Deshalb habe ich entschieden, die Sitzung durch Gamecontroller interaktiver zu gestalten, damit der Notar mehr Feedback bekommt. Zusätzlich wurde überlegt, wie Gamecontroller in den Prozess einer Review-Sitzung am besten integriert werden können, um den Einfluss von zeitverzögernden Faktoren zu eliminieren.

Ausblick auf RQ1.7 In RQ1.1 in Unterabschnitt 7.2.2 wurde der zeitliche Verlauf in einer Review-Sitzung untersucht. Hieraus ergab sich die Frage, ob das Zeitbewusstsein des Review-Teams verbessert werden kann. Deshalb wurde in Abschnitt 4.6 auf Seite 32 ein Vorschlag präsentiert, der versucht, das Zeitbewusstsein zu verbessern, indem die benötigte Zeit umfangreicher visualisiert wird. Ob durch diese Veränderung das Zeitbewusstsein gesteigert werden kann, soll in RQ1.7 untersucht werden.

Ausblick auf RQ1.8 Analog zu Forschungsfrage RQ1.5 soll untersucht werden, welchen Einfluss die umgesetzten Veränderungen auf die Motivation haben. Dabei soll besonders auf den Einfluss von Gamecontrollern auf die Motivation der Review-Teilnehmer eingegangen werden, da der Einfluss von Gamecontrollern auf die Motivation und beispielsweise die Zeit unabhängig voneinander sein können.

7.3. Zweites Review – verbesserte RevAger-Version

Aufgrund der vorherigen Erkenntnisse, den aufgetretenen Fragen und den ausgewählten Verbesserungsvorschlägen wurden folgende Forschungsfragen formuliert, welche in diesem Abschnitt untersucht werden sollen:

RQ1.6 Welchen Einfluss hat gamecontrollergestütztes Feedback auf den zeitlichen Verlauf einer Review-Sitzung?

RQ1.7 Welchen Einfluss hat die Visualisierung der Zeit auf den Verlauf einer Review-Sitzung?

RQ1.8 Welchen Einfluss hat die Verwendung von Gamecontrollern auf die Motivation?

7.3.1. Vorbereitungen für die Evaluation

Vor Beginn des Reviews wurde dem Notar die verbesserte Version von RevAger zur Verfügung gestellt. Gegenüber dem StuPro-Team wurde versucht, nicht von *Verbesserungen* zu sprechen, sondern von *Änderungen* oder *Veränderungen*, um die Personen nicht zu beeinflussen.

Zu Beginn der Sitzung wurden alle Teilnehmer über das Vorgehen instruiert (siehe Anhang B.1) und einige Testbefunde erfasst, damit die Teilnehmer sich mit den Gamecontrollern und der Präsentationsansicht vertraut machen können. Damit die Bedeutung der Tasten der Gamecontroller jederzeit klar ist, wurden die Gamecontroller mit Labeln versehen. Zusätzlich zur Beschriftung der Gamecontroller wurde jedem Teilnehmer ein Informationsblatt zur Verfügung gestellt, auf welchem die Definitionen der Befund-Klassifikationen nach Ludewig und Lichter [LL12] inklusive der zusätzlichen Klassifikation „eher kein Fehler“, abgedruckt waren (siehe Anhang B.2). Weiterhin wurde erklärt, dass bei der Befund-Klassifizierung kein Konsens vorliegen muss. Sondern dass der Notar, aufgrund des aktuellen Meinungsbilds, eine Klassifizierung auswählt. Dem Review-Team wurde mitgeteilt, dass eine Diskussion über die Klassifikation nicht zwangsläufig stattfinden muss, aber auch nicht untersagt ist. Ein Auszug der erstellten Notizen der Beobachtung der Review-Sitzung ist in Anhang B.3 dokumentiert. Die Fragen der anschließenden Interviews sind in Anhang B.4 zu finden.

Der Moderator und Notar saßen in der zweiten Sitzung am gleichen Platz wie im ersten Review. Aufgrund der geplanten Abwesenheit eines Studierenden musste in der zweiten Sitzung die Rolle des Moderators von einem anderen Studierenden übernommen werden. Die Besetzung der anderen Rollen blieb identisch.

7.3.2. Erkenntnisse bezüglich RQ1.6

Forschungsfrage RQ1.6: Welchen Einfluss hat gamecontrollergestütztes Feedback auf den zeitlichen Verlauf einer Review-Sitzung?

Bei der Verwendung der Gamecontroller sind folgende Aspekte aufgefallen:

Diskussionszeit Aus der Log-Datei der zweiten Review-Sitzung kann entnommen werden, dass die Bearbeitungsdauer der Befunde über die gesamte Sitzung relativ ähnlich war (siehe Abbildung 7.3). Die Boxplots überlappen sich vollständig, einzig das 25%-Quantil des Boxplots für die Zeit von 14:35–15:06 Uhr ist größer als die Mediane der anderen Boxplots. Die Differenz zwischen kleinstem und größtem Median beträgt 21,34s.

