

Freie Universität Berlin

Masterarbeit am Institut für Informatik der Freien Universität Berlin

Arbeitsgruppe Software Engineering

Passivität bei der Paarprogrammierung

Jeannine Darakci

Matrikelnummer: 4917809

jeannine.darakci@inf.fu-berlin.de

Betreuer/in: Linus Ververs

Eingereicht bei: Prof. Dr. Lutz Prechelt

Zweitgutachter/in: Prof. Dr. Claudia Müller-Birn

Berlin, 23. Januar 2024

Zusammenfassung

Die folgende Masterarbeit beschäftigt sich mit der Frage, welche Faktoren die Passivität in der Paarprogrammierung beeinflussen, und welche Strategien oder Maßnahmen entwickelt werden können, um diese Passivität zu minimieren oder zu verhindern.

Bei dem Material handelt es sich um Video- und Audiodateien. Die verwendete Methode ist eine qualitative Datenanalyse, die sich an der Grounded Theory orientiert. Die Analyse kam zu dem Ergebnis, dass eine Wissenslücke zwischen den Teilnehmer:innen eine wesentliche Rolle spielt, wenn es um Passivität geht. Ist eine Wissenslücke vorhanden, wird diese entgegen der Erwartungen nicht offenbart. Die Analyse hat gezeigt, dass ein Schutzmechanismus des Egos der plausibelste Grund für dieses Verhalten ist. Das Stellen von Fragen oder Formulieren von Vorschlägen, die abgelehnt werden, führen zu einer Beschädigung des Egos. Ist die Wissenslücke zu groß werden Entscheidungen ohne Verständnis evaluiert und die passiven Partner:innen neigen dazu, jedem Vorschlag zuzustimmen, um ihre Wissenslücke nicht zu offenbaren. Dies führt dazu, dass ein Einfluss auf Entscheidungen weg fällt und die Qualitätssicherung verloren geht. Ein weiterer Aspekt, der negativ beeinflusst wird, ist der Wissenstransfer. Werden Wissenslücken nicht offen kommuniziert, hat der Partner keine Möglichkeit diese zu erkennen und durch einen Wissenstransfer zu verringern. Also Folge wird die Wissenslücke immer größer bis der passive Partner den Anschluss verliert und die Zusammengehörigkeit, das Herzstück der Paarprogrammierung, verloren geht. Als Lösung sollte man einen sicheren Hafen schaffen. Dieser sorgt dafür, dass sich die passiven Personen trauen, Fragen zu stellen und sich einzubringen. Des Weiteren sollte man die passive Person bewusst an die Maus und Tastatur lassen. Dies regt sie dazu an, aktiv zu werden und Unklarheiten zu klären, die für die nächsten Schritte erforderlich sind.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

23. Januar 2024

Jeannine Darakci

J. Darakci

Inhaltsverzeichnis

1	Einleitung	9
1.1	Forschungsfrage	9
1.2	Relevanz der Forschungsfrage	9
1.3	Motivation der Arbeit	9
1.4	Aufbau der Arbeit	10
2	Grounded Theory Methodology	11
2.1	Einleitung	11
2.2	Einführung in die Grounded Theory	11
2.2.1	Offenes Kodieren	11
2.2.2	Axiales Kodieren	12
2.2.3	Selektives Kodieren	12
2.3	Leitlinien in der GTM	12
2.3.1	Theoretisches Sampling	12
2.3.2	Theoretische Sättigung	12
2.3.3	Memos	12
3	Grundlegende Definitionen	14
3.1	Einleitung	14
3.2	Paarprogrammierung	14
3.3	Entscheidungsfindung	14
3.4	Wissenstransfer	14
3.5	Ruder	15
4	Verwandte Arbeiten	16
4.1	Einleitung	16
4.2	Disengagement in pair programming: Does it matter? [7]	16
4.3	Qualitative Analysis of Knowledge Transfer in Pair Programming [12]	17
4.4	Understanding Pair Programming: The Base Layer [8]	18
4.5	Egoless Programming [11]	18
5	Methodik	19
5.1	Einleitung	19
5.2	Daten	19
5.2.1	Videos & schriftliches Material	19
5.2.2	GTM auf bestehender Datenbasis	21
5.3	Verlauf der Analyse	21
5.3.1	Identifizierung spannender Phänomene	21
5.3.2	Theoriegetriebene Stichprobenziehung	22
5.3.3	Näherer Blick auf die ausgewählten Videos	24
5.3.4	Detaillierte Analyse der Sitzungen	26
5.3.5	Memos	38
5.3.6	Zusätzliche Gedanken zu unterschiedlichen Videos	38

6	Hauptergebnisse	41
6.1	Einleitung	41
6.2	Passivität bei Entscheidungsfindungen	41
6.3	Passivität bei dem Wissenstransfer	45
6.4	Aktivität bei Ruderwechsel	48
6.5	Gründe für Passivität	51
6.6	Die Grounded Theory	53
6.7	Handlungsempfehlungen	57
7	Diskussion	59
7.1	Vergleich Ergebnisse mit bestehender Literatur/Arbeiten	59
7.1.1	Disengagement in pair programming: Does it matter?	59
7.1.2	Self-Esteem and Subjective Responses to Work Among Mature Workers: Similarities and Differences by Gender	61
7.2	Limitation der Arbeit	62
8	Zusammenfassung und Ausblick	63
8.1	Zusammenfassung	63
8.2	Ausblick	64
	Literaturverzeichnis	65

Abbildungsverzeichnis

1	Grounded Theory	53
2	Schaffen eines sicheren Hafens	58

1 Einleitung

1.1 Forschungsfrage

Welche Faktoren beeinflussen die Passivität in der Paarprogrammierung, und welche Strategien oder Maßnahmen können entwickelt werden, um diese Passivität zu minimieren oder zu verhindern?

1.2 Relevanz der Forschungsfrage

Das Gebiet der Paarprogrammierung wird schon seit einiger Zeit erforscht. Die ersten Definitionen reichen Jahrzehnte zurück, dennoch gibt es kaum Studien, die sich qualitativ mit dem Thema auseinandersetzen. Die meisten Studien befassen sich bis jetzt mit den Vorteilen der Paarprogrammierung und ihren Ergebnissen und weniger damit, was eine Paarprogrammierung genau beinhaltet. In der Arbeitsgruppe "Software Engineering" wird weiteres bereits erforscht. Die Hauptfokuse liegen hierbei auf

1. dem Wissenstransfer
2. Togetherness
3. Entscheidungsfindung

Meine Arbeit gibt einen Einblick in die Einschränkungen, die durch eine Passivität bei der Paarprogrammierung entstehen. Diese beeinflussen sowohl den Wissenstransfer, die Entscheidungsfindung als auch die Togetherness, sodass ein näherer Blick darauf relevant ist.

1.3 Motivation der Arbeit

Ich habe im Jahr 2021/22 meine Bachelorarbeit in der Arbeitsgruppe "Software Engineering" geschrieben. Die Art der Arbeit und das Thema waren so interessant, dass ich mich dazu entschloss, einen weiteren Blick auf die aktuelle Forschung in dieser Arbeitsgruppe zu werfen. Dabei kam ich auf das Thema der Paarprogrammierung, welches seit 2004 in dieser Arbeitsgruppe erforscht wird. Einer der Hauptfokuse liegt hierbei in dem Gebiet der Entscheidungsfindung, die Frage nach dem Ablauf von Entscheidungen.

Dieses Thema ist noch kaum erforscht und stellte somit eine interessante Möglichkeit dar erneut mit Hilfe von der Grounded Theory Methodology, welche ich bereits bei meiner Bachelorarbeit angewendet habe, ein Thema zu erforschen, das jede mögliche Art von Ergebnissen liefern kann. Dies machte mich neugierig.

Nachdem ich begann Entscheidungsepisoden zu analysieren, merkte ich, dass ich mehr zu dem Thema "Passivität" herausfinden wollte und lenkte meine Arbeit somit in diese Richtung.

1. Einleitung

1.4 Aufbau der Arbeit

Im Folgenden wird der Aufbau dieser Masterarbeit dargestellt.

Das zweite und dritte Kapitel beinhaltet wichtige Definitionen, die hilfreich sind, um diese Arbeit zu verstehen. Anschließend folgt im vierten Kapitel die Methodik. Es wird kurz erläutert, wie mit welchen Daten gearbeitet wurde. In Kapitel 5 werden dann die erarbeiteten Ergebnisse und Lösungsvorschläge präsentiert. In dem darauf folgenden sechsten Kapitel passiert die Diskussion. Hier werden die Ergebnisse mit bereits vorhandener Literatur verglichen und die Limitierungen offengelegt. Das siebte Kapitel formt mit der Zusammenfassung und einem Ausblick das Ende der Masterarbeit.

2 Grounded Theory Methodology

2.1 Einleitung

Dieses Kapitel gibt eine Einführung in die verwendete Forschungsmethode "Grounded Theory" und ihre Bestandteile.

2.2 Einführung in die Grounded Theory

Bei der Grounded Theory handelt es sich um eine qualitative Forschungsmethode. Sie wird vorzugsweise für Analysen verwendet, die in bisher unerforschte Themen eintauchen. Das Ziel ist das Aufstellen einer Theorie, die die in den Daten begründet ist.[10]

Bei der Grounded Theory werden qualitative Daten, wie Interviews, Beobachtungen oder Schriftstücke verwendet.

Das Besondere ist, dass zu Beginn der Analyse keine Theorie vorliegt, die untersucht werden soll, sondern ein Untersuchungsbereich, in dem der Forscher oder die Forscherin das Finden bestimmter Phänomene anstrebt[10]. Phänomene definieren Strauss und Corbin als "zentrale Idee auf eine Reihe von Handlungen gerichtet ist, um sie zu kontrollieren oder bewältigen oder zu dem Handlungen in Verbindung stehen." [10].

Eine weitere Besonderheit ist, dass die Datensammlung, Datenanalyse und Theorie in einer wechselseitigen Beziehung zueinander stehen. [3].

Der Prozess, der von Beginn an stattfindet, ist das sogenannte Kodieren der Daten. Kodieren der Daten bedeutet laut Charmaz "coding means categorizing segments of data with a short name that simultaneously summarizes and accounts for each piece of data. Your codes show how you select, separate, and sort data to begin an analytic accounting of them." (deutsch: Kodierung bedeutet die Kategorisierung von Datensegmenten mit einer Kurzbezeichnung, die gleichzeitig die einzelnen Daten zusammenfasst und abbildet. Ihre Codes zeigen, wie Sie Daten auswählen, trennen und sortieren, um eine analytische Erfassung der Daten zu beginnen.) [4]. Es gibt drei Arten der Kodierung, die im Folgenden näher erklärt werden.

2.2.1 Offenes Kodieren

Die Analyse startet mit dem offenen Kodieren. Das Ziel ist hierbei die Benennung und Kategorisierung von Phänomenen. Hierfür werden die Daten zunächst aufgebrochen, wobei Strauss und Corbin unter Aufbrechen das "Herausgreifen einer Beobachtung, eines Satzes, eines Abschnittes und Vergeben von Namen für jeden einzelnen darin enthaltenen Vorfall, jede Idee oder jedes Ereignis - für etwas, das für ein Phänomen steht oder es repräsentiert" verstehen [10]. Bei der Benennung der Phänomene ist es wichtig, die gefundenen Phänomene miteinander zu vergleichen, um gleiche Phänomene gleich zu bezeichnen. Ein weiterer Bestandteil des offenen Kodierens ist das Finden von Kategorien. Kategorien entstehen, wenn mehrere Konzepte ähnliche Muster aufweisen. Konzepte sind hierbei laut Strauss und Corbin "Konzeptuelle Bezeichnungen oder Etiketten, die einzelnen Ereignissen, Vorkommnissen oder anderen

2. Grounded Theory Methodology

Beispielen für Phänomene zugeordnet werden können.”[10].

2.2.2 Axiales Kodieren

Nach dem Prozess des Aufbrechens der Daten müssen diese auf eine neue Art wieder zusammengeführt werden. Dies geschieht bei dem axialen Kodieren. Kategorien werden mit sogenannten Subkategorien in Beziehung zueinander gesetzt. Dies geschieht mit Hilfe des paradigmatischen Modells. Hierbei wird betrachtet, in welcher Verbindung die Kategorien und Subkategorien zueinander stehen. Solche Verbindungen können laut dem paradigmatischen Modell "Ursachen, Kontext, intervenierende Bedingungen, Handlungsstrategien und Konsequenzen" [10] sein. Hierfür ist entscheidend, immer wieder auf die Daten zurückzugreifen und Beweise für diese Beziehungen zu finden. Außerdem sollen die einzelnen Kategorien weiter angereichert werden.

2.2.3 Selektives Kodieren

Sind die Kategorien und Subkategorien in Verbindung gebracht worden, geht es nun darum, die Hauptkategorie zu finden. Hierfür wird das ausschlaggebendste Phänomen ausgewählt und in den Fokus der Theorie gesetzt[10]. Ist diese Hauptkategorie gefunden, geht es um die In-Verbindung setzen dieser Kategorie mit den restlichen Kategorien. Dies kann erneut mit dem paradigmatischen Modell geschehen. Die Hauptkategorie muss so gewählt werden, dass Zusammenhänge plausibel dargestellt und begründet werden können.

2.3 Leitlinien in der GTM

2.3.1 Theoretisches Sampling

Mit "Theoretical Sampling" ist gemeint, dass die Datensammlung in Bezug auf die in der Analyse gefundenen Konzepte angepasst wird. Man beginnt mit einer Datensammlung, die offen und breit für alle Arten von Beobachtungen ist, und passt sie mit den Erkenntnissen aus der Analyse an [10]. Es werden die gezielt Daten beschafft, die noch benötigt werden, um die entwickelte Theorie anzureichern.

2.3.2 Theoretische Sättigung

Theoretical Sampling wird solange fortgeführt bis die theoretische Sättigung erreicht ist. Diese liegt vor, wenn keine neuen Erkenntnisse aus den verwendeten Daten gezogen werden können [10].

2.3.3 Memos

Udo Kuckartz bezeichnet Memos als "Form der Fixierung der eigenen Ideen und theoretischen Gedanken"[5]. Sie helfen Forscher:innen im Laufe der Analyse, indem

sie als Art Notizen verstanden werden können, die dazu dienen, Fragen, Ideen und Gedanken festzuhalten. Anfangs handelt es sich bei Memos um teilweise banale Anmerkungen und Fragen, später um Ergebnisse der Datenanalyse[5].

3 Grundlegende Definitionen

3.1 Einleitung

In diesem Kapitel werden Begriffe definiert, die für das Verständnis dieser Arbeit von Bedeutung sind.

3.2 Paarprogrammierung

Kent Beck definiert Paarprogrammierung in Extreme Programming wie folgt:

“Pair programming is a dialog between two people simultaneously programming (and analyzing and designing and testing) and trying to program better.” (deutsch: Pair Programming ist ein Dialog zwischen zwei Personen, die gleichzeitig programmieren (und analysieren und entwerfen und testen) und versuchen, besser zu programmieren.) [2]. Paarprogrammierung ist eine agile Softwareentwicklungsmethode, bei der zwei Programmierer gemeinsam an derselben Aufgabe arbeiten. Dabei sitzen sie in der klassischen Art und Weise an einem Computer und arbeiten eng zusammen, um Code zu schreiben, zu überprüfen und zu verbessern. Neben der bereits erklärten klassischen Paarprogrammierung gibt es auch “distributed pair programming”. Hier arbeiten die Paare an verschiedenen Rechnern an derselben Aufgabe.

Die Partner:innen teilen sich in beiden Varianten die Verantwortung für den Code und tauschen kontinuierlich Ideen aus, um gemeinsam effizientere Lösungen zu finden.

3.3 Entscheidungsfindung

Eine Entscheidungsfindung ist ein Prozess, bei dem im Rahmen dieser Masterarbeit zwei Personen eine Entscheidung treffen. In den meisten Fällen stehen mehrere Optionen zur Verfügung aus denen gewählt werden kann. Der Prozess besteht typischerweise aus der Identifizierung eines Problems oder einer Entscheidungsfrage, der Sammlung und Analyse relevanter Informationen, der Bewertung von Konsequenzen und der Auswahl derjenigen Option, die als am besten geeignet gilt. Die Entscheidungsfindung kann von verschiedenen Faktoren beeinflusst werden, wie zum Beispiel individuellen Präferenzen, Risikobereitschaft, soziale Einflüsse und vorhandenem Wissen.

3.4 Wissenstransfer

Definition

“Knowledge Transfer is then any attempt of the developers to close a gap in knowledge, either by exchanging knowledge they already possess or by acquiring new knowledge they still lack.” (deutsch: Wissenstransfer ist dann jeder Versuch der Entwickler, eine Wissenslücke zu schließen, entweder durch den Austausch von Wissen, das sie bereits besitzen, oder durch den Erwerb von neuem Wissen, das ihnen noch fehlt.) [12]. Mit diesen Worten erklärt Franz Zieris in seiner Dissertation “Qualitative Analysis of Knowledge Transfer in Pair Programming” den Wissenstransfer. Kurz gesagt ist der Wissenstransfer der Versuch, eine vorhandene Wissenslücke zu schließen.

