



Erste Schritte zu einer Taxonomie von Software Engineering Kontexten

*A thesis presented for the degree of **Master of Science**.*

Stephanie Hohenberg, Freie Universität Berlin, Germany

Matriculation number: 5293431

stephanie.hohenberg@fu-berlin.de

1st Februar 2021

Supervisor:

Prof. Dr. Lutz Prechelt¹, Freie Universität Berlin, Germany

Reviewers:

Prof. Dr. Lutz Prechelt², Freie Universität Berlin, Germany

Prof. Dr. Heinz Schweppe³, Freie Universität Berlin, Germany

Citation:

Stephanie Hohenberg, *Erste Schritte zu einer Taxonomie von Software Engineering Kontexten*, Freie Universität Berlin, Master Thesis, 2021

Version: 0.1

¹Dept. of Computer Science and Mathematics, Software Engineering Research Group

²Dept. of Computer Science and Mathematics, Software Engineering Research Group

³Dept. of Computer Science and Mathematics, Databases and Information Systems Group

Statutory Declaration

I declare that I have developed and written the enclosed Master thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

The Master thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Berlin (Germany), 1st Februar 2021

Stephanie Hohenberg

Abstract

Hintergrund:

Ziele:

Methoden:

Ergebnisse:

Schlussfolgerungen:

Abstract

Background:

Objectives:

Methods:

Results:

Conclusions:

Table of Contents

1	Einführung	1
1.1	Motivation	1
1.2	Ziele	2
1.3	Struktur der Masterarbeit	3
2	Hintergrund	4
2.1	Definitionen	4
2.1.1	Qualitätsfaktoren von empirischen Arbeiten	4
2.1.2	Unterscheidungsfaktoren von empirischen Arbeiten	4
2.2	Verwandte Arbeiten	5
3	Vorgehen	7
3.1	Sichtung der Publikationen	8
3.2	Identifizierung von Kontextfaktoren	9
3.3	Klassifizierung von Kontextfaktoren	11
3.4	Charakterisierung von Publikationen	12
3.5	Dokumentierung und Erstellung von Artefakten	12
3.6	Analyse der Artefakte	13
3.7	Anfertigung einer Taxonomie	13
4	Durchführung	15
4.1	Kontexte	15
4.1.1	Thema	15
4.1.2	Eingrenzung und Abgrenzung	15
4.1.3	Verallgemeinbarkeit	16
4.1.4	Details	20
4.2	Taxonomie	20
5	Evaluation	21
5.1	Schlussfolgerungen	21
5.2	Beantwortung der Recherchefragen	21
5.3	Evaluation des Vorgehen	21
5.4	Evaluation der Kontexte und der Taxonomie	21
6	Ergebnisse	22
6.1	Zusammenfassung	22
6.2	Abgrenzung	22
7	Abschluss	23
7.1	Diskussion der Ergebnisse	23
7.2	Ausblick	23
8	References	24

List of Tables

3.1	Vorkommen von Kontextfaktoren nach Zweck	10
4.1	Kontexte bezüglich der Verallgemeinbarkeit an den Stellen <i>Threats to Validity</i> (TtV), <i>Conclusion</i> (C), <i>Discussion</i> (D) <i>Limitations</i> (L), <i>Future Work</i> (FW)	17
4.2	Analyse der Diskussion zur Verallgemeinbarkeit	18
4.3	Analyse der Vorkommen von Sektionen zu <i>Threats to Validity</i> , <i>Limitations</i> (L) und <i>Future Work</i> (F)	19

List of Figures

3.1	Vorgehen	7
3.2	59 Publikationen mit Auszeichnungen	8

List of Source Codes

1 Einführung

Im Folgendem wird die Motivation zur Anfertigung dieser Masterarbeit "*Erste Schritte zu einer Taxonomie von Software Engineering Kontexten*" erläutert, Ziele werden gesetzt und die Struktur, die der Masterarbeit zu Grunde liegt, wird beschrieben.

1.1 Motivation

Die Qualität von Arbeiten in der Forschung von Software Engineering werden durch verschiedener Faktoren gemessen. Ein Qualitätsfaktor ist die Relevanz einer Arbeit. Dabei spielt die Verallgemeinbarkeit eine Rolle: die Anwendbarkeit der vorgestellten Techniken bzw. die Generalisierbarkeit und die Übertragbarkeit der empirischen Ergebnisse auf andere Kontexte.

Es kann unterschieden werden zwischen den Kontext in welchem einem empirische Studie durchgeführt wurde und die Kontexte auf welche dessen Ergebnisse sich verallgemeinern lassen und welche somit nicht nur die Relevanz der Arbeit in dem Feld Software Engineering beschreiben, sondern auch, ob die Arbeit und dessen Ergebnisse relevant für andere Arbeiten sind und für deren Beweisführung genutzt werden kann.

Jedoch fehlt in der Forschung von Software Engineering eine einheitliche Terminologie, um den Kontext einer Arbeit knapp beschreiben zu können.

Wenn der Kontext allerdings gar nicht erläutert wird, nur mit wenig Details und Sorgfalt, kann zum Einen der Kontext einer Arbeit, sowie dessen Verallgemeinbarkeit und Relevanz weniger gut eingeschätzt werden und somit können Verfasser und Verfasserinnen von anderen empirischen Arbeiten schwieriger beurteilen, ob die Ergebnisse einer Arbeit relevant für ihre eigenen Arbeiten sind. Zu Anderem können dadurch nur Annahmen über den Kontext getroffen werden.

Dies kann zur Folge haben, dass Arbeiten, die auf Ergebnisse anderer Arbeiten basieren, fehlgeleitet werden, sich fälschlicherweise an Ergebnisse anderer Arbeiten stützen und dieser fälschlicherweise zur Beweisführung nutzen.

Damit der Kontext und somit die Relevanz von Arbeiten besser beschrieben und besser eingeschätzt werden kann und damit die Forschung von Software Engineering Beweis-orientierter werden kann, soll in dieser Arbeit erste Schritte zu einer Taxonomie von Software Engineering Kontexten getan werden und eine Terminologie geschaffen werden.