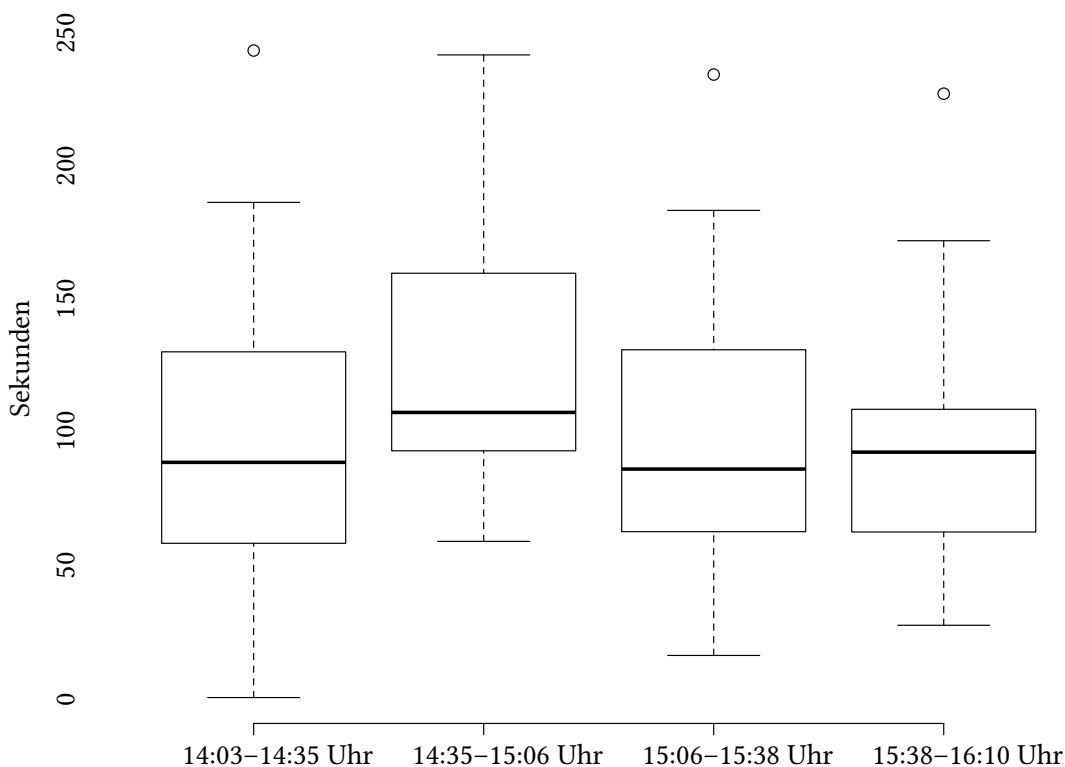


Abbildung 7.3.: Diskussionszeit im Verlauf der zweiten Review-Sitzung.

Die Teilnehmer haben in den Interviews berichtet, dass das Klassifizieren von Befunden im zweiten Review besser als im ersten Review funktioniert habe. Ein Gutachter dazu:

„Das mit den Controllern fand ich cool. [...] Und ich hatt’ das Gefühl, das geht auch schneller, weil die ganzen Abstimmungen wurden dadurch einfach erspart.“

Der Moderator hat die Änderungen im zweiten Review wie folgt bewertet:

„Es hat [...] schneller funktioniert, weil der Notar nicht darauf warten musste oder noch mal nachfragen musste, [...] sondern er hat es [= das Ergebnis der Abstimmung über die Klassifikation] gesehen.“

Parallele Dokumentation und Klassifikation In der Aussage des Moderators ist angeklungen, was in der Sitzung von mir beobachtet werden konnte: Der Notar musste teilweise nicht darauf warten, dass die Klassifikation des aktuellen Befunds vorgenommen wurde. Bei einigen Befunden war das Ergebnis der Abstimmung verfügbar, bevor der Notar die Dokumentation des Befunds abgeschlossen hatte. Ein Gutachter hat berichtet, dass die Gamecontroller dazu geführt hätten, dass die Klassifikation schneller war:

„Ich fand vor allem die Abstimmungen per Gamecontroller gut, weil [...] davor hat sich [...] niemand getraut was zu sagen, wie man das einordnet, und dann hat sich das in die Länge gezogen. [...] Da jetzt] jeder abstimmen muss und man auch nicht weiß was die anderen abstimmen, klickt man das, was man selber denkt und dann zieht's sich auch nicht so in die Länge.“

Nichtsdestotrotz konnte von mir beobachtet werden, dass der Moderator, aber auch Notar und Gutachter, explizit aufgefordert haben, die Einschätzungen abzugeben. Es kam vor, dass Gutachter vergessen hatten, ihre Einschätzung abzugeben, was zu Verzögerungen führte. Dabei kam es auch vor, dass keiner der Beteiligten wusste, wessen Einschätzung noch fehlte, sodass alle Gutachter ihre Einschätzung ein weiteres Mal abgaben. In den Interviews wurde vorgeschlagen, z. B. im Sitzungsdashboard anzuzeigen, wer bereits abgestimmt hat.

Vorgehen beim Klassifizieren Für die Review-Sitzung wurde ein festes Vorgehen beim Klassifizieren festgelegt und die Bedeutungen der Klassifikationen definiert (siehe dazu Abschnitt 7.3.1 auf Seite 53). Teilnehmer haben erzählt, dass die Diskussion über die Klassifikation geringer ausgefallen sei. Außerdem, so ein Gutachter, sei so eine Diskussionsgrundlage vorhanden gewesen:

„Da hat man schon mal sagen können, ein kritischer Fehler ist wirklich etwas, das das Ganze unbrauchbar macht [...]. Das war schon hilfreich, wenn man die Diskussionsgrundlage festgeschrieben hat.“

Das feste Vorgehen habe, laut einem Gutachter, dem Team geholfen, sich mehr auf das Review zu konzentrieren, da es „[...] das Team] ein bisschen in die Spur gezwängt [hat]“.

In der Review-Sitzung habe ich beobachtet, dass bei uneindeutigen Meinungsbildern einige Gutachter teilweise schnell ihre Einschätzung geändert haben. In den Interviews wurde von einigen Personen angemerkt, dass während der Diskussion über das Meinungsbild eine Änderung nicht möglich sein sollte. So wurde Befund 58 wie folgt klassifiziert:

15:58:44 Gutachter 1 stimmt *Nebenfehler*

15:59:14 Gutachter 2 stimmt *Hauptfehler*

15:59:16 Gutachter 3 stimmt *Kritischer Fehler*

15:59:16 Gutachter 4 stimmt *Nebenfehler*

15:59:16 Meinungsbild vollständig: *1x Kritischer Fehler, 1x Hauptfehler, 2x Nebenfehler*

15:59:19 Gutachter 4 ändert *Nebenfehler* zu *Hauptfehler*

15:59:21 Gutachter 3 ändert *Kritischer Fehler* zu *Hauptfehler*

Diskussionsfokus und Pause Die Möglichkeit, Feedback zur Review-Sitzung zu geben (siehe Abschnitt 4.5 auf Seite 31), wurde von den Gutachtern unterschiedlich genutzt. Tabelle 7.1 zeigt die Gesamtanzahl des Feedbacks pro Gutachter. Im Interview habe ich *Gutachter 2*, welcher in der

Tabelle 7.1.: Gesamtanzahl des Feedbacks pro Gutachter.

	Gutachter 1	Gutachter 2	Gutachter 3	Gutachter 4
Pause	1	87	42	0
Diskussionsfokus	4	3523	128	7
Gesamt	5	3610	170	7

Review-Sitzung durch das häufige Drücken der „Diskussionsfokus“-Taste aufgefallen war, dazu befragt. Im Interview erzählte die Person, dass sie das Drücken der „Diskussionsfokus“-Taste nicht wirklich ernst gemeint habe. Die Diskussionszeiten der Befunde in den Boxplots in Abbildung 7.4 sind deshalb nur nach der Anzahl der interagierenden Personen gruppiert; die Gesamtanzahl der Interaktionen einer Person werden nicht berücksichtigt. Trotz alledem ist die Diskussionszeit von Befunden ohne interagierende Person signifikant kleiner als bei jenen Befunden, bei denen drei Personen Feedback gegeben haben.