Dies kann durch gegenseitigen Wissensaustausch oder gemeinsames Erarbeiten von Wissen geschehen.

Aufteilung

In ihrem Paper "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues" definieren die beiden Autorinnen Maryam Alavi und Dorothy E. Leidner den Wissenstransfer, in dem sie das Wort in seine Einzelteile unterteilen: "Das Wissen" und der "Transfer"[1]. Das Wissen definieren sie als "als Informationen und Erfahrungen, die in einem bestimmten Kontext nützlich gruppiert sind" und den Transfer als "Übertragung eines Elements von einer Seite auf die andere"[1].

S- und G-Gap

Es ist wichtig, die Begriffe S- und G-Gap zu verstehen, um der Arbeit folgen zu können. In ihrem Paper "Explaining Pair Programming Session Dynamics from Knowledge Gaps" entdecken die beiden Autoren Lutz Prechelt und Franz Zieris zwei Arten der Wissenslücke, die sich durch die analysierten Programmiersitzungen ziehen[13]. Diese nennen sie S- und G-Gap. Das S steht für spezifisch, während das G für generisch steht. Das S-Gap ist eine Wissenslücke, die in Verbindung mit dem speziellen Softwaresystem steht. Das G-Gap wiederum bezeichnet eine Wissenslücke, die sich auf "allgemeine Softwareentwicklungsmethoden und -technologien" bezieht, beispielsweise die Programmiersprache[13].

3.5 Ruder

Ruder ist ein Begriff, der im Rahmen dieser Masterarbeit eingeführt wurde. Bei dem Ruder handelt es sich um das Besitzen der Maus und Tastatur. Eine Person, die das Ruder hat, entspricht dem, was im Rahmen der Paarprogrammierung als "Driver" bezeichnet wird. In ihrem Paper "Liberating Pair Programming Research from the Oppressive Driver/Observer Regime" kommen die beiden Autoren Lutz Prechelt, Stephan Salinger und Franz Zieris unter anderem auf die Schlussfolgerung, dass das Driver/Observer-Modell " mit seiner unflexiblen Zuweisung von Verantwortlichkeiten angehende Paarprogrammierer zu dysfunktionalem Verhalten verleiten kann", sodass der Begriff "Driver" einen fragwürdiges Phänomen zu sein scheint und in dieser Masterarbeit nicht verwendet wird[9].

4 Verwandte Arbeiten

4.1 Einleitung

Dieses Kapitel enthält einen kurzen Exkurs zu wissenschaftlichen Arbeiten, die sich mit Thema übergreifenden Aspekte auseinandergesetzt haben. Dabei folgt die Vorstellung der Paper nach folgendem Prinzip:

1. Welches Problem wurde angegangen?
2. Wie wurde das Problem gelöst?
3. Was hat es gebracht?
4. Wie steht es in Verbindung zu der eigenen Arbeit?

4.2 Disengagement in pair programming: Does it matter? [7]

“Disengagement in pair programming: Does it matter?” ist ein Paper, das im Jahr 2012 von Laura Plonka, Helen Sharp und Janet van der Linden verfasst wurde.

1. Welches Problem wurde angegangen?
Das Phänomen, das in dem Paper untersucht wird, wird “Disengagement in pair programming” genannt. Diesen Begriff definieren die Autorinnen wie folgt: “disengagement, where a developer is not focusing on solving the task or understanding the problem and allows their partner to work by themselves” (deutsch: “Disengagement tritt auf, wenn ein Entwickler sich nicht auf die Aufgabe oder das Verstehen der Aufgabe konzentriert und somit seinem Partner erlaubt allein zu arbeiten”). Dieses Disengagement führt laut den Autorinnen zu der Folge, dass die Zusammenarbeit des Paares behindert und die Performance negativ beeinträchtigt wird.
2. Wie wurde das Problem gelöst?
Die drei Autorinnen haben mit vier unterschiedlichen Arten von Daten gearbeitet: Fragebögen, Audio, Videomaterial und Interviews.
Die Analyse haben sie in drei Schritten durchgeführt. Im ersten Schritt haben sie die Daten herausgesucht, die ein solches Disengagement vorweisen. Dann haben sie die Gründe für dieses Disengagement gesucht und im letzten Schritt daraus abgeleitet, wie man dies verhindern kann.
3. Was hat es gebracht?
In ihrem Ergebnispart unterscheiden die drei Autorinnen zwischen Anfänger:innen und Expert:innen in Bezug auf die zu lösende Aufgabe und nicht auf die Programmier- oder Berufserfahrung im Generellen. Eine Person, die mehr in der Lage ist das aktuelle Problem zu lösen wird hierbei als “Experte” betitelt, die andere Person als Anfänger. Sie kommen auf fünf Gründe, die zu einem Disengagement führen können:

- (a) Unterbrechungen
- (b) Aufteilung der Aufgaben nach Stärke
- (c) Zu einfache Aufgaben
- (d) Sozialer Druck
- (e) Zeitdruck

Daraus leiteten sie die folgenden Lösungen ab:

- (a) Die Anfänger sollten häufiger zu Maus und Tastatur greifen.
- (b) Die Experten sollen ihre Gedanken nach außen tragen und Feedback einholen.
- (c) Die Anfänger sollen Unklarheiten klären, um nicht den Anschluss zu verlieren und um sich zu vergewissern, alles richtig verstanden zu haben.

4.3 Qualitative Analysis of Knowledge Transfer in Pair Programming [12]

“Qualitative Analysis of Knowledge Transfer in Pair Programming” ist eine Dissertation, die im Jahre 2020 von Franz Zieris veröffentlicht wurde.

1. Welches Problem wurde angegangen?
Die Dissertation befasst sich mit zwei wesentlichen Themen. Zum einen, wie ein Wissenstransfer während der Paarprogrammierung abläuft und zum anderen, was für eine Art Wissen transferiert wird.
2. Wie wurde das Problem gelöst?
Herr Zieris hat als Forschungsmethode die Grounded Theory Methodology gewählt und Videos analysiert, die Paare bei der Paarprogrammierung zeigen. Zusätzlich holte er Feldbeobachtungen und Interviews ein.
3. Was hat es gebracht?
Herr Zieris kam bei seiner Analyse auf mehrere Punkte. Zum einen kann ausgeschlossen werden, dass Wissenstransfer nur stattfindet, wenn eine Anfänger/-Experte Konstellation stattfindet
Außerdem kam heraus, dass die häufigste Art von Wissenstransfers die ist, die Wissen über das System überführt. Erst wenn dieses Wissen gedeckt ist, geht es über zu Fragen, die von generellen Programmierkenntnissen handeln.
4. Wie steht es in Verbindung zu der eigenen Arbeit?
Der Wissenstransfer spielt eine zentrale Rolle in meiner Masterarbeit, da dieser stattfinden muss, um die Wissenslücke zu schließen, welche laut meinen Ergebnissen zu einer Passivität führen kann.

4.4 Understanding Pair Programming: The Base Layer [8]

Das Buch "Understanding Pair Programming: The Base Layer" erschien am 6. Dezember 2013 und stammt von Prof. Dr. Lutz Prechelt und Stephan Salinger.

1. Welches Problem wurde angegangen?
Das Buch ist laut den Autoren als "foundation of elementary concepts useful for analyzing and understanding pair programming sessions" (deutsch: eine Grundlage für elementare Konzepte, die für die Analyse und das Verständnis von Paarprogrammierungssitzungen nützlich sind.) [8]
2. Wie wurde das Problem gelöst?
Prechelt und Salinger haben sieben unterschiedliche Programmiersitzungen analysiert.
3. Was hat es gebracht?
Als Ergebnis ist der "Base Layer" entstanden, der viele unterschiedliche Basiskonzepte enthält, die übernommen oder an die eigene Analyse angepasst werden können. Zusätzlich wird erklärt, wie diese Konzepte verwendet werden sollen.
4. Wie steht es in Verbindung zu der eigenen Arbeit?
In meiner Arbeit habe ich ebenfalls Paarprogrammierungssitzungen analysiert. Das Buch hat mir eine erste Orientierung geboten.

4.5 Egoless Programming [11]

Das Paper "Egoless Programming" erschien im Jahr 1999 und stammt von Gerald M. Weinberg.

1. Welches Problem wurde angegangen?
Das Problem, das in dem Paper angesprochen wird, ist das Ego in der Programmierung. Menschen neigen dazu, Code als ein von ihnen erstelltes Eigentum zu betrachten, dessen Kritik direkt auf sie zurückzuführen ist.
2. Wie wurde das Problem gelöst?
Das Problem wird durch die Einführung des "egoless programming" gelöst, bei dem das Aufdecken von Fehlern als Möglichkeit zur Verbesserung gesehen wird und nicht als Kritik. Personen sollen hierbei gegenseitig ihre Codes überprüfen.
3. Was hat es gebracht?
Durch das "egoless programming" schafft man eine Umgebung, in der Fehler nicht als Kritik gesehen werden. Die gegenseitige Überprüfung der Codes führt zu einer besseren Codequalität. Auch die Zusammenarbeit scheint gestärkt worden zu sein.
4. Wie steht es in Verbindung zu der eigenen Arbeit?
Das Phänomen des "Egos beim Programmieren" wird in meiner Arbeit ebenfalls angesprochen.

5 Methodik

5.1 Einleitung

Das folgende Kapitel beginnt mit einer Erklärung der vorhandenen Daten. Anschließend folgt die Methodik, die den Weg von der Datenauswahl bis zur Theoriekonstruktion beschreibt.

5.2 Daten

5.2.1 Videos & schriftliches Material

Videos

Als Quelle für meine Datenerhebung wurden mir von der Arbeitsgruppe Software Engineering Videos zur Verfügung gestellt. Diese Videos wurden seit 2007 gesammelt und in 13 unterschiedlichen Unternehmen aufgenommen. Sie dokumentieren verschiedene Entwicklerinnen und Entwickler während der Paarprogrammierung, meist zu zweit, aber auch vereinzelt zu dritt oder viert. Insgesamt kann bei der Analyse auf drei Kanäle zugegriffen werden: die IDE, die man auf dem Bildschirm der Programmierer:innen sieht, die Kamera, die auf die Programmierer:innen ausgerichtet ist und den Audiokanal.

Die Videos sind in einzelne Ordner eingeteilt. Der Name dieser Ordner stellt sich zusammen aus einem Buchstaben und dem Namen des Unternehmens. Diese beiden Teile werden durch einen Bindestrich getrennt. Um den Aufbau dieser Ordner besser zu veranschaulichen, folgt nun ein Beispiel mit dem provisorischen Unternehmen "Darakci". Der Ordner, der die Videos enthält, die in diesem Unternehmen aufgezeichnet wurden, heißt demnach beispielsweise "R-Darakci". Die Videos in dem Ordner bekommen dann die Namen RA1, RA2, RA3 und so weiter. Die Personen, die in RA1 vorzufinden sind, werden R1 und R2 genannt. In RA2 heißen sie R3 und R4. Taucht eine Person in mehreren Videos auf, behält sie ihren ersten Namen, sodass in RA3 rein theoretisch R1 und R3 auftauchen könnten. Die betrachteten Situationen sind echte Szenarien, die im normalen Arbeitsleben stattfinden, ohne künstliche Experimente oder extra für die Aufzeichnung definierte Aufgaben.

Die Länge der Videos ist sehr unterschiedlich und reicht von 30 Minuten bis zu drei Stunden, wobei es sich bei den 30 Minuten Videos meist um einzelne Teile der Sitzungen handelt, da diese oft aufgeteilt wurden.

Pre-Session Fragen

Neben den Videos gibt es zusätzliche Pre-Session Informationen, die in schriftlicher Form als Fragebogen vorzufinden sind und von den Teilnehmer:innen vor der Sitzung ausgefüllt wurden. Diese Fragebögen enthalten die folgenden Fragen:

- Klassifizierung der zu bearbeitenden Aufgabenstellung(en)
- Kurzbeschreibung der zu bearbeitenden Aufgabenstellung(en)

5. Methodik

- Die Aufgabenstellung hat folgende besondere Schwierigkeiten
- Warum eignet sich die Aufgabe für die Paarprogrammierung?
- Seit wie viel Monaten/Jahren arbeiten Sie als Entwickler?
- Seit wie viel Monaten/Jahren praktizieren Sie regelmäßig (mehrere Male im Monat) Paarprogrammierung?
- Für wie eingespielt halten Sie sich mit ihrem Partner (in Schulnoten)?
- Was sind Ihre Erwartungen an die nachfolgende Reflexion?

Post-Session Fragen

Nach der jeweiligen Session erfolgt das Ausfüllen eines zweiten Fragebogens. Hierbei geht es vor allem darum, eigene Einschätzungen der Programmierer zu erhalten und dadurch mögliche erste Hinweise auf Phänomene oder interessante Stellen im Video zu erhalten.

- Bewerten Sie inwiefern es hilfreich war die Aufgabe im Paar zu lösen (in Schulnoten)?
- Sind Sie mit der Bewertung der Aufgabe schneller vorangekommen als erwartet?
- Bitte unterteilen Sie Ihre Sitzung chronologisch in Phasen und geben Sie an wie viel Zeit Sie für die einzelnen Phasen benötigt haben. Sie können die Dauer in Prozentangaben oder in Minuten angeben. Die Dauer Ihrer Sitzung beträgt:
- Denken Sie kurz nach, welches die drei bedeutendsten Phasen sind, die Ihnen bzgl. der Sitzung einfallen? Bitte geben Sie eine kurze Begründung Ihrer Wahl an.
- Hätten Sie für bestimmte Phasen (oder innerhalb bestimmter Phasen) gerne die Paarkonstellation aufgehoben und allein gearbeitet? Geben Sie ggf. die Teilschritte und einen kurzen Grund für Ihre Entscheidung an:
- Gab es bestimmte Teilschritte, in denen Sie die Zusammenarbeit mit Ihrem Partner als besonders hilfreich empfunden haben? Geben sie ggf. die Phasen und einen kurzen Grund für Ihre Entscheidung ab.

Bei den Fragen in den Bögen handelt es sich um Fragen, die auf das Interesse der damaligen Forschung zugeschnitten sind.

Retrospektive

Die Retrospektive ist ein Gespräch zwischen Forscher:in und Programmierer:in und stellt eine individuelle Einschätzung über die aktuelle Sitzung von dem/der Forscher:in dar. Sie besteht nicht aus vordefinierten Fragen, die beantwortet werden müssen, sondern aus persönlichen, handschriftlichen Gedanken des jeweiligen Forschers beziehungsweise der jeweiligen Forscherin.

Diese schriftlichen Aufzeichnungen können betrachtet werden, um erste Hinweise auf eventuell interessante Stellen und Phänomene zu bekommen.

MAXQDA

MAXQDA ist ein Desktop-Softwareprodukt, das Forscher:innen dabei helfen soll, qualitative Datenanalyse zu betreiben. Forscher:innen binden hierzu die gewünschten Texte ein und versehen sie in der Software beispielsweise mit Codes und Memos. Des Weiteren hat man die Möglichkeit bei MAXQDA bestimmte Informationen zu den Codes zu erhalten, beispielsweise die Häufigkeit oder den Ort.

5.2.2 GTM auf bestehender Datenbasis

Bei den in 4.2.1 beschriebenen Daten handelt es sich wie bereits angeschnitten um Daten, die bereits vorhanden waren und somit nicht im Rahmen dieser Masterarbeit erhoben werden mussten.

Der erste Teil der Arbeit entstand in enger Zusammenarbeit mit Herrn Linus Ververs, einem der Doktoranden der Arbeitsgruppe. Das alleinige Angucken dieser Videos brachte gewisse Herausforderungen mit sich, da wir nur ein stiller Beobachter waren und zu keiner Zeit Fragen stellen konnten. Wenn man bedenkt, dass die Videos bereits mehr als 10 Jahre alt sind, erscheint die Kontaktaufnahme mit den Protagonisten als nahezu unmöglich. Wir mussten mit dem vorhandenen Material arbeiten und hatten keinen Einfluss darauf. Dies machte die Interpretation von Episoden umso schwieriger.

Es war an vielen Stellen unklar, wie genau die Personen Kommentare meinten. Einmal diskutierten wir 30 Minuten darüber wie ein "Hä" gemeint sein könnte.

Man muss bedenken, dass es um aktuell laufende Projekte geht und es sich somit nicht um Szenarien in den Videos handelt, die auf unsere Analyse zugeschnitten sind. Zusätzlich steigen die Videos in den meisten Fällen mitten im Projekt ein, sodass wir nur einen Bruchteil des eigentlichen Projektes mitbekamen. Dies machte es schwer, die Güte von Entscheidungen einzuschätzen.

Daher war es umso wichtiger, auszuwählen, welche Entscheidungsepisoden von Bedeutung sein könnten – eine der Herausforderungen qualitativer Forschung. "Theoretical sampling" war hier das Stichwort. Bei der Arbeit mit bereits vorhandenen Daten ermöglicht Theoretical Sampling eine zielgerichtete Auswahl von Daten, die für die Entwicklung von Theorie relevant sind. Anstatt sich auf alle verfügbaren Daten zu konzentrieren, kann man sich auf jene konzentrieren, die zur Entwicklung und Verfeinerung von theoretischen Konzepten beitragen.