1.2 Ziele

Das Ziel dieser Arbeit ist es Kontextfaktoren zu finden, welche sich auf die Anwendbarkeit der vorgestellten Technik oder auf die Generalisierbarkeit und Übertragbarkeit der empirischen Ergebnisse beziehen und welche somit den Kontext einer Arbeit beschreiben auf dessen diese verallgemeinbar ist.

Um das Ziel zu erreichen, werden in dieser Arbeit zahlreiche Publikationen gesichtet, welche innerhalb der Konferenz *International Conference on Software Engineering*, im Weiteren abgekürzt als ICSE, veröffentlicht wurden. Dabei wird sich nicht auf die Konferenz eingeschränkt, sondern auch auf das Veranstaltungsjahr der Konferenz und auf den Typ der Publikationen. Es werden nur technische Arbeiten von dem letztem Jahr 2020 der ISCE untersucht, um Kontextfaktoren aus der aktuellsten Software Engineering Literatur zu ermitteln. Es wurde sich für die ISCE Konferenz entschieden, da die Artikel, die in Rahmen dieser Konferenz veröffentlicht und vorgestellt werden, eine hohe Qualität aufweisen und somit erwartet wird, dass in den Artikeln die Kontexte beschrieben werden.

Bei der Untersuchung der Publikationen, ist nicht nur die Identifizierung der Kontextfaktoren ein Ziel, sondern auch dessen Klassifizierung und eine kritische Analyse. Kontextfaktoren werden in thematischen Faktorenklassen zusammengefasst und Korrelationen zwischen den Kontextfaktoren bzw. Faktorenklassen untersucht. Welche Kontextfaktoren bzw. Faktorenklassen tauchen oft bzw. typischerweise zusammen auf? Welche Kombinationen sind wichtig? Sind alle Faktorenklassen und Kombinationen gleich relevant? Aus welchen Kombinationen von Kontextfaktoren und Faktorenklassen lässt sich ein Kontext beschreiben, welcher die Anwendbarkeit einer Arbeit umfasst?

Aus den Kontexten, die in den Publikationen der ISCE 2020 vorkommen, soll eine Taxonomie erstellt werden und somit eine Terminologie geschaffen werden. Dafür ist es notwendig die Kontexte zu benennen, falls kein gängiger Name vorhanden ist.

Des Weiteren werden die Korrelationen zwischen Faktorenklassen und den Recherchetyp betrachtet. Da zu erwarten ist, dass sich die Kontextfaktoren innerhalb der verschiedenen Recherchetypen unterscheiden.

Außerdem ist zu erwarten, dass Kontextfaktoren an verschiedenen Stellen in einer Arbeit Erwähnung finden. Daher wird auch untersucht, wo Kontextfaktoren in Publikationen beschrieben werden und wie gründlich das getan wird, oder wo Kontextfaktoren nur implizit genannt werden.

1.3 Struktur der Masterarbeit

Die Masterarbeit besteht aus sieben Kapitel.

Zunächst wird in **Kapitel 2 Hintergrund**, Wissen zu den grundlegenden Themen *Qualitätsfaktoren empirischer Arbeiten* und *Unterscheidungsfaktoren von empirischen Arbeiten* vermittelt. Des Weiteren werden *verwandte Arbeiten* vorgestellt.

In **Kapitel 3 Vorgehen** werden die Aktivitäten beschrieben, um die gesetzten Ziele zu erfüllen: *Identifizierung von Kontextfaktoren*, *Klassifizierung von Kontextfaktoren*, *Charakterisierung von Publikationen*, *Dokumentierung und Erstellung von Artefakten*, *Analyse der Artefakte* und *Anfertigung einer Taxonomie*.

Dabei wird nicht nur erklärt, wo Kontextfaktoren vorkommen könnten, sondern auch die Lesetechnik wird vorgestellt, Recherchefragen werden aufgestellt und die Verwendung von Werkzeugen wird erläutert.

In darauf folgendem **Kapitel 4 Durchführung** wird die Realisierung des Vorgehens und die entstandenen Artefakte in Form von *Kontextfaktoren und Faktorengruppen*, sowie in Form von *Kontexte und einer Taxonomie* dokumentiert.

Anschliessend wird in **Kapitel 5 Evaluation** *Schlussfolgerungen* aus diesen Artefakten gezogen und die zuvor in Sektion 3.6 aufgestellten *Recherchefragen* werden beantwortet. Des Weiteren werden die *Kontexte*, die *Taxonomie*, sowie die *Lesetechnik* evaluiert.

In **Kapitel 6 Ergebnisse** werden die Ergebnisse dieser Arbeit zusammengefasst und abgegrenzt.

Im letzten **Kapitel 7 Abschluss** erfolgt eine *Diskussion der Ergebnisse* und ein *Ausblick*.

2 Hintergrund

Im Folgendem wird der Hintergrund der Masterarbeit vorgestellt. Dazu wird nötiges Hintergrundwissen zu den Qualitätsfaktoren empirischer Arbeiten und den Arten von empirischen Arbeiten bereitgestellt und verwandte Arbeiten werden präsentiert.

2.1 Definitionen

2.1.1 Qualitätsfaktoren von empirischen Arbeiten

Wie bereits in Kapitel 1 erwähnt ist die Relevanz einer Arbeit ein Qualitätsfaktor um empirischen Arbeiten zu bewerten. Jedoch wird in der empirischen Bewertung im Software Engineering auch über andere Qualitätsfaktoren diskutiert.

Glaubwürdigkeit kann eine Publikation erlangen durch die Gründlichkeit der Beschreibungen, Offenheit zu anderen Ergebnissen, Beschreibung von Einschränkungen, klare und anschauliche Beschreibung der Ergebnisse und der Motivation.

Relevanz erreicht eine Publikation, wenn das Thema interessant und wichtig ist. Es ist nicht überspezialisiert, sondern ist übertragbar und generalisierbar.

2.1.2 Unterscheidungsfaktoren von empirischen Arbeiten

Die Publikation der ISCE2020 stellen zu XX% eine neue Technik vor, die zu XX% mit empirischen Studien evaluiert wird. Die anderen XX% der Publikationen sind durch Recherchefragen motiviert und liefern durch empirische Studien ebenfalls empirische Ergebnisse.