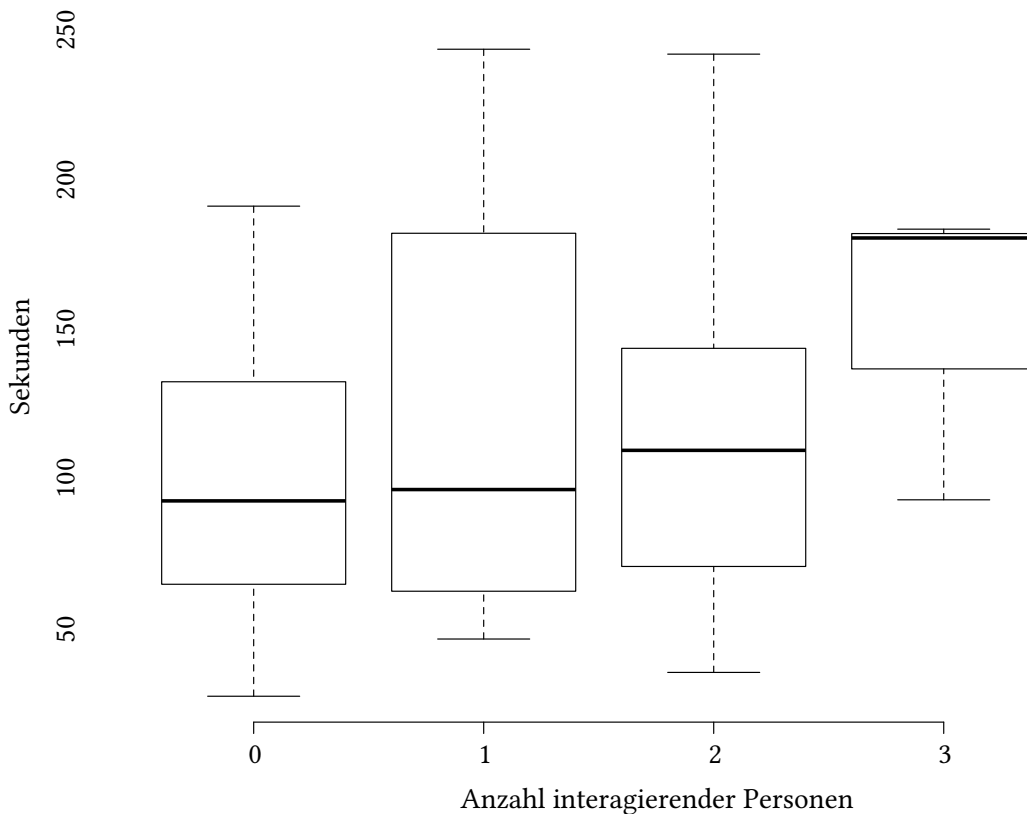


Abbildung 7.4.: Diskussionszeit mit unterschiedlich vielen interagierenden Personen.

Auf die Frage, wie oft Feedback gegeben wurde, hat ein Gutachter geantwortet:

„Bei einigen Sachen dann schon [...]. Teilweise war es auch ganz gut, weil jemand anderes das dann halt schon gemacht hat, wenn ich selber vom Thema abgekommen bin. Da war dann die Benachrichtigung auch ganz gut.“

Der Moderator hat erläutert, dass er das „Diskussionsfokus“-Feedback nicht immer sofort wahrgenommen habe und ein gesonderter Hinweis hilfreich sei. Weiterhin wurde in den Interviews vorgeschlagen, die Anzahl des erlaubten Feedbacks pro Gutachter für das ganze Review oder pro Befund zu begrenzen, da das ernst gemeinte Feedback untergegangen sei. So wurde trotz des „Pause“-Feedbacks die Sitzung nicht für eine Pause unterbrochen. Dabei hat mir ein Gutachter berichtet, dass er dieses Feedback bezüglich einer Pause ernst gemeint habe. Eine andere Person hat auf die Frage, was der Moderator hätte anders machen sollen, geantwortet:

„Zwischendrin 'ne Pause. Weil ich denk, das war schon von einigen gewünscht.“

Einsatz des Moderators In den Interviews wurde berichtet, dass der Moderator in der zweiten Review-Sitzung weniger zu tun gehabt hätte. Dies konnte ich in meinen Beobachtungen bestätigen: Der Moderator hat während der Review-Sitzung regelmäßig online Minesweeper gespielt. Besonders häufig konnte ich dies gegen Ende der Sitzung beobachten (siehe Anhang B.3 auf Seite 73). Der Moderator hat seine Tätigkeit wie folgt beschrieben:

„Ich hatte halt weniger zu tun, weil sich das ganze Review eigentlich selbst fortbewegen konnte. [...] Ich musste nur hin und wieder hinweisen, dass jetzt keine Eingabe [über den Gamecontroller] getätigt wurde oder Ähnliches.“

Antwort zur Forschungsfrage RQ1.6: Die Gamecontroller haben es ermöglicht, dass Notar und Gutachter parallel arbeiten konnten. Außerdem scheint es, dass die Diskussionszeit für Befunde über den Verlauf der Review-Sitzung nicht so stark variiert. Es scheint sinnvoll zu sein, vor Beginn eines Reviews die Bedeutungen der Klassifikationen zu definieren und den Gutachtern schriftlich zur Verfügung zu stellen. Weiterhin konnte beobachtet werden, dass bei Befunden mit längerer Diskussionszeit mehr Gutachter Feedback geben; die Möglichkeit Feedback zu geben kann aber noch optimiert werden.

7.3.3. Erkenntnisse bezüglich RQ1.7

Forschungsfrage RQ1.7: Welchen Einfluss hat die Visualisierung der Zeit auf den Verlauf einer Review-Sitzung?

In den Interviews wurde nach der Anzeige der Befund- und Gesamtzeit im Sitzungsdashboard der Präsentationsansicht gefragt. Die Antworten dazu sind unterschiedlich ausgefallen: Ein Gutachter hat berichtet, dass er ab und zu „völlig random“ danach geschaut habe. Eine andere Person erzählte, dass sie auf die Zeit geachtet habe, wenn für kleine Fehler relativ viel Zeit benötigt wurde. Speziell zur Anzeige der Gesamtzeit hat ein dritter Gutachter zum Ausdruck gebracht, dass er die verbleibende Zeit in Relation zum verbleibenden Teil des Prüflings gesetzt habe:

„Also die Gesamtzeit, denk ich, ist schon ganz gut, da haben wir auch öfters drauf geschaut. Nach der Hälfte der Zeit weiß man ja ungefähr wie weit man mit seinem Prüfling durch ist und ob man das schafft oder ob man etwas schneller vorgehen will.“

7. Evaluation

Der Moderator hat diese Aussage untermauert; allerdings habe er die Informationen nicht in seiner Argumentation verwendet,

„aber es hat mir ziemlich gut geholfen, da ich einen gewissen Überblick hatte, welche Dokumente [...] überprüft werden müssen, konnt' ich halt daran bisschen abschätzen [...]: *Wir haben jetzt die Hälfte [des Prüflings] geschafft und wir haben die Hälfte der Zeit übrig: Wir sind gut in der Zeit.*‘...“

Weiterhin erklärte er, dass das Uhr-Piktogramm ihn gelegentlich daran erinnert habe, dass sie schon länger für einen Befund gebraucht haben.