5.3 Verlauf der Analyse

5.3.1 Identifizierung spannender Phänomene

Zu Beginn unserer Zusammenarbeit gab mir Herr Ververs einen Einblick in den aktuellen Forschungsstand.

Um an Informationen zu gelangen, begannen wir damit die oben erwähnten Videos anzusehen. Zu Beginn wusste ich nicht genau, was mich erwartet und worauf ich mich fokussieren sollte. Dies ist typisch bei der Grounded Theory Methodology. Man startet mit einer offenen und breiten Fragestellung und passt diese anhand der gewonnenen Erkenntnisse an. Da Herr Ververs bereits länger an dem Thema "Entschei-

5. Methodik

dungsfindung bei der Paarprogrammierung“ forschte, entschieden wir uns hier einzusteigen.

Wir starteten damit uns auf sogenannte ‘Entscheidungsepisoden’ zu konzentrieren – bestimmte Stellen in den Videos, an denen Entscheidungen getroffen wurden. Unser Ziel war es durchgehend, möglichst vorurteilsfrei und ohne klare Erwartungen herauszufinden, wie Personen Entscheidungen treffen. Welche Faktoren führten zu einer Entscheidung? In welchem Kontext fand die Entscheidung statt? Gab es Diskussionen?

All diese Aspekte versuchten wir zu berücksichtigen, als wir uns die ersten Videos ansahen.

Beim ersten Ansehen hat man das Ziel, das Setup des Videos zu verstehen und nach ersten interessanten Phänomenen Ausschau zu halten. Wer sind die Paare und was für eine Aufgabe wollen sie erfüllen?

Zunächst begannen wir mit dem Video CA1. Herr Ververs kannte die Videos und hatte sein Interesse zum damaligen Zeitpunkt auf die Autorität von Personen gelegt. Die beiden Programmierer:innen werden C1 und C2 genannt, wobei C2 die autoritäre Rolle einzunehmen schien.

Wir markierten grundsätzlich alle Entscheidungsepisoden mit dem Code “DM” (decision making), um später einen genaueren Blick auf genau diese Stellen werfen zu können.

Danach gingen wir über zu CA2, um C2 in einer anderen Paarkonstellation zu sehen und dessen Autorität zu untersuchen. Hierbei fiel auf, dass C2 mehr Gegenwind bekam und sich nicht so einfach durchsetzen konnte, wie in CA1. Die Tatsache, dass Herr Ververs den Großteil der Videos bereits gesehen hatte, erleichterte das Auswählen der Videos, da er bereits wusste, was in welchem Video vorzufinden war.

Er schlug somit anschließend das Video CA5 vor, da es hier um eine Entscheidungs-episode ging, die durch ein Missverständnis unnötig verlängert wurde. Wir wollten herausfinden, an welcher Stelle das Missverständnis auftaucht und wieso. Uns fiel auf, dass Aussagen mit unterschiedlich viel Nachdruck getätigt wurden und begannen sie zu kategorisieren.

Nach dem Ende der Analyse ging es zum Video DA2. In DA2 besteht das Paar aus D3 und D4. Hier war das Besondere, dass wir nicht nur einzelne Entscheidungs-episoden hatten, sondern auch eine Episode mit abhängigen Entscheidungen, was uns nach dem dritten Mal ansehen, auffiel.

5.3.2 Theoriegetriebene Stichprobenziehung

Bei der Analyse und nach wiederholtem Ansehen des Videos DA2 fiel eine auffällige Charakteristik auf: die Passivität einer der beiden Programmierer. Überlegungen wie “Könnte nicht jeder diese Aufgabe übernehmen?” kamen auf und regten dazu an, einen genaueren Blick auf diese Passivität zu werfen. Es schien interessant zu sein zu untersuchen, ob es weitere Fälle gibt, in denen eine solche Passivität während einer Sitzung zum Vorschein kam. Sind Situationen denkbar, in denen die Fortsetzung der Paarprogrammierung überdacht werden sollte? Ab diesem Punkt ging ich über zu meiner eigenen Analyse und verließ die enge Zusammenarbeit mit Herrn Ververs.

Dennoch teilte ich regelmäßig meine Ergebnisse mit ihm, da er der Betreuer meiner Arbeit war.

Bei der Analyse der Videos war die Codebasis und was dort auf Produktebene passiert, für uns eher uninteressant. Wir fokussierten uns mehr auf die Prozessseite und die Dialoge, die zwischen den Partnern stattfanden.

Wie bei der GTM üblich sprang ich wieder zu der Datensammlung. Ich begann erneut, die Videos zu durchforsten, diesmal jedoch mit dem Fokus auf die Unterscheidung zwischen passiver und aktiver Haltung eines oder beider Partner. Dabei lag das Augenmerk auf der Dynamik zwischen den Partnern und dem Gefühl, das sich beim Betrachten der Videos ergibt – weniger auf dem genauen Wortlaut der Partner.

Auswahl der Videos

Ich begann während des chronologischen Ansehens der Videos meine ersten Codes zu formulieren und entschied darauf basierend, welche Videos in meine Analyse aufgenommen werden sollten. Codes, wie

- PP abbrechen?
- Mehrwert von PartnerB?
- Ruderablehnung
- Ja Sager
- Mehrwert des Partners, der nicht am Ruder sitzt?

sorgten dafür, dass ich diese Videos in meine Analyse aufnahm. Insgesamt habe ich 24 verschiedene Videos angeguckt.

- AA1
- BA1
- CA1, CA2, CA3, CA4, CA5
- DA2, DA3, DA4, DA5, DA6
- EA1, EA2, EA3, EA4, EA5, EA6, EA7
- NA1
- PA1, PA2, PA3, PA4

Den Großteil der Videos konnte ich verwerfen, da die gesuchte Passivität nicht im für mich erforderlichen Ausmaß vorzufinden war. Es gab zahlreiche Videos, in denen die Paarprogrammierung so ablief, wie man es sich vorstellen würde. Es gab zwar eine dominante Person, aber die andere brachte sich dennoch genug ein und trat nicht vollständig in den Hintergrund.

Letztendlich waren drei Videos interessant genug, um sie in meine Analyse aufzunehmen: DA2, DA4 und DA5. DA2 war mir bereits durch die ersten Analysen gut bekannt. Bei DA4 löste sich ein ähnliches Gefühl in mir aus wie bei DA2. DA5/DA6 blieb mir als besonders positiv im Gedächtnis, weshalb ich es auch in meine Analyse aufnahm – dazu später mehr.

5.3.3 Näherer Blick auf die ausgewählten Videos

Alle zu analysierenden Videos wurden in demselben Unternehmen aufgenommen, einer Firma, die sich auf die Entwicklung von Immobiliensoftware spezialisiert hat. Um die Datenschutzrichtlinien zu wahren, wurden den Protagonisten der Videos die folgenden Kürzel zugewiesen.

1. D3: Partner A in DA2
2. D4: Partner B in DA2
3. D7: Partner A in DA4
4. D6: Partner B in DA4
5. D2: Partner A in DA5/DA6
6. D8: Partner B in DA5/DA6

Zunächst folgt eine kurze Zusammenfassung der von mir ausgewählten Videos, um ein besseres Verständnis für diese zu verschaffen.

1. DA2

In dem Video "DA2" wird eine Paarprogrammierungssitzung zwischen zwei Entwicklern, D3 und D4, dokumentiert. Zu Beginn bringt D3 seinen Kollegen D4 über den aktuellen Stand des Projekts und die Zielsetzung, eine Toolbar zu implementieren, auf den neuesten Stand. Obwohl D3 bereits länger im Unternehmen tätig ist, gibt er an, in der Java-Programmierung noch ein Anfänger zu sein. Im Gegensatz dazu scheint D4 in Java erfahrener zu sein.

Während der Sitzung entscheiden sie, sich an bereits existierenden Toolbars zu orientieren. Sie suchen Hilfe bei einem Kollegen, D6, dessen Unterstützung sich jedoch als wenig hilfreich erweist. Daraufhin greifen sie auf die Hilfe von D7 zurück. D7 scheint die richtige Wahl zu sein und erklärt den beiden, was sie wissen müssen, bevor er wieder geht, sobald D4 das Konzept verstanden hat.

Nachdem D7 gegangen ist, übernimmt D4 die führende Rolle bei der Implementierung der Toolbar. Er bietet D3 während der Sitzung zweimal an, das Ruder zu übernehmen, was D3 jedoch ablehnt. Gegen Ende des Videos findet ein Refactoring statt, bei dem D3 schließlich das Ruder übernimmt.

2. DA4

Das Video dokumentiert eine Paarprogrammierungssitzung zwischen den Entwicklern D6 und D7. Die Sitzung beginnt mit technischen Problemen, die einige Zeit in Anspruch nehmen, bevor die eigentliche Arbeit beginnen kann.

D7 führt D6 in das zu lösende Problem ein und behält während der gesamten Sitzung das Ruder.

Obwohl D6 zwischendurch Vorschläge einbringt, werden diese meist von D7 mit Erklärungen abgelehnt.

Es gibt Momente, in denen D6 sich ablenken lässt und mit anderen Personen hinter ihm spricht, was für Verwirrung bei D7 sorgt. Eine Herausforderung tritt

auf, als beide auf ein Problem stoßen, das sie nicht sofort lösen können. In dieser Phase erscheint D4, um Hilfe anzubieten, jedoch ohne großen Erfolg.

D7 verlässt dreimal die Session. Das sind die drei Momente, in denen D6 zum Ruder greift. Bei dem einen Mal teilt er mit D7 seine daraus entstandene Erkenntnis.

Während der ganzen Sitzung greift D3 wiederholt in das Geschehen ein.

Am Ende beschließt D7, die Arbeit zu Hause fortzusetzen, was auf eine unvollendete Sitzung hinweist.

3. DA5, DA6

In dem Video werden D2 und D8 in einer Paarprogrammierungssitzung gezeigt, die bereits im Gange ist, als das Video beginnt. D2 hat zuvor mit einer anderen Person an der Codebasis gearbeitet und ermutigt D8, bei Unklarheiten Fragen zu stellen, da D8 neu dazukam.

D8 findet einen Fehler. Um diesen zu klären, ziehen sie die Person hinzu, mit der D2 zuvor zusammengearbeitet hatte. Diese Person bestätigt den Fehler.

D2 übergibt D8 das Ruder und nimmt es nach ca. 9 Minuten wieder. Dann folgt eine Unterbrechung, weil D8 zu einem Meeting muss.

Ab hier beginnt DA6.

D2 macht weiter und erklärt D8 anschließend die Fortschritte.

Nach einer Zeit übergibt D8 D2 das Ruder.

Mitten im Video fragt D2 die andere Person, mit der zuvor zusammengearbeitet wurde, ob Unterstützung benötigt wird.

D8 macht anschließend den Commit.

D2 möchte D8 etwas zeigen. Sie werden zweimal von der anderen Person unterbrochen.

Gegen Ende des Videos entscheidet sich D2, der anderen Person zu helfen, und lässt D8 zurück.

Dann kommt die Forscherin.

Informationen zu den Protagonisten

Jedes der Videos folgt derselben Struktur. Ein Partner, der sich besser mit der aktuellen Codebasis auskennt, führt den anderen ein. Im Folgenden werden kurz einige Fakten zu den jeweiligen Personen beschrieben.

- D3: Programmierer seit drei Monaten, PP seit drei Monaten
- D4: Programmierer seit einem Jahr, PP zum ersten Mal
- D7: Programmierer seit vier Jahren, PP keine Angabe
- D6: Programmierer seit 11 Jahren, PP seit zwei Jahren
- D2: Keine Angaben
- D8: Programmierer seit 4 Jahren und 2 Monaten, PP seit 3 Jahren und 8 Monaten

5.3.4 Detaillierte Analyse der Sitzungen

Der erste Schritt war das Codieren in dem Video selbst, um ein besseres Verständnis der Videos zu bekommen. Anschließend setzen wir Whisper ein, um die Videos transkribieren zu lassen. Whisper arbeitet lokal auf dem Rechner, sodass zu keinem Zeitpunkt Videos irgendwo hochgeladen werden mussten. Dann setze ich meine Codierung in dem Transkript fort.

Die ersten Codes

Das Betrachten von DA2 offenbarte vier Arten der Interaktion zwischen den Partnern und bildete somit den Rahmen für die Kategorien, in die das Codebuch unterteilt wurde. Diese vier Kategorien sind: "Fragen", "Antworten", "Hilfe von außen" und "Entscheidungen".

Die Codes erstellte ich darauf aufbauend. Jeder Code besteht aus zwei Teilen. Zum einen eine Beschreibung der Handlung und zum anderen, von wem sie ausging. Diese beiden Teile werden mit einem "_" getrennt: Vorschlag_aktiv.

Nun folgt eine Liste mit den ersten Codes, die ich definierte. In der Aufzählung taucht nur der Code mit dem "aktiv" Suffix auf, um unnötige Platz einzusparen. Somit ist zu bedenken, dass jeder Code genauso auch für _passiv existiert.

Fragen

In dieser Kategorie geht es darum, welcher Partner eine Frage stellt.

- Frage_Aktiv

Antworten

In dieser Kategorie lag der Fokus auf der Art der Antwort – war sie hilfreich, und von wem kam sie? Die Bewertung, ob eine Antwort als hilfreich betrachtet wurde, erfolgte anhand der Reaktion des Partners bzw. der Partnerin auf diese Antwort, da meine Kenntnisse über die Codebasis an vielen Stellen zu gering waren. Die vergebenen Codes in dieser Kategorie sind wie folgt:

- HilfreicheAntwort_Aktiv
- Überfragt_Aktiv
- UnsichereAntwort_Aktiv

Hilfe von außen

In diesem Abschnitt geht es um Situationen, in denen eine dritte Partei zum Geschehen hinzustieß, unabhängig davon, ob sie gerufen wurde oder freiwillig dazu kam. Die Codes in dieser Sektion sind wie folgt:

- Frage_Aktiv
- HilfreicheAntwort_Aktiv

Entscheidungen

In dieser Kategorie geht es um das Treffen von Entscheidungen. Dabei wird untersucht, wer sinnvolle Vorschläge einbrachte, wie die Partner jeweils darauf reagierten und wer schließlich die endgültige Entscheidung traf.

- Vorschlag_Aktiv
- Zustimmung_Aktiv
- Entscheidung_Aktiv

Die letzte Sektion handelt von dem oben bereits erläuterten Ruder und dem Wechsel zwischen den Partnern.

Übernahme_Aktiv:

- Genommen
- Angeboten
- Abgelehnt

Dieselben Codes habe ich anschließend für DA4 verwendet.

Anpassung der Codes

Bei der Analyse der Videos DA2 und DA4 wurden Codes mit den Suffixen `_Aktiv` und `_Passiv` entwickelt. Die Analyse von Video DA5/6, das als Positivbeispiel diente und somit nicht zu den anderen Videos passen sollte, sollte nicht die gleichen Codes aufweisen. Falls dies der Fall wäre, hätte etwas nicht gestimmt. Da in DA5/6 keine deutliche Aktiv/Passiv-Dynamik zu erkennen ist oder zumindest nicht in dem Maße wie in den anderen Videos, gestaltete sich die Zuordnung der Codes mit den Suffixen wie erwartet als schwierig.

Es schien, dass D2 in der Anfangsphase der aktivere Part war, was möglicherweise darauf zurückzuführen ist, dass D8 von D2 eine Erklärung zur Codebasis erhielt. Allerdings brachte sich D2 durchgehend ein und übernahm zeitweise den aktiveren Part, falls man in diesem Video überhaupt von einer solchen Unterscheidung sprechen kann. Somit erwiesen sich diese Codes als nutzlos für dieses Video, und die Idee der Unterscheidung zwischen `Partner:inA` und `Partner:inB` entstand.

Die Person, die sich mit der vorliegenden Codebasis und dem zu lösenden Problem auskennt, wird `Partner:inA` genannt und die andere Person `Partner:inB`.

Die Unterscheidung auf diese Art ermöglicht die Analyse der Teilnehmer:innen mit unterschiedlichen Ausgangssituationen.

Die Codes für die einzelnen Kategorien wurden durch umfassende und wiederholte Analysen der Videos entwickelt und werden im Folgenden für jede Kategorie näher erläutert.

Es gilt dasselbe wie zuvor: Die Codes werden nur für `Partner_A` aufgeführt, gelten aber genauso für `Partner_B`. Die neuen Codes lauteten demnach:

Fragen

1. Frage_`Partner:inA`

- **Beschreibung:**
Dieser Code wurde verwendet, um einen Moment in der Programmsitzung zu markieren, in dem `Partner:inA` `Partner:inB` eine Frage stellt.
- **Zitat:**
D4: Weil wir in der anonymen Klasse sind.
D3: Anonymen Klasse? Was ist eine anonyme Klasse?
Das Zitat befindet sich in dem Video DA2.3 und beginnt ab 0:15:19.6.
- **Kontext:**
Das Verschieben einer Klasse hat eine Fehlermeldung zur Folge. Diese Fehlermeldung befindet sich in einer "SelectionAdapter" Klasse, die D4 als anonyme Klasse deklariert, woraufhin D3 fragt, was eine anonyme Klasse ist.