Da innerhalb dieser Arbeit zahlreiche Publikationen gesichtet werden und auf die Anwendbarkeit der vorgestellten Techniken oder Übertragbarkeit der empirischen Ergebnisse betrachtet werden, werden in dieser Sektion die verschiedenen Arten empirischer Arbeiten vorgestellt und wie empirische Arbeiten unterschieden werden können.

Zu Einem lassen Arbeiten sich einordnen anhand der Art resultierender Beiträge, aber auch anhand dessen Vorgehensweise, sowie ihrer Anwendung von empirischer Methoden.

2.2 Verwandte Arbeiten

Im Folgendem wird erklärt, wie sich diese Masterarbeit in verwandten Arbeiten einbettet.

Die Masterarbeit versucht zwei der Wissenlücken zu füllen, die **Lutz Prechelt** 2019 in seiner Arbeit *"Four presumed gaps in the software engineering research community's knowledge"* herausgestellt hat und zu denen er vorschlägt zu forschen. [5] Mittels semi-strukturierten Interviews, Literaturrecherche und einen Beweis-orientierten Ansatz, wurden vier Wissenslücken aufgedeckt und vier Ziele für weitere Forschungen gesetzt. Für diese Masterarbeit ist zum Einem das Ziel interessant, dass Annahmen, welche den Umfang und die Relevanz einer Arbeit beschreiben, evaluiert werden, um einschätzen zu können, in welchen Kontexten eine Arbeit relevant ist. Zum Anderem ist das Ziel eine Taxonomie zu definieren, um eine einheitliche weitreichende Terminologie zu schaffen, das Thema dieser Masterarbeit.

Andrew Forward und **Timothy C. Lethbridge** haben 2008 in ihrer Arbeit *"A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering"* eine Taxonomie zu verschiedenen Typen von Software erstellt um empirische Beweise, die auf Software basieren, zuordnen zu können. [2]. Dazu haben sie bereits bestehende Taxonomien zu Software Typen analysiert und durch Brainstorming eine eigene Taxonomie erstellt.

Forward und Lethbridge meinen, dass der Softwaretyp einen Kontext darstellt. Doch in dieser Arbeit bezieht sich ein Kontext einer empirischen Arbeit auf mehr als den Typ der Software, die untersucht oder vorgestellt wurde. Der Typ der Software ist lediglich eine Faktorenklasse, die Kontextfaktoren bezüglich des Softwaretypes zusammenfasst und mit weiteren Faktorenklassen einen Kontext abbilden kann.

Jedoch ist diese Arbeit trotzdem für die Masterarbeit von Nutzen, um weitere Schritte hinsichtlich einer umfassenderen Taxonomie zu tun.

In der Arbeit *"Context in Industrial Software Engineering Research"* 2009 verfasst von **Kai Petersen** und **Claes Wohlin**, wurde eine Checkliste erstellt, um Wissenschaftler und Wissenschaftlerinnen zu unterstützen den Kontext ihrer Arbeiten zu beschreiben. [4] Des Weiteren haben Petersen und Wohlin Literatur der Fachzeitschrift *empirical software engineering* (EMSE) gesichtet, um zu evaluieren in wie weit Autoren und Autorinnen ihre Checkliste erfüllen.

Die Checkliste wurde unterteilt in sechs Kategorien, welche in dieser Masterarbeit als

Faktorenklassen abgebildet werden könnten. Wie die Arbeit von Forward und Lethbridge sollte die Arbeit bei der Anfertigung der Taxonomie berücksichtigt werden.

Tore Dybå, Dag I.K. Sjøberg und **Daniela S. Cruzes** verfassten 2012 ebenfalls eine Arbeit zu dem Thema Kontexte in Software Engineering: *"What works for whom, where, when, and why?: on the role of context in empirical software engineering"*. [1] Sie behaupten, dass in der empirischen Forschung zu Software Engineering, eher nach universalen Beziehungen gesucht werden, die unabhängig von dem Kontext sind. Jedoch lassen sich einige Forschungsfragen in unterschiedlichen Kontexten unterschiedlich beantworten. Sie erklären, wie eine Forschungsfragen kontextuell gefragt werden kann und dass die Beschreibung von Kontextfaktoren wichtig ist, um Wissen über mehrere Studien hinweg aufzubauen.

Die Verfasser und Verfasserinnen beziehen sich auf die Definition des Begriffes Kontext von **Gary Johns**, der in seiner Arbeit *"The essential impact of context on organizational behavior"* untersucht, welchen Einfluss ein Kontext hat und in welchen Dimensionen ein Kontext beschrieben werden kann. [3] Zwar forscht Johns in dem Fachgebiet zum organisatorischem Verhalten, jedoch sind seine Erkenntnisse und Schlussfolgerungen aufschlussreich für diese Masterarbeit.

3 Vorgehen

Im Folgenden wird das Vorgehen erläutert, um die im ersten Kapitel genannten Ziele zu erreichen. Dabei wurde folgende sieben Aktivitäten ausgearbeitet, die nicht aufeinander aufbauen, sondern in einem iterativen Prozess miteinander verzahnt sind, welcher in der Grafik 3.1 veranschaulicht und im Weiteren näher erläutert wird.

- ▶ Sichtung der Publikationen
- ▶ Identifizierung von Kontextfaktoren
- ▶ Klassifizierung von Kontextfaktoren
- ▶ Charakterisierung von Publikationen
- ▶ Dokumentierung und Erstellung von Artefakten
- ▶ Analyse der Artefakte
- ▶ Anfertigung einer Taxonomie