Antwort zur Forschungsfrage RQ1.7: Die Aussagen zur Visualisierung der Befund- und Gesamtzeit sind unterschiedlich. Einige Personen haben diese nicht beachtet, andere jedoch, wenn bei Befunden zu lange diskutiert wurde. Ferner wurde geschildert, dass die Visualisierung der Gesamtzeit geholfen hätte, sicherzustellen, dass der verbleibende Teil des Prüflings in der verfügbaren Zeit vollständig überprüft werden kann. Die Visualisierung der Zeit scheint das Zeitbewusstsein zu fördern. Es ist offengeblieben, ob es dadurch zu einer Veränderung in der Sitzung gekommen ist.

7.3.4. Erkenntnisse bezüglich RQ1.8

Forschungsfrage RQ1.8: Welchen Einfluss hat die Verwendung von Gamecontrollern auf die Motivation?

Dem Notar hat das zweite Review „ein bisschen“ mehr Spaß als das erste Review gemacht. Zudem hat er erklärt, dass der Notar auch einen Gamecontroller haben sollte:

„[...] für'n Notar wäre so ein Controller auch net schlecht gewesen [...] sonst] hat der Notar das Gefühl, es steht ihm nicht zu, auch eine Pause zu verlangen.“

Von Gutachtern wurde geschildert, dass sie das Gefühl gehabt hätten, dass die Sitzung durch die Gamecontroller schneller und abwechslungsreicher gewesen sei. Die Verwendung der Gamecontroller habe den Gutachtern Spaß gemacht:

„Man hat das Gefühl ein bisschen mehr Kontrolle zu haben als beim ersten Review. Und es hat schon Spaß gemacht, die Abstimmungen zu machen.“

Antwort zur Forschungsfrage RQ1.8: Die Verwendung von Gamecontrollern bzw. die Abstimmung darüber hat den Gutachtern Spaß gemacht und hat das Gefühl von Kontrolle vermittelt.

7.3.5. Bedeutung der Erkenntnisse

Die Verwendung von Gamecontrollern hat dazu geführt, dass Notar und Gutachter unabhängig voneinander arbeiten können. Außerdem ist die Arbeitslast des Moderators gesunken. Weiterhin ist die Diskussionsdauer für Befunde über den Verlauf der Sitzung konstant geblieben. Die Sitzung wurde von den Teilnehmern als schneller und abwechslungsreicher beschrieben.

Diese Ergebnisse sind – bedingt durch das gewählte Vorgehen – nur in einem begrenzten Rahmen gültig. Deshalb werden im nächsten Abschnitt Gefahren für die Validität beschrieben. Abschließend werden in Abschnitt 7.5 Hypothesen vorgestellt, um die allgemeine Gültigkeit der Ergebnisse ausführlicher zu untersuchen.

7.4. Gefahren für die Validität

Das gewählte Vorgehen und die Rahmenbedingung der Evaluation haben einige Schwächen. Im Folgenden wird diskutiert, wie sich diese auf die Zuverlässigkeit und Aussagekraft auswirken.

7.4.1. Moderator-Rolle

Aufgrund der geplanten Abwesenheit des Moderators im zweiten Review musste in der zweiten Review-Sitzung diese Rolle von einer anderen Person übernommen werden. Da der Moderator einen großen Einfluss auf die Review-Sitzung hat (siehe Abschnitt 2.2 auf Seite 16), kann nicht ausgeschlossen werden, dass Unterschiede zwischen den beiden Reviews auf den Moderator zurückzuführen sind.

7.4.2. Unterschiedliche Prüflinge

In den zwei Review-Sitzungen wurden unterschiedliche Prüflinge begutachtet: Im ersten Review eine Spezifikation und im zweiten Review Quellcode. Somit können die Daten des ersten und zweiten Reviews, wie beispielsweise die Diskussionszeiten von Befunden, nicht direkt verglichen werden. Es wurden also zwei verschiedene *Arten* von Prüflingen untersucht und es ist nicht bekannt, ob eine Review-Sitzung durch die Art des Prüflings geprägt ist. Weiterhin konnte das erste Review, im Gegensatz zum zweiten Review, nicht innerhalb einer Sitzung beendet werden, sondern musste in einer zweiten Sitzung vervollständigt werden.

7.4.3. Lerneffekt

Das Studienprojekt wurde für diese Evaluation ausgewählt, da die Teilnehmer bereits im 5. Semester sind und damit potenziell mehr Erfahrungen mit Reviews haben als Studierende aus dem 3. Semester. Nichtsdestotrotz können die Teilnehmer des Studienprojekts weitere Erfahrung aus dem ersten Review gesammelt haben, welche das zweite Review beeinflusst haben könnten. Um als außenstehende Person darauf keinen Einfluss zu nehmen, habe ich mit dem Projektteam nicht außerhalb der Interviews über das erste Review gesprochen.

7.4.4. Räumliche Trennung der Interview-Teilnehmer

Bedingt durch die Verfügbarkeit von Räumen und der Dauer der Interviews war es nicht möglich, sicherzustellen dass sich bereits interviewte bzw. nicht interviewte Personen räumlich getrennt voneinander aufhalten. Dadurch bestand die Möglichkeit, dass sich die Teilnehmer über den Inhalt des Interviews ausgetauscht haben.

7.4.5. Zweiter Interview-Termin

Bedingt durch den Termin für das zweite Review konnten die Interviews für dieses nicht alle am selben Tag durchgeführt werden. So wurden vier von sechs Personen direkt nach der Sitzung interviewt; die anderen zwei Personen wurden am folgenden Tag ab 09:20 Uhr interviewt. Somit kann es sein, dass die letzten zwei Interview-Teilnehmer bestimmte Eindrücke oder Ereignisse vergessen haben.