Antworten

1. HilfreicheAntwort_`Partner:inA`

- **Beschreibung:**
Dieser Code wurde verwendet, um einen Moment in der Programmsitzung zu kennzeichnen, in dem `Partner:inA` `Partner:inB` eine hilfreiche

Antwort gibt. Das "hilfreich" wurde an der Reaktion des Partner:inB eingeschätzt.

- Zitat:
D4: Mit welchem Shortcut hast du das jetzt gemacht? (...)
D3: Steuerung 1, Enter.
Dieses Zitat findet man in DA2.4 ab 0:17:40.0.
- Kontext:
D3 wendet während eines Refactoring einen Shortcut an, den D4 noch nicht kennt.
- Zitat:
D7: Ach, ich hab den Hashcode nicht überschrieben.
D6: Was ist denn das eigentlich? Was bringt man denn den Hashcode?
D7: Der Hashcode ist dafür da, das ist so was wie ein...wie ein eindeutiger Identifier für das Objekt.
Dieses Zitat ist aus dem Video DA4 ab 0:30:38.0.
- Kontext:
Bei dem Durchsehen des Codes fällt D7 ein, dass er den Hashcode noch nicht überschrieben hat, woraufhin D6 ihn fragt, um was es sich bei einem Hashcode handelt.

2. Überfragt_Partner:inA

- Beschreibung:
Dieser Code wurde verwendet, um PartnerA in einem überfragten Moment zu zeigen.
- Zitat:
D4: Ist das ein Eclipse-View oder?
D3: Ich habe wirklich keine Ahnung von dem Extension-Point. Bin ich nicht wirklich komform damit. Aber da kann ich keine großen Informationen geben.
Dieses Zitat befindet sich in DA2.1 ab Minute 0:10:12.0.
- Kontext: Diese Frage bezieht sich wahrscheinlich auf die Entwicklungsumgebung.

3. KeineAntwort_Partner:inA

- Beschreibung:
Hier folgte keine Antwort seitens Partner:inA auf Partner:inB's Frage.
- Zitat:
D4: Aber die können wir ja hier einfach als viewpart casten setzt um mit den Worten "View part casten"
D3: *Keine Antwort*
D4: Oder?
D3: *Keine Antwort*
Dieses Zitat taucht in Video DA2.3 auf und beginnt ab 0:13:20.0.

5. Methodik

- Kontext:
Die beiden Programmierer verschieben eine Klasse. Anschließend entsteht eine Fehlermeldung, die B mit dem oberen Vorschlag beheben will.

4. NichtZielführendeAntwort_Partner:inB

- Beschreibung:
Partner:inA signalisiert Partner:inB, dass die Antwort bzw. der Vorschlag nicht zielführend ist.
- Zitat:
D3: Abstract List Action?
D4: Nein, das hat ja nichts mehr mit List zu tun eigentlich, ne?
Dieses Zitat beginnt in DA2 ab 0:02:35.6.
- Kontext:
Es geht um die Benennung einer Superklasse, bei dem Extrahieren dieser.
- Zitat:
D6: Mir fällt so spontan so eine Idee wie so lauter Shadow Copies oder so.
D7: Boah. Ach Käse ey. Nee, aber das ist richtig gefr(...) Wenn wir da jetzt auch noch mit den Shadow Copies anfangen, dann ist das echt Käse.
Dieses Zitat beginnt in DA4 ab 0:39:39.0.
- Kontext:
D6 macht einen Lösungsvorschlag, welcher abgelehnt wird.

Hilfe von außen

1. Frage_Partner:inA

- Beschreibung:
Dieser Code wurde verwendet, um einen Moment in der Programmiersitzung zu kennzeichnen, in dem Partner:inA eine Frage stellt. Diese kann sich sowohl an den Experten oder die Expertin, den eigenen Partner, die eigene Partnerin oder an beide richten.
- Zitat:
D2: Wenn der auf Disagree steht, dann hatten wir hier gesagt, wir legen einen neuen an. Also kopieren den previous workflow, legen einen neuen an im Status edit. Und setzen den auf first proof. Müsste das nicht eigentlich Created heißen? Der geht ja zurück zum Ersteller.
Dieses Zitat ist aus dem Video DA5 und beginnt ab 0:08:06.0.
- Kontext:
Dem Paar ist ein Fehler im Code aufgefallen. Sie holen als Expertin eine Kollegin, die zuvor an dem Code gearbeitet hat und fragen sie.

2. HilfreicheAntwort_Partner:inA

- Beschreibung:
Dieser Code wird verwendet, wenn Partner:inA während der Anwesenheit des Experten oder der Expertin einen sinnvollen Beitrag liefert. Das kann sowohl das Beantworten einer Frage sein als auch ein Hinweis.

- Zitat:
D4: Wo kommen die Menüpunkte hin?
Experte1: Links in der Navigation.
Dieses Zitat ist aus dem Video DA2.1 und beginnt ab 0:27:22.4.
- Kontext:
Dies ist eine Frage über das Layout. Das Ziel der Sitzung ist es eine Toolbar zu bauen. D3 fragt, wo die Menüpunkte in diesem Layout auftauchen sollen.

Entscheidungen

1. Vorschlag_Partner:inA

- Beschreibung:
Dieser Code wird verwendet, wenn Partner:inA einen Vorschlag macht. Dies kann sowohl ein Vorschlag für einen nächsten Schritt als auch ein Namensvorschlag beispielsweise sein.
- Zitat:
D4: Abstract List Action?
D3: Nein, das hat ja nichts mehr mit List zu tun eigentlich, ne?
Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:35.6.
- Kontext:
Hierbei handelt es sich um einen Namensvorschlag.
- Zitat:
D2: Wir können das auch erstmal testen.
Dieses Zitat ist aus dem Video DA5.1 und beginnt bei 0:24:22.9.
- Kontext:
Hierbei handelt es sich um einen Vorschlag über den nächsten Schritt.

2. Zustimmung_Partner:inA

- Beschreibung:
Dieser Code steht für die Zustimmung von Partner:inA zu einem Vorschlag von Partner:inB.
- Zitat:
D4: Aber lass erst mal angucken, wie die das bei dem anderen gemacht haben.
D3: Ja
Dieses Zitat ist aus dem Video DA2.1 und beginnt bei 0:19:49.4.
- Kontext:
D3 und D4 wollen eine Toolbar bauen. D4 macht den Vorschlag zunächst, anzugucken, wie ihre Kollegen die Toolbars bis jetzt implementiert haben. D3 stimmt zu.

3. Entscheidung_Partner:inA

5. Methodik

- Beschreibung:
Dieser Code zeigt Stellen im Video auf, in denen Partner:inA die endgültige Entscheidung trifft.
- Zitat:
D7: Okay, ehm wir probieren das jetzt einfach mal.
Dieses Zitat ist aus dem Video DA4 und beginnt ab 1:29:43.0.
- Kontext: D7 hat eine Idee, die er erklärt. Anschließend bestimmt er, dass sein Vorschlag jetzt ausprobiert wird.

Übernahme_Partner:inA:

1. Genommen

- Beschreibung:
In solch einem Fall hat Partner:inA das Ruder selbst genommen.
- Zitat:
D4: Darf ich nochmal? Kurz
D3: Ja.
Dieses Zitat findet man in DA2.4 ab 0:20:07.0
- Kontext:
D4 möchte kurz die Maus haben und fragt dafür D3, der zu dem aktuellen Zeitpunkt die Maus hat.

2. Angeboten

- Beschreibung:
Dieser Code stellt Momente dar, in denen Partner:inA das Ruder übernimmt, nachdem dieses von Partner:inB angeboten wurde.
- Zitat:
D4: Ja, aber wenn du willst, kannst du jetzt auch...
Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:26:36.0.
- Kontext:
D4, der das Ruder hat, bietet D3 das Ruder an.

3. Abgelehnt

- Beschreibung:
Dieser Code bezeichnet einen Moment, in dem Partner:inA das angebotene Ruder abgelehnt hat.
- Zitat:
D4: Eh willst du dann nochmal machen, oder?
D3: Ick globe du bist mir involviert in dit Ganze. Ich bin da... für mich ist sie schon oberste Wissenskante.
Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:09:46.1.
- Kontext: D4 bietet D3 an den nächsten Schritt umzusetzen. Dieser lehnt ab.

Während des Codierprozesses traten zusätzlich die folgenden Codes auf, die keiner der bestehenden Kategorien zugeordnet werden konnten.

1. Abschweifen

- Beschreibung:
Dieser Code stellt Gespräche und Handlungen abseits der aktuell zu lösenden Aufgabe dar.
- Zitat:
D7: Ich dachte auch, er redet gerade mit mir haha
Dieses Zitat ist aus dem Video DA4 und beginnt ab 0:44:17.0.
- Kontext:
D6 fängt plötzlich an mit den Personen hinter ihm zu reden. Daraufhin fragt eine der Personen, ob er mit ihnen reden würde.

2. VersuchPartner:ininsBootzuHolen

- Beschreibung:
Dieser Code beschreibt Momente, in denen einer der beiden Partner:innen den anderen Partner oder die andere Partnerin versucht ins Boot zu holen.
- Zitat:
D4: Kennst du das denn mit dem OSGI?
D4: Classloading und sowas?
D3: Was? Claus? Was?
D4: Class-Loading. Claus-Loading
D3: Nicht wirklich.
D4: Also ehm, soll ich kurz sagen, oder?
D3: Naja, klar.
D4: Also wir haben ja jetzt das Problem gehabt. Wir wollten ja das erst hier in das...
Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:16:12.0.
- Kontext: D4 klärt D3 über eine Sache auf, die ihm nicht bekannt ist.

3. JaSagerPartner:inB

- Beschreibung:
In einer Episode fiel auf, dass einer der Partner überwiegend mit dem Wort "Ja" geantwortet hat. Dies zeigt dieser Code auf.
- Zitat:
D7: Das ist absoluter Bullshit.
D6: Ja
Dieses Zitat ist aus dem Video DA4 und beginnt ab 0:41:59.0.
- Kontext: Eine Situation, in der es nicht so läuft wie erhofft, führt zu dieser Aussage.

Die endgültigen Codes

Durch das Anwenden dieser Codes und das wiederholte Angucken der Videos machten sich vier Hauptkategorien bemerkbar:

1. Wissenstransfer
2. Entscheidung
3. Ruder

Ich begann meine Codes zu verfeinern und auf diese Kategorien zuzuschneiden. Hierbei entstanden dann die folgenden Codes:

Wissenstransfer

1. pull_Partner:inA

- Beschreibung: Partner:inA stellt Partner:inB eine Frage
- Zitat:

D4: Weil wir in der anonymen Klasse sind.
D3: Anonymen Klasse? Was ist eine anonyme Klasse?

Das Zitat befindet sich in dem Video DA2.3 und beginnt ab 0:15:19.6.

- Kontext:
Das Verschieben einer Klasse hat eine Fehlermeldung zur Folge. Diese Fehlermeldung befindet sich in einer "SelectionAdapter" Klasse, die D4 als anonyme Klasse deklariert, woraufhin D3 fragt, was eine anonyme Klasse ist.

2. push_Partner:inA

- Beschreibung: Partner:inA teilt sein Wissen mit Partner:inB
- Zitat:

D4: Kennst du das denn mit dem OSGI?
D4: Classloading und sowas?
D3: Was? Claus? Was?
D4: Class-Loading. Claus-Loading
D3: Nicht wirklich.
D4: Also ehm, soll ich kurz sagen, oder?
D3: Naja, klar.
D4: Also wir haben ja jetzt das Problem gehabt. Wir wollten ja das erst hier in das...

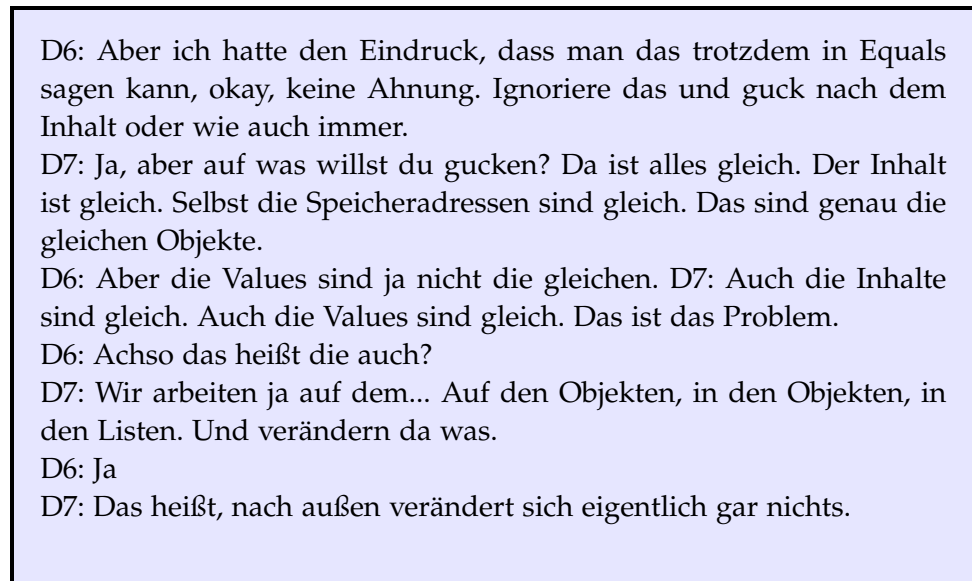


Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:16:12.0.

- Kontext:
D4 klärt D3 über eine Sache auf, die ihm nicht bekannt ist.

3. Offenbarung_Wissenslücke_Partner:inA

- Beschreibung: Partner:inA's Wissenslücke wird offenbart. Dies kann durch eine Handlung oder eigene Kommunikation entstehen.
- Zitat:



Dieses Zitat ist aus dem Video DA4 und beginnt ab 0:50:19.0.

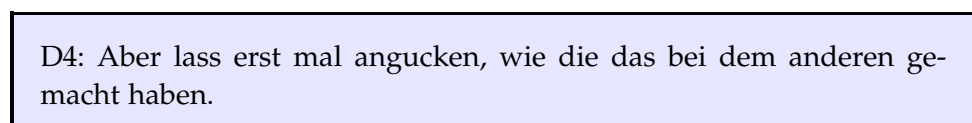
- Kontext: D6 macht einen Vorschlag, der aber nicht zu den Randbedingungen passt. Dadurch kommt zum Vorschein, dass er etwas nicht verstanden hat. D7 erklärt ihm anschließend, wo sein Fehler liegt.

Entscheidung

In Entscheidung gibt es eine Oberkategorie, die "Fähigkeit Vorschlag zu bewerten" genannt wird. Hierzu gehören die folgenden Codes:

1. Zustimmung_ohneErklärung_Partner:inA

- Beschreibung: Partner:inA stimmt ohne eine Erklärung zu.
- Zitat:



D3: Ja

Dieses Zitat ist aus dem Video DA2.1 und beginnt bei 0:19:49.4.

- Kontext:
D3 und D4 wollen eine Toolbar bauen. D4 macht den Vorschlag zunächst, anzugucken, wie ihre Kollegen die Toolbars bis jetzt implementiert haben. D3 stimmt zu.

2. Zustimmung_mitErklärung_Partner:inA

- Beschreibung: Partner:inA stimmt mit einer Erklärung zu.
- Zitat:

D7: Immer wenn der Equals gefragt wird ist false.
D6: Achso ja, weil eigentlich ist das nicht das gleiche. Ja.

Dieses Zitat ist aus dem Video DA4 und beginnt ab 0:43:39.0.

- Kontext:
D7 macht einen Vorschlag und erklärt diesen. D6 stimmt ihm anschließend zu in dem er Begründung, warum die Erklärung stimmt.

3. Ablehnung_ohneErklärung_Partner:inA

- Beschreibung: Partner:inA lehnt ohne eine Begründung ab.
- Zitat:

D4: Also Toolbar Action oder sowas nehmen? Ich weiß es nicht.
D3: Ja, naja. Toolbar Action.

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:48.0.

Die Ablehnung ist in diesem Fall "Ja, naja Toolbar Action." Dies wird deutlich, wenn man das Video sieht.

- Kontext:
Hierbei handelt es sich um einen Namensvorschlag.

4. Ablehnung_mitErklärung_Partner:inA

- Beschreibung: PartnerA lehnt mit einer Erklärung ab.
- Zitat:

D3: Abstract List Action?
D4: Nein, das hat ja nichts mehr mit List zu tun eigentlich, ne?

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:35.6.

- Kontext:
Hierbei handelt es sich um einen Namensvorschlag.

Zusätzlich gibt es den folgenden Code in der Kategorie "Entscheidungen":

1. Vorschlag_Partner:inA

- Beschreibung: Partner:inA macht einen Vorschlag.
- Zitat:

D4: Aber lass erst mal angucken, wie die das bei dem anderen gemacht haben.

Dieses Zitat ist aus dem Video DA2.1 und beginnt bei 0:19:49.4.

- Kontext: D3 und D4 wollen eine Toolbar bauen. D4 macht den Vorschlag, zunächst anzugucken, wie ihre Kollegen die Toolbars bis jetzt implementiert haben. D3 stimmt zu.

Ruder

1. Übernahme_Partner:inA

- Beschreibung: Partner:inA übernimmt Maus und Tastatur.
- Zitat:

D4: Ja, aber wenn du willst, kannst du jetzt auch...