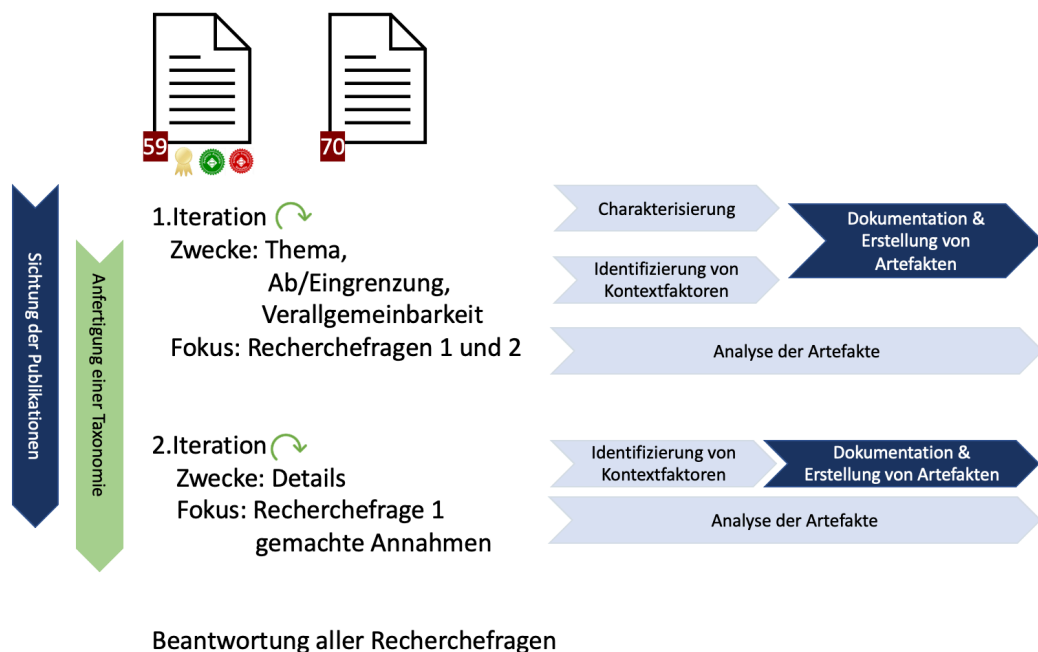


Figure 3.1: Vorgehen

3.1 Sichtung der Publikationen

Zur Identifikation von Kontextfaktoren, werden die Publikationen der technischen Artikel veröffentlicht in der ISCE 2020, in mehreren Iterationen gesichtet.

In der 1. Iteration basiert die Reihenfolge, anhand welcher die Publikationen gelesen werden, auf die Auszeichnungen und Badges, die die Publikationen erhalten haben, da angenommen wird, dass diese Publikationen eine höhere Qualität aufweisen. Von den 129 technischen Arbeiten der ISCE 2020 haben zehn Publikationen die Auszeichnung *“ACM SIGSOFT Distinguished Paper Award”* und drei Publikationen die Auszeichnung *“ACM SIGSOFT Distinguished Artifact Award”* verliehen bekommen. Außerdem wurden 34 Publikationen mit dem Badge *“ACM Artifacts Evaluated Reusable”* und 47 Publikationen mit dem Badge *“ACM Artifacts Available”* ausgezeichnet. Dabei besteht jedoch eine Überschneidung von den ausgezeichneten Publikationen, wie in der Grafik 3.2 dargestellt. Insgesamt haben 59 von den 129 Publikationen eine Auszeichnung und/oder ein Badge.



Figure 3.2: 59 Publikationen mit Auszeichnungen

Zuerst werden die 13 Publikationen mit einer *“ACM SIGSOFT”* Auszeichnung gelesen, danach die 25 Publikationen mit zwei Badges, die 21 Publikationen mit einem Badge und im Anschluss die restlichen Publikationen. Dabei ist die Reihenfolge innerhalb dieser vier Gruppen zufällig gewählt.

Während dem erstmaligen Sichten einer Publikation, soll diese charakterisiert werden, welches eine weitere Aktivität des Vorgehens darstellt und in der Sektion 3.4

näher beschrieben wird.

Unter anderem wird abhängig von dieser Charakterisierung, aber auch abhängig von den in der 1. Iteration bereits identifizierten Kontexten und von den Themengebieten einer Publikation, die Publikationen gruppiert. Innerhalb dieser Gruppen werden die Publikationen in einer 2. Iteration erneut gesichtet und kritischer analysiert.

3.2 Identifizierung von Kontextfaktoren

Da der Aufwand eine Publikation komplett in voller Länge intensiv zu lesen nicht dem Nutzen alle Kontextfaktoren zu finden entspricht und angenommen wird, dass Kontextfaktoren unterschiedlich häufig an unterschiedlichen Stellen einer Publikation vorkommen, wird sich beim Sichten der Publikationen auf diese Stellen konzentriert. Dazu wird im Folgenden weitere Annahmen getroffen zu den Stellen, an welchen Kontextfaktoren auftreten.

Es wird angenommen, dass Kontextfaktoren abhängig von dessen Zwecken an unterschiedlichen Stellen in einer konkreten Publikation vorkommen.

Denn Kontextfaktoren haben nicht nur den Zweck die Verallgemeinbarkeit einer Publikation zu beschreiben, wie in Kapitel 1 bei der Zielsetzung ausführlich erläutert, sondern auch zu anderen Zwecken. Denn es kann, wie bereits erwähnt, unterschieden werden zwischen den Kontexten in denen eine empirische Studie durchgeführt wurde und die Kontexte auf welche dessen Ergebnisse verallgemeinbar sind.

Dieser Zusammenhang zwischen dem Zweck eines Kontextfaktors und der Stelle in der Publikation wird im Weiteren erläutert und in der Tabelle 3.1 aufgeführt.

In dem Titel und in der Einleitung einer Publikation ist zu erwarten, dass Kontextfaktoren zur Charakterisierung des Themenbereichs oder zur Definition des untersuchten Problems verwendet werden.

Des Weiteren können Kontextfaktoren bei der Beschreibung einer Technik oder des Vorgehens einer empirischen Studie aufgeführt werden um Einzelheiten zu beleuchten und zu beschreiben in welchen Kontext Daten erhoben wurden.

Bei der Diskussion verwandter Arbeiten, haben Kontextfaktoren den Zweck die konkrete Publikation einzugrenzen und von anderen Publikationen abzugrenzen. An dieser Stelle können Kontextfaktoren genannt werden, die das Thema der Publikation einschränken und somit spezialisieren. Es kann beschrieben werden, in welchen Kontext die Studie nicht durchgeführt wurde und außerdem kann die Studie oder dessen Ergebnisse in Betrachtung anderer Arbeiten positioniert werden.