7.4.6. Generalisierbarkeit

Die Review-Teilnehmer waren keine repräsentative Stichprobe von Softwareentwicklern. Dementsprechend können die Erkenntnisse nicht ohne Weiteres auf die Population von Softwareentwicklern übertragen werden. Weiterhin muss berücksichtigt werden, dass an der Universität Stuttgart mehrere Studienprojekte durchgeführt werden. Die teilnehmenden Studierenden wurden durch ein Auswahlverfahren einem Projekt zugewiesen, dieses Verfahren und die Beliebtheit eines Projekts können dazu führen, dass einzelne Projekte durchschnittlich besser oder schlechter besetzt werden.

7.4.7. Genauigkeit der Log-Dateien

Beim Erstellen der Log-Dateien wurde als Startzeit für die Bearbeitung eines Befunds die Zeit der Erstellung des Befunds gewählt. In den Sitzungen konnte beobachtet werden, dass nach Erstellen eines neuen und leeren Befunds teilweise die Diskussion zum vorherigen Befund fortgeführt wurde. Gelegentlich wurden beim Dokumentieren eines Befunds auch andere Angelegenheiten diskutiert. Wegen dieser Unzulänglichkeit der Genauigkeit können die Daten nicht als exakte Repräsentation der Bearbeitungsdauer der Befunde gesehen werden. Trotzdem geben die Daten einen guten Einblick über den Verlauf der Sitzungen.

7.5. Abgeleitete Hypothesen

Auf Basis der vorangegangenen Beobachtungen und Erkenntnisse werden in diesem Abschnitt Hypothesen formuliert. Diese können in weiterführenden Arbeiten verifiziert bzw. falsifiziert werden, um weitere Erkenntnisse über die Forschungsfrage RQ1 zu erlangen. Die nachfolgenden Hypothesen sind in thematische Abschnitte gruppiert. Sie können aber nicht zwangsläufig gemeinsam untersucht werden.

7.5.1. Verwendung von Gamecontrollern zur Klassifizierung

Die Erkenntnisse bezüglich RQ1.6 haben interessante Fragen aufgeworfen bzw. mögliche Optimierungsmöglichkeiten von Review-Sitzungen gezeigt. Folgende Hypothesen untersuchen die Verwendung von Gamecontrollern zur Klassifizierung genauer.

Hypothese 1

Die Verwendung von Gamecontrollern zur Klassifizierung hat einen positiven Einfluss auf die Motivation der Review-Teilnehmer.

Hypothese 2

Die Verwendung von Gamecontrollern zur Klassifizierung erfordert weniger Aktivität des Moderators.

Hypothese 3

Die Verwendung von Gamecontrollern zur Klassifizierung ermöglicht einen höheren Grad an Parallelität beim Dokumentieren und Klassifizieren von Befunden.

Hypothese 4

Die Verwendung von Gamecontrollern zur Klassifizierung senkt die Zeit der Inaktivität der Teilnehmer.

Hypothese 5

Die Verwendung von Gamecontrollern zur Klassifizierung führt dazu, dass die Diskussionszeit pro Befund innerhalb einer Review-Sitzung konstant bleibt.

Hypothese 6

Die Verwendung von Gamecontrollern zur Klassifizierung senkt die benötigte Diskussionszeit pro Befund.

Hypothese 7

Die Verwendung von Gamecontrollern zur Klassifizierung hat keinen Einfluss auf die Korrektheit der gewählten Klassifikationen.

Hypothese 8

Das Bereitstellen der Bedeutungen von Klassifikationen in einer Review-Sitzung reduziert die Diskussionszeit pro Befund.

7.5.2. Feedback zur Sitzung mittels Gamecontrollern

In Abschnitt 7.3.2 auf Seite 53 wurde die Verwendung von Gamecontrollern evaluiert. Dabei ist deutlich geworden, dass das „Diskussionsfokus“- und „Pause“-Feedback noch nicht einwandfrei funktioniert und noch verbessert werden sollte. Die nachfolgenden Hypothesen können beispielsweise in inszenierten Review-Sitzungen, in denen einige Teilnehmer eingeweiht sind, überprüft werden. So können vergleichbare problematische Situationen reproduziert werden.

Hypothese 9

Die Möglichkeit Feedback zur Diskussion mittels Gamecontroller hilft dem Moderator, nicht konstruktive Diskussionen früher zu erkennen.

Hypothese 10

Die Möglichkeit Feedback zur Diskussion mittels Gamecontroller führt dazu, dass Gutachter nicht konstruktive Diskussionen vermeiden.

7.5.3. Visualisierung der Zeit

Bei der Visualisierung der Zeit wurde nicht klar, ob und wie groß der Einfluss der Visualisierung ist. Es ist offengeblieben, ob es dadurch zu einer Veränderung der Sitzung kam.

Hypothese 11

Die Visualisierung der Zeit führt dazu, dass die Diskussionszeit innerhalb einer Review-Sitzung konstant bleibt.

Hypothese 12

Die Visualisierung der Zeit führt dazu, dass der Moderator nicht konstruktive Diskussionen früher erkennt.

Hypothese 13

Die Visualisierung der Zeit führt dazu, dass Gutachter nicht konstruktive Diskussionen vermeiden.

7.6. Zusammenfassung

In diesem Kapitel wurde im ersten Schritt der Evaluation untersucht, wie die dokumentierten Verbesserungsvorschläge sinnvoll in den Review-Prozess integriert werden können und was dabei berücksichtigt werden muss. Das Ergebnis dieser Evaluation waren weiter zu untersuchende Forschungsfragen, welche anschließend im zweiten Review untersucht wurden. Nachfolgend wurden mögliche Gefahren für die Validität der Evaluationen erörtert. Abschließend wurden, unter Berücksichtigung der Erkenntnisse der Untersuchungen, Hypothesen formuliert.

8. Zusammenfassung und Ausblick

In dieser explorativen Arbeit wurden Verbesserungsvorschläge präsentiert, die Sitzungen in Technischen Reviews verbessern können. Nach einer ersten Evaluation wurden einige Verbesserungsvorschläge aufgrund der Erkenntnisse dieser implementiert. Zu den gewählten Ideen gehört die Verwendung von Gamecontrollern zur Interaktion und eine Präsentationsansicht mit einem Sitzungsdashboard mit Informationen zur Sitzung. Die wesentlichen Beobachtungen der zweiten Evaluation sind, dass die Klassifikation mittels Gamecontroller die Arbeitslast des Moderators senkt und die Diskussionszeit von Befunden im Verlauf der Sitzung weniger stark variiert. Ferner entsteht durch die Verwendung von Gamecontrollern das Gefühl, dass die Review-Sitzung besser strukturiert ist. Diese Beobachtungen müssen durch kontrollierte Experimente verifiziert bzw. falsifiziert werden.