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:26:36.0.

- Kontext:
D4 hat das Ruder und bietet dieses D3 an, woraufhin er zustimmt.

2. Ablehnung_Partner:inA

- Beschreibung: Partner:inA lehnt Maus und Tastatur ab.
- Zitat:

D4: Eh willst du dann nochmal machen, oder?
D3: Ick globe du bist mir involviert in dit Ganze. Ich bin da... für mich ist sie schon oberste Wissenskante.

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:09:46.1.

- Kontext: D4 bietet D3 an den nächsten Schritt umzusetzen. Dieser lehnt ab.

Es gibt Wörter, die nicht als Code vorkommen, aber eine zentrale Rolle in der Analyse gespielt haben. Diese sind "Wissenslücke zum Partner:in" und "Ego". Bei dem Analysieren der einzelnen Videos fiel auf, dass an vielen Stellen der aktivere Partner oder die aktive Partnerin über mehr Wissen zu verfügen schien und es somit eine Wissenslücke zu dem Partner beziehungsweise der Partnerin bestand. Zeitpunkte, in denen die aktive Person überfragt zu sein schien, brachten Situationen hervor, in denen sich passiv mehr einbrachte. An dieser Stelle dachte ich wiederholt über das Selbstwertgefühl der passiven Person nach und kam auf das Wort "Ego".

Im nächsten Schritt begann ich damit die oben definierten Kategorien und ihre dazugehörigen Codes in Verbindung miteinander zu bringen.

Beispielsweise folgte aus einer Ablehnung mit einer Erklärung ein Pull durch die Person, die ablehnte, da sie durch den Grund Wissen mitteilte. So hangelte ich mich durch und ging immer wieder zurück zu den Videos, bis die unten aufgeführte GT entstand.

5.3.5 Memos

Die Memos dienen zwei Hauptzwecken: Zum einen zur Dokumentation von Fragen, die während des Gesprächs mit dem Betreuer dieser Masterarbeit geklärt werden sollen, und zum anderen zur Festhaltung von relevanten Beobachtungen. Wie bereits erwähnt, basiert die Analyse mehr auf Beobachtungen und Dynamiken als auf bestimmten Wortlauten. In solchen Fällen unterstützen Memos dabei, bestimmte Beobachtungen präzise festzuhalten. Hier einige Memoexamples, die während meiner Arbeit entstandn:

- Wie stelle ich die Interaktion dar?
Hier ging es um die Frage, wie die Interaktion zwischen dem Experten und D4 dargestellt werden soll.
- D6 scheint im Boot zu sein
Ein Hinweis darauf, dass D6 noch am Prozess beteiligt ist
- Ab wann ist man ein JaSager?
- Traut sich zu diskutieren – weil keiner mehr da ist?

5.3.6 Zusätzliche Gedanken zu unterschiedlichen Videos

Achtung, die sind keine analysierten Theorien, sondern lediglich interessante Aspekte, die mir während des Anschauens der Videos aufgefallen sind.

CA1

- C1 hat eine Stunde vorher begonnen zu programmieren & C2 stößt dazu. Dennoch scheint C2 autoritärer zu sein.
- C2 arbeitet häufig ohne Rücksprachen und ignoriert C1 an vielen Stellen.
- Auch kritisiert er an vielen Stellen die geleistete Arbeit von C1.

CA2

1. C2 wird erneut eingeführt. Dieses Mal von C5. C2 kritisiert an einer Stelle während der Einführung C5. Daraufhin folgt eine Rechtfertigung von C5 und eine anschließende kurze Diskussion.
2. An einer anderen Stelle greift C2 zur Tastatur. C5 übernimmt parallel die Maus.
3. C2 will was ändern, darauf scheint Unmut von C5 zu folgen. Eine kurze Diskussion folgt, in der C5 C2 darauf hinweist, anders arbeiten zu würden.
4. Überwiegend scheinen Unstimmigkeiten aufzutauchen, wenn Implementierungen bereits da sind bzw. implementiert werden sollen, die eventuell noch nicht gebraucht werden (YAGNI).
5. C2 trifft eine Entscheidung und will die Maus greifen, aber C5 greift die Maus, woraufhin C2 die Hand zurückzieht und C5 zur Tastatur greift und beginnt zu tippen.
6. Dann folgt eine erneute Diskussionsphase über eine Implementierung von C2. C2 versucht sie zu erklären, C5 sagt bei beiden Erklärungsversuchen, dass sie es nicht versteht.
7. In CA2.2 hat C2 eine Phase durchgehend das Ruder in der Hand. Das hält C5 nicht davon ab, sich einzubringen.

CA4.1

1. C7 bekommt eine Einführung von C4. Dabei kommt C4 durcheinander und C7 versucht zu helfen.

5. Methodik

2. C4 fragt C7, ob er ans Ruder will. Keine Reaktion folgt. Daraufhin fordert C4 C7 auf, einen Test zu schreiben. C7 stimmt zu.
3. C4 wirkt an einigen Stellen passiv-aktiv auf mich, was ihre Art zu sein scheint. C7 macht den Eindruck, damit gut klarzukommen.
4. C4 kritisiert an einer Stelle die gemeinsame Arbeitsweise.
5. C7 wirkt von Minute zu Minute dominanter.
6. Hätte negativer enden können, wenn C7 vieles nicht einfach so hingenommen hätte.

6 Hauptergebnisse

6.1 Einleitung

In diesem Kapitel werden zunächst die Ergebnisse aufgezeigt. Anschließend folgt die entstandene Grounded Theory mit darauf aufbauenden Lösungsvorschlägen.

6.2 Passivität bei Entscheidungsfindungen

Eine der größten Auffälligkeiten während der Analyse war, dass die aktive einen größeren Einfluss auf Entscheidungen hatte als die passive. Im Folgenden werden Erkenntnisse aus der Analyse der Videos hervorgebracht, die diese Aussage untermauern.

Im weiteren Verlauf folgen .

1. Aktiv überwiegend Entscheidungsträger

In dem Video DA4 sind sechs wichtige Entscheidungen zu beobachten. Fünf dieser Entscheidungen betreffen die Vorgehensweise und eine bezieht sich auf die Benennung eines Elements. Interessanterweise trifft D7 alle sechs Entscheidungen. Bei einer der Entscheidungen über den nächsten Schritt fällt auf, dass D7 kurz seinen Partner anblickt und erst nach dessen Zustimmung fortfährt. Bei den restlichen fünf Entscheidungen hingegen setzt er seine Entscheidungen sofort um, ohne vorher auf eine Bestätigung oder Rückmeldung seines Partners zu warten. Dieses Verhalten deutet auf eine dominierende Rolle von D7 im Entscheidungsprozess hin.

- (a) Als erstes probieren wir mal deinen Ansatz aus. (DA4, 0:30:21.0)
- (b) Das können wir mal ganz kurz versuchen. (DA4, 0:47:13.0)
- (c) Okay, ehm wir probieren das jetzt einfach mal. (DA4, 1:29:42.0)
- (d) Wir machen das mal eben. Das ist, glaube ich, eine ganz coole Sache. (DA4, 1:30:35.0)
- (e) I... data... Binding Context. Aware. (DA4, 1:30:47.0)
- (f) Nee, wir machen ich will das schon...der Ansatz ist eigentlich ganz cool. (DA4, 1:52:21.0)

2. Zurückhaltung bei Evaluation von Vorschlägen

Es gibt zwei Antworten auf einen Vorschlag: die Zustimmung oder Ablehnung, und diese werden im Folgenden näher besprochen.

6. Hauptergebnisse

Formen der Zustimmung

Bei der Analyse der Videos wurden fünf Arten der Zustimmung beobachtet.

(a) Keine Reaktion

D4: Aber die können wir ja hier einfach als viewpart casten
setzt um mit den Worten "View part casten"
D3: *Keine Reaktion*
D4: Oder?
D3: *Keine Reaktion*

Dieses Zitat befindet sich in dem Video DA2.3 und beginnt ab 0:13:20.0.

(b) Nicken

D2 schreibt einen Kommentar, anschließend folgt dieses Szenario:

D2: Einverstanden?
D8: *nickt*.

Dieses Zitat befindet sich in dem Video DA5 und beginnt ab 0:20:31.5.

(c) Lachen

D4: Machen wir erst mal die anderen.
D3: *Lachen*

Dieses Zitat befindet sich in dem Video DA2.3 und beginnt ab 0:04:28.0.

(d) Ja-Sager

D4: Aber lass erst mal angucken, wie die das bei dem anderen gemacht haben.
D3: Ja

Dieses Zitat befindet sich in dem Video DA2.1 und beginnt ab 0:19:49.0.

(e) Satz vervollständigen

D2: Mit dem Agree werden jetzt hier alle notifiziert,...
D2 & D8 (zeitgleich): Die schon mal daran beteiligt waren.
D8: Wow.
D2: Da sind wir uns einig.
D8: Synchronsprechen

Dieses Zitat befindet sich in dem Video DA5 und beginnt ab 0:05:06.4.

Vorkommnisse in den Videos

In den Videos DA2 und DA4, die Passivität abbilden, wurden hauptsächlich die ersten vier Arten der Zustimmung beobachtet. Diese Formen der Zustimmung sind eher nachteilig, da sie eine passive Beteiligung widerspiegeln. Im Gegensatz dazu zeigt sich die letzte, effektivere Form der Zustimmung in dem Video DA5.

Bewertung der Formen

Reaktionslosigkeit, bloßes Nicken, Lachen oder einfaches "Ja-Sagen" sind suboptimale Formen der Zustimmung. Sie lassen Zweifel aufkommen, ob die Zustimmung auf echtem Verständnis der Situation beruht oder nur eine oberflächliche Beteiligung darstellt. Dies wird besonders in DA4 deutlich, als D7 einen Vorschlag macht, den D6 zunächst gutheißt, D7 jedoch bald darauf zurückzieht. Es stellt sich die Frage, warum D6 zustimmt, wenn selbst der Verfasser des Vorschlags diesen nicht befürwortet.

D7: Wir können auch immer Equals, false zurückgeben. Aber das ist ja so dirty.
D6: Equals false?
D7: Ja, immer wenn der Equals gefragt wird, ist false.
D6: Ja.
D7: Eigentlich ist das nicht das gleiche.
D6: Ja.

Dieses Zitat befindet sich in dem Video DA4 und beginnt ab 0:43:35.0.

Das Vervollständigen eines Satzes hingegen erfordert aktives Mitdenken und Verstehen des Gedankengangs, was als optimal anzusehen ist. Ein positives Beispiel hierfür ist in DA5 zu sehen: D8 stimmt einer Aussage erst zu, nachdem sie sie vollständig verstanden hat und äußert Bedenken, bis sie die Argumentation als sinnvoll erachtet.

Formen der Ablehnung

Bei der Analyse der Videos wurden zwei Arten der Ablehnung beobachtet.

- (a) Ablehnung ohne Erklärung

6. Hauptergebnisse

D4: Also Toolbar Action oder sowas nehmen? Ich weiß es nicht.
D3: Ja, naja. Toolbar Action.

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:48.0.
Die Ablehnung ist in diesem Fall "Ja, naja Toolbar Action." Dies wird deutlich, wenn man das Video sieht.

(b) Ablehnung + Erklärung

D3: Abstract List Action?
D4: Nein, das hat ja nichts mehr mit List zu tun eigentlich ne?

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:35.6.

Vorkommnisse in den Videos

Vorschlägen, die abgelehnt werden, folgt meist eine Erklärung. Das war vor allem bei den Vorschlägen der passiven Personen zu beobachten und weniger bei denen der aktiven. Dieser Code kam in DA2 & DA4 insgesamt 16-mal vor, wobei nur einer davon von einer passiven Person kam. Bei den Ablehnungen handelte es sich um Lösungsvorschläge, Hinweise, Erklärungen, Namensvorschläge und Eingabevorschläge.

Bewertung der Formen

Ablehnungen mit einer Erklärung liefern einen höheren Mehrwert, da sie den Partner oder die Partnerin darauf aufmerksam machen, was mit dem Vorschlag nicht stimmt. Dadurch können Unklarheiten geklärt werden, und die Begründung fungiert somit gleichzeitig als push.

Ausnahmen

Jedoch gab es auch Ausnahmen, die aufzeigen, dass der passive Partner oder die passive Partnerin auch Vorschläge ablehnt und standhaft bleibt, wenn der eigene abgelehnt wurde. Diese beiden Ausnahmen werden in den folgenden Zitaten aufgezeigt.

1. Ausnahme: Ablehnung durch einen passiven Partner

An einer Stelle in DA2 lehnt D3 den Vorschlag von D4 ab. Es handelt sich um einen Vorschlag zur Benennung einer Klasse. Zuvor hatte D3 einen eigenen Vorschlag gemacht, der von D4 abgelehnt wurde.

D3: Abstract List Action?
D4: Nein, das hat ja nichts mehr mit List zu tun eigentlich ne?
D4: Also Toolbar Action oder sowas nehmen? Ich weiß es nicht.
D3: Ja, naja. Toolbar Action.

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:35.6.

Die Ablehnung ist in diesem Fall "Ja, naja Toolbar Action." Dies wird deutlich, wenn man das Video sieht.

Bei dem Beispiel, in dem der passive Partner den Vorschlag nicht annimmt, handelt es sich um einen einfachen Namensvorschlag. Dieser Fall zeigt, dass der passive Partner eher bereit ist, sich in scheinbar einfachen Angelegenheiten, zu behaupten, wenn er sich in dieser Hinsicht sicher fühlt.

2. Ausnahme: D6 lässt sich nicht abwimmeln

In diesem Fall haben wir eine Konstellation, in der D6 einen Vorschlag einbringt, der von D7 zunächst abgelehnt wird. Im Gegensatz zur üblichen Entwicklung hält D6 jedoch an seinem Vorschlag fest und setzt ihn fort, bis D7 schließlich ein Verständnis dafür entwickelt und den Vorschlag akzeptiert. Dies verdeutlicht ein Szenario, in dem der zurückhaltende Partner durchaus einen Einfluss auf die Sitzung ausüben kann, indem er sich nicht sofort von dem aktiven Partner abweisen lässt. Obwohl der ursprüngliche Vorschlag am Ende nicht fruchtet, scheint er D6 zu ermutigen, da dieser daraufhin erneut einen Beitrag leistet, der dann wieder von D7 abgelehnt wird.

D6: Ansonsten können wir in der Exception von den Dingens, diese Break-point Exception, vielleicht Binding Exception oder so was like this. Oder Exception generell mal.
D7: Die haben wa hier ja alle.
D6: Vielleicht irgendwas, was intern quasi catcht.
D7: Achso du meinst so ne.
D6: Ja, kommt auch mal vor.
D7: Ja okay.. genau
D7: Du meinst so eine runtime exception irgendwas.
D6: Yeah mmh. Like this

Dieses Zitat ist aus dem Video DA4 und beginnt ab 1:01:45.6.

6.3 Passivität bei dem Wissenstransfer

Der Wissenstransfer ist ein Prozedere, bei dem Wissen vermittelt oder geschaffen wird. Eine kurze Erinnerung folgt.

Es gibt drei Arten des Wissenstransfers:

6. Hauptergebnisse

- Push: Das Teilen vorhandenen Wissens
- Pull: Aktives Einholen von Wissen durch Fragen
- Production: Hier müssen beide Personen fehlendes Wissen gemeinsam oder individuell erarbeiten.
In den analysierten Videos nutzte die aktive Person diese Formen des Wissenstransfers deutlich häufiger als die passive Person.

Im Folgenden werden Erkenntnisse aus der Analyse der Videos erläutert.

(a) **Zustimmung nach Wissenstransfer:**

Ein positives Beispiel von Wissenstransfer geschieht in DA5. D2 macht einen Vorschlag, dem D8 zunächst nicht zustimmt, da es eine Wissenslücke gibt, durch die sie D2 nicht versteht. Sie bittet um Erklärung und bekommt diese. Anschließend stimmt sie D2 zu.

(b) **Aktiv nutzt Wissenstransfer zur Schließung des S-Gaps:**

Ein Szenario, in dem ein Wissenstransfer erforderlich ist, ist das Schließen der S- und G-Gaps, also Wissenslücken.

In DA2 zeigt sich, dass D3 mit der Codebasis vertraut ist, während D4 bessere Programmierkenntnisse in Java hat. D3 stellt viele Fragen, um seine Wissenslücke (S-Gap) zu schließen, wohingegen D4 keine vergleichbare Initiative zeigt.

(c) **Push von Aktiv:**

An zwei Stellen in DA2 bietet D4 D2 Erklärungen an. Dies wirkt wie ein Versuch seinen Partner ins Boot zu holen.

D4: Kennst du das denn mit dem OSGI?
D4: Classloading und sowas?
D3: Was? Claus? Was?
D4: Class-Loading. Claus-Loading
D3: Nicht wirklich.
D4: Also ehm, soll ich kurz sagen, oder?
D3: Naja, klar.
D4: Also wir haben ja jetzt das Problem gehabt. Wir wollten ja das erst hier in das...

Dieses Zitat befindet sich in dem Video DA2.3 und beginnt ab 0:16:12.0.