Bei den Schlussfolgerungen, Nennung von Forschungsbeiträgen und bei der Diskussion zur Verallgemeinbarkeit wird unter anderem diskutiert, was die Validität der Publikation gefährden könnte und Kontextfaktoren diesbezüglich aufgestellt. Es ist zu

Erwarten, dass an dieser Stelle viele Kontextfaktoren zu finden sind, die sich konkret auf die Verallgemeinbarkeit der Arbeit beziehen.

Zweck	Vorkommen
Thema	Titel, Einleitung, Recherchefragen
Detail	Beschreibung, Vorgehen, Datenerhebung
Abgrenzung	Diskussion verwandter Arbeiten
Verallgemeinbarkeit	Schlussfolgerungen, Forschungsbeiträge, Diskussion

Table 3.1: Vorkommen von Kontextfaktoren nach Zweck

Bei dem erstmaligen Sichten der Publikationen der ISCE 2020, wird sich auf die in Tabelle 3.1 aufgelisteten Stellen beschränkt.

Um sich einen Überblick über eine Publikation zu schaffen, werden zuerst die Titel der einzelnen Sektionen einer konkreten Publikation gelesen und zugeordnet, zu welchen Zweck Kontextfaktoren in einer konkreten Sektion vorkommen könnten. Anschließend werden die Sektionen, von denen vermutet wird, dass dort Kontextfaktoren vorkommen, in folgender Reihenfolge abhängig des Zweckes gelesen: Thema, Abgrenzung, Verallgemeinbarkeit.

Zunächst werden Sektionen, die sich auf das Thema der Publikation beziehen gelesen, um nicht nur Kontextfaktoren bezüglich des Themas zu finden, sondern auch um eine schnelle Einleitung in das Thema der Publikation zu erlangen. Anschließend werden die Sektionen bezüglich der Abgrenzung gesichtet, wie die Diskussion verwandter Arbeiten, um das Thema einzugrenzen. Danach werden die Sektionen bezüglich der Verallgemeinbarkeit, wie die Schlussfolgerungen und Diskussion gelesen. Dabei wird analysiert, welchen Beitrag die Publikation leistet. Erst in der 2. Iteration werden die Sektionen der Publikation angeschaut, die Details erläutern, und eine kritischere Analyse erfolgt.

Bei dieser Lesetechnik, werden die Publikationen innerhalb zwei Iterationen nacheinander gesichtet. Das hat den Vorteil, dass es zu keinem Kontextwechsel kommt, wie es bei der Alternative der Fall wäre, wenn in vier Iterationen die Publikationen lediglich hinsichtlich eines Zweckes gesichtet werden, sprich zuerst werden alle Publikationen hinsichtlich des Themas gelesen, danach alle hinsichtlich der Verallgemeinerung, und so weiter. Dadurch könnte das Verständnis einer konkreten Publikation gemindert werden, sodass implizite Kontextfaktoren nicht entdeckt werden könnten. Jedoch hätte die Alternative den Vorteil, dass Beziehungen zwischen Kontextfaktoren verschiedener Publikationen bezüglich eines Zweckes besser erkannt werden könnten.

Da durch die Lesetechnik eventuell nicht alle Kontextfaktoren gefunden werden, wird in Kapitel 5 die Lesetechnik evaluiert und abgeschätzt, wie viele Kontextfaktoren unerkannt blieben als *false negative*. Eine maximale bzw. vollständige Abdeckung ist jedoch nicht das Ziel dieser Arbeit.

Eine weitere Schwierigkeit ergibt sich dadurch, dass lediglich die Autorin dieser Masterarbeit alle Publikationen sichtet, Kontextfaktoren identifiziert und klassifiziert. Somit besteht zwar ein Grad an Einheitlichkeit. Jedoch könnten Verzerrungen und Bias entstehen.

Dies könnte verhindert werden, wenn mehrere Autoren und Autorinnen beteiligt wären und diese Aktivität unabhängig voneinander ohne gegenseitige Beeinflussung für eine zufällig ausgewählte Teilmenge der Publikation durchführen würden. Wenn sich dann noch die Teilmengen überschneiden, sodass jede Publikation mehrmals gesichtet wird, hat es den Vorteil, dass nicht nur mehr Kontextfaktoren identifiziert werden könnten, sondern auch, dass die interne Validität der Arbeit gestärkt wird.

Da jedoch nach der eidesstaatlichen Erklärung diese Arbeit komplett alleine durchgeführt wurde, ist dieser Ansatz nicht möglich. Allerdings wäre es vorstellbar, dass eine Folgearbeit geleistet werden kann, um Ergebnisse dieser Arbeit zu evaluieren und die interne Validität dieser Arbeit zu stärken.

Um die Qualität der Ergebnisse zu sichern, werden Argumentationsstränge so gut wie möglich offengelegt und es wird beschrieben, wie aus den Textstellen konkreter Publikationen Schlussfolgerungen entstanden sind. Dadurch kann der Leser bzw. die Leserin dieser Masterarbeit sich eine eigene Meinung über die Ergebnisse bilden und die Argumentation verfolgen und bewerten.

3.3 Klassifizierung von Kontextfaktoren

Im Anschluss der Sichtung einer konkreten Publikation werden die identifizierten Kontextfaktoren dieser Publikation klassifiziert und Faktorengruppen werden gebildet. Dabei wird ein Oberbegriff gefunden, der den Kontextfaktor beschreibt. Ein Beispiel dafür wäre, wenn eine Publikation ¹ Softwareprojekte untersucht, die durch das Kriterium *Nutzung von Java* ausgewählt wurden, lässt sich der Kontextfaktor *Java* erkennen und ein passender Oberbegriff dafür, wäre *Technologien*, welche die Faktorengruppe für alle Kontextfaktoren die Technologien beschreiben bildet.

Da bei den Publikationen die zuerst gesichtet werden, noch keine Klassen bzw. Faktorengruppen bestehen, weil die Faktorengruppen fortschreitend nach jeder Publikation ausgearbeitet werden, müssen die entstandenen Faktorengruppen und dessen zugeordneten Kontextfaktoren regelmäßig evaluiert und angepasst werden. Da alle Kontextfaktoren, die bereits zu einer Faktorengruppe zugeordnet sind, sich mit dem neuklassifizierten Kontextfaktor einer anderen Publikation ähneln müssen und die Faktorengruppe zu allen zugeordneten Kontextfaktoren passen muss.

Dabei kann es vorkommen, dass eine Faktorengruppe umbenannt wird auf einen mehr generalisierten Begriff, um alle Kontextfaktoren zu umfassen, oder dass eine

¹z.B. in der Publikation XXX

Faktorengruppe auf zwei spezialisierte Faktorengruppen aufgeteilt wird, sodass eine Hierarchie von Faktorengruppen entstehen könnte.

Wie bereits in Kapitel 2 Sektion 2.2 erwähnt, sollen ebenfalls verwandte Arbeiten evaluiert werden, um Kontextfaktoren und Faktorenklassen zu finden. Außerdem können geeignete Benennungen für Faktorengruppen übernommen werden.

3.4 Charakterisierung von Publikationen

Bei dem Leseprozess, sollen nicht nur Kontexte einer Publikation identifiziert werden. Die Publikation soll auch charakterisiert werden, damit Korrelationen zwischen Kontextfaktoren und den Publikationen erkannt werden können.

Im Gegensatz zu der Klassifizierung von Kontextfaktoren in Faktorengruppen, dessen Anzahl im Leseprozess steigt, werden die Charakterisierungskriterien und Optionen nicht fortschreitend auserarbeitend, sondern vorab definiert.

Da bereits in Kapitel 2 *Hintergrund* Sektion 2.1.2 erläutert wurde, wie sich empirische Arbeiten unterscheiden können, wird hier nicht mehr näher darauf eingegangen, sondern auf diese Sektion verwiesen. Anhand der Unterscheidungsfaktion von empirischer Arbeiten wurden Charakterisierungskriterien ausgewählt, welche in Kapitel 2 *Hintergrund* Sektion 2.1.2 aufgelistet wurden.

3.5 Dokumentierung und Erstellung von Artefakten

Um diesen iterativen Prozess, welcher in Grafik 3.1 dargestellt ist, zu bewerkstelligen und eine Übersicht von Kontextfaktoren und Faktorengruppen zu ermöglichen und eventuell bereits eine Hierarchie von Faktorengruppe abbilden zu können, werden die Werkzeuge *MaxQDA* und *Microsoft Excel* genutzt.

Mit Hilfe von *MaxQDA* wurden die Publikationen gesichtet und markiert. Dabei wurden verschiedene Farben für verschiedene Zwecke genutzt.

- ▶ rot für die Kontexte
- ▶ gelb für Notizen, Verständnis
- ▶ blau für Zitate bzgl der Verallgemeinbarkeit
- ▶ lila für Zitate bzgl Future Work oder Ergebnisse
- ▶ grün für die Charakterisierung der Publikation

Im Anschluss einer Sichtung wird mittels *Microsoft Excel* die Publikation charakterisiert, und Kontext erfasst. Dabei werden Tabellen erstellt.

Es wurde sich für diese Werkzeuge entschieden, da diese für die Autorin kostenlos verfügbar sind. Die entstandenen Artefakte sind im mx20.-Format und im xlsx.-Format zugänglich ².

3.6 Analyse der Artefakte

Im nächsten Kapitel 4 *Durchführung* wird nicht nur die bereits beschriebene Lesetechnik angewendet und das bereits erläuterte Vorgehen durchgeführt, sondern auch kritisch analysiert. Diese Analyse soll anhand folgender Recherchefragen geleitet werden, welche so wie das Vorgehen in Kapitel 5 *Evaluation* evaluiert werden und somit beantwortet werden.

Recherchefrage 1: Welche Kontextfaktoren werden in den Publikationen erwähnt? In wie weit beschreiben Autoren und Autorinnen den Kontext ihrer Publikationen? In welcher Form kommen Kontextfaktoren vor? Sind diese explizit oder implizit erkennbar? Sind andere Zwecke als, die in der Tabelle 3.1 genannten, erkennbar?

Recherchefrage 2: In wie weit diskutieren Autoren und Autorinnen über die Verallgemeinbarkeit ihrer Publikationen, Ergebnisse und Techniken? Bestehen Lücken in den Diskussionen zur Verallgemeinbarkeit? Wie unterscheiden sich die Kontextfaktoren zum Zweck der Verallgemeinbarkeit zu den Kontextfaktoren anderer Zwecke?

Recherchefrage 3: Unterscheiden sich die Kontexte und Kontextfaktoren abhängig von den Arten der Publikationen und von der Vorgehensweise der empirischen Studie?

Bei der kritischen Analyse soll vor Allem die erstellten Artefakte genutzt werden, um relevante Daten für die Beantwortung der Recherchefragen sichtbar zu machen und auswerten zu können. Des Weiteren werden Korrelationen und Zusammenhänge in geeigneten Grafiken visualisiert oder in Tabellen zusammengefasst.

3.7 Anfertigung einer Taxonomie

Im Anschluss des bereits erläuterten Vorgehen werden ersten Schritte zu einer Taxonomie von Software Engineering Kontexten gemacht. Dabei soll wie bei dem Schritt zur *Analyse der Artefakte*, welcher in Sektion 3.6 dieses Kapitels erläutert wurde,

²Link zum github XX

das Datenbank Management System genutzt werden, wo bereits durch die Verbindungen zwischen Faktorengruppen eine Hierarchie erkennbar sein kann.

Jedoch besteht ein Kontext nicht nur aus einer Faktorengruppe, sondern aus mehreren. Daher soll analysiert werden, in welchen Kombinationen Faktorengruppe in den Publikationen vorkommen und in wie weit diese Kombinationen sich häufen. Dabei soll nicht nur der Kontext erkannt werden, in der eine empirische Arbeit durchgeführt wurde, sondern auch die Kontexte auf welche diese verallgemeinbar ist.

Wie bei dem Schritt zur *Analyse der Artefakten*, welcher in Sektion 3.6 dieses Kapitels erläutert wurde, sollen geeignete SQL-Abfragen geschrieben werden, um die Kontexte aufzudecken.

Ebenfalls soll, wie im Schritt zur *Klassifizierung von Kontextfaktoren*, welcher in Sektion 3.3 dieses Kapitels beschrieben wurde, verwandte Arbeiten genutzt werden, um die Kontexte und die Taxonomie zu evaluieren oder zu ergänzen.

4 Durchführung

Im Folgendem wird die Realisierung des im vorherigen Kapitel beschreibenden Vorgehen dokumentiert. Dafür werden die gesichteten Publikationen analysiert und die entstandenen Artefakte beschrieben.

4.1 Kontexte

4.1.1 Thema

4.1.2 Eingrenzung und Abgrenzung

4.1.3 Verallgemeinbarkeit

Im Folgendem werden die Kontextfaktoren beschrieben, die zu dem Zweck der Verallgemeinbarkeit zugeordnet wurden. Des Weiteren wird untersucht, in wie weit Autoren und Autorinnen über die Verallgemeinbarkeit ihrer Annahmen diskutieren.

Vorrangig werden lediglich die 59 Publikationen untersucht, die mit einem ACM Award für *Distinguished Paper* oder *Distinguished Artefact* ausgezeichnet wurden und/oder einen Badge erhalten haben *ACM Artifacts Evaluated Reusable* oder *ACM Artifacts Available*. Es wird angenommen, dass diese Publikationen eine höhere Qualität aufweisen und mehr auf die Verallgemeinbarkeit eingegangen wird.

Wie bereits in Kapitel 3 beschrieben, wird angenommen, dass die Kontexte mit dem Zweck der Verallgemeinbarkeit an bestimmten Stellen in einer Publikation vorkommen. Daher wurden lediglich folgende Stellen berücksichtigt: *Threats of Validity*, *Conclusion*, *Limitations*, *Future Work* und *Discussion*.

Innerhalb 34 Publikationen wurden 35 Kontextfaktoren identifiziert. Davon beziehen sich 24 Kontextfaktoren auf die Verallgemeinbarkeit, 8 auf die Abgrenzung des Themas, zwei auf das Thema selbst und ein Kontextfaktor beschreibt Details der Publikation. Die Kontextfaktoren bezüglich der Verallgemeinbarkeit wurden in der Tabelle 4.1 zusammengefasst und in vier Faktorengruppen klassifiziert: Anwendung, Zielgruppe Programmiersprachen und Testtypen.

ID	Kontextfaktor	Klassifikation	Vorkommen
b535	Crowdtesting Plattform	Anwendung	TtV
b535	funktionale Tests	Testtypen	TtV
b535	Usability Tests	Testtypen	TtV
b535	Security Tests	Testtypen	TtV
b535	Performanz Tests	Testtypen	TtV
a435	open-source Systeme	Anwendung	TtV
a435	kommerzielle close-source Systeme	Anwendung	TtV
a949	convolutional neural network	Anwendung	TtV
b110	TensorFlow, Keras, Pytorch	Anwendung	TtV
b159	Philly und andere DL Plattformen	Anwendung	TtV
a691		Programmiersprachen	L
a309	Entwickler/innen, industrielle Teams	Zielgruppe	TtV
a481	große, industrielle Apps	Anwendung	C
a026	other domains with partially ordered constructs	Anwendung	C
a075	other applications and grammar	Anwendung	TtV
b459	SMT solver, other classes of solvers	Anwendung	C, FW
b086	Python, other languages, language-agnostic	Programmiersprachen	D, L
b061	specific scenarios	Anwendung	C
b447		Programmiersprachen	TtV
b447	real-world Java programs	Anwendung	C
a847	active analyzer and real world programs	Anwendung	C
a752	OSS Datensatz mit Puppet Skripten	Anwendung	TtV
a752	across languages	Programmiersprachen	TtV
a457	Android Apps	Anwendung	TtV
a666	Softwareentwicklungsfirmen, Arbeitsplatz	Zielgruppe	C
b183	große, open-source Softwareprojekte	Anwendung	TtV
a641	open-source Projekte, close-source Projekte	Anwendung	TtV
a210	Java und Python	Programmiersprachen	L, T
a714	control applications	Anwendung	TtV
a714	convolutional neural network	Anwendung	TtV

Table 4.1: Kontexte bezüglich der Verallgemeinbarkeit an den Stellen *Threats to Validity* (TtV), *Conclusion* (C), *Discussion* (D) *Limitations* (L), *Future Work* (FW)

In ungefähr 3/4 der Publikationen (72,8 %) ist die Sektion *Threats to Validity* vorhanden. In 41,8% dieser Publikationen wird diese Sektion in interne, externe und construct Validität aufgeteilt und in 20,9% in interne und externe Validität. Jedoch wird meistens mehr auf die interne Validität eingegangen und beschrieben welche Maßnahmen getroffen wurden, um diese zu stärken und um Fehler, Bias oder Effekte zu vermeiden. Die externe Validität wird meist nur knapp beschrieben und es werden sehr wenig Aussagen zur Verallgemeinbarkeit getätigt.

Es fällt auf, dass anstatt über die Verallgemeinbarkeit zu diskutieren, die Studie und das Vorgehen erneut beschrieben wird. In vielen Publikationen (39,5%) wird auf die Evaluation verwiesen, wie auf die Popularität der genutzten Benchmark ¹ und die Auswahl der evaluierten Testsubjekte. In 8 Publikationen (18,6 %) wird auf die Diversität oder auf die Limitierung der Teilnehmenden verwiesen. ²

	#
Aufteilung in internal, external und construct validity	18
Aufteilung in internal und external validity	9
Beschreibung wie die interne Validität bestärkt wird	7
Sektion: <i>Threats to Validity</i>	43
Keine Sektion <i>Threats to Validity</i> vorhanden	16
Limitations	15
Future Work	24
Kein Fokus auf Verallgemeinbarkeit	2
Keine Stellungnahme zur Verallgemeinbarkeit	11
Behauptung: erstmalige Studie	26
Verweis auf der Diversität / Limitierung der Teilnehmenden	8
Verweis auf andere Studien (scope, scale)	2
Verweis auf eigenen Scope, Scale	7
Verweis auf Vorgehen, Beschreibung der Studie	7
Verweis auf Evaluation	17
Verweis auf Implementierung	3

Table 4.2: Analyse der Diskussion zur Verallgemeinbarkeit

¹"Our use of this well-known and widely-used benchmark of real-world defects aims to ensure our results generalize" - "*Causal Testing: Understanding Defects' Root Causes*" von Brittany Johnson, Yuriy Brun und Alexandra Meliou [a087]

²"To bolster these observations we subsequently used a more diverse interview sample, which included participants from both large and small employers." - "*A Tale from the Trenches: Cognitive Biases and Software Development*" von Souti Chattopadhyay, Nicholas Nelson, Audrey Au, Natalia Morales, Christopher Sanchez, Rahul Pandita und Anita Sarma [a654]

In zwei Publikationen wird ausgesagt, dass der Fokus der Studie nicht auf die Verallgemeinbarkeit liegt ³ und in 11 Publikationen (18,6%) wird ausgesagt, dass die Ergebnisse möglicherweise nicht generalisierbar sind. ⁴ In diesen Aussagen wurden zwar Kontextfaktoren gefunden bezüglich der möglichen Verallgemeinbarkeit. Jedoch werden wenig Aussagen über diese Kontexte gemacht und außerdem wird nicht über die Verallgemeinbarkeit der Ergebnisse auf diese Kontexte argumentiert.

Des Weiteren fällt auf, dass in mehr als 1/4 der Publikation (27,1 %) die Sektion *Threats to Validity* nicht vorhanden ist. Da Kontextfaktoren auch in anderen Sektionen vorkommen können, wurde untersucht, ob neben der *Conclusion*, Sektionen vorhanden sind, um die Begrenzungen der Arbeit und die zukünftige Arbeit zu erläutern. In 10 der 16 der Arbeiten (62,5%) sind solche Sektionen oder Passagen vorhanden. Im Vergleich mit den Publikationen mit *Threats to Validity*, welcher in Tabelle 4.3 dargestellt ist, ist ein Unterschied in der Häufigkeit der Sektion *Limitations* bemerkbar. Es kann daher geschlussfolgert werden, dass in Publikationen ohne *Threats to Validity* mehr über die Begrenzungen der Arbeit diskutiert wird, statt über die Validität und die Verallgemeinbarkeit.

	#	<i>Limitations</i>	<i>Future Work</i>	L + F
<i>Threats to Validity</i> vorhanden	43	3 (6,9%)	12 (27,9%)	6 (13,95%)
<i>Threats to Validity</i> nicht vorhanden	16	4 (25%)	4 (25%)	2 (12,5%)
insgesamt	59	7 (11,8%)	16 (27,1%)	8 (13,5%)

Table 4.3: Analyse der Vorkommen von Sektionen zu *Threats to Validity*, *Limitations* (L) und *Future Work* (F)

Es ergibt sich eine Schnittmenge von 27 Publikationen (45,7%), in denen nicht oder nur oberflächlich auf die Verallgemeinbarkeit eingegangen wird, sodass Lücken in der Diskussion der Verallgemeinbarkeit bestehen. Lediglich in 25 Publikationen (42,3%) wurden Kontexte bezüglich der Verallgemeinbarkeit gefunden.

³"While desirable, generalizability was not the main focus of this study" - "A Tale from the Trenches: Cognitive Biases and Software Development" von Souti Chattopadhyay, Nicholas Nelson, Audrey Au, Natalia Morales, Christopher Sanchez, Rahul Pandita und Anita Sarma [a654]

⁴"Our findings might not be generalizable to other systems." - "Towards the Use of the Readily Available Tests from the Release Pipeline as Performance Tests. Are We There Yet?" von Zishuo Ding, Jinfu Chen und Weiyi Shang [b435]

4.1.4 Details

4.2 Taxonomie

5 Evaluation

Im Folgendem werden Schlussfolgerungen basierend auf den Artefakten des vorherigen Kapitel 4 gezogen und die Recherchefragen, welche in Kapitel 3 formuliert wurden, werden beantwortet. Des Weiteren werden nicht nur die Artefakte evaluiert, sondern auch das Vorgehen.

5.1 Schlussfolgerungen

5.2 Beantwortung der Recherchefragen

5.3 Evaluation des Vorgehen

5.4 Evaluation der Kontexte und der Taxonomie

6 Ergebnisse

Im Folgendem werden die Ergebnisse aus den vorherigen zwei Kapitel zusammengefasst und es erfolgt eine Abgrenzung der Masterarbeit.

6.1 Zusammenfassung

6.2 Abgrenzung

7 Abschluss

In diesem Kapitel werden abschliessend die Ergebnisse aus dem vorherigen Kapitel diskutiert und es wird ein Ausblick auf weitere Arbeiten gegeben.

7.1 Diskussion der Ergebnisse

7.2 Ausblick

8 References

- [1] **Tore Dybå, Dag I.K. Sjøberg, and Daniela S. Cruzes.** “What Works for Whom, Where, When, and Why? On the Role of Context in Empirical Software Engineering”. In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM '12. Lund, Sweden: Association for Computing Machinery, 2012, pp. 19–28. ISBN: 9781450310567. DOI: 10.1145/2372251.2372256. URL: <https://doi.org/10.1145/2372251.2372256>.
- [2] **Andrew Forward and Timothy C. Lethbridge.** “A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering”. In: *Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds*. CASCON '08. Ontario, Canada: Association for Computing Machinery, 2008. ISBN: 9781450378826. DOI: 10.1145/1463788.1463807. URL: <https://doi.org/10.1145/1463788.1463807>.
- [3] **Gary Johns.** “The Essential Impact of Context on Organizational Behavior”. In: *Academy of Management Review* 31 (Apr. 2006). DOI: 10.5465/AMR.2006.20208687.
- [4] **Kai Petersen and Claes Wohlin.** “Context in Industrial Software Engineering Research”. In: *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. ESEM '09. USA: IEEE Computer Society, 2009, pp. 401–404. ISBN: 9781424448425. DOI: 10.1109/ESEM.2009.5316010. URL: <https://doi.org/10.1109/ESEM.2009.5316010>.
- [5] **Lutz Prechelt.** *Four presumed gaps in the software engineering research community's knowledge*. 2019. arXiv: 1911.09971 [cs.SE].