Ausblick

Diese Arbeit kann dahingehend fortgeführt werden, dass die aufgestellten Hypothesen in kontrollierten Experimenten untersucht werden. Außerdem können die umgesetzten Verbesserungsvorschläge noch optimiert werden. Darüber hinaus können weitere Vorschläge implementiert und in einem ähnlichen Rahmen evaluiert werden.

Danksagung

Abschließend möchte ich mich bei den Personen bedanken, die mich beim Erstellen dieser Arbeit unterstützt haben: Bei meinem Betreuer Dr. Ivan Bogicevic, für das gute Feedback zu meiner Arbeit, das Bereitstellen der Gamecontroller und sein Interesse an diesem Thema. Bei Daniel Kulesz, der mir gutes und interessantes Feedback zu meinen Verbesserungsvorschlägen gab und RevAger zu GitHub migriert hat. Ebenfalls möchte ich mich bei dem StuPro-Team bedanken, dass ich ihren Review-Sitzungen beiwohnen durfte. Weiterhin danke ich meinem Prüfer Prof. Dr. Stefan Wagner für das Prüfen dieser Bachelorarbeit.

Literaturverzeichnis

- [08] *IEEE Standard for Software Reviews and Audits*. Norm IEEE STD 1028-2008. IEEE Computer Society, 2008. DOI: 10.1109/ieeestd.2008.4601584 (zitiert auf S. 13–17, 19, 37).
- [BB13] A. Bacchelli, C. Bird. „Expectations, outcomes, and challenges of modern code review“. In: *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013, S. 712–721. DOI: 10.1109/icse.2013.6606617 (zitiert auf S. 17–19).
- [BBH+08] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp. „Motivation in Software Engineering: A systematic literature review“. In: *Information and software technology* 50.9 (2008), S. 860–878. DOI: 10.1016/j.infsof.2007.09.004 (zitiert auf S. 30).
- [DH63] N. Dalkey, O. Helmer. „An experimental application of the Delphi method to the use of experts“. In: *Management science* 9.3 (1963), S. 458–467. DOI: 10.1287/mnsc.9.3.458 (zitiert auf S. 21, 23, 30, 33).
- [ELL+92] S. G. Eick, C. R. Loader, M. D. Long, L. G. Votta, S. Vander Wiel. „Estimating software fault content before coding“. In: *Proceedings of the 14th international conference on Software engineering - ICSE '92*. ACM. ACM Press, 1992, S. 59–65. DOI: 10.1145/143062.143090 (zitiert auf S. 16).
- [Fag02] M. Fagan. „A history of software inspections“. In: *Software pioneers*. Springer, 2002, S. 562–573. DOI: 10.1007/978-3-642-59412-0_34 (zitiert auf S. 13, 16, 19).
- [Fag76] M. Fagan. „Design and code inspections to reduce errors in program development“. In: *IBM Systems Journal* 15.3 (1976), S. 182–211. DOI: 10.1007/978-3-642-59412-0_35 (zitiert auf S. 13, 17, 19, 32).
- [Gre02] J. Grenning. „Planning poker or how to avoid analysis paralysis while release planning“. In: *Hawthorn Woods: Renaissance Software Consulting* 3 (2002) (zitiert auf S. 22, 23).
- [Hof13] D. W. Hoffmann. *Software-Qualität*. Springer-Verlag, 2013. DOI: 10.1007/978-3-642-35700-8 (zitiert auf S. 17).
- [IEE89] IEEE Computer Society. *1028-1988 - IEEE Standard for Software Reviews and Audits. Evolution of the Standard*. 1989. URL: <http://ieeexplore.ieee.org/document/29123/versions> (zitiert auf S. 13).
- [JT98] P. M. Johnson, D. Tjahjono. „Does Every Inspection Really Need a Meeting?“ In: *Empirical Software Engineering* 3.1 (1998), S. 9–35. ISSN: 1573-7616. DOI: 10.1023/A:1009787822215. URL: <http://dx.doi.org/10.1023/A:1009787822215> (zitiert auf S. 15, 17, 19).

- [KSR17] S. Khandelwal, S. K. Sripada, Y. R. Reddy. „Impact of Gamification on Code review process: An Experimental Study“. In: *Proceedings of the 10th Innovations in Software Engineering Conference*. ACM. 2017, S. 122–126. DOI: 10.1145/3021460.3021474 (zitiert auf S. 22, 30).
- [Lig09] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Springer Science & Business Media, 2009. DOI: 10.1007/978-3-8274-2203-3 (zitiert auf S. 13, 16, 19).
- [LL12] J. Ludewig, H. Lichter. *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. dpunkt. verlag, 2012 (zitiert auf S. 13–16, 19, 29, 32, 53).
- [MKAH16] S. McIntosh, Y. Kamei, B. Adams, A. E. Hassan. „An empirical study of the impact of modern code review practices on software quality“. In: *Empirical Software Engineering* 21.5 (2016), S. 2146–2189. DOI: 10.1007/s10664-015-9381-9 (zitiert auf S. 18, 19).
- [MM98] F. Macdonald, J. Miller. „A comparison of tool-based and paper-based software inspection“. In: *Empirical Software Engineering* 3.3 (1998), S. 233–253. DOI: 10.1023/A:1009747104814 (zitiert auf S. 21, 23, 29, 30).
- [MMB+95] F. Macdonald, J. Miller, A. Brooks, M. Roper, M. Wood. „A review of tool support for software inspection“. In: *Computer-Aided Software Engineering, 1995. Proceedings., Seventh International Workshop on*. IEEE. 1995, S. 340–349. DOI: 10.1109/case.1995.465300 (zitiert auf S. 21, 23).
- [PL03] D. L. Parnas, M. Lawford. „The role of inspection in software quality assurance“. In: *Software Engineering, IEEE Transactions on* 29.8 (2003), S. 674–676. DOI: 10.1109/tse.2003.1223642 (zitiert auf S. 13, 17).
- [RB13] P. C. Rigby, C. Bird. „Convergent contemporary software peer review practices“. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM. 2013, S. 202–212. DOI: 10.1145/2491411.2491444 (zitiert auf S. 18, 19).
- [Reva] RevAger Developers. *Die Wurzeln von RevAger*. URL: <https://www.revager.org/#page-de-development> (zitiert auf S. 11).
- [Revb] RevAger Developers. *RevAger Features*. URL: <https://www.revager.org/#page-de-features> (zitiert auf S. 11).
- [SHP+14] F. Streit, L. Haddad, T. Paul, J. Frank, A. Schäfer, J. Nikitopoulos, C. Akdeniz, F. Lederbogen, J. Treutlein, S. Witt et al. „A functional variant in the neuropeptide S receptor 1 gene moderates the influence of urban upbringing on stress processing in the amygdala“. In: *Stress* 17.4 (2014), S. 352–361. DOI: 10.3109/10253890.2014.921903 (zitiert auf S. 32).
- [SRK16] S. K. Sripada, Y. R. Reddy, S. Khandelwal. „Architecting an extensible framework for Gamifying Software Engineering concepts“. In: *Proceedings of the 9th India Software Engineering Conference*. ACM. 2016, S. 119–130. DOI: 10.1145/2856636.2856649 (zitiert auf S. 22, 23).
- [SS12a] L. Singer, K. Schneider. „Influencing the adoption of software engineering methods using social software“. In: *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 2012. DOI: 10.1109/ICSE.2012.6227088 (zitiert auf S. 22, 23, 29).