(d) **D6 verkleinert Gap allein:**

Im weiteren Verlauf von DA4 kommen D7 und D6 an einen Punkt, an dem sie nicht weiterkommen. Während D7 eine Raucherpause einlegt, ergreift D6 die Initiative und nutzt die Maus, um eine bestimmte Stelle zu untersuchen. Diese Eigeninitiative führt zu einer wichtigen Erkenntnis, die er anschließend mit D7 teilt.

Dies zeigt, dass D6 bedeutende Beiträge leisten kann, wenn er ausreichend

informiert ist. Es bleibt jedoch unklar, warum er nicht aktiv nach fehlenden Informationen fragt, sondern abwartet, bis er allein ist.

- (e) **Problematiken durch Nichtkommunikation von Wissenslücken:**
 In DA4 kann nicht klar bestimmt werden, ob D6 oder D7 ein S-Gap oder G-Gap haben. D7 übernimmt die Einführung, was auf ein S-Gap bei D6 hindeuten könnte. Es bleibt jedoch ungewiss, ob das G- oder S-Gap das größere Hindernis für D6's aktive Beteiligung ist. Dies führt zu der Erkenntnis, dass selbst Forscher:innen, die das Videomaterial möglicherweise hundertfach analysieren, Schwierigkeiten haben, genau zu bestimmen, welches Gap D6 daran hindert, sich aktiver einzubringen. Diese Analyse wurde über zwölf Mal durchgeführt, ohne eindeutige Schlüsse ziehen zu können. Dies unterstreicht die Herausforderung, Wissenslücken in Echtzeit zu identifizieren.
- (f) **Ja-Sager:** Folgt auf eine Erklärung ein einfaches ja, sagt das nichts darüber aus, ob die Person gegenüber die Erklärung tatsächlich verstanden hat oder nur diesen Eindruck vermitteln möchte. Es könnte auch sein, dass das "Ja" sagen einprogrammiert ist und der Partner oder die Partnerin dem oder der anderen gar nicht mehr wirklich folgt. Dies würde dazu führen, dass kein Wissenstransfer stattfindet und die Wissenslücke weiterhin bestehen bleibt, ohne, dass der Partner bzw. die Partnerin dies versteht. Alleine in DA4 taucht der Code "Ja-Sager_PartnerA" 66-mal auf.

Passivität bei Experten

Einen Spezialfall des Wissenstransfers in den Videos stellt das Fragen eines Experten oder einer Expertin dar. An einigen Stellen der Videos, wenn die Programmierer:innen nicht weiter kamen, holten sie einen Experten oder eine Expertin zur Hilfe. Eine Person, die bereits mit der Codebasis vertraut war und aushelfen sollte.

- (a) **D4 stellt 15 Fragen:**
 Die erste Sache, die in dieser Phase des Videos auffiel, war, wie viele Fragen D4 den Experten stellte. Es schien so, als wolle er ihm jede mögliche Information entziehen. Dies könnte wiederum auf die im vorherigen Unterkapitel erläuterte S-Gap Thematik zurückgeführt werden. D4 stellt zwei Experten, die im Laufe der Sitzung dazustoßen, 15 Fragen. D3 im Gegensatz dazu vier.
- (b) **Interaktion zwischen D4 & D7:**
 D4 dominiert die Gespräche mit dem Experten D7 durch seine vielen Fragen, wodurch D3 etwas ins Abseits gerät. Die Bildschirmposition verstärkte diesen Effekt, indem D7, der zwischen D3 & D4 steht, zu D4 geneigt ist und beim Sprechen auf ihn ausgerichtet bleibt. Den Höhepunkt findet das Ganze am Ende von DA2.1, als D7 dabei ist, den nächsten Schritt zu erklären und dabei von D4 unterbrochen wird, der seinen Satz vollendet. D7 stimmt zu und entfernt sich anschließend, da seine Aufgabe erfüllt zu sein scheint,

während D3 nur zu lachen beginnt.

D3 wirkt in diesen Interaktionen weitgehend abwesend.

(c) **Folge: D3 steigt aus:**

Als Konsequenz dieser beschriebenen Interaktion schiebt D3 D4 die Tastatur und Maus zu und kommentiert dies mit den Worten "Ick sehe, du bist mehr mitgekommen als ich". Dies kann auf eine Form der Entmutigung oder das Gefühl der Überforderung bei D3 hindeuten, hervorgerufen durch die dominante Rolle von D4 in der Interaktion mit den Experten.

(d) **Der Experte (D4) als aktiver Teilnehmer anstatt passiver Berater:**

Dieses Szenario spielt sich in DA4 ab. Hier bietet D4, der neben D6 & D7 zu sitzen scheint, seine Hilfe aktiv an und beteiligt sich immer wieder am Gespräch. Es gibt Phasen in der Sitzung, in denen D4 mehr beiträgt als D6, obwohl er nicht als Experte hinzugezogen wird. In diesen Phasen beginnt D7, seine Ideen und folgenden Schritte in Richtung D4 zu kommunizieren, bevor er sie umsetzt.

6.4 Aktivität bei Ruderwechsel

Die Steuerung von Tastatur und Maus, metaphorisch als "Rudern" bezeichnet, spielt eine zentrale Rolle in der Dynamik von "Driver Observer"-Szenarien. Die reine Konzentration auf das Ruder kann von anderen wichtigen Geschehnissen ablenken, was zu einem Übersehen wesentlicher Aspekte führen könnte. Trotz dieser potenziellen Gefahr ist die Steuerung von Tastatur und Maus entscheidend für die Aktivität und Teilnahme im Arbeitsprozess. In den analysierten Videos gibt es nur eine Tastatur und eine Maus, was bedeutet, dass der Arbeitsfluss ins Stocken geraten würde, wenn die Person am Ruder inaktiv bleibt. Im Gegensatz zu den oben erläuterten Passivitäten, ist es nicht möglich am Ruder passiv zu sein, ohne, dass der Partner oder die Partnerin dies mitbekommt und der Prozess ins Stocken kommt.

Am Ruder: 2 Möglichkeiten

Wenn man das Ruder übernimmt und auf Schwierigkeiten stößt, gibt es zwei grundlegende Verhaltensweisen:

(a) **Kommunikation von Unwissen:**

Die Person am Ruder gibt offen zu, dass sie überfordert ist und nicht weiterweiß.

(b) **Produktion von Unsinn:**

Anstatt ihre Unsicherheit zuzugeben, beginnt die Person, sinnlose Aktionen durchzuführen. Dies wird allerdings oft als demütigender empfunden, weshalb die meisten Menschen vermutlich die erste Option bevorzugen.

Im Folgenden werden Erkenntnisse aus der Analyse der Videos vorgestellt.

(a) **Aktives Fragen am Ruder**

Das folgende Szenario stellt dar, inwiefern es hilfreich ist, das Ruder zu

haben. Ein gutes Beispiel für effektive Ruderführung zeigt sich in DA5, als D8 das Ruder übernimmt und sofort Fragen stellt, um die Aufgabe zu verstehen und anschließend umsetzen zu können. Durch das Stellen von Fragen kann D8 den Prozess besser verstehen und direkt beeinflussen, da sie die Aufgabe selbst ausführt.

D2: Jetzt darfst du auch nochmal tippen. Jetzt habe ich dich, glaube ich, auf den Stand gebracht. Auf den Stand der Dinge.
 D8: Wir haben da einen Subject.
 D2: Ich kann das erst entknoten.
 D8: For.. So ich gerne einmal durch und wenn ich einen finde, was mache ich damit? D2: Eigentlich müsste er ja nur genau einen gefunden haben.
 D8: Achso
 D2: Wenn er brav war, dürfte er nur eine Aktivität haben.
 D8: Dann sage ich Assert.
 D2: Genau
 D8: Also nur danach prüfen?
 D2: Eigentlich ja. Nur, dass es.. Eigentlich würd sogar reichen, dass es genau eine Aktivität gibt.
 D8: Achso kann ich hier sehen bei Introspection. Also was da drin steht.
 D2: Mh ja.
 D8: Ich bin immer noch bei meinen alten sysouts.
 D2: Wir können das debuggen.

Dieses Zitat stammt aus dem Video DA5 und beginnt ab 0:22:35.2.

(b) **Aktiv überwiegend am Ruder**

In DA2 und DA4 beobachtet man, dass der aktivere Partner länger das Ruder behält. In DA2 übernimmt der aktive Partner das Ruder nach der Experteninteraktion und behält es bis zum Refactoring. In DA4 behält der aktive Partner das Ruder durchgehend, abgesehen von Momenten, in denen der passive Partner allein ist. In DA5 sieht man einen häufigeren Wechsel.

(c) **Lehnt Ruder mehrfach ab**

Eine interessante Erkenntnis findet sich in DA2.3. D3 lehnt das Ruder, das ihm von D4 angeboten wird, zweimal ab. Dies begründet er mit mangelndem Vertrauen in seine Fähigkeiten. Andere Videos zeigen, dass Personen tendenziell aktiver mitarbeiten, sobald sie das Ruder übernehmen. D3 entscheidet sich jedoch dagegen, diese Chance zu nutzen, um eine aktivere Rolle einzunehmen. D4's Angebot des Ruders entspricht einer guten Handlungsweise, um die Zusammenarbeit zu verbessern.

Situation1:

D4: Eh willst du dann nochmal machen, oder?
D3: Ick globe du bist mir involviert in dit Ganze. Ich bin da... für mich ist sie schon oberste Wissenskante.

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:09:46.1.

Situation2:

D4: Also sag, wenn du jetzt machen willst oder sowas.
D3: Ne mach mal erstmal weiter ich sage wenn ick wieder voll drinne bin denn schrei ick schon.

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:21:02.0.

(d) **Bei Refactoring: Nimmt Ruder an**

Nachdem D3 das Angebot, das Ruder zu übernehmen, zunächst zweimal abgelehnt hat, stimmt er zu, als es um eine Refactoring-Aufgabe ging. Dieses Verhalten bestätigt die Annahme, dass D3 wahrscheinlich mehr Initiativen zeigen würde, wenn er ein besseres Verständnis der Gesamtsituation hätte.

(e) **Schwieriges Refactoring gibt er ab**

Während eines Refactorings, bei dem D3 Klassen aufräumt, stößt er auf eine komplexe Klasse und entscheidet sich, diese Aufgabe an seinen Partner zu übergeben. Dies deutet darauf hin, dass D3 das Refactoring dieser spezifischen Klasse als schwieriger empfindet als die bisherigen Aufgaben. Es wäre förderlich für seinen Lernprozess, nach Erklärungen zu fragen, statt die Aufgabe abzugeben.

D3: Okay, die lasse ich erstmal für dich.

Dieses Zitat findet man in dem Video DA2.4. Es beginnt ab 0:18:31.0.

(f) **Umsetzung mit 4 Möglichkeiten**

Implementierungen, während man die Tastatur und Maus bedient, können nach den folgenden vier Möglichkeiten umgesetzt werden. In den folgenden Fällen hat die aktivere Person das Ruder.

i. Erklärung vorher:

Dies bietet die beste Gelegenheit für die passive Person, aktiv mitzudenken und sich durch eigene Vorschläge oder Einwände einzubringen, noch bevor der Partner oder die Partnerin mit der Umsetzung beginnt. Dies sorgt für eine proaktive Zusammenarbeit.

- ii. Erklärung währenddessen:
Hierbei kann die passivere Person den Prozess aktiv verfolgen und mitdenken.
- iii. Erklärung danach:
Diese Methode dient dem reinen Wissenstransfer.
- iv. Keine Erklärung:
Ohne Erklärung bleibt die passivere Person nur involviert, wenn er oder sie den Prozess von sich aus versteht oder aktiv nachfragt.

6.5 Gründe für Passivität

(a) Verhalten von Aktiv

Die Passivität kann teilweise durch das Verhalten des aktiven Partners oder der aktiven Partnerin bedingt sein. Häufige Ablehnungen von Vorschlägen können das Selbstvertrauen schwächen und zu einer zurückhaltenden Beteiligung führen. Besonders die Interaktion zwischen der aktiven Person und einem Experten oder einer Expertin könnte den passiven Partner oder die passive Partnerin sich übergangen fühlen lassen, was wiederum dazu führen kann, dass er oder sie Wissenslücken nicht eingesteht und den Anschluss verliert. Ein weiterer Punkt ist, dass Aktiv in DA4 Passiv zu keinem Zeitpunkt die Maus und Tastatur anbot. Das Ruder hätte dafür sorgen können, dass Passiv sich mehr einbringt.

(b) Ego

Der nach der Analyse am plausibelsten erscheinende Grund für das Verbergen von Wissenslücken ist ein Schutzmechanismus des Egos. Das Stellen von Fragen oder Formulieren von Vorschlägen, die abgelehnt werden, können zu einer Beschädigung des Egos führen. Dies wird vermieden, indem Wissenslücken nicht offenbart werden. Wissenslücken offen zu kommunizieren, bedeutet zuzugeben, dass man etwas nicht weiß, das der Partner oder die Partnerin wissen könnte. Man gesteht somit, in einem Bereich inkompetenter zu sein.

Eine Beobachtung aus den Videos, die für diese Behauptung spricht, ist beispielsweise der Moment aus DA4, in dem D6 versucht seine Wissenslücke zu verringern als D7 den Arbeitsplatz verlässt. Dies weist daraufhin, dass D6 durchaus Interesse daran hat sich an dem Prozess zu beteiligen, jedoch nicht nach den Informationen, die ihm hierfür fehlen, fragt.

Eine weitere Beobachtung ist, dass sich gegen Ende von DA4 immer mehr Leute Richtung Feierabend verabschiedet haben. Je leerer der Raum wurde, desto mehr schien sich D6 einzubringen. Er lässt sich gegen Ende überraschenderweise auf eine Diskussion mit D7 ein. Dies könnte ebenfalls auf die Egothematik hinweisen. D6 traut sich scheinbar mehr Beiträge zu leisten, wenn weniger Menschen um ihn herum sind.

Eine andere Stelle ist eine Phase, in der D7 nicht weiterweiß. In dieser Phase folgen acht von insgesamt elf Vorschlägen von D6. Dies könnte damit

6. Hauptergebnisse

zusammenhängen, dass D6 sich nicht inkompetenter fühlt als D7.

In D2 soll D3 D4 in die Codebasis einführen und verfällt anschließend in eine passive Haltung. Dies könnte ebenfalls für einen Schutzmechanismus des Egos sprechen. Die Tatsache, dass D3 zu Beginn der "Experte" ist, könnte ihn daran hindern, sein G-Gap D4 gegenüber zu schließen. Schließlich müsse er der Person Fragen stellen, die er einarbeiten soll.

Ein positives Beispiel bietet wiederum DA5. D2 ermutigt D8, zu Beginn der Session, aktiv Fragen zu stellen und sagt ihr, dass es vollkommen normal sei, da sie noch nicht an dem System gearbeitet habe. So ein Verhalten bietet eine Art sicheren Hafen und nimmt eventuell sogar die Angst vor dem Fragen.

D2: Frag ruhig, wenn du... du bist ja jetzt dazu gekommen. Ich hatte es vorher mit Simone gemacht. Ein Teil hat Simone auch alleine gemacht. Deswegen muss ich mich da auch durchwühlen. Aber frag ruhig.

Dieses Zitat ist in DA5 vorzufinden und beginnt ab: 0:03:51.7

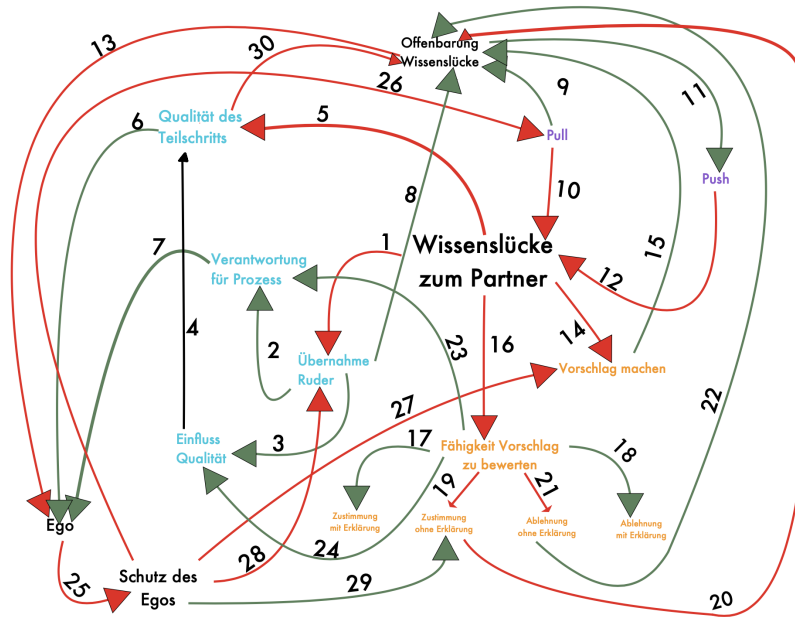


Abbildung 1: Grounded Theory

6.6 Die Grounded Theory

Die gesamte Grounded Theory ist aus Sicht der passiven Person zu betrachten. Die roten Pfeile bedeuten, dass sich ein Konzept negativ auf ein anderes auswirkt. Die grünen Pfeile stellen einen verstärkenden Effekt dar und führen zu einer Erhöhung des betrachteten Phänomens.

Die Grounded Theory baut hierbei auf den bereits in der Analyse verwendeten Videos DA2, DA4 und DA5 auf. Im Hauptfokus der GT steht die Wissenslücke zum Partner bzw. der Partnerin. Diese wirkt sich negativ auf mehrere Aspekte aus.

Zum einen wirkt sie sich negativ auf die Übernahme des Ruders aus. Wenn eine Person das Gefühl hat, über weniger Wissen als sein Partner oder seine Partnerin zu verfügen, sieht er oder sie eher davon ab, das Ruder zu übernehmen.(1)

Situation1:

D4: Eh willst du dann nochmal machen, oder?
 D3: Ick globe du bist mir involviert in dit Ganze. Ich bin da... für mich ist sie schon oberste Wissenskante.

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:09:46.1.

Situation2:

6. Hauptergebnisse

D4: Also sag, wenn du jetzt machen willst oder sowas.
D3: Ne mach mal erstmal weiter ich sage wenn ick wieder voll drinne bin denn schrei ick schon.

Dieses Zitat ist aus dem Video DA2.3 und beginnt ab 0:21:02.0.

Ein Ruder zu übernehmen, bringt mehrere Folgen mit sich. Zum einen übernimmt man, wenn auch nur zum Teil, eine stärkere Verantwortung für das Produkt bzw. den Prozess.(2) In dem Moment, in dem eine Person an der Tastatur sitzt und aktiv Code schreiben oder verändern muss, greift sie in den Prozess ein und verändert das Produkt. Somit entsteht ebenso ein direkter Einfluss auf die Qualität des Produktes.(3) Traut man sich das Ruder zu übernehmen und hat kaum eine Wissenslücke zu dem Partner oder zu der Partnerin, entsteht mit einer gewissen Wahrscheinlichkeit ein qualitativ hochwertiges Produkt. Dies kann ein Code, ein Prozessschritt oder aber auch ein Designvorschlag sein. (5).

Die hohe Qualität und die Verantwortungsübernahme im Allgemeinen stärken das eigene Ego. Man hat dies selbst erschaffen und kann stolz darauf sein.(6,7) Wenn die Wissenslücke zum Partner oder zur Partnerin jedoch hoch ist und man das Ruder übernimmt, hat man zwei Möglichkeiten. Entweder man produziert qualitativ schlechten Code, da ein erforderliches Wissen fehlt, oder man offenbart die eigene Wissenslücke. (5,8) Hierfür gibt es in DA5 ein Beispiel, bei dem D8 das Ruder übernommen hat und die eigene Wissenslücke offenbart hat, indem sie nach den nächsten Schritten gefragt hat.

D2: Jetzt darfst du auch nochmal tippen. Jetzt habe ich dich, glaube ich, auf den Stand gebracht. Auf den Stand der Dinge.
D8: Wir haben da einen Subject.
D2: Ich kann das erst entknoten.
D8: For.. So ich gerne einmal durch und wenn ich einen finde, was mache ich damit? D2: Eigentlich müsste er ja nur genau einen gefunden haben.
D8: Achso
D2: Wenn er brav war, dürfte er nur eine Aktivität haben.
D8: Dann sage ich Assert.
D2: Genau
D8: Also nur danach prüfen?
D2: Eigentlich ja. Nur, dass es.. Eigentlich würd sogar reichen, dass es genau eine Aktivität gibt.
D8: Achso kann ich hier sehen bei Introspection. Also was da drin steht.
D2: Mh ja.
D8: Ich bin immer noch bei meinen alten sysouts.
D2: Wir können das debuggen.

Dieses Zitat stammt aus dem Video DA5 und beginnt ab 0:22:35.2.

Dieses Beispiel zeigt zwei wichtige Punkte. Zum einen kann die Offenbarung

der Wissenslücke durch einen pull, also das Stellen konkreter Fragen, geschehen.(9)

Zum anderen wirkt ein solches Pull der Wissenslücke entgegen, da in dem geschilderten Szenario D8 durch die Antworten von D2 das fehlende Verständnis bekommt, um die Aufgabe zu bewältigen. (10) Das neue Wissen hilft ihr nicht nur dabei, die Aufgabe zu bewältigen, sondern auch qualitativ hochwertiger Code zu produzieren. (8)

Das Offenbaren einer Wissenslücke führt also auch zum Push.(11) Der Partner oder die Partnerin versteht, an welchen Punkten es einem an Wissen fehlt, und kann die Wissenslücke durch das Teilen der wichtigen Informationen und Erklärungen verringern. (12)

Das Offenbaren einer solchen Wissenslücke wirkt sich jedoch negativ auf das eigene Ego aus. Die Person muss zugeben, an einer Stelle fehlendes Wissen zu haben, welches der Partner oder die Partnerin gegenüber jedoch besitzt.(13) Dies erklärt die oben bereits genannte Theorie, dass ein Partner oder eine Partnerin mit einer Wissenslücke somit eher abgeneigt ist, das Ruder zu übernehmen.(1) Auch bei der Entscheidungsfindung kann eine Wissenslücke eine Art Behinderung sein. Zum einen schreckt die Person mit einer gewissen Wissenslücke davor zurück eigene Vorschläge einzubringen,(14) da bei einer Ablehnung des Vorschlags durch den Partner oder die Partnerin die eigene Wissenslücke offenbart werden könnte. (15) Dies kann passieren, sobald ein Vorschlag abgelehnt wird, der nicht zur aktuellen Codebasis passt oder fehlerhaft ist. Hier ein kleines Beispiel:

D6: Aber ich hatte den Eindruck, dass man das trotzdem in Equals sagen kann, okay, keine Ahnung. Ignoriere das und guck nach dem Inhalt oder wie auch immer.

D7: Ja, aber auf was willst du gucken? Da ist alles gleich. Der Inhalt ist gleich. Selbst die Speicheradressen sind gleich. Das sind genau die gleichen Objekte.

D6: Aber die Values sind ja nicht die gleichen. D7: Auch die Inhalte sind gleich. Auch die Values sind gleich. Das ist das Problem.

D6: Achso das heißt die auch?

D7: Wir arbeiten ja auf dem... Auf den Objekten, in den Objekten, in den Listen. Und verändern da was.

D6: Ja

D7: Das heißt, nach außen verändert sich eigentlich gar nichts.

Dieses Zitat ist aus dem Video DA4 und beginnt ab 0:50:19.0.

Aber auch die Fähigkeit einen Vorschlag des Partners zu bewerten, leiden unter der Wissenslücke zum Partner.(16) Es gibt drei Arten, auf einen Vorschlag zu reagieren. Keine Reaktion, eine Zustimmung oder eine Ablehnung. Die Zustimmung und Ablehnung können mit Erklärung erfolgen und ohne. Hat man eine kleine bis keine Wissenslücke zum Partner oder zur Partnerin, ist man durchaus in der Lage, einen Vorschlag zu bewerten, und tut dies mit einem gewissen

6. Hauptergebnisse

Verständnis. (17,18)

Hat man jedoch eine Wissenslücke zum Partner oder zur Partnerin und versteht somit den Vorschlag oder die Materie, in der der Vorschlag getätigt wird, nicht, neigt man eher dazu zuzustimmen, da dies auch ohne Erklärung funktioniert.(19) Hier passt das oben genannte "Ja Sager" Phänomen, bei dem D 66-mal mit einem "Ja" auf einen Vorschlag reagiert hat. Eine solche Zustimmung schützt den Unwissenden vor der Offenbarung seiner Wissenslücken, da der Partner oder die Partnerin nicht erkennen kann, ob man den Vorschlag tatsächlich verstanden hat oder ob man der Beteiligung wegen zustimmt.(20)

Eine Ablehnung mit einer Wissenslücke und somit ohne eine Erklärung ist auch möglich. (21) Sie hingegen bringt das Risiko einer Offenlegung der Wissenslücke mit. Bei einer Ablehnung wird, wie bereits erwähnt, eine Begründung erwartet. Kann man die Ablehnung nicht erklären oder gibt eine unsinnige Erklärung kann dies dazu führen, dass der Partner oder die Partnerin erkennt, dass einem Wissen fehlt. (22) Dies ist jedoch nicht immer der Fall. In DA2.2 gibt es ein Szenario, in dem D3 einen Vorschlag ablehnt, ohne eine Begründung zu liefern.

D3: Abstract List Action?

D4: Nein, das hat ja nichts mehr mit List zu tun eigentlich ne?

D4: Also Toolbar Action oder sowas nehmen? Ich weiß es nicht.

D3: Ja, naja. Toolbar Action.

Dieses Zitat ist aus dem Video DA2.2 und beginnt ab 0:02:35.6.

Die Ablehnung ist in diesem Fall "Ja, naja Toolbar Action." Dies wird deutlich, wenn man das Video sieht. Diese Zusammenstellung aus 'Ablehnung' und 'keine Erklärung' wird jedoch eher umgangen, um eine Wissenslücke nicht zu offenbaren. Die Fähigkeit einen Vorschlag zu bewerten, bringt auch eine gewisse Verantwortung für das Produkt mit sich, da man selbst direkt auf den Entscheidungsprozess einwirkt und entscheidet, ob Vorschläge umgesetzt werden und auf welche Art. (23)

Zusätzlich hat jede Entscheidung dadurch auch einen Einfluss auf die Qualität des Endproduktes. (24) Zu guter Letzt einen Blick auf die negative Folge der Offenbarung der Wissenslücke. Das Konzept wirkt sich wie bereits erklärt negativ auf das Ego aus. Taucht dies häufiger auf, kann ein Bedürfnis entstehen, das eigene Ego zu schützen, was weitreichende Folgen mit sich bringen kann, da alles vermieden wird, was das Ego weiter beschädigen würde.(25) Somit kann dies dazu führen, dass die Person mit der Wissenslücke keine weiteren Fragen stellt, um nicht noch mehr in den Augen des Partners, der Partnerin oder der Leute in der Umgebung zu sinken. (26) Aus demselben Grund kann die besagte Person aufhören Vorschläge zu machen (27) und sogar das Ruder zu übernehmen. (28) Einen verstärkenden Effekt hat hingegen dieser Schutzmechanismus auf die Zustimmung ohne Erklärung. Die Gründe hierfür wurden bereits erläutert. Bringt man einen qualitativen Teilschritt hervor, wirkt dies der Offenbarung einer Wissenslücke entgegen.

Wesentliche Einsichten aus der GT:

(a) **Ruder zwingt aktiv zu werden:**

Wenn man das Ruder übernimmt, ist man gezwungen, aktiv Ergebnisse zu erzielen, sei es beim Programmieren oder bei Prozessentscheidungen. In solch einer Situation ist man gezwungen, entweder die Wissenslücken zu kommunizieren oder sie zu verbergen. Letzteres kann jedoch dazu führen, dass bei dem Versuch, das fehlende Wissen zu kaschieren, unzureichende Ergebnisse erzielt werden und die Wissenslücke letztendlich enthüllt wird. Man produziert also ein unzureichendes Ergebnis und offenbart die eigene Wissenslücke, was eine doppelte Belastung für das Ego darstellt. Im Gegensatz dazu führt das offene Eingestehen von Wissenslücken nur zu einer einmaligen Beeinträchtigung des Egos.

(b) **Ruderablehnung führt zu einer verpassten Chance:**

Wenn man eine Wissenslücke hat und sich entscheidet, das Ruder abzulehnen, verpasst man die Möglichkeit durch praktische Erfahrung und die Zusammenarbeit mit besser informierten Partnern oder Partnerinnen neues Wissen zu erlangen.

(c) **Ablehnung der eigenen Vorschläge führt zum Rückzug:**

Wiederholte Zurückweisungen eigener Ideen durch den Partner oder die Partnerin können zu einem Infragestellen der eigenen Fachkompetenz führen. Personen mit geringerem Selbstwertgefühl neigen dann dazu, sich zurückhaltender zu verhalten, um ihre Unsicherheiten nicht zu offenbaren.

(d) **Zustimmung ohne Erklärung gefährdet Qualitätssicherung:**

Eine Zustimmung ohne Erklärung oder zusätzlichen Input kann bedeuten, dass die Person nicht wirklich versteht, was vorgeht, und lediglich den Anschein von Verständnis erwecken möchte. Dies führt dazu, dass ein Einfluss auf Entscheidungen weg fällt und die Qualitätssicherung verloren geht.

(e) **Verringerung von Wissenslücken durch deren Offenbarung:**

Das Offenlegen von Wissenslücken informiert den Partner oder die Partnerin über fehlendes Wissen. Eine darauf folgende Erklärung kann dazu beitragen, diese Wissenslücke zu verringern.

(f) **Große Wissenslücken machen Partner zum Entscheidungsträger:**

Ist die Wissenslücke so groß, dass man Entscheidungen ohne Verständnis trifft oder nicht mehr reagiert, macht dies den Partner oder die Partnerin zum Entscheidungsträger, da man vom Mitspracherecht nicht Gebrauch machen kann.

6.7 Handlungsempfehlungen

(a) **Sicherer Hafen**

Die Wissenslücke zum Partner bzw. zur Partnerin hat sich als deutlichster Grund für die Passivität von Personen herauskristallisiert. Existiert diese Wissenslücke ist es umso wichtiger, sie zu verringern. Das Kaschieren einer Wissenslücke zum Schutz des Egos hat zur Folge, dass kein Pull stattfindet. Was zu einer größeren Wissenslücke führen kann. Eine größere Wissenslücke verstärkt den negativen Effekt auf das Ego, wenn sie offenbart wird,

6. Hauptergebnisse

und verhindert dadurch noch mehr das Einholen von Informationen. Um dem entgegenzuwirken, muss ein sicherer Hafen geschaffen werden, damit sich die Personen mit der Wissenslücke trauen, Fragen zu stellen. Die aktiveren Teilnehmer:innen könnten hierfür die passiveren dazu ermutigen, Fragen zu stellen. Wird eine Frage gestellt, ist es dann umso wichtiger diese ohne Überheblichkeit oder Frustration zu beantworten, um das Ego des Gegenübers nicht zu beschädigen. Die Offenbarung einer Wissenslücke oder ein abgelehnter Vorschlag sollten das Ego nicht beeinflussen. Das Schaffen des sicheren Hafens soll auch dazu beitragen, dass Personen sich trauen, mehr Ideen einzubringen. Erfolgt eine gute Idee der passiven Seite, sollte dies als positiv hervorgehoben werden.



Abbildung 2: Schaffen eines sicheren Hafens

(b) Sinn der Paarprogrammierung

Bei der Paarprogrammierung liegt das Paar im Hauptfokus. Ergebnisse sollen gemeinsam erzielt werden. Sollte eine Person bemerken, dass sie überwiegend den Beitrag leistet, wäre das der Zeitpunkt, die andere Person in das Geschehen mit einzubeziehen. Hierfür kann sie der passiveren Person beispielsweise das Ruder übergeben. Dadurch wird der passive Teilnehmer bzw. die passive Teilnehmerin angeregt, aktiver zu werden. Um keine schwachsinnigen Ergebnisse zu liefern, wird sie die Wissenslücke im besten Fall offenbaren, da dies das kleine Übel zu sein scheint, im Gegensatz dazu, sich mit schlechten Ergebnissen zu blamieren. Sollte die Person das Ruder ablehnen, muss man sich darüber bewusst sein, dass auch dies auf eine Wissenslücke hindeuten könnte. Man sollte klären, wieso es zu dieser Ablehnung kam.

Eine weitere Möglichkeit besteht darin, die passivere Person aktiv in Entscheidungsprozesse einzubeziehen, anstatt Entscheidungen überwiegend allein zu treffen oder ein "Ja" als Antwort zu akzeptieren.

7 Diskussion

7.1 Vergleich Ergebnisse mit bestehender Literatur/Arbeiten

7.1.1 Disengagement in pair programming: Does it matter?

Dieses Paper habe ich zu Beginn meiner Arbeit gelesen. Meine Fragestellung beschäftigte sich zu dem damaligen Zeitpunkt noch damit, wie Entscheidungen getroffen werden, sodass es keinen direkten Mehrwert zu liefern schien. Im Laufe meiner Analyse bewegte sich mein Thema in die Richtung der Passivität. Total in der Analyse vertieft, verlor ich die Existenz dieses Papers aus den Augen. Mein Gehirn schien die Informationen nicht gespeichert zu haben, da die Informationen, wie oben bereits beschrieben, keinen wirklichen Nutzen für mich lieferten.

Bei dem Aufschreiben dieser Arbeiten beschloss ich, alle Paper erneut durchzusehen. Das, was ich in diesem Paper las, ähnelte meinen Erkenntnissen. Die Ergebnisse des Papers überschneiden sich teilweise mit meinen Ergebnissen und der Datensatz ist derselbe.

Im folgenden diskutiere ich die Gemeinsamkeiten und Unterschiede zwischen dem Paper "Disengagement in pair programming: Does" it matter?" und meiner Masterarbeit hervor. Dabei gehe ich auf die Punkte Daten- und Datenerhebung, Methodik und Ergebnisse ein.

1. Daten und Datenerhebung

- Unterschiede

Die Autorinnen in dem Paper hatten Zugang zu Interviews, die nach der jeweiligen Session stattfanden, und ziehen mehrere Schlüsse aus den Aussagen, die in diesen Interviews entstanden sind. Ich hatte zu keinem Zeitpunkt Zugriff auf diese Interviews und habe meine Analyse einzig und allein auf die Dialoge und Verhaltensmuster in den Videos gestützt. In Interviews können Menschen dazu neigen, Aussagen zu tätigen, die sie in ein besseres Licht führen. Dies kann meist aber nicht überprüft werden, sodass man gezwungen ist, dem Glauben zu schenken. Bei der Videoanalyse war das anders. Die Personen sind an ihrem Arbeitsplatz, in ihrer gewohnten Umgebung und werden über einen längeren Zeitraum aufgenommen. In dem Buch "The Base Layer" wird erwähnt, dass die Paare keine Anzeichen dafür zeigen, dass sie sich bei der Arbeit akut über die Aufnahmen bewusst sind [8].

Das oben erwähnte Zitat ist das einzige, das in dem Paper und in meiner Arbeit auftaucht. Wir scheinen also nicht dieselben Stellen analysiert zu haben.

- Gemeinsamkeiten

Die Gemeinsamkeiten sind der Datensatz, der verwendet wurde. Frau Plonka selbst hat einen Großteil Daten erhoben, die ich in meiner Analyse verwenden durfte.

Das Paper enthält ein Zitat, welches sehr ähnlich zu meinem klingt. Es ist

7. Diskussion

auf Englisch übersetzt, aber die Übersetzung legt nahe, dass es sich um dieselbe Stelle handelt.

Paper:

Dave and Mark are working together on a task. Dave is driving and explaining to Mark a concept that is new to him. Mark does not reply or react to Dave's explanation. Dave keeps explaining and driving. When Dave finishes his explanation he keeps driving in silence for a little while, then turns around to Mark and asks: "Do you want to drive?" Mark is shaking his head: "I'm not sure what you're doing at the moment. I'm a bit lost here. So you continue driving and I let you know when I know again what's going on."

Masterarbeit:

D4: Also sag, wenn du jetzt machen willst oder sowas.
D3: Ne mach mal erstmal weiter ich sage wenn ick wieder voll drinne bin denn schrei ick schon.

Auch die Pre- und Postsession Fragebögen werden in dem Paper erwähnt. Jedoch spielten diese in meiner Analyse keine wirkliche Rolle, sodass dieser Punkt vernachlässigt werden kann.

2. Analyse

- Unterschiede

In dem Paper wird zu keinem Zeitpunkt erwähnt, welche Forschungsmethode gewählt wurde. Es werden lediglich die drei Arbeitsschritte "Videos mit Disengagement finden, Gründe für Disengagement finden, Lösungen für Vermeidung finden genannt", aber zu keinem Zeitpunkt wie genau vorgegangen wurde, um auf die Ergebnisse zu kommen. Außerdem tauchen keinerlei Codes auf, was auf eine GT hätte hinweisen können. Es erweckt den Eindruck, als hätte man die Schlüsse teilweise aus vereinzelt Aussagen getroffen.

Ich habe für meine Analyse das Forschungsdesign "Grounded Theory Methodology" verwendet. Ich habe die Videos codiert und aus den Codes Konzepte und daraus Theorien entwickelt.

- Gemeinsamkeiten

Die Gemeinsamkeit ist die Zielsetzung und die einzelnen Schritte. Ich habe ebenfalls zunächst nach Videos gesucht, die meine zu untersuchende Passivität vorweisen, und in diesen Videos dann unter anderem nach Gründen für die Passivität.

3. Ergebnisse

- Unterschiede

Das Paper kommt auf fünf Gründe, die zu einem Disengagement führen können. Lediglich eines davon kommt auch in meinen Ergebnissen vor: der soziale Druck. Die Autoren führen an, dass sich Anfänger zurückhalten während einer Paarprogrammierung, um "nicht dumm dazustehen". Dies entspricht meiner Erkenntnis von "Offenbarung einer Wissenslücke kann das Ego beschädigen". Diese Begründung stützen die Autoren auf zwei Aussagen, die in den Interviews getätigt wurden, auf die ich keinen Zugriff hatte.

Einer der Gründe, der genannt wird, ist "Simple Tasks". Dies steht zum Teil im Widerspruch zu meinen Ergebnissen. Während einfache Aufgaben in dem Paper dazu führen, dass Personen aussteigen und ihrem Partner diese überlassen, fand ich heraus, dass einfache Aufgaben dafür sorgen, dass passive Personen wieder aktiv werden und somit eine positive Folge entsteht.

- Gemeinsamkeiten

Das Paper kommt auch auf die Schlussfolgerung, dass das Ruder einen positiven Effekt hat. Es zwingt die "losgelöste" Person dazu sich einzubringen. Auf diese Erkenntnis bin ich auch gestoßen. Folglich erwähnt sowohl das Paper als auch meine Masterarbeit dies als Handlungsempfehlung.

Beide Arbeiten kommen auf den Entschluss, dass es einen Vorteil bringt, wenn man der passiven Person bzw. "novice" Fragen stellt. In dem Paper liegt der Fokus hierbei auf dem Stellen von Fragen, in denen der Novice nicht sicher zu sein scheint, den Partner und sein Vorhaben verstanden zu haben. In meiner Arbeit liegt der Fokus hierbei auf dem Schließen von Wissenslücke, die nicht unbedingt auf den Aussagen des Partners beruhen.

Sie kamen auch darauf, dass Partner häufig nicht mitbekommen, dass ihr Partner ausgestiegen ist bzw. nicht mitkommt.

Zusammenfassend lässt sich sagen, dass wir zwar mit demselben Datensatz, aber an überwiegend anderen Stellen im Datensatz gearbeitet haben. Die Methodik war eine andere. Die Ergebnisse und Empfehlungen stützen sich bei meiner Arbeit auf die von mir aus den Daten konstruierte Grounded Theory, während die Schlüsse aus dem Paper hauptsächlich aus den Interviews stammen zu scheinen, ohne in Kontext gebracht zu werden. Die Tatsache, dass wir mit unterschiedlichen Forschungsmethoden auf ähnliche bis gleiche Ergebnisse gekommen sind, erhöht die Glaubwürdigkeit dieser Ergebnisse.

7.1.2 Self-Esteem and Subjective Responses to Work Among Mature Workers: Similarities and Differences by Gender

Dieses Paper untersucht unter anderem Faktoren, die sich auf das Selbstwertgefühl von Arbeitern auswirken, und kommt zu dem folgenden Ergebnis:

"Sowohl die Arbeitszufriedenheit als auch die Arbeitsidentität als kompetent tragen dazu bei, das Selbstwertgefühl von Personen, die dem Rentenalter näher rücken, zu

stärken“[6]. Die unterstützt meine These, dass die Tatsache, wie man von seinem Umfeld wahrgenommen wird, sich auf das Selbstwertgefühl, in dieser Masterarbeit "Ego" genannt, auswirkt.

7.2 Limitation der Arbeit

Im folgenden werden meine Limitierungen aufgezeigt.

1. Einseitige und geringe Datenquelle
Eine der größten Limitierungen dieser Arbeit ist die Tatsache, dass nur drei Videos für die Analyse herangezogen werden konnten, von denen lediglich zwei das analysierte Phänomen enthielten. Dies ist nicht repräsentativ genug.
Dem habe ich versucht entgegenzuwirken, indem ich 23 Videos angesehen habe.
2. Videos aus einem Unternehmen
Eine zweite Limitierung ist, dass die drei Videos in demselben Unternehmen aufgenommen wurden. Es kann somit nicht ausgeschlossen werden, dass Beobachtungen eventuell damit zusammenhängen und in anderen Unternehmen anders ausfallen würden.
Hierbei ist jedoch zu beachten, dass es sich dennoch um unterschiedliche Personen in den Videos und somit personenunabhängige Ergebnisse handelt, sodass diese Limitierung abgeschwächt werden kann.
3. Theoretical Sampling
Theoretical Sampling hätte das Analysieren weiterer Sitzungen mit dem Fokus auf die Wissenslücke geboten. Hierbei hätte man Partner:innen analysieren müssen, die mit einer besonders großen Wissenslücke starten, um dann beobachten zu können, ob eine zunehmende Passivität sichtbar wird. Dies habe ich versucht umzusetzen, indem ich mehrere verschiedene Videos in Hinblick darauf angeguckt habe. Da dies in keinem weiteren Video der Fall war, hätte ich ab diesem Zeitpunkt neue Daten sammeln müssen, was aus zeitlichen Gründen nicht möglich war, sodass die zwei verwendeten Videos besonders tief analysiert wurden.
4. Kein Kontakt zu Versuchsteilnehmer:innen
Eine Limitierung war, dass die Videos zum einen nicht von mir aufgenommen wurden und zum anderen viele Jahre zurücklagen. Dies verhinderte das in Kontakttreten meinerseits mit den Personen aus den Videos, um Fragen zu klären, die im Rahmen der Analyse aufgekommen sind.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die Masterarbeit begann mit der Untersuchung von Entscheidungsepisoden in der Paarprogrammierung und entwickelte sich zu der Frage "Welche Faktoren beeinflussen die Passivität in der Paarprogrammierung, und welche Strategien oder Maßnahmen können entwickelt werden, um diese Passivität zu minimieren oder zu verhindern?".

Bei den Daten handelt es um Video- und Audioaufzeichnung und schriftliche Informationen in Form von Fragebögen und handschriftlichen Notizen. Diese wurden im Rahmen der Arbeitsgruppe "Software Engineering" (Freie Universität Berlin) über mehrere Jahre aufgenommen.

Analysiert wurden drei Videos, von denen zwei ein passives Verhalten eines Partners vorweisen. Das dritte agiert als Positivbeispiel für eine Paarprogrammierungssitzung entgegen den beiden Videos.

Als Forschungsmethode wird die Grounded Theory Methodology verwendet.

Die Analyse kam zu dem Ergebnis, dass eine Wissenslücke zwischen den Teilnehmer:innen eine wesentliche Rolle spielt, wenn es um Passivität geht. Ist eine Wissenslücke vorhanden, wird diese entgegen der Erwartungen nicht offenbart. Die Analyse hat gezeigt, dass ein Schutzmechanismus des Egos der plausibelste Grund für dieses Verhalten ist. Das Stellen von Fragen oder Formulieren von Vorschlägen, die abgelehnt werden, führen zu einer Beschädigung des Egos. Dies wird vermieden, indem Wissenslücken nicht offenbart werden, was einige Nachteile mit sich bringt.

Ist die Wissenslücke zu groß werden Entscheidungen ohne Verständnis evaluiert und die passiven Partner:innen neigen dazu, jedem Vorschlag einfach zuzustimmen, um ihre Wissenslücke nicht zu offenbaren. Dies führt dazu, dass ein Einfluss auf Entscheidungen weg fällt und die Qualitätssicherung verloren geht. Als weitere Folge nimmt die aktive Person eine noch aktivere Position ein.

Ein weiterer Aspekt, der negativ beeinflusst wird, ist der Wissenstransfer. Werden Wissenslücken nicht offen kommuniziert, hat der oder die Partner:in keine Möglichkeit diese zu erkennen und durch einen Wissenstransfer zu verringern. Als Folge wird die Wissenslücke immer größer bis die passive Person den Anschluss verliert und die Zusammengehörigkeit, das Herzstück der Paarprogrammierung, verloren geht.

Abschließend lässt sich sagen, dass durch die Passivität zwei wesentliche Vorteile der Paarprogrammierung (Wissenstransfer und das Treffen besserer Entscheidungen) verloren gehen.

Eine Handlungsempfehlung ist das Schaffen eines sicheren Hafens. Es soll kommuniziert werden, dass das Stellen von Fragen oder Einbringen von Vorschläge keine negativen Folgen mit sich bringt und im Gegenteil als lehrreich empfunden wird, siehe "egoless pairprogramming"[11]

Das Übergeben von Maus und Tastatur an die passive Person regt diese an sich aktiv an dem Prozess zu beteiligen, da der Prozess ins Stocken kommt, sollte diese regungslos dasitzen. Hat die Person eine Wissenslücke muss sie diese offenbaren, um etwas Qualitatives zu produzieren, und beschützt somit ihr Ego stärker als wenn sie die Wissenslücke nicht offenbaren würde.

8.2 Ausblick

Bei dem Betrachten der Analyse und der Limitierungen fällt auf, dass es sich in dieser Arbeit überwiegend um Existenzbeweise handelt. Beweise, die die Existenz der Phänomene zeigen, aber näher untersucht werden sollten.

Hierfür ist es sinnvoll, eine neue Datensammlung zu starten mit dem Fokus auf die Passivität der Personen. Man sollte dabei folgende Punkte beachten:

1. Aufnahmen in unterschiedlichen Unternehmen

Es ist sinnvoll, Videos in verschiedenen Unternehmen aufzunehmen, um zu verhindern, dass Beobachtungen beispielsweise aufgrund der Unternehmenskultur auftauchen.

2. Mehrere Videos von derselben Person

Mehrere Sitzungen von derselben Person können dabei helfen, ihr Verhalten besser einzuordnen. Es könnte vorkommen, dass eine Person eine passive Haltung vorweist, da sie an dem Morgen eine schlechte Nachricht bekommen hat und sie deshalb abgelenkt zu sein scheint oder einen schlechten Tag hat. Das Aufnehmen derselben Person zu unterschiedlichen Zeiten kann solche Fälle aufdecken, falls die Person in den anderen Videos diese passive Haltung nicht zeigt. Somit können persönliche Gründe ausgeschlossen werden.

3. Personen in unterschiedlichen Konstellationen

Personen sollten mit unterschiedlichen Partner:innen aufgenommen werden, um zu untersuchen, ob die passive Haltung mit den Partner:innen zusammenhängen könnte. Hier kann untersucht werden, ob die Wissenslücke der Grund für die Passivität ist. Man wählt Teilnehmer:innen, die eine kleine bis keine Wissenslücke vorweisen, und untersucht, ob dieselben Verhaltensweisen auftauchen wie bei Partner:innen mit großen Wissenslücken.

4. Interviews nach der Sitzung und Analyse

Es sollten mehrere Interviews stattfinden. Das Erste erfolgt direkt nach der Sitzung. Hier sollen die Einschätzungen und Gedanken der Personen in den Videos eingesammelt werden. Während der Analyse beschäftigt man sich zum ersten Mal als Forscher:in mit den Videos und kommt sowohl auf Erkenntnisse als auch auf Fragen. Es wäre sinnvoll, diese in weiteren Interviews mit den Programmierer:innen durchzusprechen, um den Interpretationsspielraum so klein wie möglich zu halten.

Des Weiteren wäre es empfehlenswert, eine Analyse zu starten, die sich mit dem Schutzmechanismus des Egos, der aktuell als plausibelster Grund aufgeführt wird, beschäftigt, um diese Beziehung zu manifestieren.

Eine weitere interessante Frage wäre, wie oft ein solches passives Verhalten tatsächlich vorzufinden ist.

Literaturverzeichnis

Literatur

- [1] Maryam Alavi und Dorothy Leidner. "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues". In: *MIS quarterly* 1 (März 2001), S. 107–. DOI: [10.2307/3250961](https://doi.org/10.2307/3250961).
- [2] Kent Beck und Cynthia Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2004.
- [3] Kathy Charmaz. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis (Introducing Qualitative Methods series)*. 1. Aufl. SAGE Publications Ltd, 2006.
- [4] Uwe Flick. *The SAGE Handbook of Qualitative Data Analysis*. SAGE Publications Ltd, 2013.
- [5] Udo Kuckartz. *Einführung in die computergestützte Analyse qualitativer Daten*. Springer Nature Switzerland AG, 2010.
- [6] Elizabeth J. Mutran u. a. "Self-esteem and subjective responses to work among mature workers: similarities and differences by gender". In: *J Gerontol B Psychol Sci Soc Sci* 52.2 (März 1997), S89–S96. DOI: [10.1093/geronb/52b.2.s89](https://doi.org/10.1093/geronb/52b.2.s89).
- [7] Laura Plonka, Helen Sharp und Janet van der Linden. "Disengagement in Pair Programming: Does it Matter?" In: *Proceedings - International Conference on Software Engineering* (2012), S. 496–506.
- [8] Lutz Prechelt und Stephan Salinger. *Understanding Pair Programming: The Base Layer*. BoD — Books on Demand, Norderstedt, Germany, Dez. 2013.
- [9] Stephan Salinger, Franz Zieris und Lutz Prechelt. "Liberating pair programming research from the oppressive driver/observer regime". In: *2013 35th International Conference on Software Engineering (ICSE)*. 2013, S. 1201–1204.
- [10] Anselm L. Strauss und Juliet Corbin. *Grounded Theory: Grundlagen Qualitativer Sozialforschung*. BELTZ Psychologie Verlags Union, 1996.
- [11] Gerald M. Weinberg. "Egoless Programming". In: *IEEE Software* 16.01 (Jan. 1999), S. 118–120.
- [12] Franz Zieris. "Qualitative Analysis of Knowledge Transfer in Pair Programming". Diss. Freie Universität Berlin, Fachbereich Mathematik und Informatik, Dez. 2020, S. 463.
- [13] Franz Zieris und Lutz Prechelt. "Explaining Pair Programming Session Dynamics from Knowledge Gaps". In: *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. 2020, S. 421–432.