- [SS12b] L. Singer, K. Schneider. „It was a bit of a race: Gamification of version control“. In: *Games and Software Engineering (GAS), 2012 2nd International Workshop on*. IEEE. 2012, S. 5–8. DOI: 10.1109/gas.2012.6225927 (zitiert auf S. 22).
- [Vot93] L. G. Votta Jr. „Does every inspection need a meeting?“ In: *ACM SIGSOFT Software Engineering Notes* 18.5 (1993), S. 107–114. DOI: 10.1145/167049.167070 (zitiert auf S. 16, 17, 19, 26, 28–30).
- [WKCJ00] L. Williams, R. R. Kessler, W. Cunningham, R. Jeffries. „Strengthening the case for pair programming“. In: *IEEE software* 17.4 (2000), S. 19–25. DOI: 10.1109/52.854064 (zitiert auf S. 17).

Alle URLs wurden zuletzt am 4. September 2017 geprüft.

A. Anhänge zum ersten Review

A.1. Anonymisierter Auszug aus den Notizen der Beobachtung

*Diese Notizen stellen kein Transkript dar. Wörtliche Zitate sind durch Anführungszeichen gekennzeichnet („wörtlich“) und inhaltliche Zitate sind in Sternen eingeschlossen (*inhaltlich*).*

10:42: Es wird viel gelacht.

10:42: Moderator: *Sind wir uns bei Befund 43 einig?*

10:42: Befund ist unbewertet.

Notar: *Sobald ihr mir was sagt.*

10:43: Jemand macht einen Spaß und sagt „Rechtschreibfehler“.

10:44: *Fehlerart?*

Gutachter 1: *kritischer Fehler*

Es werden keine weiteren Meinungen geäußert oder abgestimmt.

10:44: Befund über Satzbau und Rechtschreibung wird erstellt.

10:45: Moderator: „Ist dies eigentlich unser letztes Dokument?“

10:46: Notar: *Können wir weitermachen?*

10:46: Das Team diskutiert über die dritte Stunde.

10:47: Moderator: *Schließen wir um 11:15 Uhr das Review und setzen einen neuen Termin fest oder schaffen wir das noch?*

10:48: Moderator: *Zu Anforderung 13 etwas? 14? 15?*

10:48: Notar dreht sich auf dem Stuhl und reibt sich die Augen. Es wird diskutiert.

10:49: Moderator: *Zu Anforderung 15?*

10:49: Moderator: *Das ist ein Nebenfehler.*

10:50: Moderator: *Hat jemand was zu Anforderung 15? 16? 17?*

Gutachter: *Bei 16 hab ich etwas.*

10:51: Moderator: *Zu Anforderung 18?*

Gutachter: *Rechtschreibfehler*

10:51: Moderator: *Zu Anforderung 19? 20?*

10:52: Moderator: *Zu Anforderung 21? 22?*

Gutachter 1: *Ja. Testfall wird doppelt erwähnt.*

A.2. Interviewfragen

1. Was hat dir an RevAger gefallen?
2. Was hat dir in RevAger nicht gefallen?
3. Notar: Wie war die Bedienung von RevAger? Erzähl mal.
Gutachter/Moderator: Konntest du dem Verlauf (Protokoll) immer folgen? Waren notwendige Informationen vorhanden?
4. Wie sehr hat euch RevAger, deiner Meinung nach, beim Review unterstützt bzw. gehindert?
5. Gab es Befunde, die ihr besprochen, aber nicht dokumentiert habt? Wenn ja, welche? Warum?
6. Gutachter: Gab es Befunde, die du nicht erwähnt hast? Wenn ja, warum?
7. Gab es Momente, wo du das Gefühl hattest, dass sich die Diskussion im Kreis dreht und wenn ja, welche waren das? Skala: 0–20 %, 20–40 %, 40–60 %, 60–80 %, 80–100 %.
8. Gutachter: Wie oft warst du bei der Bewertung eines Befunds anderer Meinung, also egal, ob du es gesagt hast oder nicht? Skala: 0–20 %, 20–40 %, 40–60 %, 60–80 %, 80–100 %.
Moderator: Wie oft hattest du beim Protokollieren eines Befunds das Gefühl, dass sich die Diskussion im Kreis dreht? Skala: 0–20 %, 20–40 %, 40–60 %, 60–80 %, 80–100 %.
9. Gutachter: Wie oft hast du deine Meinung dann auch geäußert? Skala: 0–20 %, 20–40 %, 40–60 %, 60–80 %, 80–100 %.
Moderator: Wie oft hast du moderierend eingegriffen? Skala: 0–20 %, 20–40 %, 40–60 %, 60–80 %, 80–100 %.
10. Kannst du mir beschreiben, wie du den zeitlichen Verlauf der Sitzung wahrgenommen hast? Habt ihr für manche Dinge besonders lange/kurz gebraucht?
11. Wie bewertest du die Länge von dem Review? Wäre eine Pause notwendig gewesen?
12. Hattest du Spaß? Beschreibe! Gab es etwas, was dich (de-)motiviert hat?
13. Gibt es sonst etwas, was nicht gut lief? Möchtest du noch was sagen?

B. Anhänge zum zweiten Review

B.1. Instruktionen und Hinweise für die Teilnehmer

Hinweise und Erklärungen für alle Teilnehmer

1. Kommentare zur Sitzung/Befundliste/Protokoll zeigen und erklären
2. Präsentationsansicht zeigen und verschiedene Bereiche erklären:
 - Vorheriger Befund
 - Aktueller Befund
 - Dashboard
3. Jeder bekommt einen Gamecontroller
4. Informationsblatt austeilen
5. Vorgehen beim Klassifizieren von Befunden erläutern:
 - a) *Alle* stimmen über Gamecontroller ab
 - b) Notar wählt Klassifikation aus
 - c) Diskussion nur bei Bedarf, aber
 - es müssen alle abgestimmt haben und
 - der Moderator moderiert die Diskussion
6. Tastenbelegung erklären (siehe Informationsblatt):
 - Pause
 - Diskussionsfokus
 - Klassifikationen
7. Hinweis: Joysticks bitte nicht bewegen
8. Teilnehmer auffordern, einen Testbefund zu verfassen.
9. Gibt es noch Fragen?

Hinweise für den Moderator

- Zusätzliche Hinweise bezüglich Dashboard:
 - Auf die Zeitanzeige und Uhr achten.
 - Auf das Feedback achten. Auf Feedback eingehen, wenn nötig.
- Zum Vorgehen beim Klassifizieren von Befunden:
 1. Diskussion nur erlauben, wenn alle abgestimmt haben.
 2. Diskussion moderieren.

B.2. Informationsblatt für die Teilnehmer

1. *Alle* stimmen über Gamecontroller ab
2. Notar wählt Klassifikation aus
3. Diskussion nur bei Bedarf, aber
 - es müssen alle abgestimmt haben und
 - der Moderator moderiert die Diskussion

Pause

(auf Tastendruck wird verzögert reagiert)

- Müde?
- Toilettenpause?

Hauptfehler (=hF)

- Prüfling merklich beeinträchtigt
- *Sollte* vor Beginn des nächsten Sprints behoben werden

Diskussionsfokus (=Fokus)

(auf Tastendruck wird verzögert reagiert)

Drücken bei:

- Lösungsdiskussionen
- Off-Topic-Diskussionen
- Diskussion „dreht sich im Kreis“
- Bitte weitermachen
- ...

Nebenfehler (=nF)

- Prüfling kaum beeinträchtigt
- *Kann* bei Gelegenheit behoben werden

eher kein Fehler

- Unsicher, ob es ein (kleiner) Fehler ist
- Wird in Korrekturphase noch mal angeschaut und ggf. korrigiert

kritischer Fehler (=kritF)

- Prüfling unbrauchbar
- *Muss* vor Beginn des nächsten Sprints behoben werden

Gut

- Abschnitt/... enthält keinen Fehler

B.3. Anonymisierter Auszug aus den Notizen der Beobachtung

*Diese Notizen stellen kein Transkript dar. Wörtliche Zitate sind durch Anführungszeichen gekennzeichnet („wörtlich“) und inhaltliche Zitate sind in Sternen eingeschlossen (*inhaltlich*).*

- 15:53:** Befund 56: Moderator spielt Minesweeper
- 15:54:** Befund 56: Moderator spielt Minesweeper
- 15:55:** Befund 56: Moderator spielt Minesweeper
- 15:55:** Gutachter: *LOL. Ich glaube, das ist ein Bug im ... (Browser?)*
- 15:55:** Gutachter: *Wer hat noch nicht abgestimmt?*
- 15:56:** Gutachter zu anderem Gutachter: *Wenn man sich anmeldet.*
- 15:56:** Moderator spielt Minesweeper. Gutachter surft auf 9gag(?)
- 15:57:** Moderator spielt Minesweeper. Ein Gutachter ist auf 9gag(?)
- 15:57:** Moderator: *Wir warten auf Input.*
Moderator: *Wir haben noch 6 Minuten.*
- 15:58:** Moderator spielt Minesweeper. Ein Gutachter teilt via WhatsApp ein Bild
- 15:59:** Notar: *Abstimmen!*
- 15:59:** Gutachter: *Jetzt haben wir es geklärt.*
(Ergebnis war nicht eindeutig. Der Gutachter hat dann sein Vote geändert)
- 16:00:** Zwei Gutachter schauen auf einen Laptop. Testen sie etwas?
- 16:01:** Moderator: *Kleine Frage: Wie viele Anmerkungen bleiben noch?*
- 16:02:** Unentschieden
- 16:03:** Befund 62 wird dokumentiert
- 16:03:** Befund 62: Ein Gutachter zählt einen Countdown
- 16:03:** Moderator hält seine Hände hoch (als wolle er was sagen)
- 16:04:** Das Team macht weiter mit dem Review, obwohl die Zeit abgelaufen ist.
- 16:04:** Ein Gutachter schleudert einen Kaltgerätestecker wie ein Lasso durch die Luft.
- 16:04:** Notar: *Abstimmung! Ach egal*
- 16:05:** Moderator spielt Minesweeper
- 16:05:** Gutachter ist auf 9gag(?) und teilt Bilder per Whatsapp(?)
- 16:05:** Notar und ein Gutachter sprechen über Befund
- 16:06:** Ein Gutachter schaut beim Moderator auf den Laptop.
- 16:06:** Notar: *Könnte die vierte Person auch noch abstimmen?*
- Abstimmung vollständig. Er schreibt noch.

B.4. Interviewfragen

1. Gutachter/Moderator: Wie findest du die Präsentationsansicht?
Notar: Wie findest du die Änderungen bezüglich Bedienung?
2. Wie sehr hat euch RevAger beim Review unterstützt/gehindert? Im Vergleich zum 1. Review?
3. Wie hast du den zeitlichen Verlauf der Sitzung wahrgenommen? Habt ihr für etwas besonders lange/kurz gebraucht? Woran lag es, dass es länger/kürzer gebraucht hat? Vergleich 1. Review
4. Wie fandest du das Vorgehen beim Klassifizieren? Was war gut/schlecht? Woran lag es (Gamecontroller/Vorgehen/Definitionen)? Haben die Definitionen geholfen?
5. Wie bewertest du die Länge von dem Review?
6. Gutachter: Wie fandest du das Benutzen der Gamecontroller? Erzählen lassen...
7. Gutachter: Wie findest du es, dass die Gesamt-/Befundzeit im Dashboard angezeigt wird?
Moderator: Hast du die Gesamt-/Befundzeit bei Entscheidungen bezüglich Moderation einbezogen?
Notar: Hat dir die Anzeige der Klassifikationen geholfen?
8. Gutachter: Wie fandest du die Möglichkeit Feedback zu geben? Hast du sie genutzt?
Moderator: Hast du das Feedback bei Entscheidungen bezüglich Moderation einbezogen?
Notar: Hättest du mehr Feedback gebraucht?
9. Hattest du Spaß? Beschreibe! Gab es etwas, was dich (de-)motiviert hat?
10. Gutachter: Hat dir die Verwendung der Gamecontroller Spaß gemacht?
Notar/Moderator: Glaubst du, den anderen hat das Verwenden der Gamecontroller Spaß gemacht?
11. Gibt es sonst etwas, was nicht gut lief? Möchtest du noch was sagen?

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift