Blekinge Institute of Technology
Doctoral Dissertation Series No. 2025:03
ISSN 1653-2090
ISBN 978-91-7295-496-0

Good-Enough Requirements Engineering

Julian Frattini

DOCTORAL DISSERTATION
for the degree of Doctor of Philosophy at Blekinge Institute of Technology to be publicly defended on 2025-02-28 at 13:00 in J1630, Valhallavägen 1, 37140 Karlskrona

Supervisors
Prof. Daniel Mendez, Blekinge Institute of Technology, Sweden, and fortiss GmbH, Germany
Dr. Michael Unterkalmsteiner, Blekinge Institute of Technology, Sweden

Faculty Opponent
Prof. Fabiano Dalpiaz, Utrecht University, Utrecht, Netherlands
Grading Committee
Prof. Per Runeson, Lund University, Sweden
Prof. Natalia Juristo, Polytecnic University of Madrid, Spain
Dr. Jennifer Horkoff, Chalmers University of Technology, Sweden

Abstract

Background: High-quality requirements are considered crucial for successful software development endeavors as the requirements' purpose is to inform subsequent activities like implementation or testing. Requirements quality defects have been shown to incur significant costs for remediation, scaling up even to project failure. At the same time, the effort to improve the quality of requirements must be justified. Organizations developing software, therefore, need to understand when their requirements artifacts are of "good enough" quality, i.e., they need to be able to identify the optimum between over- and under-engineering.

Problem: The body of knowledge in requirements quality does not yet offer solutions that would allow organizations to identify that optimum due to three shortcomings: (1) there is no generally accepted, theoretical foundation to describe requirements quality that can serve as a basis to coordinate distributed research efforts and the synthesis of evidence in the field, (2) the scientific practice currently applied in the field is of limited rigor to draw reliable conclusions from existing empirical contributions, and (3) the field lacks empirical evidence that can be aggregated to form a holistic view of requirements quality. These are potential causes for the lack of adoption of requirements quality research in practice.

Goal: In this cumulative, publication-based thesis, we address these three shortcomings and aim to contribute to a more evidence-based approach to requirements quality research grounded in scientific theory.

Method: First, we develop a theoretical foundation by adopting and integrating existing software engineering theories. Second, we evaluate the state of the art of data analysis and open science in the field and provide guidelines to improve these practices. Third, we demonstrate the application of these guidelines and conduct a controlled experiment to contribute additional empirical evidence to the field.

Results: The resulting set of analytical theories specifies requirements quality and provides a structure for future empirical contributions. Our evaluation of the state of the art shows both the need for a common theoretical foundation as well as support for applying rigorous research practices. Our empirical studies contribute to these needs and illustrate the complexity of the impact that requirements quality defects have on subsequent activities. Finally, we develop a method for the effective aggregation of empirical results.

Conclusion: Our theoretical, methodological, and empirical contributions help to coordinate a productive and constructive research agenda on requirements quality that is based on evidence and grounded in theory. This allows for rigorous and practically relevant research that ultimately informs organizations on how to engineer good-enough requirements.

Keywords: Requirements Engineering, Requirements Artifacts, Requirements Quality

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2025:03

Good-Enough Requirements Engineering

Julian Frattini

Doctoral Dissertation in Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN

Blekinge Institute of Technology
Department of Software Engineering

Blekinge Institute of Technology Doctoral Dissertation Series No. 2025:03
ISBN 978-91-7295-496-0
ISSN 1653-2090
urn:nbn:se:bth-27382

Printed in Sweden by Media-Tryck, Lund University, Lund 2025

Acknowledgements

This thesis would not have been possible without the continuous support and contributions of many extraordinary people, to whom I am deeply grateful.

First and foremost, I owe my deepest thanks to my two supervisors, Daniel Mendez and Michael Unterkalmsteiner. I have been privileged to be advised by two fantastic researchers who supported me with clear vision, endless guidance, and encouragement. They always found the right balance between supporting and challenging me. I am confident that there was no way of capitalizing more on these five years as a Ph.D. for my professional growth than through the supervision I have received from Daniel and Michael, and I can only hope to become a researcher with such a positive impact on my environment as them.

Additionally, I am grateful for the many fantastic colleagues I had the pleasure of working with. I am thankful for my fellow Ph.D. students who made the journey much more fun, like Ehsan Zabardast, Michael Dorner, Andreas Bauer, Lukas Thode, and Felix Jedrzejewski, the fantastic senior researchers, among which I am particularly grateful for Davide Fucci's endless support, and the good souls of our department, Svetlana Živanović, Anna Eriksson, and Monique Johansson. Finally, I am thankful to Michael Mattsson and Tony Gorschek for opening every door possible for me and enabling my research.

Beyond my direct colleagues, I am thankful for the great collaborators I had the pleasure of working with. Among these, I am particularly grateful for Jannik Fischbach and Lloyd Montgomery, two fellow researchers who greatly inspire me. This thesis would not have been possible without them paving the way and actively supporting me.

Furthermore, I owe thanks to the company partners of the SERT project who collaborated with us. Most importantly, I thank Parisa Yousefi from Ericsson Karlskrona, who has supported the idea of good-enough requirements engineering from the start and gave me insights into the real world of software engineering that would have otherwise been inaccessible.

Finally, I am grateful for my family, who supported me on this journey and ensured that I stayed sane. Most importantly, I thank my wife, Anja. Her unrelenting support from the start to the finish of my Ph.D. journey is why I prevailed even in the most frustrating times. Without her, many of the following pages would have been empty.

Preface

Our endeavor to understand, specify, and operationalize the concept of requirements quality under the leitmotiv "good-enough requirements engineering" has taken many turns and saw several research roadmaps drafted and discarded. This preface briefly retells the evolution of the research agenda from a procedural perspective, i.e., how our perspective evolved over time. The scientific, more holistic perspective, i.e., the coherent big picture that orders the containing contributions narratively, not procedurally, follows in the introductory chapter Chapter 1.

The original goal of the endeavor was to develop support for companies to determine whether or not their requirements artifacts are good enough. The journey started out by adopting the stance on requirements quality that was predominant in the research field, focusing on quality factors of requirements artifacts, e.g., passive voice, ambiguous pronouns, and many others.[1] However, while collecting existing requirements quality factors and structuring them in an ontology (see Chapter II), we noticed that the relevance of the factors proposed in the literature varied strongly. Most contributions to the field of requirements quality simply claimed that a quality factor is important and only very few made an—often unsystematic—effort to prove the relevance of their proposed factor in an empirical fashion. At the same time, both our own experience from company collaborations and recent, related work pointed towards a lack of trust in research results from the requirements quality field.

Although the original roadmap foresaw the extension of the collection of requirements quality factors and the attempt to automate their detection and remediation, this conundrum incentivized us to step back. The need for a theoretical foundation that ensures the *relevance* of contributions to the requirements quality research field emerged.

We drew inspiration from the adjacent research field of source code quality, which also studied the quality of artifacts from the software engineering context. This much more mature research field allowed us to adopt and further develop an activity-based requirements quality theory (see Chapter I). The theory dictates that the relevance of the previously collected requirements quality factors depends on how they *impact subsequent software development activities*. In other words: a quality factor of a requirements artifact (e.g., the use of passive voice) is only relevant if it has an effect on any activity that uses the requirements artifact (e.g., if deriving test cases

[1] See `http://reqfactoront.com/content/factors`.

from it would take more time). The potential of this theory shifted our focus away from quality factors, with which literature already abounds, onto these potentially impacted activities (see Chapter III).

With the theoretical foundation of requirements quality established (Chapters I to III), we planned to contribute empirical evidence to the field that subscribes to this foundation. However, reviewing the few empirical studies in the field exhibited several opportunities to also advance the *rigor* of commonplace scientific practices. Hence, we dedicated a significant amount of our time to meta-research. Particularly, we reviewed and tried to improve the availability of research artifacts from the requirements quality literature (Chapter IV) and the state of data analysis (Chapters V and VI) with methodological contributions.

With both the theoretical foundation of requirements quality established and reasonable advances to scientific practice contributed, we were finally confident of producing rigorous and relevant empirical evidence that contributed to the original goal. The resulting controlled experiment yielded insights into the impact of two particular requirements quality factors (passive voice and ambiguous pronouns) on one particular software development activity (domain modeling) (Chapter VII). It serves as a demonstration of both the adherence to the revised scientific practices, which ensures the rigor of the conclusions, and to the theoretical foundation, which ensures the relevance of the conclusion.

We realized that the amount of empirical studies required to produce a reliable body of knowledge about requirements quality was insurmountable within the scope of a single Ph.D. program. Hence, instead of producing one more piece of empirical evidence, we rather turned our attention to the challenge of enabling a constructive long-term, distributed community effort working toward one larger goal. The final contribution to this thesis (Chapter VIII), thus, proposes a research synthesis framework that allows integrating separate, evolving pieces of quantitative, empirical evidence to more general variance theories.

In hindsight, the original goal of this Ph.D. thesis turned out to be significantly more multi-faceted than originally assumed. While, at this point, we are unable to answer precisely when requirements are good enough, we hope that our theoretical foundation, methodological advice, and synthesis framework paved the way for rigorous and relevant research answering the question.

List of Papers

Paper I

Frattini, J., Montgomery, L., Fischbach, J., Mendez, D., Fucci, D., & Unterkalmsteiner, M. (2023). Requirements quality research: a harmonized theory, evaluation, and roadmap. Requirements engineering, 28(4), 507-520. DOI: 10.1007/s00766-023-00405-y.

Paper II

 Frattini, J., Montgomery, L., Fischbach, J., Unterkalmsteiner, M., Mendez, D., & Fucci, D. (2022, August). A live extensible ontology of quality factors for textual requirements. In 2022 IEEE 30th International Requirements Engineering Conference (RE) (pp. 274-280). IEEE. DOI: 10.1109/RE54965.2022.00041.

Paper III

 Frattini, J., Fischbach, J., Fucci, D., Unterkalmsteiner, M., & Mendez, D. (2024, June). Measuring the Fitness-for-Purpose of Requirements: An initial Model of Activities and Attributes. In 2024 IEEE 32nd International Requirements Engineering Conference (RE) (pp. 398-406). IEEE. DOI: 10.1109/RE59067.2024.00047.

Paper IV

Frattini, J., Montgomery, L., Fucci, D., Unterkalmsteiner, M., Mendez, D., & Fischbach, J. (2024). Requirements quality research artifacts: Recovery, analysis, and management guideline. Journal of Systems and Software, 112120.
DOI: 10.1016/j.jss.2024.112120

Paper V

Frattini, J., Fucci, D., Torkar, R., & Mendez, D. (2024, April). A second look at the impact of passive voice requirements on domain modeling: Bayesian reanalysis of an experiment. In Proceedings of the 1st IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering (pp. 27-33). DOI: 10.1145/3643664.3648211

Paper VI

Frattini, J., Fucci, D. & Vegas, S. (2024, October). Crossover Designs in Software Engineering Experiments: Review of the State of Analysis. In Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 482-488). DOI: 10.1145/3674805.3690754

Paper VII

Frattini, J., Fucci, D., Torkar, R., Montgomery, L., Unterkalmsteiner, M., Fischbach, J., & Mendez, D. (2024, November). Applying Bayesian Data Analysis for Causal Inference about Requirements Quality: A Controlled Experiment. Empirical Software Engineering. DOI: 10.1007/s10664-024-10582-1

Paper VIII

Frattini, J., Mendez, D., Fischbach, J., Fucci, D., & Unterkalmsteiner, M. (2025). Managing Variance Theories in Software Engineering. *Submitted to* Transactions on Software Engineering. arXiv: 2412.12634.

Author's contribution to the papers

The chapters of this compilation thesis are based on eight publications. Julian Frattini is the main author of all eight publications compiled in this thesis. As the main author, he took the main responsibility for conceptualization, methodology, software, formal analysis, visualization, validation, and writing of all contributions. Despite the author taking the main responsibility, the introductory Chapter 1 is written in plural form to emphasize the collaborative nature of all research endeavors. He and the co-authors describe their contributions to the chapters in detail, utilizing the contributor role taxonomy *CRediT*:

Paper I

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Lloyd Montgomery (Conceptualization, Methodology, Investigation, Writing: review & editing), Jannik Fischbach (Conceptualization, Methodology, Investigation, Writing: review & editing), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition), Davide Fucci (Conceptualization, Methodology, Writing: review & editing), Michael Unterkalmsteiner (Conceptualization, Writing: review & editing, Supervision).

Paper II

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Investigation, Resources, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Lloyd Montgomery (Conceptualization, Methodology, Investigation, Data Curation, Writing: review & editing), Jannik Fischbach (Conceptualization, Methodology, Investigation, Data Curation, Writing: review & editing), Michael Unterkalmsteiner (Conceptualization, Writing: review & editing, Supervision), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition), Davide Fucci (Conceptualization, Methodology, Writing: review & editing).

Paper III

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Investigation, Resources, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Jannik Fischbach (Conceptualization, Methodology, Investigation, Data Curation, Writing: review & editing), Davide Fucci (Conceptualization, Methodology, Writing: review & edit-

ing), Michael Unterkalmsteiner (Conceptualization, Writing: review & editing, Supervision), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition).

Paper IV

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Investigation, Resources, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Lloyd Montgomery (Conceptualization, Methodology, Resources, Writing: original draft, Writing: review & editing), Jannik Fischbach (Conceptualization), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition), Davide Fucci (Conceptualization, Methodology, Writing: review & editing), Michael Unterkalmsteiner (Conceptualization, Methodology, Writing: review & editing, Supervision).

Paper V

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Resources, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Davide Fucci (Conceptualization, Methodology, Validation, Formal Analysis, Writing: review & editing), Richard Torkar (Conceptualization, Methodology, Validation, Formal Analysis), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition).

Paper VI

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Davide Fucci (Conceptualization, Methodology, Validation, Writing: review & editing, Supervision), Sira Vegas (Conceptualization, Methodology, Validation, Resources, Writing: review & editing).

Paper VII

with contributions by Julian Frattini (Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Davide Fucci (Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Writing: review & editing), Richard Torkar (Methodology, Validation, Formal Analysis), Lloyd Montgomery (Conceptualization, Data Curation, Writing: review & editing), Michael Un-

terkalmsteiner (Conceptualization, Methodology, Writing: review & editing, Supervision), Jannik Fischbach (Conceptualization, Resources), Daniel Mendez (Conceptualization, Writing: review & editing, Supervision, Funding acquisition).

Paper VIII

with contributions by Julian Frattini (Conceptualization, Methodology, Formal Analysis, Investigation, Data Curation, Writing: original draft, Writing: review & editing, Visualization, Project administration), Jannik Fischbach (Conceptualization, Writing: review & editing), Daniel Mendez (Conceptualization, Methodology, Writing: original draft, Writing: review & editing, Supervision, Funding acquisition), Davide Fucci (Conceptualization, Writing: review & editing), Michael Unterkalmsteiner (Conceptualization, Writing: review & editing, Supervision).

Table of Contents

1 Introduction

1.1 Overview

Requirements artifacts—e.g., systematic requirements specifications, use cases, or user stories—are used as input to other activities of software development. For example, developers implement the functionality described in a use case and testers derive test cases from acceptance criteria of user stories. Therefore, the quality of requirements artifacts impacts subsequent software development activities [1]. For example, an *ambiguous* requirements specification may cause the subsequent activity of *implementing* the requirements to produce an incorrect solution [2]. Remediating this subsequent impact (i.e., re-implementing incorrect source code) often requires much more effort than remediating the cause (i.e., clarifying the ambiguous requirements specification) [3, 4].

At the same time, effort spent on improving the quality of requirements needs to be justified. Requirements artifacts are a means-to-an-end [1], and any effort that exceeds meeting this end can be considered a waste [5]. Consequently, companies aim to ensure a *good-enough* level of requirements quality that minimizes the risk of incurring this impact while also avoiding over-engineering the requirements specifications. Requirements quality research aims to support companies in attaining this good-enough level. To this end, requirements quality research is dedicated "to understand and define measurable attributes of requirements quality, to improve requirements quality through the creation of intervention techniques, and to evaluate those techniques/interventions." [6]. However, previous studies have noticed several shortcomings in the current state of requirements quality research [1, 7, 8] which impede its adoption in practice [9]. This thesis is dedicated to identifying existing shortcomings and addressing several of them to propel requirements quality research into a more rigorous and relevant trajectory.

This first chapter of the cumulative thesis introduces the reader to the overall research area, explains the overarching research endeavor in the scope of the thesis, and illustrates how the individual contributions in the subsequent chapters are connected to the endeavor. In this chapter, Section 1.2 introduces the fundamentals of requirements engineering, requirements artifacts, and requirements quality. Section 1.3 explains the gaps identified in the current state of research and practice, and Section 1.4 the goals and research questions in the scope of this thesis that address a subset of these gaps. Section 1.6 lists the individual publications included

in this cumulative thesis and how they contribute to achieving those goals. Finally, Section 1.8 critically reflects on the results, including implications, limitations, and potential future work, before we conclude in Section 1.9.

1.2 Background

The following subsections introduce the fundamental terminology of the research domain in which this thesis is embedded.

1.2.1 Requirements Engineering

Requirements engineering (RE) is the "systematic, iterative, and disciplined approach to develop explicit requirements and system specifications that all stakeholders agree on" [10]. As such, RE aims to explore and understand the *problem space* of a software development project (i.e., *why* and *what* to develop), but not the solution space [11] (i.e., *how* to develop a system). Still, researchers and practitioners often struggle to confine efforts into respective spaces [12], which results in solution-oriented requirements, i.e., requirements that do not describe the problem but rather already propose a solution. These solution-oriented requirements pose a significant risk as they entail a commitment to a solution without a full understanding of the problem to solve [13], which is one form of quality defect in a requirements artifact.

Traditional RE activities include requirements elicitation, analysis, specification, and validation and verification [14–16]. Regardless of the software process model employed during a software development project, some fundamentals of requirements and RE remain universally valid. This includes the aforementioned focus on the problem- instead of the solution space as well as the general process of obtaining requirements (i.e., elicitation), improving and documenting them (i.e., analysis and specification), ensuring that they reflect the original intentions (i.e., validation), and ensuring that the developed product or service meets those requirements (i.e., verification).

One source of confusion about requirements is that the established terminology refers to "requirements" as both the *needs or constraints* imposed by a stakeholder and *their physical manifestation* in artifacts (e.g., documentation) [10]. We explicitly refer to the physical manifestation as a "requirements artifact" and limit the meaning of "requirement" to a need or constraint to avoid confusion [17].

Because RE requires significant effort and its impact is difficult to trace precisely [18], practitioners often challenge the necessity of applying RE methods and how they are supposed to be executed. Several studies report practitioners' reluctance to commit effort to RE since they perceive it as a waste of time [19] or generally not constructive [20]. This happens despite multiple large-scale studies having shown that negligence of RE exhibits significant risk for the subsequent software develop-

ment process [18, 21].

1.2.2 Requirements Artifacts

Two major schools of thought exist in RE: activity orientation and artifact orientation. Activity orientation emphasizes the process of RE and prescribes a set of interconnected techniques and methods to achieve its goal [22]. Artifact orientation, on the other hand, emphasizes the artifacts and their relationships produced during the RE phase while remaining agnostic about how these artifacts are produced or used [23].

Requirements artifacts are defined as "a work product that is produced, modified, or used by a sequence of tasks that have value to a role" [24]. They are characterized by their physical representation, syntactic structure, and semantic content [24]. Artifacts may include more comprehensive software requirements specifications, as commonly seen in plan-driven software processes, and user stories, as seen in agile software processes. Artifacts are decomposable, i.e., one artifact may consist of several sub-artifacts. For example, a systematic requirements specification artifact may contain several sub-artifacts of the type use case.

In this thesis, we subscribe to artifact orientation and focus mainly on natural language (NL) requirements artifacts. Because the RE phase involves a heterogeneous set of stakeholders with varying levels of technical background and requirements artifacts have to be understood by all involved stakeholders, NL requirements artifacts have emerged as the most understandable and applicable syntactic structure [25]. While requirements artifacts of different syntactic structures—for example, specified using formal languages [26, 27], models [28], or other media like videos [29]—offer distinct benefits, NL remains the most prominent form of specifying requirements [21].

1.2.3 Requirements Quality

The quality of requirements impacts subsequent software development activities [30]. These impacts have been empirically investigated both at a high level, i.e., connecting practitioners' self-reported experiences and perceptions of requirements quality to problems including project success or failure [18] and at a lower level, i.e., connecting specific linguistic occurrences in requirements artifacts to time and budget overrun [31].

Within the paradigm of artifact-oriented RE, requirements artifacts carry the responsibility to communicate the requirements to subsequent software development activities. This renders requirements artifacts as eligible subjects to quality assurance (QA). Requirements quality research is dedicated to guiding this QA by understanding the impact that properties of requirements artifacts have on subsequent activities [6]. Traditionally, this manifests in the proposal of guidelines associating

specific linguistic patterns with good or bad quality and, hence, advocating for or against the usage of these patterns [32]. For example, the use of passive voice is often advised against in RE textbooks [33] given that it omits information and, consequently, negatively impacts subsequent activities like domain modeling [34].

A critical property of QA in RE is the phenomenon of scaling costs for defect removal. The longer a defect persists in software development, the more expensive it becomes to fix it [4, 35]. For example, an ambiguous requirements artifact might take a couple of hours to clarify with the relevant stakeholders, while an incorrect implementation built based on a misunderstanding of that requirements artifact may take several days to rework [36]. If that defect is only noticed after the product or service has already been deployed, then the cost of remediating it becomes even greater and may not only be paid in monetary resources but also in reputation and trust. A seminal study by Boehm et al. [3] estimated a cost increase by a factor of 10 per phase that the defect survives. This study is both dated and was conducted in a more plan-driven context, but there is no reason to assume that the fundamental principle of cost increase—regardless of the actual factor of exponentiation—has changed.

1.3 Gaps

Requirements quality research should support practitioners in deciding whether their requirements artifacts are good-enough. Achieving good-enough requirements entails finding an optimum between under- and over-engineering the requirements artifacts. As Fricker et al. summarize, "[i]nadequately specified requirements lead to ambiguity and misunderstandings that cause large corrective costs down the development road. However, too much detail and quality improvement retards the delivery of development results while also increasing specification costs and unnecessarily constraining the solution space." [5]. Traditional requirements quality research concerns itself with providing practitioners tools and methods to identify when the optimum of good-enough requirements engineering is reached. Yet, the current state of research and practice is subject to several shortcomings noted in previous research [2, 7, 8, 37], and elaborated in the subsequent chapters Chapters I to VIII. The following subsections Sections 1.3.1 to 1.3.3 summarize these shortcomings.

1.3.1 Gap 1: Insufficient Theoretical Foundation for Requirements Quality Research

A mature scientific discipline is governed and coordinated by a set of commonly accepted theories [38]. Depending on the purpose of the theories, these fulfill different roles in guiding the scientific practice. Gregor et al. differentiate four different primary purposes of theories [39]:

1. **Analysis and Description**: providing a description of the phenomena of interest and of the relationship between them

2. **Explanation**: explaining how, why, and when phenomena occur

3. **Prediction**: estimating what will happen in the future under certain conditions

4. **Prescription**: prescribing methods and structures for the utilization of knowledge in practice

In their role within a scientific discipline, analytic and descriptive theories frame the phenomena of interest and provide uniform terminology to communicate about them. Explanatory theories contribute a causal understanding of the interrelation of the phenomena. Predictive theories inform about potential consequences, while prescriptive theories guide the utilization of the procured knowledge in practice.

The scientific discipline of requirements quality lacks, so far, a common, sophisticated theoretical foundation [40]. Contributions to the field declare no reference to any overarching theory to the best of our knowledge. This results in several aspects of the discipline to diverge. For example, repeatedly studied phenomena like *requirements quality factors*, i.e., metrics evaluating the quality of requirements artifacts, are referred to with different names (e.g., requirements smell, requirements indicator, and others) [41]. Moreover, similar requirements quality factors are often described differently in separate studies, resulting in competing, incommensurable definitions [41]. Empirical studies about requirements quality also lack adherence to any explanatory or predictive theory that would put these phenomena into relation with each other and specify the context in which they hold. In the context of requirements quality research, this manifests as a lack of coherence when describing the impact that quality factors have, i.e., what consequences they cause [1]. All this terminological and conceptual heterogeneity makes the synthesis of individual studies to more general and valid conclusions impossible [42], constraining requirements quality research to a fragmented, incidental endeavor.

1.3.2 Gap 2: Immature Scientific Practice

Assuming that the discipline of requirements quality research receives a governing set of commonly accepted theories, the rigor of the applied research methodology determines the quality of scientific contributions. Contributions lacking rigor will provide little value to the scientific discipline regardless of their adherence to theoretical foundations. This necessitates the continuous development and evolution of empirical methods as seen by the ACM Empirical Standards [43] or scientific forums like the Empirical Software Engineering journal [1] and the Empirical Software

[1] `https://link.springer.com/journal/10664`

Engineering and Measurement conference series. [2] While reviewing literature in the scope of our studies, we encountered several shortcomings threatening the validity of contributions. Three of these shortcomings emerged as particularly significant to the research we conducted.

Lack of Adherence to Open Science Firstly, we identified a lack of adherence to open science principles [44]. While reviewing literature about requirements quality factors [41], we collected research artifacts connected to them. These research artifacts include data sets that were often described to be manually annotated, as well as tools that automatically detect and remove requirements quality factors [41]. However, the majority of research artifacts have become unavailable or have never been disclosed in the first place [45]. Availability of research artifacts is a necessary precondition for reproducibility [46]. Hence, the lack thereof inhibits the ability of a research community to independently review published results. Furthermore, the unavailability of research artifacts inhibits replication and their evolution.

Simplistic Statistical Tools The second identified shortcoming is the reliance on empirical studies on simple statistical tools for their data analysis. The requirements quality literature, just as the encompassing requirements engineering and software engineering, mostly resorts to out-of-the-box frequentist approaches null-hypothesis significance tests (NHSTs) [47]. These approaches reduce complex data to unreasonably restrictive, often binary, results (e.g., a p-value) [48]. Additionally, NHSTs are often applied without any consideration of causality and, therefore, merely represent associative, correlational inferences. This threatens the validity of the conclusions drawn from data within the scope of empirical studies.

Mismatch of Study Design and Data Analysis The third identified shortcoming is the mismatch between the study design and the way that the resulting data is analyzed. In particular, we noticed this mismatch when reviewing empirical studies that conducted an experiment with a crossover design, i.e., an experiment where every unit received every level of the treatment but in different orders [47]. The crossover design allows to control between-subject variability by studying differences between units rather than between treatment groups, but it also incurs several new threats to the validity of the results [49]. For example, if the units of analysis are human participants, a learning effect may affect the observed results as the experiment continues. To mitigate these threats, Vegas et al. proposed guidelines for the design and analysis of crossover-design experiments [50]. However, the adherence to these guidelines varies strongly, which undermines the validity of threat mitigation.

[2] `https://conf.researchr.org/series/esem`

1.3.3 Gap 3: Lack of Empirical Evidence

While the field of requirements quality does receive empirical contributions [6], it lacks—also in consequence of the above—contributions that adhere to a theoretical foundation, apply appropriate scientific practices and provide insight into the impact of factors of requirements quality. A systematic study of empirical evidence about requirements quality revealed that most studies propose approaches and tools to *improve* requirements quality, but only few attempt to actually *define* and *understand* quality and its impact [6]. Studies proposing approaches and tools to improve requirements quality are popular in the natural language processing for requirements engineering (NLP4RE) domain [51] as shown by the excessive number of tools proposed in the recent years [52]. However, these tools lack empirical evidence for the relevance of the quality factors that they detect or remove. Otherwise, the tool does not perform a meaningful task regardless of its de facto accuracy. Requirements quality research needs to produce more empirical evidence about impact to contribute relevant guidance both for researchers aiming to build automatic solutions and for practitioners aiming to ensure the quality of their requirements artifacts.

1.4 Goals and Research Questions

This thesis is dedicated to addressing the gaps 1-3 described in Section 1.3. To this end, we aim to achieve the following goals Goals 1-4 described in Sections 1.4.1 to 1.4.4. Every goal is further specified in terms of associated research questions. Figure 1.1 visualizes the relationships between goals, gaps, and contributions (presented in Section 1.6) in the scope of this thesis.

1.4.1 Goal 1: Theoretical Foundation for Requirements Quality Research

We aim to provide the requirements quality research domain with a theoretical foundation that describes both the relevant constituents of requirements quality as well as the relationships between them. This theoretical foundation—consisting of several theories of different types fulfilling different purposes [39]—shall provide a frame for any future research endeavor and place them into a clear relationship both to the studied phenomena and to other contributions. We assign priority to *analytic theories* to establish a conceptualization of the phenomena of interest and a shared vocabulary for them. This entails one analysis theory taking the form of an ontology and describing which general concepts are relevant to requirements quality and how they interact, e.g., requirements artifacts, quality factors, and affected activities [2]. It further entails analysis theories taking the form of taxonomies and classification structures to collect the instances of each concept, e.g., which requirements quality

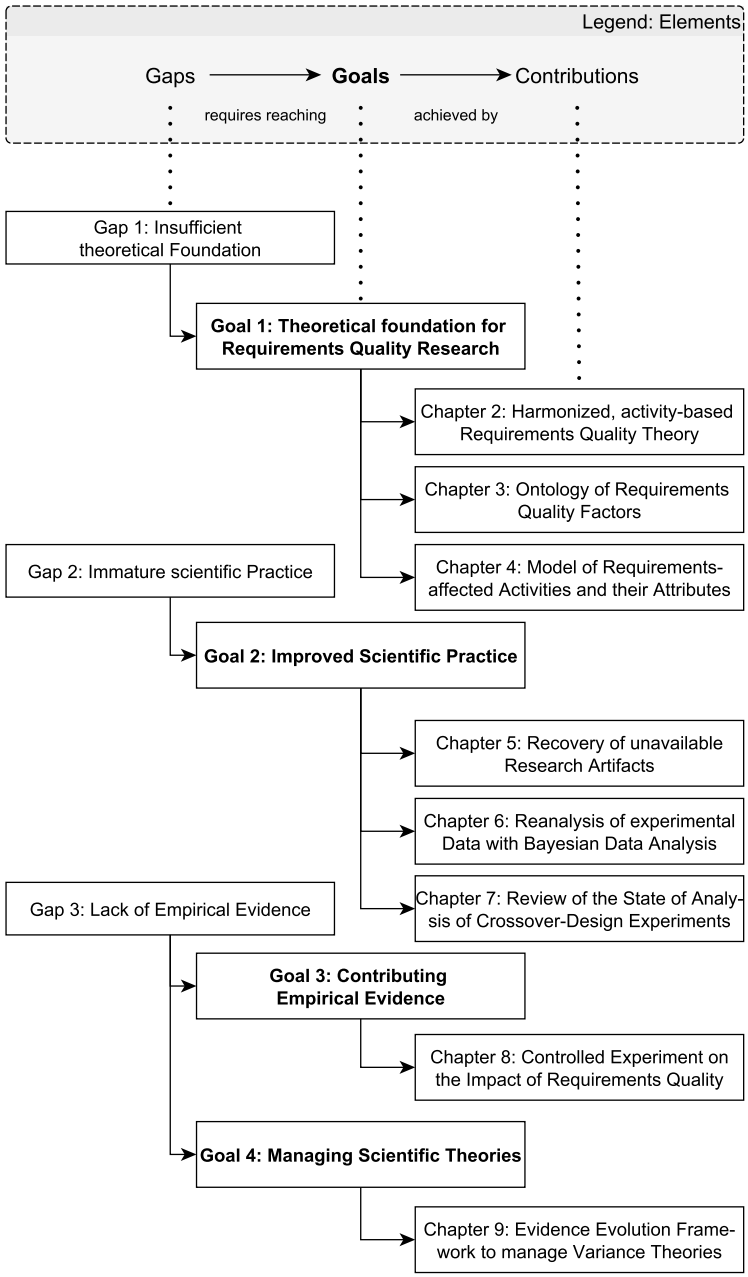


Figure 1.1: Gaps, goals, and contributions of this thesis

factors are discussed in literature [41]. Publicly disclosing all material used to specify these theories allows them to evolve organically with emerging research from the research community. Achieving this goal addresses gap 1 (Section 1.3.1) by answering the research questions stated in Table 1.1.[3]

Table 1.1: Research questions in the scope of Goal 1 (Research questions marked with an asterisk (*) are explicitly stated as such in the respective chapters, the others are imputed for the sake of the narrative.)

Chapter	ID	Research Question
Chapter I	RQ1.1	What is requirements quality?
	RQ1.2*	How are the concepts of the requirements quality theory reported in requirements quality literature?
Chapter II	RQ2.1	What is the structure of requirements quality factors?
	RQ2.2	Which requirements quality factors are reported in literature?
Chapter III	RQ3.1*	Which software development activities are affected by requirements artifacts?
	RQ3.2*	By which attributes are requirements-affected activities evaluated?

1.4.2 Goal 2: Improved Scientific Practice

We aim to survey and critically reflect on current scientific practices in the requirements quality research domain and to propose improvements that increase the rigor and relevance of future contributions. Lack of rigor in applying research methods will render future contributions—despite adherence to theoretical foundations—unreliable and prevent the requirements quality research domain from advancing. Hence, we dedicated a significant part of our research efforts not only to advancing the content of requirements quality, but also the scientific practice by which the latter is produced. This goal is not constrained to the requirements quality research domain. Though we draw motivation for it from the requirements quality literature and our claims about the observed shortcomings might not generalize to other fields of SE research, we are confident that other fields might similarly benefit from the advances. Achieving this goal addresses gap 2 (Section 1.3.2) by answering the research questions stated in Table 1.2.

Table 1.2: Research questions in the scope of Goal 2

Chapter	ID	Research Question
Chapter IV	RQ4	What is the state of artifact availability in requirements quality research?
Chapter V	RQ5	How do more rigorous methods for statistical causal inference revise previous claims about the impact of requirements quality?
Chapter VI	RQ6	To what extent do SE experiments adhere to data analysis guidelines?

RQ5 is answered by the specific example of the impact that the use of passive voice in functional requirements specifications has on the domain modeling activity [34]. For more rigorous methods for statistical causal inference, we chose the use of a framework for statistical causal inference [53, 54] and Bayesian data anal-

[3]The ID numbering system is only valid throughout this Chapter 1 to put them into relation.

ysis [55]. We answer RQ6 for experiments employing a crossover design [47] and assess their adherence to the analysis guidelines by Vegas et al. [50].

1.4.3 Goal 3: Contributing Empirical Evidence

We aim to contribute empirical evidence of our own to the research domain of requirements quality. The observed lack of empirical evidence about the understanding of requirements quality [6] necessitates empirically studying the phenomena in different real-world contexts. An important aspect of this goal is not only the contribution of evidence but also the demonstration of *how* to contribute evidence *when subscribing to* the theoretical foundation. This way, studies guide future contributions to adhere to the theories that form the theoretical foundation of the research domain, which ensures their coherence and homogeneity. Achieving this goal addresses gap 3 (Section 1.3.3) by answering the research questions stated in Table 1.3.

Table 1.3: Research questions in the scope of Goal 3

Chapter	ID	Research Question
Chapter VII	RQ7.1	To what extent do quality defects in NL requirements specifications impact subsequent activities?
	RQ7.2	Do context factors influence this impact of quality defects on activities?

We answer RQ7.1 and RQ7.1 by the specific example of the impact of passive voice and ambiguous pronouns on the domain modeling activity.

1.4.4 Goal 4: Managing Variance Theories

Achieving the aforementioned goals will address the identified gaps but evoke a new challenge. Assuming that our contributions allow for further empirical evidence (goal 3) following more rigorous scientific practices (goal 2) based on a common, theoretical foundation (goal 1) in the research field of requirements quality, the potential to integrate these individual contributions into larger, more valid conclusions emerges. To anticipate this development and facilitate an effective management of evidence-based variance theories and, thus, to allow for the scientific community to advance based on a more coherent body of scientific knowledge, we extrapolate a fourth goal. We aim to provide the requirements quality research domain with support for synthesizing multiple pieces of empirical, quantitative evidence to more generally valid variance theories. Achieving this will aid the requirements quality research community to direct their collaborative effort toward a common, greater goal.

Table 1.4: Research questions in the scope of Goal 4

Chapter	ID	Research Question
Chapter VIII	RQ8	How should a research community synthesize empirical, quantitative evidence to produce valid variance theories?

1.5 Methods

We employ the research methods detailed in Table 1.5 to address the previously stated research questions. We justify and describe all research methods in detail in each respective chapter where they were applied. The subsections Sections 1.5.1 to 1.5.4 summarize only non-conventional choices with an impact on the overall thesis.

Table 1.5: Applied approaches per chapter and research question

Chapter	RQ	Approach
Chapter I	RQ1.1	Theory adoption [56]
	RQ1.2	Survey [57]
Chapter II	RQ2.1 & RQ2.2	Taxonomy development [58]
Chapter III	RQ3.1 & RQ3.2	Literature review [59], case study [60], thematic synthesis [61]
Chapter IV	RQ4.1	Artifact recovery analysis [45]
	RQ4.2 & RQ4.3	Bayesian data analysis [62]
Chapter V	RQ5	Reanalysis [63], Bayesian data analysis [62]
Chapter VI	RQ6	Forward snowball sampling [64]
Chapter VII	RQ7.1 & RQ7.2	Controlled experiment [47, 50], conceptual replication [63], Bayesian data analysis [62]
Chapter VIII	RQ8	Constructive research, focus group

1.5.1 Theory Adoption

Constructing theories is the main way of assembling and refining general knowledge [38] and the presence and use of theories are often seen as an indicator of a scientific discipline's maturity [65]. A "[t]heory provides explanations and understanding in terms of basic concepts and underlying mechanisms, which constitute an important counterpart to knowledge of passing trends" [66]. To achieve goal 1, the development of a requirements quality theory is necessary. While SE research is often not considered rich in theory [67, 68], a common method applied in the rare cases of theory development is grounded theory [69]. However, we opt to obtain our central analytic theory [39] via the less common *theory adoption* and two supporting analytic theories using taxonomy development [58]. A theory can be adopted if the phenomena of the original theory are consistent with the phenomena in the target discipline [56]. In the case of requirements quality, we were able to draw heavy inspiration from the field of software quality [70] as previously noted by Femmer et al. [2]. Software quality research shows parallels to requirements quality research in that both disciplines aim to assess the quality of software artifacts [24] for subsequent activities. The maturity of the software quality discipline [71–73] allowed us to adopt the theory for requirements quality.

1.5.2 Literature Review

Several of our goals require a review of relevant literature. For example, in Chapter III, we survey controlled experiments from the RE literature to determine the activities in which requirements artifacts can be involved as one of our data sources. In several cases [40, 41, 74], we were able to reuse an existing set of relevant primary studies from a previous literature review where the search criteria matched ours [6]. However, in the cases where we had to employ a search strategy of our own, we deviated from the de-facto standard in SE literature, a query-based database search following the guidelines of Kitchenham et al. [75]. Instead, we opted to employ the survey method proposed by Sjøberg et al. [59], which proposes to make a pre-selection of relevant venues (i.e., journals and conferences) and query these specifically. While this sacrifices recall, it constrains the large number of false positives with which a standard keyword-based search would have rendered the search strategy unusable.

1.5.3 Bayesian Data Analysis

Several of our research questions require the application of inferential statistics. SE researchers most commonly apply simple frequentist tools like analysis of variance (ANOVA) for this task, the simplest representative being the Student's t-test. However, these frequentist approaches are not only mostly void of any explicit causal considerations, but also reduce the complex data to unnecessarily narrow statistics (e.g., a p-value) [48]. Hence, we instead utilize Bayesian data analysis (BDA) within a framework for statistical causal inference [62, 76]. The framework for statistical causal inference provides a systematic way of dealing with confounders and reducing bias in a data analysis [54, 77]. The use of BDA ensures transparency of statistical assumptions and preserves any uncertainty inherent to the data [62]. These properties have led to a call for a paradigm shift from frequentist to explicitly causal Bayesian methods in several disciplines, including software engineering [48, 78]. However, BDA has not yet seen significant adoption in software or requirements engineering research due to its steep learning curve [54]. Chapter VII contains an extensive demonstration of applying BDA in RE, though this thesis does not claim to provide a pedagogical introduction to the topic. For this, we refer the interested reader to adequate textbooks [62] and more elaborate guidelines [48, 54, 79, 80].

1.5.4 Replication

Finally, we contribute replications in the scope of this thesis [81, 82]. In SE research, Baldassarre et al. [63] provided a commonly accepted distinction between types of replications, which include internal, external, close, differentiated, and conceptual replications. The latter type occurs when only the hypothesis of the original experiment and replication are similar while all relevant elements of experimental design

(i.e., site, experimenters, apparatus, operationalization, and population) differ [83]. Conceptual replications are often dismissed in SE research, as the number of changed elements makes it impossible to trace disagreeing results to a single change [84]. However, we avoid dismissing conceptual replications categorically, as any study where the outcome would be considered diagnostic evidence about a claim from prior research is a type of replication [85]. In this case, conceptual replications can be particularly valuable if they produce similar results, as they strengthen the external validity of the original claims regarding all elements of experimental design. Finally, we also conduct a reproduction and a reanalysis in Chapter V, i.e., using existing data from a previous study, we investigate the hypothesis with the same analysis method (reproduction) and a different analysis method (reanalysis). The latter is sometimes also referred to as a test of robustness [85] as it assesses whether different analysis methods produce the same results. Both reproductions and reanalyses are, just as conceptual replications, rather rare in SE research

1.6 Contributions

In the scope of this thesis, we aim to provide three kinds of contributions. *Theoretical contributions* expand the theoretical foundation of the requirements quality research field (reaching goal 1). *Methodological contributions* advance the research methods (reaching goal 2). *Applications* demonstrate the usability of the two aforementioned contributions in practice (reaching goals 3 and 4). Each type contains several contributions. Each contribution corresponds to one publication and is represented in a separate chapter of this thesis. In Figure 1.1, contributions are listed on the right. Their grouping by type of contribution corresponds to the three gap statements on the left.

1.6.1 Theoretical Contributions

In Chapter I, we develop an analytic theory of requirements quality that serves as the foundation for requirements quality research. We adopted this theory from established software quality theories [71, 73], contextualized it for RE [1], and refined it with further developments [86, 87]. Most significantly, the theory emphasizes an *activity-based* perspective of requirements quality [2], i.e., it postulates that the quality of a requirements artifact depends on how it influences the activities in which this artifact is used in the subsequent software development process [30]. Additionally, our theory emphasizes the influence of *context* [87]. Requirements quality is not universal and highly depends on the involved people, the developed product, the application domain, and many other factors that need to be respected when determining whether a requirements artifact can be considered good or bad [88]. Figure 1.2 visualizes the main concepts of the harmonized requirements quality theory [40]. This re-

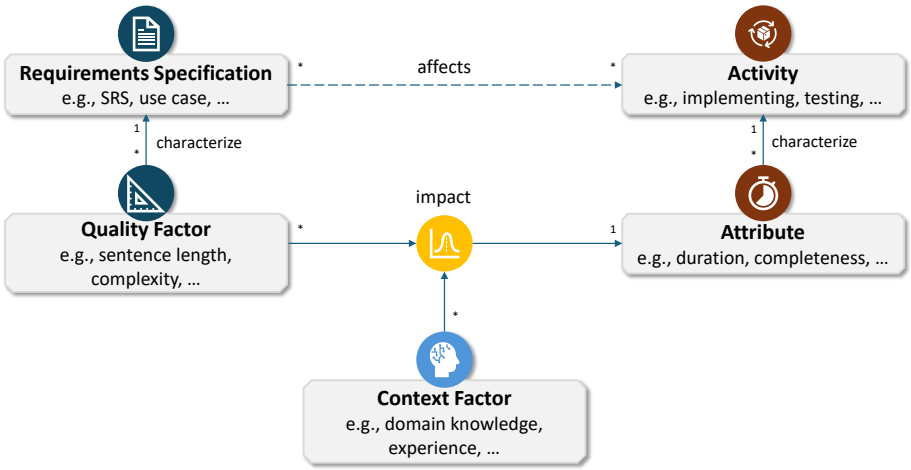


Figure 1.2: Core Concepts of the Activity-based Requirements Quality Theory

quirements quality theory defines the fundamental constructs and relationships [38] that can be used to specify prediction theory [39] (also known as a variance theory [89]), i.e., a theory predicting what will happen without explaining why.

In Chapters II and III, we develop two distinct classification schemes for the two major elements of the requirements quality theory [40]: requirements quality factors and requirements-affected activities and their attributes. Requirements quality factors (left side of Figure 1.2) represent properties of requirements artifacts and are a common concept in requirements quality literature [6]. Factors like sentence length [90], passive voice [34], and ambiguous pronouns [91] have been explored by multiple studies in the literature because researchers attribute (often negative) consequences to them. For example, the use of passive voice is suggested to challenge subsequent activities like modeling [34] and development [33]. Requirements-affected activities (right side of Figure 1.2) are those subsequent activities that use requirements artifacts as input, e.g., implementing or testing [2]. Their attributes are their measurable properties, e.g., duration or completeness. Both types of properties are often used in requirements quality literature, but there is no systematic overview of them. In these contributions [41, 92], we initiate a systematic classification of existing requirements quality factors (Chapter II) and requirements-affected activities and their attributes (Chapter III). These classifications serve as analysis theories [39] aimed at describing and conceptualizing the constructs relevant to the previously mentioned core analytic theory [40].

1.6.2 Methodological Contributions

In Chapter IV, we address the issue of unavailable research artifacts in requirements quality publications. Research artifacts like data sets and implementations are a vital contribution to the field [51]. Data sets serve as benchmarks for new tools and encode the ground truth about requirements quality phenomena, e.g., by annotating quality issues in requirements specifications. Implementations serve as actionable tools that can be applied in practice to transfer the knowledge generated by research to industry. However, many artifacts presented in publications become unavailable over time or have never been available [45]. In this contribution [74], we conduct an artifact recovery initiative to improve the availability of research artifacts. We then analyze these artifacts to gain insights into the reasons for artifact (un-) availability Finally, we develop concise guidelines to increase the community's awareness of open science practices and, thus, improve the availability of future research artifacts.

In Chapter V, we address the issue of simplistic data analyses employed in SE publications. Most analyses of quantitative data employing inferential statistics are limited to simple, implicit hypotheses (consisting of only one independent and one dependent variable), which are tested via out-of-the-box frequentist methods like null-hypothesis significance tests [47]. These analyses lack both an explicit causal framework and sophisticated statistical methods. In this contribution [81], we re-analyze a controlled experiment [34] by employing both an explicit framework for statistical causal inference and Bayesian modeling [62] to revise the claims of the original publication.

In Chapter VI, we address the issue of analyzing a complex type of controlled experiment: the crossover-design experiment. In this particular design, all levels of the treatment are applied to every experimental unit but in different orders [50]. This way, every participant acts as their own control group and between-subject variance can be factored out of the analysis. However, the design incurs new threats to the validity of the conclusions and requires more attention during the data analysis [49]. Vegas et al. [50] have provided explicit guidelines to counteract these threats. In this contribution [93], we survey publications citing the guidelines by Vegas et al. and assess the degree to which these publications adhere to the guidelines.

1.6.3 Applications and Transfer

In Chapter VII, we address the scarcity of empirical evidence in the requirements quality research domain. To this end, we conduct a conceptual replication of the previously re-analyzed controlled experiment [34]. We extend the experiment by investigating not only the impact of passive voice but also of ambiguous pronouns [91], and extend the sampling strategy to involve practitioners. Furthermore, we employ a crossover design to account for between-subject variance [50] and conduct Bayesian data analysis for more sophisticated statistical insights [48, 62]. In this contribu-

tion [82], we demonstrate an advanced approach for generating empirical evidence about requirements quality.

Finally, in Chapter VIII, we address the anticipated issue of synthesizing quantitative, empirical evidence to obtain more valid variance theories. To this end, we first define empirical, quantitative evidence as a tuple consisting of a causal hypothesis, collected data, and an appropriate analysis method. Then, we define a framework specifying the relationships between two pieces of empirical, quantitative evidence in terms of three types of evolution listed in Table 1.6. Every type of evolution is defined by which part(s) of the original piece of evidence it changes. For example, a *replication* applies the same analysis method under the same causal hypothesis to a new data set. Depending on the type of evolution, the new piece of evidence strengthens different aspects of the validity of the overall claim. For example, a replication that comes to the same conclusion on a different data set improves the external validity of the hypothesis, as it is shown to hold in a different context.

Table 1.6: Types of evolution of empirical, quantitative evidence

Type	Hypothesis	Data	Method	Conclusion
Replication	same	new	same	Improved external validity
Revision	new	new (if necessary)	same	Improved internal validity
Reanalysis	same	same	new	Improved conclusion validity

This framework extends the practice of research synthesis in SE which is currently mostly limited to meta-analyses of replications [94]. We apply the framework to synthesize previous research on requirements quality [34, 81, 82] to demonstrate how it can be used to obtain more valid variance theories.

1.7 Errata

Discussions after the publication of the individual contributions that compose the chapters—for example, in the scope of presentations at conferences—have led us to reconsider some formulations and framings. In the following, we briefly discuss all errata we are aware of.

Chapter I classifies the requirements quality theory as an explanatory and prescriptive theory [39]. In hindsight, we argue that it is neither. The theory is not explanatory as it does not yet explain phenomena, which would require properly explaining the reason for the relationships proposed in the requirements quality theory [40]. Furthermore, we exercise caution in calling the theory prescriptive, as it lacks any procedural guidance on how to apply the theory. While this is a future goal, as we later discuss in Section 1.8.4, we constitute that the requirements quality theory does not yet deserve classification as either an explanatory or prescriptive theory. Rather, the requirements quality theory should be understood as an *theory for analysis and understanding* [39], as it postulates general relationships between con-

cepts on a meta-level [89]. The theory primarily serves to enable *prediction theories* about specific quality factors and their impact on activities and their attributes. Our demonstration of applying the theory in an empirical study about the impact of passive voice and ambiguous pronouns on domain modeling [82] is an early example of a prediction theory, as it estimates how the domain modeling activity will be affected by the requirements quality factors without explaining why.

Chapter III introduces a terminological inconsistency between requirements and requirements artifacts. The model of requirements-affected activities and their attributes aims at collecting common activities performed once the requirements have been elicited and specified [92]. Technically, the population of interest are activities affected by *requirements artifacts*, not by *requirements*, and the model should be called "a model of requirements *artifact*-affected activities and their attributes."

Chapters V and VII present the application of Bayesian data analysis to RE phenomena. Both contributions conflate the approach of Bayesian data analysis and statistical causal inference [54]. Due to the recency of Bayesian methods in SE research [48] and the terminological confusion surrounding the early adoption [80], there is no clearly established and commonly accepted relationship between these statistical concepts. After further revision of literature from other fields, we agree that a better framing would be that Bayesian data analysis is a method for statistical causal inference and that the two approaches are not completely disjoint [62]. We attempted to remediate the terminological confusion in Chapter VIII.

1.8 Discussion

In the following sections, we discuss the contributions within the scientific and practical context. Section 1.8.1 outlines the anticipated implications for research and Section 1.8.2 the implications for practice. Section 1.8.3 acknowledges limitations and Section 1.8.4 presents aspired future work to address these limitations.

1.8.1 Implications for Research

1.8.1.1 Implications on Disciplinary Culture

Our research endeavor draws near the concept of *research paradigms* popularized by Thomas Kuhn [95]. Placing the work of this thesis in terms of Kuhn's framework shows how we understand our work in the larger context of the evolution of our scientific field. We briefly explain the concepts introduced by Kuhn and apply them to our endeavor to outline the anticipated implications of the research in the scope of this thesis on the research culture in the field of requirements quality.

Central to Kuhn's proposal is the concept of a research *paradigm*, which constitutes three components [95]:

1. general theoretical assumptions and laws
2. the techniques for their application that the members of a particular scientific community adopt
3. general methodological prescriptions (e.g., that any serious attempt of contribution should match the paradigm to the real-world context)

While we are not aware of any explicit attempts at defining a paradigm in the field of requirements quality research or, for that matter, software engineering research, systematic reviews of the field [6, 40, 41] hint at an implicit paradigm that emerged through publication patterns. The general theoretical assumptions that most of the contributions to the field share are that the quality of requirements artifacts matters and companies require support in the detection and removal of quality defects. The techniques for their application focus mainly on the definition of requirements quality factors [41] and the development of tools to detect violations against them [52]. Methodological prescriptions are limited and mostly implicit, but many contributions follow a similar pattern containing the following:

- the proposal of a requirements quality factor,
- a mostly non-empirical, often anecdotal justification of its relevance,
- the annotation of a data set with instances of violations against that quality factor, and
- the architecture and evaluation of a tool detecting and/or remediating these violations.

Methodological support for some of these steps exist, e.g., guidelines for the evaluation of tools [96], but are scarce and rarely connected to the overall paradigm. Because the theories governing the current paradigm largely ignore subsequent impact, the paradigm is not equipped to produce conclusions about it.

Thomas Kuhn defines an *anomaly* as a "puzzle within a paradigm that resists resolution" [95], i.e., a phenomenon that cannot be explained within the current paradigm. The anomaly that the current paradigm of requirements quality research experiences is the fact that—despite the general theoretical assumption that requirements artifact quality matters—companies do not seem to uptake the results of the requirements quality research field [1, 7, 8].

Anomalies can evoke a *crisis* when they either strike at the fundamentals of a paradigm and resist any attempt of removal or when they are important with respect to some pressing social need. The former is the case in requirements quality research. Despite the continuous efforts to produce support for detecting and removing requirements quality defects [51, 52], research results do not unfold the impact in practice that the general theoretical assumptions of the current paradigm would expect.

Resolving a crisis requires a scientific *revolution* where one theoretical structure is replaced by another, potentially incommensurable one. Femmer et al. heralded this revolution by proposing the notion of *activity-based* requirements quality [2, 97]. With this thesis, we aim to advance the revolution set in motion by making explicit a new paradigm, therefore furthering a *paradigm shift*. The paradigm is expected to resolve the aforementioned crisis and consists of the following constituents:

1. General theoretical assumptions and laws: The paradigm is governed by the explicit activity-based requirements quality theory [40] and the taxonomies organizing its constituents [41, 92].

2. The techniques for their application: The experimental approach [82] and methodological guidelines [74, 81, 93] advise on how to contribute evidence to the paradigm.

This enables two ways of interacting with the paradigm. The first is what Kuhn terms a *normal science*. Following the techniques for the application of the theoretical assumptions and laws, constructive efforts to improve the knowledge accumulated within the paradigm can be undertaken by any researcher. In practice, this means that any researcher subscribing to the activity-based notion of requirements quality [40], i.e., the main theory governing the paradigm, can apply empirical research methods to contribute new evidence answering the core question about the impact of the quality of requirements artifacts. Another contribution in the scope of a normal science would be to extend the knowledge structures like the requirements quality factor ontology [41] or the model of requirements-affected activities and their attributes [92]. By subscribing to these two analysis theories that describe the elements of the requirements quality theory [40], independent and dependent variables relevant to the research domain become more complete and precise. This extends the common vocabulary and improves the measurement instruments used within the paradigm.

The second way of interacting with the paradigm is by initiating a new revolution after identifying an anomaly that produces a crisis. Should researchers identify an anomaly, i.e., an observation that resists explanation within the current paradigm, the ensuing crisis needs to be resolved. Specifically, this means that a new theory would relieve the activity-based requirements quality theory and introduce a new paradigm.

While the extension of existing knowledge structures resembles a paradigm shift in that it changes the foundation of a research endeavor, these two changes are distinguished in their magnitude. This can be explained in terms of *research programs* as introduced by Imre Lakatos [98]. According to Lakatos, the theoretical foundations of a research program are composed of a *hard core* of essential, irrevisable theories, surrounded by a *protective belt* of supporting theories that may be subject to

change. In our context, we may well argue that the requirements quality theory [40] constitutes the irrevisable hard core of the program while the knowledge structures listing quality factors [41], activities, and attributes [92] form the protective belt. Hence, adjusting the latter does not incur a crisis necessitating a paradigm shift but rather represents a significant change that remains consistent with the hard core of our paradigm.

1.8.1.2 Implications on Methodology

Complementary to the anticipated implications for the particular research domain of requirements quality, we strive to make a contribution to methodological discussions in the software engineering research community as well. We aim to add to ongoing initiatives pursued by the ISERN network[4] and the ACM empirical standards [43] by improving the design, execution, and documentation of empirical research methods. Our methodological contributions support four particular initiatives:

1. **Open Science**: Our recovery of unavailable research artifacts and our guidelines to improve their availability [45, 74] aim to support researchers in properly disclosing artifacts connected to their studies.

2. **Causal Inference**: Our demonstration of applying an explicit framework for statistical causal inference [81, 82] shall support the endeavor in SE to abandon correlational studies and attempt inferring causal claims [54].

3. **Data Analysis**: Our comparison of frequentist and Bayesian methods [81, 82] and the review of guideline adherence for crossover-design experiments [93] provide additional guidelines for reliable data analysis.

4. **Research Synthesis**: Our framework of the evolution of empirical, quantitative evidence [99] extend the current research synthesis practices beyond meta-analysis [94] and allow a more structured approach to arriving at valid variance theories.

We hope that our work enhances the community's awareness of these methodological discussions. Furthermore, we hope to provide the community with demonstrations and tools that make more rigorous approaches usable. We properly documented and archived all supplementary material of each work to ease the replication of our work and increase its accessibility.

By participating in annual community meetings like the ISERN and national SiREN meeting[5] we actively disseminated our contributions and participated in ongoing methodological discussions. The publication and archival of all replication packages connected to our studies further increase their usability.

[4] International Software Engineering Research Network, see `https://isern.iese.de/`

[5] `http://sirensweden.org/`

1.8.2 Implications for Practice

By reaching goals 1-4, we hope to enable the transfer of knowledge about requirements quality to software engineering practitioners dealing with requirements artifacts. The transferred knowledge takes the form of accumulated research results about requirements quality phenomena. This way, practitioners obtain recommendations synthesized from multiple empirical studies about the impact of requirements quality factors. Practitioners can use these recommendations to design requirements writing guidelines.

The requirements quality theory [40], the plethora of available factors [41, 92], and the advanced statistical methods for data analysis [81, 82] pose considerable complexity to advance the field of requirements quality. Consequently, we designed the requirements quality framework [99] such that it hides this complexity and offers a simple interface for researchers to communicate their results to practitioners.

1.8.3 Limitations

Our work is still subject to the following limitations. Firstly, while our studies are focused on NL requirements artifacts, most of the artifacts involved in our studies represent functional requirements. While our approach is neither limited to functional nor to NL requirements artifacts, the empirical evidence generated during this thesis pertains mostly to NL requirements artifacts specifying functional requirements.

The contribution presented in Chapter VII does not fully achieve goal 3, i.e., the provision of a significant amount of empirical evidence. Being merely one study, it rather represents one step towards reaching goal 3 and aims to entice replications. Additionally, Chapter VII demonstrates how to contribute to the proposed paradigm. Therefore, we cannot claim that we have fully reached goal 3 in the scope of this thesis.

On a similar note, this demonstration of producing empirical evidence according to the paradigm of the requirements quality theory is limited to a controlled experiment [82]. This research method offers the highest control over the factors of interest and, therefore, supports our claim of causality. On the other hand, we acknowledge that controlled experiments are expensive and do not scale well [100]. Comparable guidelines on how to contribute to the proposed research paradigm using observational instead of experimental studies is still missing.

Furthermore, our elaboration of the theoretical foundation includes taxonomies for only two out of three classes of variables: The requirements quality factor ontology [41] structures requirements quality factors, and the model of requirements-affected activities and their attributes [92] structures activities and attributes. We did not develop a taxonomy for the third class of variables, the context factors (bottom of Figure 1.2). Context factors span a variety of human factors, organizational aspects, and properties of a system's application domain [88]. Critically, many of

these factors are latent variables with unclear operationalization, but their influence on requirement quality phenomena is strongly suggested, given the importance of human factors in RE [87].

Finally, we acknowledge that our final contribution in Chapter VIII, the framework for managing scientific theories, is currently strongly tailored to support the use cases of researchers but not of practitioners. Practitioners are similarly important stakeholders in the framework as they are supposed to utilize it to receive research results that researchers feed into the framework. Studying the applicability of this approach from the practitioners' view fell out of the scope of this thesis.

1.8.4 Future Work

Our most imperative future work will be to maintain the requirements quality framework and orchestrate empirical research in the requirements quality research domain. We aim to shepherd this research endeavor beyond this thesis. Our immediate course of action is to generate attention for the requirements quality theory and its constituents, as well as the requirements quality framework as an integration platform. Additionally, we aim to disseminate our advice on generating empirical evidence in seminars and tutorials.

To address the second of the limitations mentioned in Section 1.8.3, we aim to complement our experimental studies with observational studies. This way, we aim to provide additional guidance to scholars and an alternative for generating new evidence via experiments. To this end, we are actively recruiting company partners and investigating requirements quality phenomena in their respective contexts. We aim to make use of advanced statistical methods to draw causal inferences from observational data that still conform to the requirements quality framework [80].

To address the third of the limitations mentioned in Section 1.8.3, we aim to develop a taxonomy of context factors relevant to requirements engineering, similar to our previous ontology [41] and taxonomy [92]. This taxonomy of context factors shall guide a systematic exploration of the impact that context has on requirements quality. The main challenges will be the elicitation of relevant factors and a valid operationalization of those factors. We envision pooling this knowledge from both the experience of subject matter experts from the RE research domain as well as extensive empirical studies from practice.

Finally, we aim to extend our theoretical contributions to strengthen the proposed research paradigm. Currently, our paradigm consists of three analysis theories:

1. the **requirements quality theory** [40] describing the relationship between requirements quality concepts,

2. the **requirements quality factor ontology** [74], a classification system of qual-

ity factors, and

3. the **model of requirements-affected activities and their attributes** [92], a classification system of activities and attributes.

Additionally, the application of those theories in our empirical contribution [82] (Chapter VI) represents a first step towards a prediction theory, estimating the impact of two specific requirements quality factors. According to the categorization of Gregor [39], two types of theories are not covered by our paradigm. Firstly, our paradigm lacks an *explanation* theory. While our prediction theory supports obtaining a systematic understanding of *what* happens (e.g., what impact a passive voice requirement will have on the domain modeling activity), it cannot explain *why* this impact happens. Adopting theories from linguistics and social sciences will be necessary to explain such phenomena. Secondly, our paradigm lacks a *design and action* theory. The requirements quality framework presented in Chapter VIII provides an interface for knowledge synthesis and translation, but it does not prescribe *how* to enact the recommendations, as mentioned in Section 1.8.3. Once the requirements quality framework has matured and accumulated more empirical evidence worth synthesizing, we plan to investigate the reception and use of this knowledge.

1.9 Conclusion

Requirements quality research aims to support software engineering practitioners in deciding whether their requirements artifacts are good-enough. To achieve this goal, requirements quality research requires a paradigm shift to ensure that it studies relevant issues in a productive manner. This paradigm must encourage focusing on relevant phenomena (i.e., how requirements artifacts impact subsequent activities), using valid research methods to produce new empirical evidence, and facilitating constructive, distributed, yet coherent research endeavors. These endeavors ultimately integrate into more general and valid propositions that provide valuable decision support for practitioners. In the scope of this thesis, we take several steps in this paradigm shift. Particularly, we make (1) theoretical contributions by developing a harmonized requirements quality theory and taxonomies for its constituents, (2) methodological contributions by improving research methods, and (3) practical contributions by demonstrating the application of our advancements. We are confident that adherence to this paradigm will propel requirements quality research in its trajectory to produce meaningful research that aids practitioners.

Paper I

Requirements Quality Research: a harmonized Theory, Evaluation, and Roadmap

Abstract

High-quality requirements minimize the risk of propagating defects to later stages of the software development life cycle. Achieving a sufficient level of quality is a major goal of requirements engineering. This requires a clear definition and understanding of requirements quality. Though recent publications make an effort at disentangling the complex concept of quality, the requirements quality research community lacks identity and clear structure which guides advances and puts new findings into an holistic perspective. In this research commentary we contribute (1) a harmonized requirements quality theory organizing its core concepts, (2) an evaluation of the current state of requirements quality research, and (3) a research roadmap to guide advancements in the field. We show that requirements quality research focuses on normative rules and mostly fails to connect requirements quality to its impact on subsequent software development activities, impeding the relevance of the research. Adherence to the proposed requirements quality theory and following the outlined roadmap will be a step towards amending this gap.

Keywords: Requirements Quality, Theory, Survey

1 Introduction

The empirical evidence of the impact of requirements engineering (RE) on the software development life cycle has shown that the quality of requirements artifacts and processes influences project success and budget adherence [18, 21, 101]. Moreover, the cost of defects introduced during the RE phase of a project is reported to scale exponentially the longer they remain undetected [102]. This necessitates quality assurance techniques capable of detecting RE defects as soon and as reliably as possible.

Requirements quality research is dedicated to supporting the software engineering process with the means to evaluate and improve the quality of requirements, mainly focusing on requirements artifacts [24]. However, recent systematic investigations of requirements quality literature revealed a lack of rigor and relevance of these contributions [6, 41]. Moreover, the impact of the quality factors proposed in literature (i.e., requirements writing rules) remains largely unexplored in practice [41], hindering its adoption in industry [1, 7–9].

Existing quality theories and frameworks are too abstract to guide requirements quality research at an operational level [103, 104]. These theories often only divide quality into sub-categories without any means of applicability. In this paper, we argue for the need for a theoretical and operationalizable foundation of requirements quality research. We review the closely related software quality research and draw parallels to requirements quality research to consolidate a harmonized requirements quality theory. Additionally, we survey requirements quality literature with respect to the theory to reveal current shortcomings. Accordingly, we make the following contributions:

1. A harmonized requirements quality theory serving as a theoretical foundation for requirements quality research.

2. A survey of requirements quality research revealing if and how concepts of the theory are reported in the state of the art, but also emphasizing shortcomings.

3. A consequent research roadmap aimed at mitigating these shortcomings.

The rest of this manuscript is organized as follows: Section 2 illustrates the evolution of software quality research and draws the parallel to requirements quality research. In Section 3, we derive a harmonized requirements quality theory from this comparison. This theory is used to evaluate the state of requirements quality research in Section 4 and reveal current shortcomings. The consequent research roadmap to mitigate these shortcomings is presented in Section 5 before concluding in Section 6.

2 Software Quality Research

Software quality research follows a similar premise as requirements quality research. It is necessary to control the quality of software artifacts (e.g., source code) as it impacts the overall quality of the development life cycle and the final product. This premise aligns with the aim of requirements quality research. To show commonalities and differences between these two research fields, we review the evolution of software quality research in Section 2.1 and draw a parallel to requirements quality research in Section 2.2. We reach conclusions about the necessary direction the latter needs to take.

2.1 Evolution of Software Quality Research

Software quality research revolves around assessing the quality of software artifacts [105]. In the following, we describe the evolution of the field according to Broy et al. [105] and Deissenboeck et al. [71].

Guidelines and Metrics-based approaches Guidelines are the simplest approach for controlling the quality of software artifacts. For example, the Java coding conventions [106] prescribe—among other suggestions—how to name and structure Java files. However, guidelines commonly fail to significantly impact software quality, likely because they lack the motivation for their relevance [70]. For example, the aforementioned suggestions are justified because "[c]ode conventions improve the readability of the software" [106] without any empirical evidence of that claim. Furthermore, guideline conformance is difficult to assess and hence seldom done in practice [71]. The latter shortcoming was addressed by introducing metrics-based approaches where metrics were devised to measure relevant attributes of software artifacts. Among others, *lines of code* [107] and *cyclomatic complexity* [108] were used to evaluate software quality automatically. Nevertheless, most metrics continue to lack justification of their relevance [105, 109–111].

Quality Models To overcome the relevance shortcoming, quality models aggregated metrics into hierarchical trees of criteria [35, 112]. The leaf nodes are specific enough to be operationalized as an evaluation metric, while the aggregation into higher-level quality characteristics provided the justification for their relevance. For example, low-level concepts such as *structuredness* and *conciseness* of code were justified by their aggregation to *understandability* and *maintainability*, which were widely accepted as relevant software quality characteristics [35]. However, hierarchical models suffered from unclear decomposition rules and constrained levels of granularity, which were either too abstract to be operationalized or too detailed, disconnecting the applicable metrics from their rationale [71, 105].

Quality Meta-Models The popularity of quality models necessitated a structure for the proposed models [113]. Meta-models like the Goal Question Metric approach by Basili et al. [114] and the factor-strategy quality meta model by Marinescu and Ratiu [115] provide this overarching structure. Deissenboeck et al. [116] contribute the DAP classification for quality models, which categorizes the aim of a quality model to be to *define* (D), *assess* (A), or *predict* (P). The publication further relates quality meta-models to quality models as the "model of the constructs and rules needed to build specific quality models." [116].

Activity-based Quality Models In addition to the shortcomings that existing quality models continued to suffer, the elements populating these models were found to

be heterogeneous [71]—i.e., properties of a *system* were mixed with properties of *activities in which the system is used*. For example, the maintainability branch in the software quality characteristics tree by Boehm et al. [117] contains both system properties like the *structuredness* of a software artifact, but also attributes of activities in which these artifacts are used, like *modifiability*. The latter describes the *activity* of *modifying* an artifact rather than a system property, despite the adjective's nominalization suggesting otherwise.

So far, no clear rule for distinguishing a system from an activity property has been proposed. We derived two heuristics from the implicit argumentation of previous publications [71]. First, if a property involves an additional agent (e.g., *testability* involves a *test engineer*, *modifiability* involves a *modifier*, although not necessarily human), then it represents how the system is used—i.e., an activity property. The second heuristic comes in the form of a syntactical criterion:

- Nominalized adjectives (e.g., structured-ness, concise-ness) tend to be **system properties**
- Nominalized verbs (e.g., modify-ability, access-ability, augment-ability) tend to be **activity properties**

Interpreting activity properties as system properties ignores an underlying impact relationship. For example, interpreting *modifiability* as the *system* property of how receptive it is to change omits that actual system properties (e.g., whether the system is digital or analog or who has writing access rights) *impact* the ability of a stakeholder to modify the system, which is an activity property.

To address the issue of heterogeneous properties, Deissenboeck et al. introduced *activity-based quality models* [71, 105], which separate system properties from activity properties and form two distinct, orthogonal dimensions. The model expresses quality as the impact of system properties on activity properties. Figure 1 visualizes a simplified version of the quality model [71], showing how code clones impact the modification sub-activity and expressive identifiers impact the concept-location sub-activity.

The activity-based quality model was successfully applied to usability [118], security [119], and service-oriented architecture [120] before Wagner et al. distilled a comprehensive activity-based meta-model in the scope of the Quamoco project [72, 121]. In parallel, the original use case of the activity-based quality model, which focused on maintainability, received extensive tool support [73, 122] contributing evidence to the operationalization of quality models in practice [123].

Activity-based quality models solve limitations of previous quality models at the cost of increased complexity, which manifests in additional challenges to operationalize and communicate the notion of quality [124]. However, the complexity of these models is necessary to tackle the faceted concept of quality [124, 125]. Research continuously tackles the inability of activity-based quality models to assess ar-

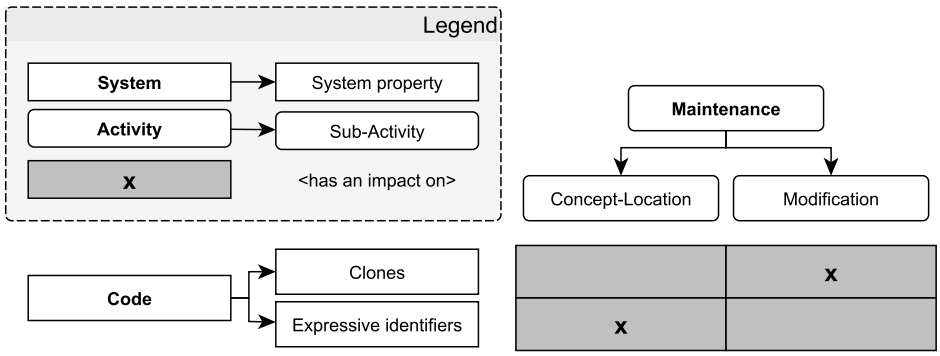


Figure 1: Excerpt from the activity-based quality model for maintainability

tifact quality and distinguish quality levels [126]. For example, weights empirically derived from historical data replaced expert-based propositions [127], and Bayesian networks were utilized to model the impact relationships [128].

2.2 Mapping to Requirements Quality Research

In the following, we draw a parallel of the evolution of quality research between the areas of software engineering and requirements engineering.

Metrics and Quality Models Similar to software quality, requirements quality research historically originated from proposing metrics like *passive voice* of requirements sentences [34] or *sentence length* [90], which are associated with bad quality of requirements specifications. Frattini et al. [41] collected these quality factors and indicated their limitations. Most existing publications either fail to gauge the impact of these metrics [129] or explicitly disregard their relationship [130]. Requirements quality models [131, 132] integrate these factors into larger frameworks but often remain vague on their notion of impact.

The investigation of impact is often limited to a comparison between the quality factor and practitioners' subjective, general perception of the quality of the requirements entities [133]. Wilson et al. contribute a first impact matrix between quality indicators and quality attributes [134], but the latter suffers from the same system and activity properties heterogeneity. Similarly, Yang et al. state that "[a]mbiguity is therefore not a property just of a text, but a conjoint property of the text and of the interpretations held by a group of readers of that text" [36], exposing the necessary distinction between system and activity properties.

Activity-based Requirements Quality A large portion of requirements quality research exhibits the same shortcomings identified and overcome by software quality

research, namely that (1) requirements quality factors lack relevance due to their unknown impact, which in turn inhibits adoption in practice, and (2) the terminology of requirements quality aspects confuses system and activity properties.

Femmer et al. apply the activity-based quality perspective to requirements engineering by proposing the activity-based requirements engineering quality model (ABRE-QM) [2]. This model leverages the insights from activity-based software quality models [70–72] and shows that the quality of requirements depends on the impact they have on the activities in which they are used. However, despite the authors' call for action [97], ABRE-QM saw little adoption in research as demonstrated in recent systematic investigations of the requirements quality literature [6, 41].

The ABRE-QM example above raises the concern that requirements quality researchers do not properly utilize the activity-based approach successfully employed in software quality research. In this manuscript, we want to encourage further research on this approach by presenting a revised requirements quality theory, a thorough investigation of the requirements quality literature verifying the hypotheses from previous studies [6, 41], and a consequent research roadmap.

3 Requirements Quality Theory

We generated a harmonized requirements quality theory (RQT) by consolidating the evolution of software quality models described in Section 2.1, their application in requirements engineering as described in Section 2.2, and alignment to the established Quamoco quality model [121]. In terms of theory types [39], the RQT is both *explanatory*, as it explains the notion of requirements quality, and *prescriptive*, as it prescribes how to report contributions to requirements quality. The building blocks of the theory are described in Section 3.1 and illustrated with an example in Section 3.2.

3.1 Theory

The concepts that constitute this theory are visualized in Figure 2, and each concept is described in Table 1. The model represents an evolution of the original activity-based requirements engineering quality model (ABRE-QM) proposed by Femmer et al. [2]. Here, we present changes to the original model.

The artifact-related section of the model (left part of Figure 2) is largely equivalent to the original publications [2, 71]. Entities represent requirements artifacts of different granularity [24], which can be decomposed into further entities. For example, a requirements specification can be decomposed into sections, which in turn consist of paragraphs and sentences or requirements. We consider an artifact to be a high-level requirements entity and hence do not explicitly add the *artifact* to the model, deviating from the original [2]. Similarly, factors can be decomposed into

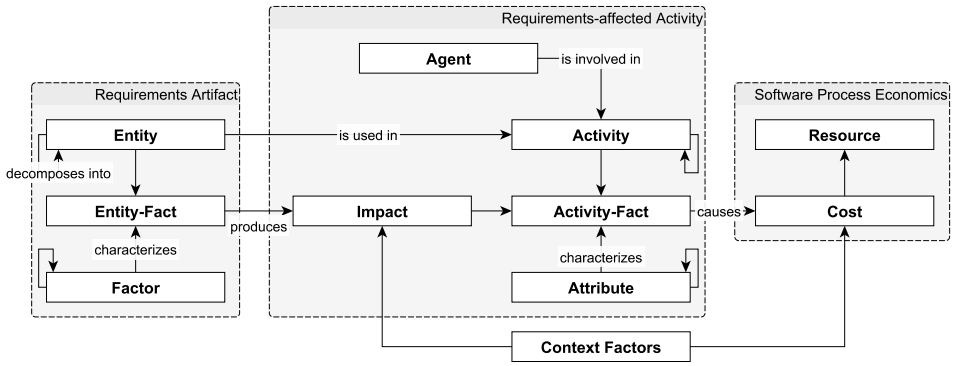


Figure 2: Concepts of the Requirements Quality Theory.

Table 1: Explanation and origin of theory concepts.

Concept	Explanation	Origin
Entity	A requirements artifact or part thereof	[2]
Factor	``[A] normative metric which maps a textual requirement of a specific granularity'' [41] to a numerical output	[2, 71]
Entity-Fact	A composition of one entity and one factor	[71]
Agent	Any person, group of people, or automatism involved in an activity	[2]
Activity	An activity in which the entity is used	[71]
Attribute	A measurable property of an activity	[118]
Activity-Fact	A composition of one activity and one attribute	
Impact	The impact of a fact on an activity-fact	[2, 71]
Context Factor	A factor describing the context of the impact relationship	[87, 135]
Cost	The magnitude of cost associated with an activity-fact	[135]
Resource	The resource affected by the economical impact	[86, 135]

sub-factors to accommodate composite factors. For example, Antinyan et al. [136] position their proposed quality factor of *conjunctive complexity* as a sub-factor of *syntactical complexity*.

The activity-related section of the model (middle part of Figure 2) again adapts the original models [2, 71]. The concept *activity* does not represent common requirements activities, like elicitation, analysis, and validation [137], but rather every process that takes a requirements entity as input and produces an output. This includes some requirements activities (like analysis and validation, which use requirements as input) but not others (like elicitation, which often does not presuppose existing requirements). Hence, we rather refer to them as *requirements-affected activities*. These further include implicit sub-activities (e.g., *understanding* and *interpreting* an entity), which can be aggregated with other, more explicit sub-activities (e.g., *test case design*) to form high-level activities (e.g., *validation*). The decomposition relationship of the activity concept accommodates this aggregation. To accommodate not only human actors involved in activities but also any automatism like requirements processing tools [138] we abstract the concept of *stakeholder* to *agent*.

We generalized the impact concept in this theory. While previous models as-

sumed that impact is categorical (i.e., the occurrence of a fact has either a positive, negative, or no impact at all, like in Figure 1 [71] or linear (i.e., the larger the evaluation of a quality factor, the better/worse is its quality), we consider the impact to model any kind of relationship between Entity-facts and Activity-facts. This opens up the theory to more complex relationships, which can model the actual impact more accurately and allows to compare the impact of quality factors with each other.

Two concepts were added to the model. First, the impact was related to an *Activity-fact* composed of an activity and an attribute as proposed by Winter et al. [118]. This way, the structure of the variables on the two sides of the impact relationship is mirrored. Furthermore, the necessity to associate an impact with a measurable property of an activity is emphasized. Second, context factors also influence the impact of an Entity-fact on an Activity-fact. As recognized by previous publications [87, 135], the impact differs depending on external factors related to, among others, the organization and the people involved [88].

The economic section of the model (right part of Figure 2) is a novel addition to previous iterations of the activity-based models [2, 71, 121]. As long as the subsequent *economic* impact of an Activity-fact is unknown, the Entity-fact that produces the Impact on this Activity-fact will remain neglected [86, 135]. Hence, the software process economics perspective introduces a *Cost* for a specific *Resource* such as time or money.

3.2 Example

In this section, we illustrate the RQT with a fictitious example to demonstrate its application. The example is additionally visualized in Figure 3.

In this example, a customer's requirements were elicited and documented in a requirements specification containing the entity *user story 42*. One relevant quality factor used by the organization responsible for implementing the requirements is template *conformance*, which prescribes that all user stories must follow the Connextra template [139] "As a <role> I want to <goal> so that <benefit>." This quality factor maps the entity to a categorical value, containing—among others—the values *conform*, *missing role*, and *missing all elements*. In this example, the role is omitted from the user story. Hence, the quality factor template conformance is evaluated to *missing role*, which constitutes the entity-fact (yellow box in Figure 3).

The organization uses this user story in a subsequent, requirements-affected *development* activity, where a different stakeholder—the developer—is responsible for translating the entity into code. This activity can be decomposed into two distinct sub-activities: *understanding* the entity and *programming* the respective implementation.

One desired attribute of the activity understanding is *determinism*—i.e., a requirements entity should have only one unique interpretation. Possible variations of

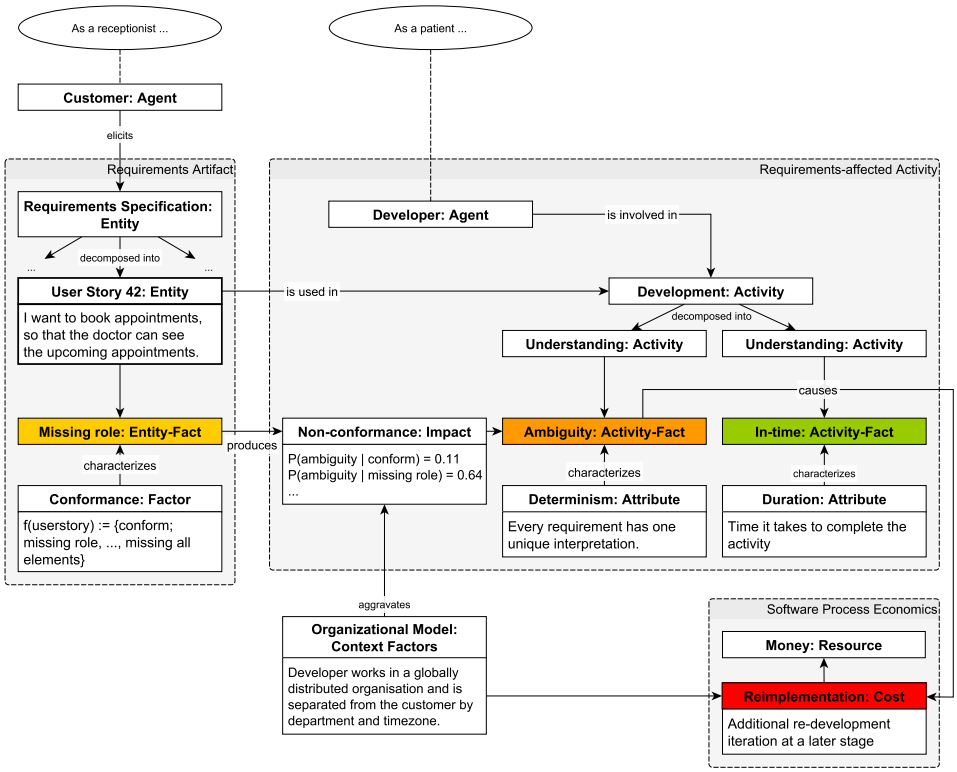


Figure 3: Exemplary instantiation of the theory

the interpretation and, therefore, the subsequent translation of a requirement must be avoided. Because the *conformance* quality factor is evaluated to *missing role* on the *user story* entity, the *understanding* activity is less *deterministic*, as the developer can make a different assumption about the role implied by the requirement. The understanding activity has become ambiguous, which constitutes the *activity-fact* (orange box in Figure 3).

The relationship between the entity-fact and the activity-fact is the *impact* of the quality factor. Instead of limiting the impact concept to categorical values (e.g., either *has an impact* or *has no impact*), the RQT enables more complex impact relationships. In this fictitious example, the quality factor value *missing role* is associated with a 64% chance of making the understanding sub-activity ambiguous. This relationship can be determined empirically via experimental research investigating the likelihood of the different values of the conformance quality factor reducing the determinism of the understanding sub-activity.

The programming sub-activity may go unaffected by the entity-fact that the conformance has a value of *missing role* (green box in Figure 3): regardless of the agent's interpretation of the requirements entity, the programming sub-activity will remain

unaffected in respect to the relevant attribute *duration* under the assumption of a similar user interface for both roles. Whether the feature is coded for the role receptionist (as the customer intended) or patient (as the developer assumed) does not significantly change the duration of the sub-activity if the user interfaces only barely differ.

The significant impact on understanding is influenced by the organizational model, which is one relevant *context factor*. Since the organization is globally distributed and the two involved agents are unlikely to have informal interactions, the impact is amplified. In contrast, in a small organization where all involved agents share an office, the impact can be alleviated as missing information is recovered through informal communication. Similarly, the software development process model may significantly influence the impact of the quality factor, and the use of an agile approach may reduce the impact by encouraging communication between the customer and developer. The context factors significantly influence the impact and, therefore, have to be included in the relationship between entity-facts and activity-facts.

The reduced determinism of the understanding activity has an economic effect—i.e., the less deterministic the activity is, the more the implementation needs to be revised, which costs money and time (red box in Figure 3). Context factors influence the extent of this effect as, for example, a re-implementation can be more costly in larger organizations due to organizational overhead.

For the sake of brevity, the example omits the following aspects: (1) the example limits the number of elements populating the relationship. More quality factors of the entity, activities, attributes of activities, and context factors are possibly involved in the relationship. (2) Interaction effects between quality factors and context factors are plausible but not reported here.

However, the example demonstrates how adherence to this activity-based RQT elevates requirements quality factors from normative rules (i.e., user stories must conform the template for the sake of it) to empirically-backed impact predictions (i.e., user stories must conform the template to mitigate ambiguous interpretations and avoid implementation cost).

4 State of research

Despite the publication of the ABRE-QM [2] and its authors' proposition to adapt the quality meta-model for future requirements quality research [97], recent systematic reviews raised concerns regarding a perspective on requirements quality limited to the artifact-related section of the model (left part of Figure 2) [6, 41].

To validate these concerns, we formulate the following research question. **How are the concepts of the requirements quality theory reported in requirements quality literature?** Answering this research question requires extracting information from a population of publications; accordingly, we employ survey research as our

approach to gain insight into the current state of research. We follow the survey guidelines by Molléri et al. [57] and report our survey in the following subsections. All supplementary material for replicating this study is available in our replication package[1].

4.1 Survey Objects

The target population of our survey is the requirements quality literature dealing with quality factors in requirements artifacts. Frattini et al. [41] conducted a systematic study on requirements quality factors, including a sample of 57 primary studies. To our knowledge, this is the only sample that fulfills our aforementioned requirements. This classifies the sampling as non-probabilistic, more specifically convenience sampling [57].

4.2 Study Design

We follow the recommended practices for the survey research process and report our steps accordingly [57]. However, we disregarded steps that only apply to surveys with human subjects, such as *participant recruitment* and *response management*.

We derived the *definition of the research objectives* in the form of the research question directly from previous research [6, 41, 97]. We established a *study plan*, rigorously documenting all research progress and justifications for any deviations during the process. We *identified and characterized the population* of our survey and executed our *sampling plan* as described in Section 4.1.

For our *instrument design*, we maintained two artifacts. We created an extraction guideline based on the RQT concepts. Each concept of the RQT was associated with one or more categorical variables, each containing a set of codes that represented *if* and *how* the concept was reported. The codes were created ad hoc in the first iteration of extraction and refined based on discussions and theoretical background in the second iteration.

The extent of the codes varied. The codes that represent how the concept *entity* is reported are, for example, *explicit* and *implicit*. An entity is either reported explicitly if its scope and form are clear. It is reported implicitly if the authors just report that the factor applies to a "requirement" without defining whether this is a single or multiple natural language sentence, whether the language is constrained or not, or whether it assumes a full sentence at all.

The codes of other concepts were more complex and grouped into distinct categories. For example, the codes of the concept *Factor* were split into two groups, representing both the *explicitness* when reporting a factor (i.e., whether the factor is explicitly *reported* or *referenced* from another publication) and the *form* in which

[1] Available at `https://doi.org/10.5281/zenodo.8167598`.

the factor is reported (i.e. if the factor is represented with a *textual description* or defined using a logical or mathematical *formula*). The extraction guideline containing all codes, explanations, and examples can be found in the replication package.

The first author extracted the appropriate code for each concept in the requirements quality theory from each publication. The extractions for each publication in the sample were recorded in a spreadsheet. For *instrument validation*, the second author of this manuscript independently performed the extraction task using the guideline on six ($\approx 10\%$) publications randomly sampled from the survey objects. The second author performed the extraction on two of these six publications as training, and the remaining four were used to calculate the inter-rater reliability between the first and second author.

The task overlap achieved an percentage agreement [140] of 83.3%, whereas Cohen's Kappa yields a *moderate* agreement of 54.2%. As Cohen's Kappa is unreliable for uneven marginal distributions [141], we calculated the more robust S-Score [142]—yielding a *good* agreement of 76.8%—which we deem sufficient for assessing the inter-rater reliability.

We used the codes in the *data analysis* phase to generate descriptive statistics on which we based our interpretation of the state of requirements quality. These form a quantified foundation for interpreting the state of requirements quality literature with respect to the research question. For final *reporting*, we adapted established reporting guidelines [57] and disclosed all material in a reusable replication package.

4.3 Study Results

Figure 4 visualizes the distribution of the relevant codes among all concepts included in the requirements quality theory. Each concept is overlaid with a bar representing how many of the 57 publications contained the concept. The row below each concept represents its dimensions derived from the appropriate codes.

Though both entities and factors are explicitly reported in all 57 publications of the sample, a large portion ($24/57 = 42.1\%$) of entities is reported implicitly—i.e., the entity's scope is not clear. This occurs mostly because authors attach the reported quality factor to the entity *requirement* without specifying the scope or form of the entity. Montgomery et al. [6] have already noted this shortcoming in the requirements quality literature and it represents a terminological ambiguity in the research domain.

Seventeen out of 57 publications (29.8%) do not report any impact on activities (code *N/A*) and hence neglect the practical relevance of the proposed quality factors. Agents are only reported in 14 (24.6%) of all publications. Activities are—when reported—predominantly elicited *ad hoc* ($37/40 = 92\%$) and rarely *systematically*—i.e., when activities impacted by a quality factor are discussed, the identification of activities has no systematic approach. Attributes are also only rarely reported ($8/57 = 14\%$).

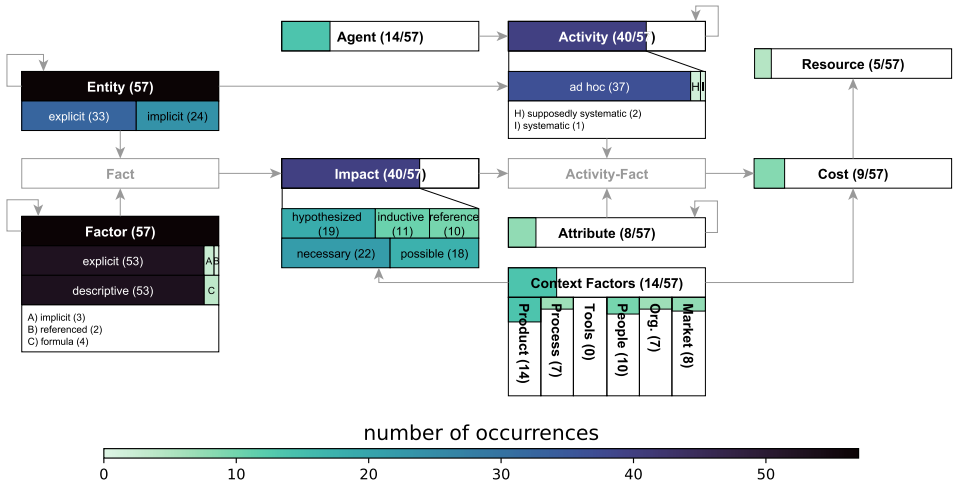


Figure 4: Survey results depicting the distribution of codes.

We grouped the codes classifying how *impact* is reported into four distinct dimensions, two of which are reported here. The *evidence* for the impact—when at all reported—is dominantly hypothesized ($19/40 = 47.5\%$) and rarely either inductive ($11/40 = 27.5\%$) or referenced ($10/40 = 25\%$), i.e., draws the evidence from another publication. Previous studies [6, 41] have also noted this dominance of anecdotal, non-empirical evidence. The *modality* of impact relationships is balanced between *necessary* and *possible*—i.e., the impact of quality factors is reported almost equally often to be certain or potential. The remaining two dimensions of impact (*generality* and *frame of reference*) yielded no additional insight into the surveyed objects and are hence not reported here but contained in the replication package.

Context factors are almost completely neglected and only reported to a degree varying between zero (no publication reports the influence of any *tools*) and 24.6% (14 out of 57 publications reporting *product*-related factors, e.g., the system's size or type).

Both *cost* and *resources* are reported only rarely ($9/57 = 15.8\%$ and $5/57 = 8.8\%$ respectively) and, if so, only hypothesized or referenced, never determined empirically. Money and time are mentioned as the resources affected by activity impact, and the cost is only estimated in terms of expected change (e.g., "*reduction* of the time spent" [130]) or general magnitude (e.g., "*significant* amounts of money" [30]).

4.4 Interpretation

In this section, we interpret the results presented in Section 4.3 and answer the research question.

Publications in the requirements quality literature adhere to the RQT to a varying

degree. While all publications in the sample mentioned both an entity and a quality factor, activity-related concepts, context factors, and the economic impact are often neglected. Failing to consider the context factors severely threatens the external validity of the proposed quality factors [87, 135] and neglecting the economic impact risks undermines their acceptance [86, 135].

Context factors and economic impact are arguably more challenging to investigate [31]; however, we emphasize that the lack of activity perspective when proposing quality factors is critical for several reasons. The complete negligence of a quality factor's impact limits the factor to a normative, unmotivated prescription and challenges its practical relevance [2], which in turn promotes skepticism regarding requirements quality factors in industry [1, 7–9].

The survey emphasized two additional shortcomings in the field of requirements quality research. First, the tendency to elicit activities *ad hoc* when discussing the impact of requirements quality factors bears the risk of missing other important impacts. Most publications discuss a hypothesized impact of a quality factor on a non-systematically selected activity or set of activities. This selection is usually justified by anecdotal or folkloric circumstances, like "[a]mbiguous requirements may bring about misinterpretations among stakeholders, and prompt a few issues" [143].

While these impact relationships are neither empirically proven nor falsified, the non-systematic selection of activities can disregard other impact relationships. Femmer et al. [2] demonstrated that a systematic elicitation of activities could reveal both positive and negative impacts by the same quality factor. For example, the factor *free of UI design details*, which states that an "artifact should describe the problem domain instead of the solution domain" [2], will positively affect maintainability, as UI details are volatile in the beginning and require a lot of change management if specified in a requirement. Conversely, the same factor negatively impacts understandability, as the presence of UI design makes requirements more comprehensible.

Second, while activities are not reported consistently, attributes of activities are reported even less. Attributes represent measurable characteristics of activities; for example, the activity *understanding* can be quantified by its attribute *level of agreement* [136, 144] or a *readability index* [145]. Neglecting the quantifiable attributes of activities impedes an empirical evaluation of a quality factor impact because it omits the measurement instrument for the dependent variable (i.e., the activity-fact) in the impact relationship [118].

We conclude that the requirements quality theory is implicitly embedded in the requirements quality literature. However, insufficient adherence to it results in several limitations when reporting new requirements quality factors. While the artifact-centric theory concepts are commonly covered, activity-centric concepts, context factors, and economic concepts receive less attention, which decreases these publications' practical relevance. With this study, we empirically confirm the concerns voiced in previous investigations of the requirements quality literature [6, 41].

4.5 Threats to Validity of this Research

We discuss the threats to validity proposed by Wohlin et al. [47] and extended by Molléri et al. [57].

Internal Validity We acknowledge a threat to internal validity due to sampling of publications. The method of object selection [6, 41] is deemed sufficiently rigorous to derive an initial theory.

Construct Validity The constructs in this study—i.e., the elements of the theory—are established strictly following mature quality theories from the field of software quality. This ensures the alignment between the underlying theory and measurement constructs.

The lack of a theory to which the surveyed publications could have adhered when reporting quality factors resulted in the concepts of requirements quality often being embedded implicitly, complicating the extraction task. We minimized the resulting threat to internal validity through independent labeling and calculating appropriate inter-rater reliability metrics [141].

External Validity The selected sample of publications [41] is constrained to empirical contributions to requirements quality research [6]. This limits the conclusion validity of the type of evidence for the *impact* concept, as non-empirical work could contribute *theoretical* evidence for impact relationships. For example, the impact of quality factors like *nominalization* [146] can be derived deductively by referring to valency reduction caused by nominalization [147]. While publications utilizing linguistic theory are unknown to the authors, a valid conclusion regarding this type of evidence requires a more thorough extension of the sampling strategy.

5 Research Roadmap

Femmer et al. proposed an initial research roadmap detailing how to advance the field of requirements quality research [97]. Based on concerns of previous studies [6, 41] and the survey of the state of research reported in this study, we assess and update the three suggested steps by Femmer et al. [97]:

1. Creation of "a reference artifact and a usage model" eliciting typical entities, activities, and agents.
2. Creation of "a taxonomy of quality factors" as a central, accessible repository of quality factors.
3. Creation of "a taxonomy of impacts" as a catalog of impacts from quality factors onto activities.

We reflect on these proposed research streams in Sections 5.1 to 5.3 and add three further proposals in Sections 5.4 to 5.6. Because these research streams are grounded in the experiences from the software quality research, we expect contributions to them to promote requirements quality research that is relevant to practice.

5.1 Artifact and Usage Model

Mendez et al. have contributed a reference artifact model for requirements engineering [24, 148] based on their fundamental positioning on artifact orientation [23, 149]. The AMDiRE approach constitutes a domain-agnostic reference for artifact types and serves the purpose requested by Femmer et al. [97] in that it can be tailored towards any industry context to model an artifact structure.

While the elicitation of human [150] and non-human, automatic agents [51] has been addressed, a reference model for activities requires explicit attention in literature. More importantly, with the update of the requirements quality theory over the initial ABRE-QM [2], we argue that a reference model for requirements-affected activities needs to provide *attributes* to quantify each activity. Such attributes enable an empirical assessment of the impact of quality factors.

Additionally, a majority of publications reporting an impacted activity mention some variation of *understanding* or *interpreting* ($32/40 = 80\%$). We assume that every requirements-affected activity comprises an initial *interpretation* sub-activity. However, such composition is obscured by the lack of a proper reference model for requirements-affected activities accounting for their aggregated nature.

It is conceivable that the *interpretation* sub-activity is most prone to defects, which explains the research community's focus on *ambiguity* [6], as ambiguity represents the non-determinism of an interpretation. We argue that a proper reference model for requirements-affected activities accounting for their aggregated nature can steer research towards identifying critical sub-activities—i.e., the ones most prone to impacting subsequent activities.

5.2 Taxonomy of Quality Factors

Requirements quality factors [41, 97] are the cornerstone of artifact-centric quality assurance. The requirements quality factor ontology proposed by Frattini et al. [41] furthered this research stream. Although the ontology is in an early stage and requires additional iterations, quality factors and related objects—such as data sets and automation approaches—are now collected in a central repository.

5.3 Taxonomy of Impacts

The taxonomy of impacts that Femmer et al. [97] deem the necessary final step of the roadmap has to be extended. Previous quality models—including the ABRE-QM [2]—consider only categorical or, at most, linear impact relationships. Therefore, a taxonomy seemed sufficient to record "a list of well-examined effects of quality factors on activities" [97]. We argue that the impact relationship can be more complex and requires a more general representation—i.e., rather than aiming for a taxonomy of impacts, we argue for developing an *impact framework*.

Given the evaluation of quality factors on requirements entities on one side and the evaluation of activity attributes on the other side, the impact relationship between these variables can be formulated as a regression problem. Instead of relying on experts to hypothesize the (categorical) type or (linear) extent of an impact, more complex relationships can be determined using, for example, Bayesian data analysis [62]. Consequently, this research stream aims to develop an impact framework capable of determining these impact relationships based on statistical instruments given sufficient data.

5.4 Context Factors

Context factors must be considered in the impact relationship to operationalize the requirements quality theory [87]. Large-scale endeavors acknowledge the importance of context factors in regard to requirements quality [18], yet no unified collection of context factors relevant to requirements engineering exists. Established sets of software engineering context factors [88, 151] can be used as a starting point but require a dedicated investigation from the requirements engineering perspective.

A clear set of relevant context factors can support developing reporting guidelines for empirical studies on requirements quality and enable context-driven research [152]. While empirical software and requirements engineering publications typically strive for generalizability [151], scoping an empirical study according to the given context factors allows the data collected in that study to be integrated into the impact framework as outlined in Section 5.3. Conversely, reporting the limited scope of a study enables a general requirements quality theory that can be assembled from multiple studies in well-defined contexts.

5.5 Economic Impact

With the addition of economic concepts in the requirements quality theory, a research stream should be dedicated to the economic impact of activity facts. The impact relationship between quality factors and activities already benefits the acceptance of those factors for quality assurance in practice [97]. Adding an economic perspective—i.e., what amount of which resource a change of a certain activity-fact

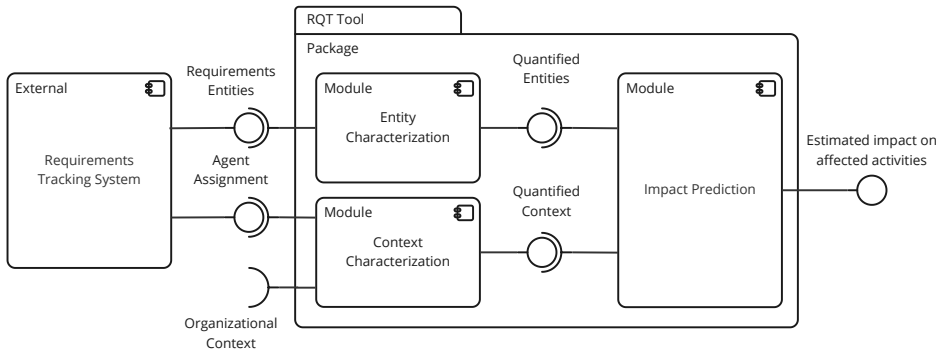


Figure 5: Architectural overview of the proposed tool-support

entails—can further bridge the gap between the normative, artifact-centric quality factors on one side and an economic decision-making process on the other side [86]. Since the purpose of quality factors is to support quality assurance in industry, understanding this economic perspective is of high priority despite the complexity of the topic.

5.6 Tool support

We aim to make the RQT applicable to the industrial context through the development of tool support. The components necessary to realize this tool support are visualized in Figure 5. The goal of the tool is to estimate the impact of requirements entities and their context on the attributes of requirements-affected activities.

To this end, the tool needs an interface to the requirements entities, context information about the involved agents, and context information about the organization. The former two are often available in a requirements tracking system like Jira[2] [153], while the latter a company likely has to generate and provide manually.

Once provided with the necessary information, the tool characterizes both entities and context, i.e., quantifies the natural language requirements entities and the elusive factors determining the context. The quantified entities and context serve as input to the impact prediction model as described in Section 5.3, estimating the impact on the attributes of the requirements-affected activities, which in turn enables quantifying the economic impact as described in Section 5.5.

The realization of this tool depends on the previously described streams of research to identify valid quality factors (Section 5.2), context factors (Section 5.4), and activity attributes (Section 5.1). For the tool to provide an automated impact prediction the following automation modules must be realized:

[2] `https://www.atlassian.com/software/jira`

1. Automatic entity characterization: a shared architecture to automatically evaluate the requirements quality factors collected in the quality factor ontology [41]
2. Automatic impact prediction: an accessible statistical model estimating the impact of quantified entities and context on affected activities, trained on historical data.

Developing this tool while adhering to open science principles will allow scholars to propose new quality and context factors, customize relevant activity attributes, and contribute historic data to improve the impact estimation of the prediction model. We invite contributions to the implementation and maintenance of the tool via its dedicated repository on Github[3].

6 Conclusion

In this manuscript, we investigated the software quality literature and the application of the activity-based quality perspective to the requirements engineering domain. We extend the work of Femmer et al. [2] by proposing an evolved and harmonized requirements quality theory, and assess the adherence of the requirements quality literature to this theory. Our survey confirms the bias towards artifact-centric and the negligence of activity-centric concepts, which was noted in previous secondary studies [6, 41]. Finally, we update the requirements quality research roadmap initiated by Femmer et al. [97] to guide future contributions in the requirements quality research domain.

We are confident that the harmonized requirements quality theory provides the necessary guidance to propel requirements quality research and establish a common understanding of quality that is operationalizable in practice. We invite fellow researchers to contribute to the theory and the requirements quality research field in adherence to it.

[3] Available at `https://github.com/JulianFrattini/rqt-tool`. An archived version is accessible at `https://doi.org/10.5281/zenodo.8167541`.

Paper II

A Live Extensible Ontology of Quality Factors for Textual Requirements

Abstract

Quality factors like passive voice or sentence length are commonly used in research and practice to evaluate the quality of natural language requirements since they indicate defects in requirements artifacts that potentially propagate to later stages in the development life cycle. However, as a research community, we still lack a holistic perspective on quality factors. This inhibits not only a comprehensive understanding of the existing body of knowledge but also the effective use and evolution of these factors. To this end, we propose an ontology of quality factors for textual requirements, which includes (1) a structure framing quality factors and related elements and (2) a central repository and web interface making these factors publicly accessible and usable. We contribute the first version of both by applying a rigorous ontology development method to 105 eligible primary studies and construct a first version of the repository and interface. We illustrate the usability of the ontology and invite fellow researchers to a joint community effort to complete and maintain this knowledge repository. We envision our ontology to reflect the community's harmonized perception of requirements quality factors, guide reporting of new quality factors, and provide central access to the current body of knowledge.

Keywords: Requirements Engineering, Requirements Quality, Quality Factor, Ontology

1 Introduction

Context. A requirements quality factor [154] is a normative metric which maps a textual requirement of a specific granularity to a scale which informs about the quality of this input. Because quality factors can be calculated entirely on textual input and do not necessarily need to consider the perspective of any stakeholder who

is intended to use the requirement, factors are an efficient tool for early estimates of requirements quality, often even eligible for full automation. This satisfies the need for detecting potential defects in textual requirements at an early stage, as the cost for addressing these defects increases the longer they stay undetected, putting the project success at risk when treated poorly [18]. The applicability of quality factors is corroborated by the plethora of existing tools which automate their detection [6, 130]. Among the popular requirements quality factors are *passive voice* [130], where the use of a verb in passive voice is associated with ambiguity of a requirement due to the omission of the subject within a sentence, and *sentence length* [90], where exceeding a specific threshold of words or characters in a sentence is associated with complexity due to the sentence becoming increasingly hard to comprehend.

Problem. Requirements quality research is lacking a holistic perspective on quality factors and a central repository containing the existing body of knowledge to enable reuse and evolution. These two gaps result in challenges such as concurrent work on same or similar quality factors instead of reusing and advancing those already established. For example, *anaphora* or *anaphoric ambiguity* is described as "an expression used, in language, to refer to another expression" [155], "a linguistic expression that refers to a preceding utterance in text" [36], and "whenever a pronoun (e.g., he, it, that, this, which, etc.) refers to a previous part of the text" [90]. While a certain degree of similarity between all three competing descriptions is apparent, the lack of consensus on the definition is bound to introduce ambiguity to the understanding of tFhe quality factor. Furthermore, another challenge of requirements quality research is the proposal of shallow quality factors neglecting practical relevance due to insufficient or anecdotal evidence [6].

Approach. We take the first step at tackling these problems by defining requirements quality factors, their related elements like data sets and automatic detection approaches, and the relationships between the elements. These elements and their relationship constitute our *domain* of interest. Next, we formalize this domain into an *ontology* where each element is represented by an individual *taxonomy* initially derived from literature. Using a set of 105 primary studies from the area of empirical research on requirements quality [6], we rigorously improve the structure of the ontology by applying established guidelines [58] and extracted eligible *objects* to populate the ontology. Finally, the refined structure of the ontology as well as all extracted objects are stored centrally in a repository and visualized through a connected web interface.

Structure. After discussing related work in Section 2, we elaborate the long-term objectives of this research direction in Section 3 by explaining the domain of requirements quality factors as well as the corresponding ontology development method in Section 4. Section 5 describes the first step taken towards this long-term objective in the scope of this work by presenting the process and results of the first prototype at ontology development. Challenges are outlined in Section 6 before calling for action in Section 7 to involve the requirements quality research community

in a joint effort at advancing and maintaining a harmonized vision of requirements quality before concluding in Section 8.

2 Related Work

The concept of requirements quality factors has been implicitly used in many publications over the last years: Femmer et al. [130], for instance, introduce nine *requirements smells*, which indicate quality violations in textual requirements. Din and Rine [145] propose a metric for requirements complexity, which is referred to as a *requirements indicator*. Ormandjieva et al. [156] gather several *quality characteristics* to define the quality of requirements text. We continue using the term *quality factor* which was applied in this context by Femmer et al. [2], since the term avoids the negative connotation that for example *requirements smell* evokes, opening the concept of quality factors up to also represent positive impacts on requirements quality, and since the term is well-embedded into a larger context of requirements quality [2].

Several sets of requirements quality factors have already been proposed in literature, among which are–as previously mentioned–the requirements smells proposed by Femmer et al. [130], the quality user story framework introduced by Lucassen et al. [157], and the framework for quality measurement developed by Génova et al. [158]. Previous attempts at establishing a subject-based classification for requirements quality are to the best of our knowledge limited to an approach by Saavedra et al. [159], which is, however, on a coarser granularity and elicits only high-level requirements quality aspects like *correctness*, *completeness*, and others. The work most comparable to our approach has been conducted by Femmer et al. [30], where 129 industrial requirements writing rules were classified regarding their eligibility for automation. Our own work differs from theirs in that (1) we aim at integrating quality factors established in peer-reviewed literature instead of in industrial writing rules [30] into a holistic ontology, while (2) considering the eligibility of the individual factors for automation only as one of many sub-goals. Further (3), as our endeavour shall lay the groundwork for a long-term community initiative, one main contribution is to publicly disclose all of our results for an effective maintenance and evolution of the ontology by the community.

3 Long-Term Objective

We begin by framing the long-term objective of our initiative. While this objective is out of scope of this paper, it guides the design and implementation of the prototype.

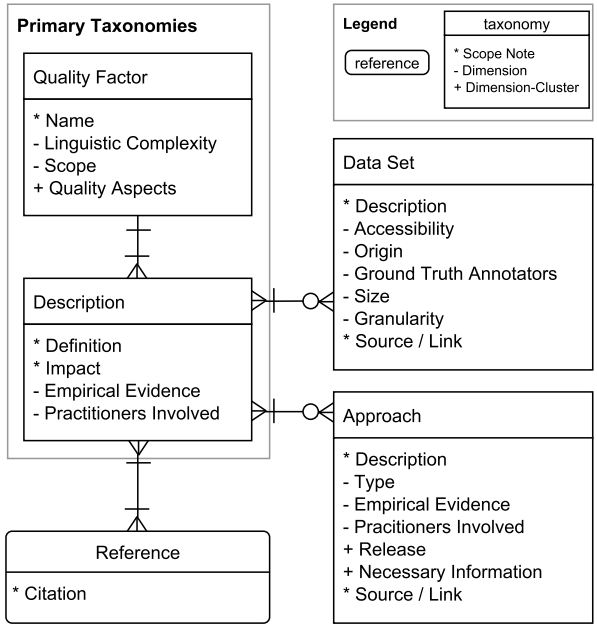


Figure 1: Schema of the ontology structure in Crow's foot notation

3.1 Establishing a Requirements Quality Factor Domain

Harmonizing the perspectives on requirements quality factors presupposes understanding the domain of related elements in which factors are embedded. We conceptualize four relevant elements during initial investigations of the available literature (see Figure 1). We consider a *requirements quality factor*–as described in the introduction–as a normative metric which maps a textual requirement at a specific level of granularity to a scale which informs about the quality of this input, where the level of granularity represents different ranges of text (e.g., words, sentences, or documents) and the scale is an often binary categorization of whether the factor has a positive or negative impact on specific aspects of quality (e.g., ambiguity, consistency) [159]. The lack of an explicit definition of this concept so far, however, led to quality factors only being referred to implicitly in literature. This resulted in the abstract concept of quality factors being instantiated predominantly as *descriptions* of varying levels of formality in literature. Consequently, the abstract element of a quality factor is related to one or more description elements which define the factor.

Evaluating textual requirements artifacts against these descriptions of quality factors is a way of estimating the quality of the requirements. In several cases, this evaluation can be automated: in our domain, we denote an approach for automatically detecting violations against a quality factor as an *approach*, which is associated to at least one description since approaches often automate the detection of several quality factors. These approaches are evaluated on *data sets*, which may have information about certain quality factors embedded into them, for example through the annotation

of violations against a set of quality factors.

We deem these four elements relevant for achieving our objective: the quality factors serve as a conceptual anchor for our objects of interest, descriptions make the factors tangible and comprehensible, and data sets as well as approaches facilitate application and reuse of the factors. We do not claim exhaustive completeness of the domain but rather use it as a starting point for the first iteration. Hence, we focus on extracting these four types of elements from existing literature.

3.2 Guiding the Ontology Development

In order to formalize these domain elements, we select the simplest subject-based classification system capable of representing the domain elements and their relationships [160]. Hence, we formalize our domain of interest as an ontology, where each element is represented by a taxonomy. All *objects* contained in each taxonomy are classified in a fixed number of *dimensions* specific to that taxonomy. As visualized in Figure 1, an object contained for example in the *quality factor* taxonomy is classified among others by the dimension *scope*. Each object takes exactly one value per dimension, where the set of all possible values of a dimension is called the *characteristics* [160]. The dimension *scope* contains the characteristics *word*, *sentence*, and others. We denote the collection of all dimensions and characteristics of a taxonomy as its *structure*. The structure of the ontology is the collection of all taxonomy structures.

Strictly categorical dimensions are not able to represent certain attributes of an object in the context of our ontology. First, references between two objects of different taxonomies require indexing each object, where indices are not meaningful characteristics. Second, textual attributes like a natural language description of a quality factor can also not be represented by a finite set of characteristics. We therefore extend the attributes of our subject based-classification by indices as well as *scope notes* as commonly used in thesauri [160], which enables a proper description of objects. Each quality factor object for example contains a scope note *name* to associate the object with a unique label.

For the sake of brevity in notation we also introduce *dimension-clusters*, which consist of a list of dimensions and a list of characteristics, where the latter applies to each dimension. A dimension-cluster abbreviates similar dimensions, e.g., the dimension cluster *quality aspects* of the quality factor taxonomy contains dimensions like ambiguity, complexity, and verifiability, which all can take the characteristics *impacted positively*, *impacted negatively*, or *not impacted* individually.

We translate the rigorous taxonomy development guideline proposed by Nickerson et al. [58] and extended by Kundisch et al. [161] to the larger scale of our ontology. None of the aspects in which our ontology design extends the design of a taxonomy contradicts the abstract guidelines, since (1) the ontology simply consists

of four individual taxonomies and (2) the extraction guideline is applicable to the additionally included index and scope note attribute as well.

4 Provisional Ontology Design

We take a step towards our long-term objective by defining requirements and exit criteria of the ontology creation process.

4.1 Meta-Characteristics

As defined in the guideline [58], the design of the classification system is rooted in the identification of the users which are intended to use the ontology, and their goals, which these users are supposed to achieve with the ontology. We consider two types of users to interact with the ontology: *researchers* dedicated to advancing the field of requirements quality research and *practitioners* aiming to apply results emerging from this research in order to evaluate the quality of their requirements artifacts. The goals of these two abstract users constitute the meta-characteristics [58] and represent the high-level requirements for the ontology. The following goals (G) are formulated in the user story template:

- **G1:** As a researcher or practitioner, I want to find explanations to available requirements quality factors so that I understand how they inform about requirements quality.
- **G2:** As a researcher or practitioner, I want to find available resources connected to a quality factor so that I can reuse these resources for my own work.
- **G3:** As a researcher, I want to identify gaps in literature so that I can tailor my own research to provide valid contributions.
- **G4:** As a researcher or practitioner, I want to find who is working on specific quality factors so that I can establish a collaboration.

These goals strictly apply to the *ontology*. The obvious, overarching goal to evaluate the quality of requirements artifacts applies to the *quality factors* and is independent of our goals.

4.2 Exit criteria

The exit criteria, which indicate the completeness of the iterative ontology development method, are also derived from Nickerson et al. [58]. In this, we aim to achieve the following objective ending conditions (condensed from [58]): (1) All objects or a representative sample of objects have been examined, (2) no object, dimension

or characteristic was merged or split in the last iteration, (3) at least one object is classified under every characteristics of every dimension, (4) no new dimensions or characteristics were added in the last iteration, and (5) every dimension is unique in every taxonomy of the ontology and every characteristic is unique in its dimension.

By documenting all changes to the ontology in each iteration, we can objectively decide when these ending conditions are met. We have explicitly excluded the objective ending criterion "Each cell (combination of characteristics) is unique and is not repeated" [58], since the extension of the ontology as described in Section 3.2 entailed the inclusion of attributes that are not dimensions, i.e., indices and scope notes. Two objects can hence have the same combination of characteristics among all dimensions, but be distinct due to different scope note values. In addition, we also aim to achieve the following subjective ending conditions [58]:

- **Concise**: the number of dimensions needs to be meaningful yet manageable
- **Robust**: the dimensions and characteristics need to provide for differentiation among objects of interest
- **Comprehensive**: all objects within the domain of interest can be classified
- **Extendable**: new dimensions and characteristics can be easily added
- **Explanatory**: the dimensions and characteristics explain the objects

5 Prototype of the Ontology

We developed a prototype of the ontology to (1) illustrate the usability of the structure, repository, and tool, and to (2) contribute the first step towards the long-term objectives.

5.1 Iterative process

We illustrate the approach outlined in Section 3.2 by adopting the iterative ontology development process as extended from Nickerson et al. [58]. Accordingly, either an empirical-to-conceptual or conceptual-to-empirical approach has to be chosen. We chose the former approach for the initial iteration, as a significant understanding of the domain has already been established along previous engagement with requirements quality research. We distilled an initial structure of the ontology based on relevant literature [30, 159, 162].

An empirical-to-conceptual approach was chosen for the subsequent four iterations, as we aim to extract eligible objects for the four taxonomies from established literature. For this prototype, we selected the set of primary studies gathered in a recent systematic mapping study on empirical requirements quality research by Mont-

gomery et al. [6] as our data to extract from. This publication is the only secondary study to our knowledge which explicitly investigates requirements quality and, thus, serves as a reliable collection of peer-reviewed primary studies. The three first authors distributed the set of 105 eligible studies among each other and split the resulting subset into four iterations. During each iteration, they extracted all relevant objects based on an extraction guideline, which was initiated during the first iteration and maintained according to ontology development protocol [58, 161]. Publications had to at least contain one eligible quality factor based on the definition established in Section 1 and an according description. Data set and approach objects were extracted when eligible according to the extraction guideline. At the end of each iteration, the three extracting authors convened together with the fourth author and discussed necessary changes to the ontology structure in case objects were encountered which could currently not be framed by the taxonomies.

The set of references in [6] is heavily biased towards empirical work. To confirm that the ontology is also robust when considering non-empirical work, we conducted a final iteration considering publications that were excluded in the reference selection phase of [6]. The inclusion of non-empirical work, e.g., [163], did not challenge the structure of any taxonomy, strengthening our confidence in the robustness of the ontology.

After this final iteration, all relevant exit criteria were fulfilled, which indicated the completion of the ontology creation process in the scope of this work. The objective ending conditions were fulfilled as the documentation of the final iteration of the protocol showed no violation against any of the five conditions. The subjective ending conditions were assessed and agreed upon by the first four authors to a reasonable extent of this prototype; for example, the ontology was deemed *concise* since the number of dimensions of each taxonomy is compliant with the seven plus two rule [58, 164], and *robust* since the inclusion of non-empirical work did not challenge any taxonomy structure. The final assessment of the subjective ending conditions applies to the future version of the ontology and will be discussed in the outlook in Section 7.2.

5.2 Current State

The schema of the ontology structure at the current stage of development is shown in Figure 1 with the structure and relationship between all four included taxonomies. A thorough explanation of all attributes (dimensions, dimension-clusters, scope notes, and references), the eligible characteristics, and their corresponding extraction rule can be found in our replication package[1] and on our web interface[2]. We limit the following explanations to the most important of these attributes and illustrate them

[1] Replication package at `https://doi.org/10.5281/zenodo.6583690`

[2] The application can be accessed at `http://www.reqfactoront.com`

with a running example.

Requirements quality factors are characterized by a *name* and *scope*, which is the dimension representing the granularity of input necessary in order to decide the quality factor. As an example, one publication by Femmer et al. [30] contains multiple quality factors, one of which is named *containing subflows*. The scope of this factor is *use case*, as a full use case is necessary to decide whether at least one subflow is contained or not. Quality factors are further characterized by the dimension-cluster quality *aspect*: it is necessary to denote the impact which the calculated value of a quality factor has on the activities in which the requirement is used, which is framed by the notion of activity-based requirements quality [2]. The factor *containing subflows* is reported to have a *negative* effect on the aspect *understandability*, because subflows "force the reader to jump between different positions in the text in order to read through the use case, which can be argued to lead to less readable use cases that are harder to understand" [30], but also a *positive* effect on *maintainability*, because "if parts of the flow change, they only need to be changed in one location (the subflow), and not in each use case" [30]. The notion of aspects is further explained in Section 6. The set of quality aspects is a harmonized superset of aspects used in established literature [6, 159, 165, 166]. We make no claim about the completeness and granularity of this set, as we consider quality aspects as a connected, but distinct element in a larger domain. Using a harmonized superset, we provide an interface for future research in this subsequent domain of requirements quality, for example, on their interrelationships [162].

Descriptions are instantiated by a scope note for both the *definition* of the quality factor and also its *impact*. Since a rigorous framework of requirements quality factors has been absent in requirements quality research, textual descriptions of what a quality factor means and how it impacts subsequent development activities are most common. The factor *containing subflows* is defined as "Subflows are mechanisms for reuse that enable the author of a use case to extract a certain set of steps into a reusable subflow to prevent copy-and-paste reuse [...] in the use cases" [130], while the impact is described as mentioned in the previous paragraph about aspects. Description objects are further annotated on whether the according publication provides *empirical evidence* for its relevance and whether *practitioners were involved* in its inception or development, as these dimensions help identifying quality factors that are empirically informed. Since all quality factors in [30] were derived from an industrial requirements writing guideline, they explicitly have *practitioners involved* and their use in practice serves as *empirical evidence*.

Data sets are characterized by their *origin*, which reflects whether the data is from industry, academia, or mocked, and who embedded the information (called *ground-truth annotators*) of quality factor violations in the data, i.e., whether the authors themselves, practitioners, or students annotated the violations. Femmer et al. [30] report on one data set from a large software project at a German reinsurance company, whose *origin* is *practitioner data*. Since the data bears no annotations,

the data set has no ground-truth annotators. The *size* and *granularity* of the data set represents the number and type of contained elements. The aforementioned data set contains 51 objects of the granularity *document*. Finally, the *accessibility* of a data set reflects to what degree the data set can be used. If available, the corresponding link or reference to the source is given. The described data set is *private* and has no link or source given.

A similar approach for characterizing the *accessibility* is done for an approach object, in addition to the type of *release* (source code, tool, API, or other). Approaches are further characterized by their *type* (rule-based, machine learning or deep learning) and the *necessary information* utilized to conduct the automatic evaluation (e.g., POS tags, dependency tags, or other). Finally, approaches are–similar to descriptions–classified regarding the *empirical evidence* they provide and whether *practitioners were involved* in the evaluation. The approach *Smella*, described in another publication by Femmer et al. [130], is a *proprietary tool* detecting smells with a *rule-based* algorithm using *POS tags* and *lemmatization*.

In the following paragraphs we describe how the current state of the ontology and its contained objects address the meta-characteristics. All conclusions are drawn based on the limited subset that was selected for this prototype [6], hence the inferences are not necessarily universal. In addition, the conclusions are currently limited to a quantitative evaluation of the ontology's structure and content. A qualitative evaluation involving the intended users of the ontology will be necessary to determine its usability.

Addressing G1. The association of a quality factor object with at least one description object provides an overview over all proposed explanations of a quality factor. Out of the 105 primary studies from the initial set [6], 59 contained at least one eligible quality factor. In total, 206 unique quality factors were extracted and associated to 258 descriptions. Consequently, 172 quality factors are associated to exactly one description. On the other hand, nine quality factors were described in three or more occasions: anaphora, coordination ambiguity, vagueness, passive voice, referential integrity, subjectivity, nocuous ambiguity, multiple interpretations, and consistency.

Addressing G2. The association of a quality factor object with data set and approach objects allows to find available resources for reuse and evolution. The 105 primary studies describe 56 unique data sets and 36 approaches. However, only 9 of 56 data sets are publicly available (i.e., have the characteristic *available in paper* or *open access link* in the dimension *accessibility*), while most data sets are either private or not disclosed. Only 5 of the 36 approaches are publicly available (i.e., have the characteristic *open access* or *open source* in the dimension *accessibility*), while most approaches are not disclosed at all. These numbers highlight a dire condition of open source in the requirements quality research landscape, which inhibits the use and reuse of existing resources. Filtering by the dimension *accessibility* supports identifying available resources and exposing undisclosed contributions.

Addressing G3. As well as achieving G2 in the aforementioned way to identify which quality factors are not yet annotated in a data set or automatically detected with an approach implementation, the dimensions *empirical evidence* and *practitioners involved* can be used as a filter to identify objects that lack empirical validation. Out of the 258 descriptions, only 82 are devised based either on empirical evidence, i.e., by assessing how well the metrics correspond to the subjective perception of requirements quality in a survey [167], or by involving practitioners in the design process of the quality factor [136]. In addition, only 92 of the extracted 258 description objects contain an explicit *impact* of the quality factor. The significance of this lack of an impact description is further emphasized in Section 6.

Addressing G4. The association of description, data set, and approach object to references allows to trace every contribution to the corresponding authors, which can be used to connect to researchers who have contributed in a specific area of requirements quality research.

5.3 Repository and Tool

The initial results are recorded in a first version of a maintainable tool: both the structure and the objects of the ontology are stored in a publicly accessible data repository hosted on GitHub[1]. The structure is represented by structure files defining the attributes of each taxonomy. The objects are stored in form of *extractions*, where each extraction is associated to one reference and contains an arbitrary number of extracted objects according to the existing taxonomies. The current status of the repository is retrieved by an interactive web application which processes the data and visualizes it in a human-readable and -comprehensible way[2], fulfilling the elicited goals through filters and links. The repository can be easily maintained using the version control offered by GitHub: contributions to both the structure and the content of the ontology can be made by adding new extraction elements for either existing or new references. This way, new publications can be included or already included publications can be revised, supporting an inclusive and collaborative approach at harmonizing the perspective on requirements quality factors.

6 Threats and Challenges

Transparent Ontology Design Process A major challenge in developing any subject-based classification is the lack of transparency of the process [161], where the process obscures the rationale behind design decisions. Since our ontology is both meant to facilitate collaboration and a community-driven maintenance and evolution, we mitigate this threat by disclosing all process documentations[1].

Shared Understanding of Extraction Guidelines As with any extraction task, the subjective nature of interpreting literature according to an extraction guideline is inherently prone to misunderstandings. Even though the initial set of extracted objects are neither the main contribution of this work nor assumed to be permanent, we assured a common understanding by assigning primary studies which were already processed by one author to another author in order to calculate an overlap and quantify the agreement. This way, each of the first three authors additionally extracted relevant objects from two already processed primary studies, such that every extractor had an overlap with every other extractor. All relevant attributes were evaluated: dimensions were assessed by equivalence, scope notes were assessed by similarity using sequence matching scaled to range [0, 1]. The six primary studies resulted in 799 extracted individual values, on which an agreement of 85.03% between all authors was achieved. This agreement assures a sufficiently common understanding of the extraction guidelines.

Requirements Quality Research Framework As mentioned in Section 1, requirements quality factors are purely normative and evaluate textual input based on metrics which are often arbitrary. The relationship between these metrics and the actual impact on the quality is more complex: as identified in previous research on specific quality factors [34], a violation against the rule entailed by a quality factor may or may not lead to an actual impact on the requirements quality depending on numerous context factors. For example, the use of passive voice might not lead to an ambiguous interpretation in a small-scale development unit if the stakeholders which are intended to use the written requirement can reconstruct the omitted subject of the sentence anyway.

This relationship has been framed by Femmer et al. in the form of activity-based requirements engineering quality models [2], where a violation against a quality factor only potentially leads to an impact on an activity in which the requirement is meant to be used. The relevance of a quality factor is dependent on the likelihood of an impact on subsequent activities under the given context factors.

This imposes a necessary interface on the requirements quality factor ontology: ultimately, every quality factor should be associated to a specific impact on specific activities given specific context factors in order to determine the relevance of the factor. The state of research in this respect is currently relatively poor, as shown in the preliminary results of Section 5.2, and most publications proposing quality factors are satisfied with determining the impact of a factor based on educated guesses or anecdotal evidence. Therefore, our ontology currently only records explicitly stated impacts in the dimension-cluster *quality aspect* of the taxonomy *quality factor*. However, improving the information about the potential impact of quality factors is an anticipated extension point of our ontology once research in this domain has advanced.

7 Limitations and Call for Action

We discuss the limitations of the prototype to highlight the distance between this first step and the long-term objective. Further, we propose a community effort to bridge this distance.

7.1 Limitations of the current Approach

Incompleteness of Publications The lack of a shared terminology impedes identifying what the complete set of publications would be, as quality factors have been addressed with different names and in different approaches. Hence, the list of publications to extract eligible objects from is far from complete. The systematic mapping study on empirical requirements quality research by Montgomery et al. [6] is to our knowledge the only secondary study which makes an attempt at comprehending the research domain of requirements quality. Currently not considered publications could potentially add relevant objects to the ontology or challenge its structure. Since the domain of requirements quality research is only loosely coherent by an explicit identity, the effort to comprehend and order relevant research is, as we argue, an extensive undertaking.

Overload of factors The result addressing goal G1 presented in Section 5.2 raises the question about the relevance of this large number of unique quality factors. The included publications show a large variation in the degree of evidence for their relevance, as also noticed by Montgomery et al. [6], which ranged from purely anecdotal justifications over references to established literature [168] to sound empirical evaluations [34]. We decided not to exclude publications with lacking evidence of relevance at the cost of a manageable number of resulting factors, mainly because no mature research approach to reliably determine a quality factor's relevance exists yet.

7.2 Call for Action

One hope we associate with this RE@Next! contribution is to appeal for participation in a coordinated community effort aimed at tackling this task. The extension of this task to a community effort makes the extensive undertaking of identifying all relevant literature surmountable. In addition, it ensures to include diverse perspectives on the matter, contributing to establish a harmonized vision. This will additionally lead to healthy scrutiny and subsequent evolution of the ontology structure, for example by including the dimension *language* for quality factors, as publications discussing quality factors in languages other than English begin to emerge [169]. Finally, involving as many parts of the implicit requirements quality research community as possible is bound to establish an explicit, shared identity of the research domain in the process.

The community effort will be initialized by interested members of the require-

ments quality research community committing to it. We anticipate this effort to span over several years, though a consistent commitment is not mandatory. Coordinated by the first authors of this paper, systematic strategies for identifying previously not considered publications will be developed, distributed, and executed. Once confidence in the completeness of the publications will have been reached, the iterative ontology creation process described in Section 5.1 will be scaled up and continued by the members involved in the community effort. The ultimate deliverable of this community effort will be a sufficiently complete and robust ontology structure and content–assessed jointly using the objective and subjective exit criteria–which reflects the harmonized perspectives of the requirements quality research community.

This also lays the groundwork for addressing the relevance-problem of requirements quality publications: after the space of quality factors has expanded during the community effort, this same community shall be involved in developing a reliable research approach for determining the relevance of a quality factor. This method will be used to condense the space of quality factors again to a manageable number of relevant objects, addressing the second limitation mentioned in the Section 7.1. Finally, a complete yet concise set of applicable and relevant quality factors contained in the final version of the ontology fulfilling goals G1-G4 can be delivered.

8 Conclusion and Outlook

This paper presents the long-term objective of a harmonized vision on requirements quality factors in the form of an ontology, relating four taxonomies to represent the four elements quality factor, description, data set, and approach of the domain containing quality factors for textual requirements. The extraction of eligible objects from 105 primary studies as well as a central repository and accessible web interface are the first step towards this long-term objective.

Establishing a harmonized perspective on the structure of quality factors and related elements as well as a central repository containing a sufficiently complete set of relevant objects is an extensive task necessitating a community effort, making this task surmountable and also including diverse perspectives on the domain. The final version of the ontology will then serve as a conceptual framework for future research, a reliable resource for practitioners to base requirements quality assurance on, and a tool for requirements quality education.

Paper III

Measuring the Fitness-for-Purpose of Requirements: An initial Model of Activities and Attributes

Abstract

Requirements engineering aims to fulfill a purpose, i.e., inform subsequent software development activities about stakeholders' needs and constraints that must be met by the system under development. The quality of requirements artifacts and processes is determined by how fit for this purpose they are, i.e., how they impact activities affected by them. However, research on requirements quality lacks a comprehensive overview of these activities and how to measure them. In this paper, we specify the research endeavor addressing this gap and propose an initial model of requirements-affected activities and their attributes. We construct a model from three distinct data sources, including both literature and empirical data. The results yield an initial model containing 24 activities and 16 attributes quantifying these activities. Our long-term goal is to develop evidence-based decision support on how to optimize the fitness for purpose of the RE phase to best support the subsequent, affected software development process. We do so by measuring the effect that requirements artifacts and processes have on the attributes of these activities. With the contribution at hand, we invite the research community to critically discuss our research roadmap and support the further evolution of the model.

Keywords: Requirements Engineering, Requirements Quality, Literature Review, Interview Study, Activity

1 Introduction

Requirements engineering (RE) is a means to an end and aims to fulfill a purpose, i.e., to inform subsequent activities of the software development life cycle about the

needs and constraints of relevant stakeholders [1]. Therefore, requirements artifacts and processes must be fit for purpose. This fitness for purpose is determined by the attributes of the software development activities that are affected by requirements artifacts or processes [40]. For example, a requirements specification is considered fit for purpose when *implementing* (activity) its implied features works *correctly*, *completely*, and *quickly* (attributes), among other attributes. In that sense, we should judge the quality of requirements (and RE) based on the extent to which they are fit for purpose, i.e., how they impact the attributes of requirements-affected activities [2]. Still, research on requirements quality is dominated by studies aiming to determine the quality of a requirements specification solely based on normative metrics [41].

Recent endeavors to nuance requirements quality research with this activity-based perspective are promising [1, 2], but have so far not seen adoption in practice [40]. One reason for this is the lack of an overview of software development activities that are affected by requirements engineering as well as their measurable attributes. This gap was acknowledged in previous requirements quality research [2, 170] and is one milestone on requirements quality research roadmaps [1, 40]. The overview of the activities that are potentially affected by RE would offer guidance on which activities determine the fitness for purpose of RE processes and artifacts. Furthermore, an overview of the activities' attributes would offer guidance on how to measure their performance. Consequently, we formulate the following research questions:

- **RQ1**: Which software development activities are affected by requirements artifacts?

- **RQ2**: By which attributes are requirements-affected activities evaluated?

This paper initializes the endeavor to create and maintain an overview of requirements-affected activities and attributes answering the research questions. As the first step, we inductively construct an initial model from three distinct data sources (Section 3). The model contains 24 activities like *implementing*, *testing*, and *estimating effort*, and characterizes them with 16 attributes including *duration*, *completeness*, and *correctness* (Section 4). The paper further describes how to apply the model in research and practice and how future research will advance the endeavor (Section 5). We disclose all material, data, and source code[1] to facilitate this community endeavor.

[1] Archived at `https://zenodo.org/doi/10.5281/zenodo.10869626`

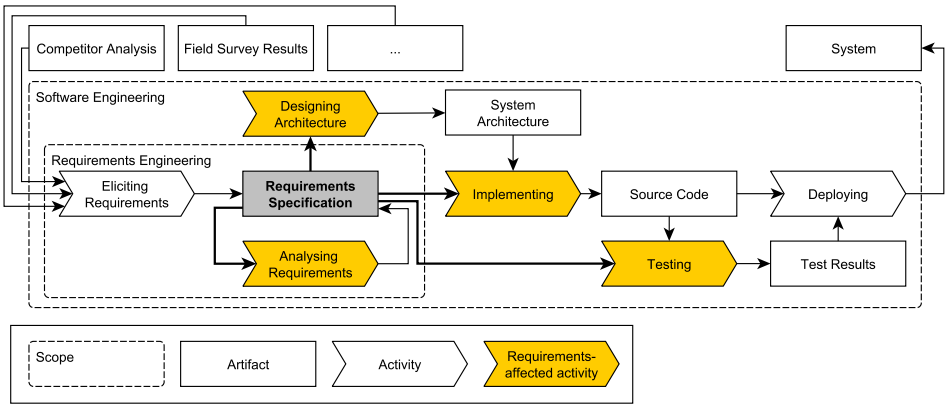


Figure 1: Simplified example of SE-relevant activities

2 Background and Related Work

2.1 Requirements Use in SE

We consider as an *activity* any SE-relevant process performed by a (human or software) agent that uses one or more input artifacts and produces one or more output artifacts [2]. Figure 1 visualizes a simplified overview of SE activities, the artifacts they use as an input and produce as an output, and their scope. For example, the *implementing* activity receives several input artifacts like a requirements specification and system architecture to produce output artifacts like source code.

We consider an activity *requirements-affected* if at least one of its input artifacts is a requirements artifact (yellow activities in Figure 1). The aforementioned implementing activity is requirements-affected because it considers a requirements specification as an input. In the simplified example in Figure 1, the requirements elicitation and the deployment activity are not requirements-affected. It is, however, possible that the requirements elicitation activity may be affected by requirements artifacts of previous projects and sprints or that explicit deployment requirements exist.

2.2 Requirements Quality

Since requirements artifacts are used as input to requirements-affected activities, the artifacts' quality affects the quality of these activities and their output [21]. For example, a vague requirements specification may lead to incorrect or missing features and reduced customer acceptance [18]. These quality defects are more expensive to fix the later they are addressed [3]: Revising a vague requirements specification is less expensive than redeveloping a faulty system built on it. Therefore, organizations aim to detect and remove requirements quality defects as early as possible [6].

However, requirements quality research focuses predominantly on normative quality factors [41] that do not consider an impact on affected activities [6, 40]. For example, the use of *passive voice* is often advised against in literature [33, 158, 171] despite a lack of empirical evidence for its negative consequences [34, 81, 172]. This fosters skepticism of organizations to adopt requirements quality research [9, 173].

To address this issue, Femmer et al. proposed the perspective of *activity-based* requirements quality [2]. This perspective entails that requirements are only as good as they support the activities in which they are used [1], i.e., requirements quality depends on the performance of requirements-affected activities. Specifying requirements quality as fitness-for-purpose to support affected activities necessitates requirements quality research to understand requirements-affected activities, i.e., it requires identifying and measuring activities affected by a requirements artifact [40].

Without a systematic elicitation of requirements-affected activities prior to investigating the quality of a requirements artifact, researchers risk drawing incomplete conclusions. For example, Ricca et al. investigate the effect of screen mock-ups on requirements comprehension [174] and conclude that providing screen mock-ups improves the understandability of requirements. Femmer et al. confirm this effect but contrast that they simultaneously have a negative effect on requirements maintainability [2]. Systematic studies on activity-based requirements quality agree that an overview of requirements-affected activities and their attributes is necessary to advance relevant requirements quality research [1, 2, 40].

2.3 Related work

Requirements engineering literature contains several studies about the impact of requirements quality on subsequent software development activities. For example, Kamata et al. [31] and Zowghi et al. [175] empirically investigated the impact of requirements quality on project success measured in time and cost overrun. Similarly, Knauss et al. studied the impact of requirements quality on project success measured by customer satisfaction [176]. These studies generalize the affected activities and summarize their effect on the overall project outcome.

Studies focusing on more specific activities include Chari et al. investigating the impact of requirements defects on injected software defects [177], and Femmer et al. relating the use of passive voice to the domain modeling activity [34]. On the other hand, some studies expand the scope of affected activities. Damian et al. conducted a longitudinal case study observing a full project development lifespan and measured the tradeoffs of a revised RE process on several activities like communication, effort estimations, and implementation [101]. Mendez et al. conducted a large-scale, global survey of perceived problems in RE and their effects on activities, including designing, implementing, and organizing [18].

Research on traceability between software development artifacts constitutes an-

other closely related domain. Several secondary studies have summarized traceability research and identified artifacts that are commonly connected [178, 179]. Although requirements artifacts are prominent targets of trace links, they are typically connected to other artifact types, not the activities that produce them [178]. These artifact types can be used to infer the producing activities, though the inferred activities typically remain on a very high level [179]. Furthermore, this limitation excludes by design all activities that do not necessarily or only rarely produce artifacts, like, for example, informal reviewing, modifying existing artifacts, assessing feasibility, or estimating effort.

In summary, none of these previously mentioned primary studies systematize the affected activities and their attributes but rather select the studied impact based on the availability of data or anecdotal hypotheses, and traceability research exhibits significant limitations regarding the identification of these activities. Only two studies known to the authors attempt to explicate the affected activities. Femmer et al. elicited the activities affected by specific requirements artifacts at a case company and determined the qualitative impact of requirements defects on them [2]. In a similar study, Frattini investigated requirements quality factors relevant to a case company and their impact on subsequent activities [170]. Both studies prototype a model of requirements-affected activities for the specific context but acknowledge the need for a more systematic and comprehensive overview.

3 Goal and Early Method

One goal of activity-based requirements quality research is to create and maintain a comprehensive model of requirements-affected activities and their attributes exhibiting the following properties [1, 40]:

1. **Applicability**: The model can represent all requirements-affected activities and attributes in any given SE context.
2. **Suitability**: The model can be used to evaluate relevant activities by means of their attributes.
3. **Extensibility**: The model can be extended with new activities or attributes.
4. **Usability**: The model can be accessed and comprehended by software engineers.

In this study, we contribute the first version of this model. Since we are not aware of any systematic prior work collecting requirements-affected activities and their attributes [1, 40], we surveyed different data sources for textual descriptions of SE activities that use requirements artifacts as input. From these textual mentions, we

inductively construct a model of requirements-affected activities and their attributes by employing thematic synthesis as proposed by Cruzes and Dybå [61].

3.1 Data Collection

To ensure the property of applicability as mentioned above, we collected data from three distinct sources described in the following three subsections: a systematic review of experimentation literature (Section 3.1.1), an interview study (Section 3.1.2), and a literature study on software process models (Section 3.1.3).

3.1.1 Systematic Literature Review

The first source of textual descriptions of requirements-affected activities and their attributes that we considered were scientific studies reporting controlled experiments in which the experimental task involves human subjects and considers requirements as an input artifact. These experimental tasks simulate requirements-affected SE activities performed by practitioners. The dependent variables in these experiments are eligible attributes describing the performance of the activity. We adopted the systematic literature survey method employed by Sjøberg et al. [59].

Database selection. To ensure that our database search for eligible primary studies targets publications relevant to SE we pre-selected eligible journals and conferences (from hereon out collectively called *venues*) from the CORE ranking[2] whose field of research is software engineering. To ensure a high quality of the primary studies, we constrained the venues to those of rank A* or A. A few select venues of lower rank that are particularly relevant to the topic constituted an exception. These included the *Requirements Engineering Journal*, the *Journal of Software: Evolution and Process*, the *International Working Conference on Requirements Engineering: Foundation for Software Quality*, the *International Conference on Product-Focused Software Process Improvement*, and the *Euromicro Conference on Software Engineering and Advanced Applications*, which all have a core rank of B. Additionally, we removed all venues that host computer science rather than SE topics. This task was performed by three authors in conjunction to ensure reliability. The final database selection contained 35 venues (10 journals and 25 conferences).

Database search. We performed a keyword-based database search for each included venue with the keywords *experiment** as well as *requirement** (or the synonyms *srs* or *specification**). These keywords limited the retrieved primary studies to those (1) describing an experiment and (2) involving requirements at least to some degree. We executed the database search via Scopus[3] and in four cases, where Scopus did not index publications of that venue, via the ACM Digital Library.[4] The search

[2] `https://www.core.edu.au/`

[3] `https://www.scopus.com/search/form.uri?display=advanced`

[4] `https://dl.acm.org/`

string per venue consisted of the two sets of keywords as well as a limitation to the venue via its title. For example, the search string for the ACM Computing Surveys journal in Scopus looked as follows: `SRCTITLE ( computing AND surveys ) AND TITLE-ABS-KEY ( requirement* OR srs OR specification* ) AND TITLE-ABS-KEY ( experiment* )`. The search per venue returned between 1 (e.g., from the *European Conference on Object-Oriented Programming*) and 175 (from the *Journal of Systems and Software*) primary studies for a total of 1446 studies.

Inclusion. Next, we performed an inclusion phase to ensure the following properties of primary studies expressed by the two inclusion (I1 and I2) and four exclusion criteria (E1-E4):

- I1: The primary study presents an experiment with human subjects as one of its core contributions.
- I2: The experimental task uses a requirements specification as an input.
- E1: The experimental task is a requirements review.
- E2: The study is not written in English.
- E3: The publication is not available via the university's access program.
- E4: The study is a duplicate of or extended by an already included study.

I1 ensures that eligible primary studies present a proper experiment (regardless of whether it is controlled or quasi) that involves human subjects. Otherwise, the experimental task would not simulate an SE activity, the concept of interest. This excludes, for example, experiments in which machine learning algorithms of different configurations are compared on a classification task. I2 ensures that the activity is requirements-affected. E1 explicitly excludes requirements review tasks, i.e., requirements defect detection and removal activities. The purpose of identifying requirements-affected activities is to optimize the affecting requirements in a way that improves their impact on the activities. This optimization process is the requirements review. Hence, we excluded these studies to avoid a circular impact, i.e., suggesting to optimize requirements for the reviewing activity, which is exactly this optimization. E2 and E3 exclude inaccessible studies, and E4 removes content duplicates. Primary studies were considered for the next data analysis step when they met all two inclusion and none of the exclusion criteria. The first author conducted the inclusion step based on the title, abstract, and keywords. Out of 1446 primary studies, 145 (10.3%) were included. To ensure the reliability of this subjective process, the second author independently performed the inclusion task on 75 (i.e., 5.2%) randomly selected studies. We calculate the inter-rater agreement using Bennett's S-Score [142], which is robust against uneven marginal distributions [141].

The inter-rater agreement yields a value of 92%, which we deem sufficient to instill confidence in this subjective task.

Data Extraction. The first author reviewed all 145 included primary studies and extracted, for each human-subject experiment in each study, (1) the description of the experimental task and (2) all dependent variables measured to evaluate the performance of the task. The description of the experimental task constituted the source of requirements-affected activities, and the dependent variables were the source of their attributes. While reviewing the full text of the studies, 22 studies revealed to not, in fact, meet all inclusion criteria other than the title, abstract, and keyword had suggested. We excluded these 22 studies from further processing.

Additionally, we excluded extractions where the attribute description did not *imply a valuation*. Because our goal was to identify attributes that quantify the *performance* of their respective activity, eligible attributes must be *valuating*—i.e., values of that attribute must imply a degree of performance. While attributes do not necessarily have to be measured on an interval scale (i.e., it is not important to associate an interval of the attribute, like a certain amount of minutes for the attribute *duration*, with a specific level of quality), it has to be at least on an ordinal scale—i.e., the sign of the interval is important (more minutes of duration is bad, less minutes of duration is good). For example, if the dependent variable of an experiment investigating the activity of *estimating effort* is the *estimated amount of hours* [180], then this data point(i.e., pair of activity and attribute) was excluded as a higher or lower value of that attribute does not automatically make it good or bad due to the lack of ground truth. If, instead, the dependent variable was *precision*, i.e., how close the estimated amount of hours is to actual implementation time, then the data point would be included as a higher value of precision (i.e., an estimation that is closer to the actual time) is better. This process eliminated 12 descriptions of non-valuating attributes. To assess the validity of this process, the third author independently repeated the task on a sample of 12 data points, which consisted of 6 random samples from each of the two classes (valuation vs. no valuation), and we measured the inter-rater agreement using Cohen's Kappa [181] since the classes have an even marginal distribution [141]. The first overlap achieved a Cohen's Kappa of only 33.3%, which emphasized the complexity of the task. The two authors reconvened, discussed the differences, reformulated the exclusion criteria, and repeated the labeling. The second overlap achieved a score of 83.3%, which represents a sufficient reliability of the step.

The extraction produced 142 descriptions of experimental tasks and 355 descriptions of dependent variables. Several experimental tasks were evaluated via multiple dependent variables, which is why the 355 resulting data points contain repeated descriptions of experimental tasks.

3.1.2 Interview Study

The second source of textual descriptions of requirements-affected activities and their attributes that we consider were reports from industry practitioners about the usage of requirements specifications in subsequent SE activities. To this end, we evaluated the transcripts of a previously conducted interview study [170].

Interview Participants. The first author conducted the interview study in a large, globally distributed software development organization that specifies requirements using both free-form and constrained natural language (use cases) prior to each development cycle. A contact at the organization provided a sample of eight software engineers directly responsible for processing requirements specifications and developing solution specifications based on them. These eight engineers represent the majority of personnel in their role in the team that was involved in the study. The interview participants had an average of 3.5 years of experience in their role, 7.5 years with the organization, and 15.3 years as software engineers.

Interview. The original purpose of the interview was to identify which quality defects practitioners perceive in the requirements specifications that they process [170]. Because the elicitation of quality defects entailed mentioning what kind of subsequent activity is affected by this defect, the generated data served to identify requirements-affected activities and their attributes. For example, stating that vague requirements lead to a delay of the testing phase contains the requirements-affected *testing* activity and its attribute *duration*. To guide the semi-structured interview, we developed a protocol. The protocol contained, among demographic questions, one prompt per type of requirements quality. The types of requirements quality were derived from Montgomery et al. [6] and covered, among others, ambiguity, completeness, and traceability.

Data Extraction. All eight one-hour-long interviews were recorded, automatically transcribed using a speech-to-text conversion tool,[5], and verified by the first author. Then, the first author extracted from the transcripts each mention of an activity affected by a requirements quality defect and how this effect was measured. The extraction produced 55 descriptions of affected activities but no descriptions of how this effect was measured on them.

3.1.3 Literature Study

The third source of textual descriptions of requirements-affected activities and their attributes that we consider were descriptions of software process models. Software process literature describes processes and products of the SE life cycle and, hence, contains information about which activities are affected by requirements. Since software process literature is fairly mature [182], we have access to reliable summaries of process models.

Literature. We selected the book “Software Process Definition and Manage-

[5] `https://www.descript.com/`

ment" by Münch et al. [183] as a reliable summary of software process literature. The first author reviewed the descriptions of all seven lifecycle models, which cover the waterfall model [184], iterative enhancement [3], prototyping, the spiral model [185], the incremental commitment spiral model [186], Unified Process [187], and Cleanroom Development [188]. The first author extracted all textual mentions of requirements-affected activities and their attributes as prescribed by the lifecycle model. This extraction produced 21 textual descriptions of activities and one explicit description of an attribute.

3.2 Data Analysis

Coding. The data collection phase over the three sources culminated in a table containing 218 textual descriptions of requirements-affected activities and 356 textual descriptions of their attributes. In the absence of a prior theory or model of requirements-affected activities, we resorted to an inductive coding process [61]. The first and third authors jointly established the level of granularity of the codes that were applied to the textual descriptions and documented this process in a guideline. The first author then performed the coding process independently and, upon completion, verified the assigned codes with the third author. For each pair of textual descriptions of an activity and attribute, we coded four concepts:

1. Activity: the requirements-affected activity
2. Activity attribute: a property evaluating an activity
3. Artifact: an output artifact produced by the activity
4. Artifact attribute: a property evaluating an artifact

The distinction of artifacts from activities was necessary since some activities were not evaluated directly but rather by the artifacts they produced. For example, *duration* is an attribute of the *implementing* activity, but several studies additionally evaluate that activity by measuring the *coupling* (artifact attribute) of the resulting *source code* (artifact).

Consolidation. The inductive coding process produced 24 unique codes for activities, 16 for activity attributes, 21 for artifacts, and 26 for artifact attributes. The first and third authors then created an abstraction hierarchy of identified activities and artifacts based on the guide to the software engineering body of knowledge [189]. For example, both the *planning* and the *estimating effort* activities are sub-types of the more abstract *managing* activity [189]. We decided to merge the activities *interpreting* and *understanding* with *comprehending* as none of the data sources sufficiently distinguished between them. Future studies differentiating them properly are necessary.

Once the hierarchy emerged, we associated each activity and artifact with the respective attributes that our data sources reported to characterize them. Whenever all activities or artifacts of a hierarchical group shared an attribute, we moved it to the higher-level activity or artifact for conciseness. Additionally, we made educated assumptions about the transferability of some attributes. For example, even though our data did not contain an instance of *duration* being evaluated on every activity, it is safe to assume that every activity can be characterized and evaluated by its duration. This step introduces slight subjectivity but improves the applicability of the model.

3.3 Data Availability

To achieve the goals of usability and extensibility of the resulting model, we disseminate it via GitHub.[6] The repository contains a reference to all considered data sources, guidelines and protocols for the data extraction, and a specification of the current model of requirements-affected activities and their attributes. More importantly, it contains guidelines on how to contribute new or revise existing activities and attributes. Using the version control system of GitHub[7] we will foster a collaborative evolution of the model.

4 Results

4.1 Requirements-affected Activities and their Attributes

Figure 2 visualizes the initial model of requirements-affected activities and their attributes. The model is structured like a UML class diagram and makes use of the inheritance relationship. An activity, represented as a UML class, that inherits from another activity also exhibits its attributes. For brevity, artifacts are excluded from the visualization. The replication package contains an extended model that includes the artifacts. The root of the inheritance tree is the abstract activity *processing*, which represents every executable activity. The model contains several activities that are commonly considered in research as requirements-affected activities, like modeling, prioritizing, implementing, and testing. Another prominent spot is taken by the merged activity *comprehending*, which dominates the distribution of activities among both experimental literature and interview statements. This correlates to the prominence of ambiguity among the attributes of requirements quality in empirical research [6] and is supported by the fact that this activity precedes every other activity [2]. The model, furthermore, contains several less commonly investigated activities. For example, Murakami et al. investigate the activity of *code review* in which subjects are provided with a requirements specification [190]. *Consolidating* larger

[6] Available at `https://github.com/JulianFrattini/gere-r3a`

[7] `https://docs.github.com/en/get-started/using-git/about-git`

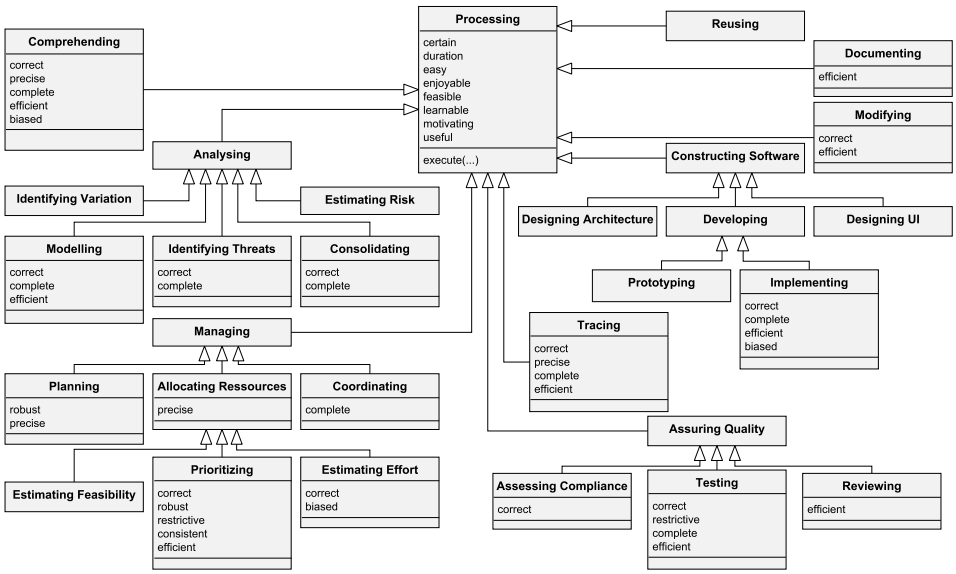


Figure 2: Model of requirements-affected activities and their attributes

sets of requirements to identify a semantically equivalent subset [191, 192] is another rare example. The model also contains activities that did not appear in experimental studies but were reported by interview participants or prescribed by software process literature. The activity of *prototyping* is such an example that was both mentioned during the interviews and as part of lifecycle models. Furthermore, the following activities were all named by interview participants but not considered in the experimentation literature: *coordinating* internal stakeholders based on a requirement, *reusing* artifacts like source code based on a new requirement, and *estimating feasibility* of a requirement. The attributes recorded in the model also show a varying distribution of prevalence. The most commonly encountered attributes of an activity are *duration*, *correctness*, and *completeness*. These represent both simple-to-measure and critical properties of most activities. Additionally, we observed several attributes related to the effect that the activity has on the executing agent, for example, how *certain* an agent feels when executing the activity, how *easy*, *enjoyable*, *motivating*, and *useful* they perceive it to be, and how *learnable* the activity was. Rarely mentioned attributes include how *robust* an activity is against errors and how *biased* an activity becomes given some controlled stimulation.

4.2 Implications

4.2.1 Implications for Research

The results contain multiple implications for requirements engineering and, specifically, requirements quality research. Firstly, the distribution of activities and at-

tributes among the three data sources hints at potential research gaps. For example, the above mentioned activities of prototyping, coordinating, reusing, and estimating have not appeared in the sample of primary studies. Secondly, the model provides guidance for comprehensive measurements of the software development life cycle with respect to the impact of requirements artifacts and processes. As determined by Femmer et al., only a holistic view of all requirements-affected activities will reliably determine the impact of any treatment in requirements artifacts or processes [2]. This affects all comparative studies in requirements engineering, i.e., all controlled and quasi-experiments aiming to evaluate the impact of a quality defect or the benefit of a new method. Only by measuring this impact on all requirements-affected activities in terms of their attributes and summarizing the total benefit or drawback, a holistic decision on the benefit or harm of any treatment can be made. While we certainly do not suggest that any comparative study from here on out must necessarily consider all 24 activities simultaneously, the model of requirements-affected activities provides at least a framework that allows integrating the results of multiple studies investigating the effect of the same treatment on different activities to one, overall conclusion.

4.2.2 Implications for Practice

The resulting initial model of requirements-affected activities and their attributes may serve practitioners as an overview of activities to measure when attempting to understand the fitness for purpose of their requirements. The model emphasizes the diversity of activities that may be affected by requirements but also the diversity of metrics by which they can be evaluated. While attributes like completeness, correctness, and duration are likely to be covered in key performance indicators of organizations, attributes like usefulness, ease of use, and learnability may often be neglected. Further practical use of the model for quantitative comparisons requires future work and will be discussed in Section 5.

4.3 Limitations

This study exhibits the following limitations. Firstly, the data extraction phase was only performed by one researcher. This introduces the possible risk that relevant information from the bodies of text was missing from the textual descriptions that were later coded. Secondly, the interview study was not performed with the research questions stated in Section 1 in mind. Instead, the main theme of the interview study was centered around the broader scope of requirements quality [170]. However, confirming previous studies that proposed that requirements quality inevitably depends on requirements-affected activities [1, 2, 40], the responses of interview participants naturally contained information that contributed to answering our research questions. Hence, we deem the interview data as an eligible data source for this study. Thirdly,

every step of the study where we depart from purely summarizing and reporting data and instead interpret it introduces researchers' bias. This is particularly evident in the conscious merging of the understanding, interpreting, and comprehending activity but also in the assumption about the transferability of several activities' attributes. This step was necessary to elevate the model beyond a systematic summary toward an evaluation framework as demanded in previous research roadmaps [1, 40]. We documented all interpretative steps and disclosed them in our replication package to allow other researchers to scrutinize these decisions. Finally, we address the threat to external validity. Full generalizability was out of the scope of the goals of this study, but we, nevertheless, briefly discuss all threats to external validity in order to justify the research plan as presented in Section 5. One threat to the generalizability stems from the sampling of the literature survey, which only considers a specific set of SE-relevant venues and categorically excludes workshops. Additionally, the literature review is limited to experiments and excludes other methods like case studies. Another threat stems from the sample of interview participants, which represent only one team of only one company.

5 Research Plan

5.1 Model Extension

The limitations mentioned in Section 4.3 necessitate the extension of the model to achieve goals 1 (applicability) and 2 (suitability) stated in Section 3. Both the applicability and the suitability are inhibited by the potential incompleteness of the model. Hence, we plan to repeat the early method presented in Section 3. Two immediately planned extensions are (1) repeating the systematic literature survey on workshop papers and (2) replicating the interview study in different companies and teams. Because of the extensive documentation of data collection methods for both empirical data (i.e., interview transcripts) and meta-research (i.e., primary studies), as well as the data analysis protocol, we anticipate that the model extension can be distributed well within our network of researchers interested in requirements-affected activities.

5.2 Model Maintenance

Goals 3 (extensibility) and 4 (usability) stated in Section 3 are fulfilled by the design of the chosen dissemination strategy. The authors of this study will maintain the GitHub repository containing the current content and structure of the model.

5.3 Model Validation

The most significant step of future work is to validate whether the model achieves the four goals stated in Section 3.

Validating applicability. To test whether the model can represent all requirements-affected activities and attributes in any given SE context, we plan to conduct multiple case studies in different company contexts. Once the model is deemed sufficiently extensive, we trace requirements artifacts in each case company to every instance of reuse. The process of tracing requirements artifacts to activities using these artifacts as input shall happen both directly, i.e., by interviewing involved stakeholders, but also indirectly, i.e., by observing which stakeholder accesses the artifact and then following up on the purpose. The latter accounts for requirements-affected activities that stakeholders are not actively aware of, i.e., in case they unconsciously retrieve information to execute an activity without considering that this makes the activity requirements-affected. We constitute that the model achieves goal 1 if we do not encounter any requirements-affected activity that has no semantic equivalent in the model.

Validating suitability. To test whether the model can be used to evaluate relevant activities by means of their attributes, we plan to conduct an empirical study involving all surveyed case companies. Given the already detected requirements-affected activities, we evaluate these via the attributes associated with the activities in our model to quantify their performance. We aim to produce two types of empirical investigations from this data. Firstly, we aim to survey the activities and generate an overview of attribute values for all affected activities. This overview provides an absolute comparison of the activities and answers questions like "Which activity phase takes the longest time" or "Which development activity is perceived as the least enjoyable?". Secondly, we aim to conduct quasi-experiments at the case companies investigating whether certain properties of requirements artifacts or properties have an impact. For example, the subject of the experiments could be the comparison between two types of template systems for requirements specification [193] or the avoidance of specific linguistic structures like passive voice [170]. The subject of the experiments will be aligned with current questions and endeavors of the case companies to optimize their requirements engineering artifacts or process in an evidence-based manner. The results of the quasi-experiments will be measured in terms of differences in attribute values of all affected activities. This overview will provide companies with a summary of the effect that the proposed change has on all affected activities. We constitute that the model achieves goal 2 if the results generated by the surveys and quasi-experiments are accepted by the respective case companies.

Validating extensibility. To test whether the model can extended with new activities or attributes, aim to involve additional researchers in the model extension presented in Section 5.1. By distributing the task beyond the authors of this study, we determine how easily other researchers can extend the model. We constitute that

the model achieves goal 3 if external researchers extend the model successfully.

Validating usability. To test whether the model can be accessed and comprehended by software engineers, we plan to facilitate external replications of the validation of goals 1 and 2. This not only validates whether the model achieves goal 4 but also extends the empirical evidence about the impact of requirements on affected activities in different company contexts. We constitute that the model achieves goal 4 if external researchers successfully replicate the empirical studies.

6 Conclusion

Requirements artifacts and processes fulfill a specific purpose in the software development lifecycle, that is, to inform subsequent activities about the needs and constraints imposed by stakeholders on the system under development [2]. How fit requirements artifacts and processes are to fulfill their purpose, i.e., how well they benefit these requirements-affected activities, can be effectively determined when (1) all affected activities are known and (2) the performance of these activities can be evaluated. The need for a systematic overview of (1) requirements-affected activities as well as (2) the attributes which quantify their performance has been well recognized in requirements quality literature [2, 170] and evoked the call for a comprehensive model [1, 40].

We answer this call by proposing an initial model of requirements-affected activities and their attributes systematically derived from three distinct data sources. The model aims to support both researchers by guiding empirical studies concerning the impact of requirements artifacts and processes but also practitioners by offering an overview of attributes that may serve as key performance indicators of their requirements-affected activities. We envision that this model will be extended and evolved by the requirements engineering community to provide an applicable and suitable model for the task. We will actively maintain the presented resources to enable and foster this community endeavor.

Paper IV

Requirements Quality Research Artifacts: Recovery, Analysis, and Management Guideline

Abstract

Requirements quality research, which is dedicated to assessing and improving the quality of requirements specifications, is dependent on research artifacts like data sets (containing information about quality defects) and implementations (automatically detecting and removing these defects). However, recent research exposed that the majority of these research artifacts have become unavailable or have never been disclosed, which inhibits progress in the research domain. In this work, we aim to improve the availability of research artifacts in requirements quality research. To this end, we (1) extend an artifact recovery initiative, (2) empirically evaluate the reasons for artifact unavailability using Bayesian data analysis, and (3) compile a concise guideline for open science artifact disclosure. Our results include 10 recovered data sets and 7 recovered implementations, empirical support for artifact availability improving over time and the positive effect of public hosting services, and a pragmatic artifact management guideline open for community comments. With this work, we hope to encourage and support adherence to open science principles and improve the availability of research artifacts for the requirements research quality community.

Keywords: Requirements Engineering, Artifact, Availability, Bayesian Data Analysis, Guideline

1 Introduction

Requirements quality research is a sub-domain of requirements engineering research specifically focused on assessing and improving the quality of requirements specifications [6]. Requirements quality research depends on research artifacts: annotated

data sets are used as the ground truth about quality violations, and *implementations* are deliverable artifacts (e.g., tools) that are often intended to be transferred into industry in order to support the management of requirements quality.

However, recent research exposed that the majority of research artifacts are not available anymore or have never been [6, 41, 51]. This inhibits the progress of the requirements quality research domain, as new contributions cannot reuse existing data sets for benchmarking their approach or evolve existing implementations. Instead, they have to resort to recreating already proposed solutions.

In this work, we aim to address the problem both retrospectively (i.e., recovering unavailable artifacts of prior publications) and prospectively (i.e., offering guidance for future publications). To this end, we make the following contributions:

1. Artifact recovery (C1): a two-phase recovery initiative of previously unavailable research artifacts, resulting in 10 recovered data sets and 7 recovered implementations. This improves the availability of research artifacts in the requirements quality research domain, recovering lost opportunities for the reproduction of empirical results or reuse of developed tools.

2. Evaluation (C2): an empirical evaluation of the reasons for artifact unavailability using Bayesian data analysis. The results help to better understand the potential barriers in disclosing artifacts and steer future open science initiatives within the requirements quality research community.

3. Open Science Artifact Management Guideline (C3): a concise and pragmatic guideline summarizing recommendations on how to collect, document, license, archive, and share research artifacts. The guideline supports authors of scientific work in disclosing research artifacts to maintain their availability and increase their impact.

An initial version of the first contribution—the first of the two phases of the artifact recovery initiative—has been published in the *Natural Language Processing for Requirements Engineering* (NLP4RE) workshop[1] [45]. In this paper, we extend the artifact recovery initiative by a second phase (C1) and add two more contributions (C2 and C3) to deepen the insights from the gathered data and provide actionable guidance for future publications. We reused minor parts of the original manuscript in a verbatim manner, such as the presentation of related work or terminological definitions.

The rest of this manuscript is structured as follows: Section 2 introduces the background on both the requirements quality research domain and open science principles. Section 3 describes the two-phase initiative to recover unavailable research artifacts, and Section 4 empirically evaluates the gathered data to infer reasons for

[1] `https://nlp4re.github.io/2023/`

the unavailability of these artifacts. Finally, Section 5 presents a concise guideline on artifact management, offering support in preserving artifact availability, before concluding the paper in Section 6.

Data Availability Statement

All study material, including our data, code, and evaluation reports, are accessible in our replication package, archived at `https://doi.org/10.5281/zenodo.7708570`. The guideline presented in Section 5 is archived at `https://doi.org/10.5281/zenodo.8134402` and a collaborative version is accessible at `https://bit.ly/OSAMG`.

2 Background

2.1 Open Science in Software Engineering

A vital property of scientific work is reproducibility [44, 194], i.e., the ability to "duplicate the results of a prior study using the same material as were used by the original investigator" [195], as it strengthens the robustness of scientific findings contributed to a field of research [195–197]. One necessary precondition of reproducibility is the availability of study materials, including study protocols to reproduce an investigation, data to re-evaluate statistical claims, or source code to recreate tools. To emphasize this dependency, Minocher et al. introduced a four-stage model of reproducibility [46] consisting of *data recoveribility*, *data usability*, *analytical clarity*, and *agreement of results*. The model is sequential, i.e., *analytical clarity* of data is meaningless if the data is not *recoverable*. Hence, all reproducibility hinges on the availability or recoverability of research artifacts. Furthermore, the model applies to all types of research. While the *agreement of results* is more straightforward to obtain for studies involving quantitative data, studies dealing with qualitative data should at least allow an external reviewer to review the analysis process and understand how the authors arrived at their conclusions.

Open science is an initiative dedicated to ensuring public availability of research artifacts [44]. Within open science, the facets of *open access* for publications, *open data* for data sets, and *open source* for source code are most relevant to software engineering [198], where each facet of open science entails different techniques and best practices to disclose its respective type of research artifacts. Several governmental research funding agencies, including ones of the European Union, made open access to scientific results (including data, tools, etc.) mandatory.[2]

Open science entails its own set of challenges. Most notably, adherence to open science principles requires additional effort in light of documenting and disseminat-

[2] `https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science/open-access_en`

ing research artifacts [199] and is sometimes even impossible, as sensitive data may not be shareable [200]. However, because of the importance of replication and reproduction for a scientific field [201] and the contribution of open science toward these properties [44], the software engineering research community has come to the agreement that open science is not only worth but rather imperative to pursue [202].

Recent endeavors to incentivize scholars to follow open science principles include open science badges [202–204] and registered reports [205]. The introduction of artifact evaluation tracks at premiere SE conferences like ESEC/FSE[3] greatly benefited the availability of research artifacts in SE research [202]. However, several sub-domains of the SE research community are still in the process of adapting and implementing open science principles [44], and the unavailability of artifacts remains common in areas like requirements quality research [6, 51]. Prominent reasons for the unavailability of artifacts include the sensitivity of data [202], corresponding authors changing their affiliation and consequently losing access to their artifacts [199, 206], or authors not seeing any benefit in sharing their artifacts [207]. While some reasons for the unavailability of artifacts (e.g., the sensitivity of company-owned data) may well require significant effort to cope with or are unavoidable, other reasons (e.g., loss of artifact, lack of diligence) can be circumvented easily by following guidelines [44] and making use of modern tools, e.g., Zenodo, for artifact sharing [208].

2.2 Requirements Quality Literature

Artifacts produced in requirements engineering (RE)—e.g., systematic requirements specifications, use cases, or user stories—have a significant impact on downstream software development activities [21], potentially causing project delay or even failure [209]. Consequently, requirements artifacts merit quality assurance [6]. The requirements quality literature is dedicated to providing an understanding as well as the support for measuring and improving the quality of requirements [6]. One popular approach to this is the proposal of *quality factors* [41]. Requirements quality publications often propose one or more quality factors—e.g., the use of *coordination ambiguity* leading to divergent interpretations [91] or the use of *passive voice* causing the omission of information [34]—, provide a *data set* where instances of violations against that quality factor are annotated, and finally present an *implementation* (i.e., an algorithm or full-fledged tool) to detect or remove these instances automatically.

These artifacts—both data sets and implementations—represent essential contributions facilitating empirical research and technology transfer. While the (annotated) data sets are the main drivers for developing new and improving existing implementations for quality factor detection, implementations are the research deliverables to be deployed in industry for actual integration and improvement of the software engi-

[3] `https://conf.researchr.org/series/fse`

neering process [202].

However, the degree to which artifacts are disclosed in the requirements quality literature varies [6]. The majority of research artifacts are simply *unavailable*, i.e., authors present them in a publication but provide no access to them [51]. *Proprietary* artifacts, i.e., those turned into a commercial product, or *private* artifacts, i.e., those not disclosed due to sensitivity, constitute an excused exception to this group. Among those artifacts that are actually disclosed, i.e., hosted in an accessible way and referred to from the manuscript, several are *broken* since the URL pointing toward the artifact does not resolve anymore. Other artifacts are only available *upon request*, and their access depends on the author's correspondence and upholding the promise made in the article. Among those artifacts that are actually *reachable* (i.e., referred to with a non-broken source link), implementations where the source code is hosted openly and contains an open source license (i.e., *open source*) are most common. For smaller data sets, it is also common that they are *available in the paper* themselves, e.g., in the form of a table in the manuscript. In the best case, authors have properly *archived* their artifact, which implies (1) an immutable URL, (2) permanent hosting, and (3) unrestricted accessibility. Only very few services (e.g., Zenodo [208]) offer this level of hosting.

Recent systematic studies revealed that a significant amount of the artifacts presented in the requirements quality literature are not available[4] anymore or have never been [6, 41, 51]. Table 1 reports the availability status of 57 data sets (D) and 36 implementations (I) extracted from the 57 primary studies of a previously-published literature review on requirements quality factors [41]. Most notably, the table visualizes that a large portion of data sets (87.8%) and implementations (80.6%) are not available anymore or never were (i.e., status *broken* or worse). A significant portion of data sets (42.1%) are excused due to containing private or sensitive data, but for the majority of implementations (77.8%), no valid excuse for their unavailability is provided.

The visualization of the poor artifact availability [6, 41] shown in Table 1 revealed that the requirements quality research domain suffers from a similar reproducibility crisis as other fields of research had [202]. While SE research, in general, has advanced over the last 10 years [199, 204], the requirements quality research domain suffers from a limited adoption of open science principles [44]. The consequently limited reproducibility of research in the field [46] undermines the robustness of scientific findings [195–197] and the technology transfer of scientific results into practice [210]. There is a clear gap regarding the state of open science in the requirements quality research domain.

[4]Where available means a status of *Upon request* (see Table 1) or better.

Table 1: Availability status of requirements quality artifacts [41]

Status	Explanation	Datasets		Implementations	
Archived	The artifact is hosted in a service that satisfies the following criteria: (1) immutable URL (cannot be altered by the author or someone else), (2) permanent (the hosting organization has a mission to maintain artifacts for the foreseeable future), and (3) accessible (there is a DOI pointing to the real data source URL)	1	1.7%	0	0%
Open Source	[only for implementations] The source code is disclosed and contains an open-source license (the artifact has a license which grants access and reuse)	-		5	13.9%
Available in paper	[only for data sets] The data set is contained in the manuscript itself	5	8.8%	-	
Reachable	The artifact is reachable now but missing some of the *archived* or *open source* aspects	1	1.7%	1	2.9%
Upon request	Authors claim the artifact is available upon request	0	0%	1	2.9%
Broken	A link to the artifact is contained in the paper, but it does not resolve	10	17.5%	1	2.9%
Unavailable	An artifact is presented, but no indication on how to access it is provided	15	26.3%	27	77.8%
Private	The authors state that an artifact exists but is private for some reasons (such as industry collaboration with private data, etc.)	24	42.1%	0	0%
Proprietary	The artifact is proprietary, and access is granted upon payment	1	1.7%	1	2.9%
Total		57		36	

3 Artifact Recovery

We aim to improve the *availability* of research artifacts by (a) requesting authors of publications containing unavailable artifacts to recover their artifacts and (b) requesting authors of publications containing available but not archived artifacts to improve their artifacts' availability. To this end, we ask the following research question: to what degree can research artifacts from the requirements quality research domain be recovered by approaching their owners (RQ1)? In Section 3.1, we document the design of the artifact recovery initiative. In Section 3.2, we present the results of this initiative. In Section 3.3, we discuss the threats to validity of these results.

3.1 Recovery Process

Section 3.1.1 describes the selection of primary studies from which we recover the artifacts. We detail our approach of contacting corresponding authors in Section 3.1.2 and maintaining contact in Section 3.1.3. Section 3.1.4 explains the evaluation of the collected data and Section 3.1.5 documents the involvement of the NLP4RE community for the extension of the study. Figure 1 visualizes the two-phase process.

3.1.1 Study sample selection and preparation

Prior studies [6, 41] provided an existing selection of primary studies relevant to the requirements quality literature. In particular, the subject of our recovery request is the artifacts from a set of primary studies that we used to construct an ontology of

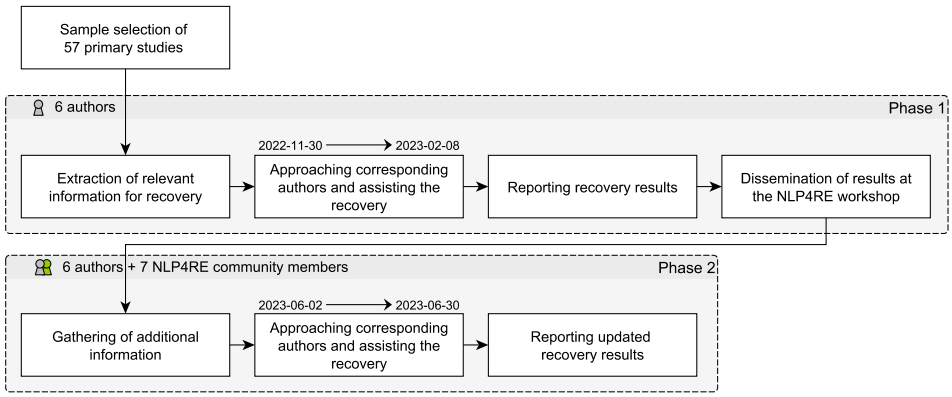


Figure 1: Overview of the two-phase recovery process

requirements quality factors [6]. To develop this ontology, we collected manuscripts reporting quality factors from an original set of publications reported in another secondary study [6]. Extracting data sets and implementations from such publications revealed the unfortunate state of artifact availability in the first place. Reusing this set of artifacts qualifies our sampling strategy as *convenience sampling* [211].

In the initial ontology-creation study [41], we extracted only the name of the artifact, its availability, and—in case it was accessible—its source link. To enable recovery requests for each unavailable resource, we additionally extracted the following information for each artifact:

- **Corresponding author**: each artifact was associated with a corresponding author responsible for its availability.
- **Mention**: each artifact was associated with its verbatim mention in the manuscript.

Additionally, we corrected information about one data set and three implementations that persisted in the previous study.[5] In three spreadsheets, we collected data about

- (1) **authors** (n=35), specified by their name and email address,
- (2) **data sets** (n=57), and (3) **implementations** (n=36), specified by the publication which contains it, its verbatim mention, the corresponding author, and its current availability.

3.1.2 Approaching the authors

We created a Python script that automatically assembles one email to each corresponding author. This email contained the following elements:

[5] All corrections are documented in the replication package.

1. Header: an explanation of our endeavor and a request to contribute to open science (or alternatively explain why recovery is impossible).

2. Artifact list: a list of artifacts for which the corresponding author was responsible

3. Instructions: brief instructions on how to properly disclose artifacts according to open science principles as well as the offer to assist them in the process

4. Contact: a way to reach out to us

We developed the instructions based on our collective knowledge regarding open science, relevant literature [212], and artifact evaluation guidelines of SE conferences. These instructions are a condensed version of the management guideline presented later (see Section 5) and are contained in our replication package. The development of the instructions is described in more detail in Section 5.1.

In the initial phase of the recovery attempt, we approached the authors in a first mail on the 30th of November 2022, followed by a reminder on the 13th of December, and a final reminder on the 11th of January 2023. For authors who did not respond to our request until the final reminder, we additionally contacted their co-authors to increase the likelihood of response. We concluded the recovery process on the 8th of February 2023, yielding a time frame of 70 days.

3.1.3 Correspondence

We kept close contact with the authors we approached by responding in a window of 24 hours within workdays. During this process, we clarified concerns and offered our help. We processed and recorded the information contained in the authors' answers in a spreadsheet file. We tracked the response status to our request in an additional column, denoting the request as either *undeliverable*, *unanswered*, *answered*, or *completed*. We labeled a recovery request as *completed* once the corresponding author, for all their artifacts, either improved their availability or explained the inability to recover or disclose them.

Furthermore, we documented the dates of the first email sent, the first response received, and the completion of the request alongside the number of emails sent by the author in addition to the updated availability status of the artifacts or, eventually, the author's explanation for not taking the recommended actions. Two authors coded these explanations independently and came to an absolute agreement on the types of reasons for non-recovery. When the corresponding author's email address was no longer used, we reached out via personal contacts or social networks like Twitter and LinkedIn.

3.1.4 Evaluation

To evaluate the artifact recovery process, we generated statistics of the following data from the documentation in our tables.

1. Correspondence (i.e., author response time and frequency) to evaluate the effort of the recovery process.
2. Recovery request success (i.e., change in artifact availability) to evaluate the success of the recovery process.
3. Reason for non-recovery (i.e., authors' responses excusing the recovery) to evaluate the reasons inhibiting adherence to open science principles.

We evaluated the data by generating descriptive statistics from our documentation.

3.1.5 Dissemination and Crowd-Sourcing

The results of the first phase of the recovery request showed initial success but also room for improvement [45] (more details in Section 3.2). The first author of this article presented these results at the 6th Workshop on Natural Language Processing for Requirements Engineering[6] (NLP4RE) co-located with the 29th International Working Conference on Requirement Engineering: Foundation for Software Quality[7] (REFSQ). The visualization of the dire previous state of open science in the research field, but also the initial success of the artifact recovery, inspired the workshop attendees to contribute to the recovery initiative. Three hypotheses about the remaining room for improvement emerged:

1. Corresponding authors might not have responded to the recovery request because its sender was unknown to them, and they were unable to verify the trustworthiness of the request.
2. Corresponding authors might be more likely to react to an alternative email address than the corresponding email address printed in a published article. For example, some workshop attendees were aware of email addresses of unresponsive authors through which they could personally reach out.
3. Artifacts lost to corresponding authors might have been acquired by members of the community at the time they were available.

These three hypotheses invited crowd-sourcing the recovery initiative, as the more established members of the research community are likely to have a better connection to the corresponding authors of unavailable artifacts (addressing hypotheses 1

[6] https://nlp4re.github.io/2023/
[7] https://2023.refsq.org/

and 2), and they might have acquired resources and could still recover them when corresponding authors have already confirmed their status of unavailability (addressing hypothesis 3). To this end, we compiled two lists: one containing unresponsive corresponding authors and one containing artifacts that corresponding authors claimed to be lost. The lists were distributed to seven attendees of the NLP4RE workshop who expressed interest in contributing to the recovery initiative. All of those contributors are experienced and established scholars in the research domain.

The contributors partook in the artifact recovery initiative by providing the authors of this paper with additional information, establishing contacts, and recovering artifacts. Once a new contact was established, the artifact recovery request was handled as in the first phase of the initiative. Once a contributor retrieved an artifact, the owner of the artifact was determined and approached to request permission for archiving it via Zenodo.[8]

We initiated the second, crowd-sourced phase of the artifact recovery initiative on the 2nd of June 2023, sent a reminder on the 19th of June 2023, and concluded all requests on the 30th of June 2023. We evaluated the recorded data similar to the first phase described in Section 3.1.4. Note that we merely used the above-mentioned three hypotheses to design the second phase of the artifact recovery initiative and did not empirically evaluate them since we prioritized the recovery of research artifacts over investigating research community dynamics.

3.2 Results

3.2.1 Correspondence

Phase 1 Out of the 35 approached corresponding authors, 19 (54.3%) answered the recovery request, and 13 (37.1%) completed it. We could not reach three (8.6%) authors despite searching for a valid contact. The distribution of correspondence status is visualized as the blue bars in Figure 2.

Phase 2 During the second phase, 5 additional corresponding authors were identified, as it became clear that 5 artifacts were actually owned by other, previously not considered authors. Out of these 40 approached corresponding authors, 30 (75.0%) answered the recovery request, and 21 (52.5%) completed it. 2 (5.0%) authors remained unreachable and 8 (20.0%) requests remained unanswered. The distribution of correspondence status is visualized as the orange bars in Figure 2.

It took, on average, 23.8 days for a corresponding author to reply to our request and 29.25 additional days to complete the request. On average, a request was resolved in an exchange of 3 emails with the corresponding author. The distributions of these statistics are visualized in Figure 3 and Figure 4, respectively.

[8] `https://zenodo.org/`

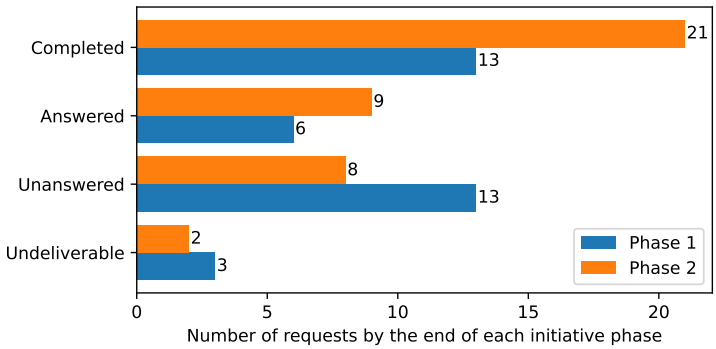


Figure 2: Status of correspondence

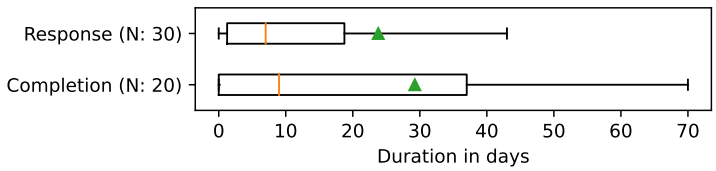


Figure 3: Distribution of time for correspondence in days (excluding outliers)

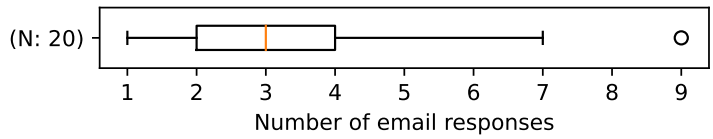


Figure 4: Distribution of frequency of correspondence in the number of emails

3.2.2 *Artifact Recovery Success*

Table 2 summarizes the total number of artifacts that were either recovered or where their unavailability was confirmed by the corresponding authors of primary studies over the two phases. An artifact counted as *availability improved* if its availability at the end of the respective phase was higher than at the beginning of the study according to Table 1. The availability of 10 data sets was improved through the two recovery phases. Two of these 10 data sets were already available before but on a lower level of Table 1, and eight data sets were newly recovered. This increases the availability of data sets from 12.3% (7/57, 1 archived) to 26.3% ((7+8)/57=15/57, 8 archived). Similarly, the availability of 8 implementations was improved. Five of these were not available before. Additionally, one previously available implementation became unavailable during the study (see the explanation for this below), such that the availability of implementations improved from 19.4% (7/36, 0 archived) to 30.5% ((7+5-1)/36=11/36, 6 archived). Authors further confirmed the unavailability of 30 (52.6%) data sets and 10 (27.7%) implementations and provided reasons for

the inability to recover or disclose them.

Table 2: Total number of artifacts clarified after each of the two phases

Availability	Phase 1 (D)	Phase 1 (I)	Phase 2 (D)	Phase 2 (I)
Availability improved	7 (12.3%)	7 (19.4%)	10 (17.5%)	7 (19.4%)
Unavailability confirmed	21 (36.8%)	6 (16.7%)	30 (52.6%)	10 (27.7%)
Total	28 (49.1%)	13 (36.1%)	40 (70.1%)	17 (47.2%)

Figures 5 and 6 visualize the total success of the two-phase recovery initiative. The heatmap considers all artifacts (data sets in Figure 5 and implementations in Figure 6) where the corresponding author completed the recovery request. The number in a cell represents the number of artifacts for which the original availability (on the y-axis) has been updated to the new availability (on the x-axis). The count of artifacts whose availability remained the same (e.g., because an author confirmed that the artifact could not be made more available) is reported on the diagonal (shaded gray). An improvement in the availability of an artifact contributes to cells to the right of the diagonal, a deterioration of the availability to the left. Consequently, the "Availability improved" count in Table 2 is the sum of all cell values to the right of the diagonal, and the "Availability confirmed" count is the sum of all cell values on the diagonal and to the left of the diagonal.

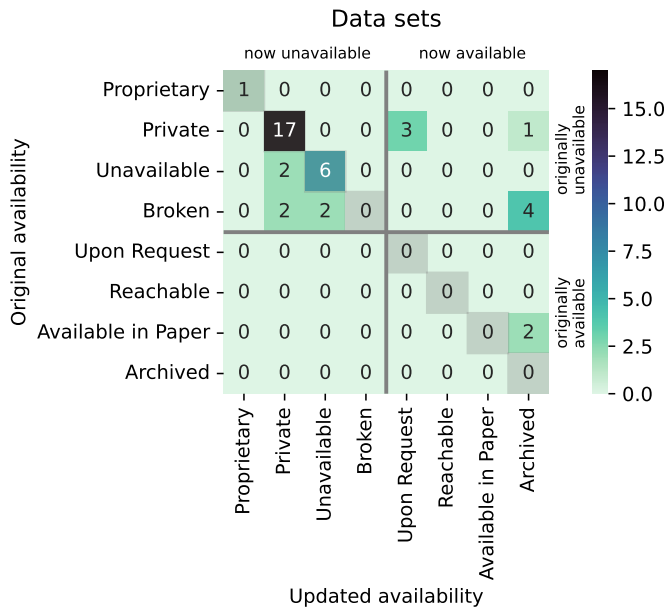


Figure 5: Change of availability in data sets

For example, one implementation was previously available upon request [213]. Now that the authors archived the implementation following open science princi-

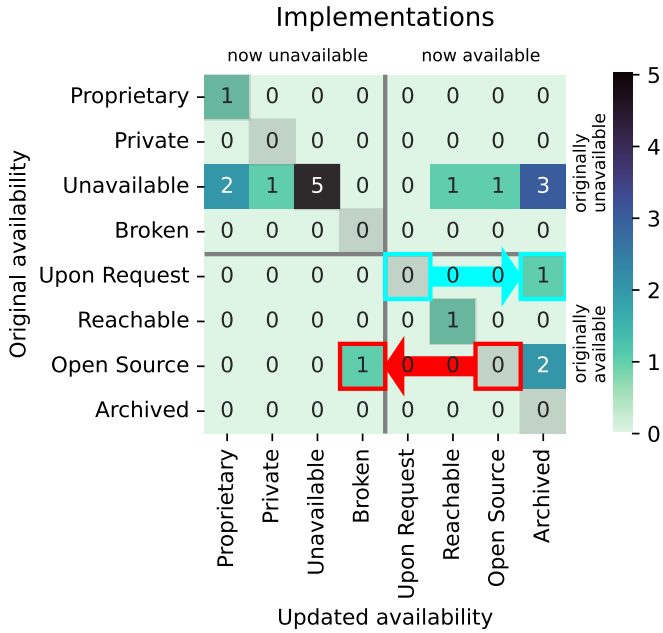


Figure 6: Change of availability in implementations

ples,[9] the entry moved three cells to the right (see the cyan arrow in Figure 6). On the other hand, another implementation called RETA [214] was available at `http://sites.google.com/site/retanlp/` during the first phase of the study [45]. While checking during the second phase, the URL did no longer resolve and the link became *broken*. The entry, therefore, moved three cells to the left (see the red arrow in Figure 6). The authors explained the loss of the artifact with their change of affiliation, which caused the website not to be maintained anymore. The implementation could sadly not be recovered at the time.

The inability to recover or disclose artifacts was reported as follows: among 30 unrecoverable data sets, 19 were lost (i.e., the author could not find them anymore or the contact of whom the author assumed had the data was unreachable), and 11 could not be disclosed due to sensitive contents. Among the 10 unrecoverable implementations, 4 became proprietary, and 6 were lost.

The recovered artifacts (i.e., all artifacts that counted towards "Availability improved" in Table 2), their new location, and the original publication presenting them are listed in Table 3 (data sets) and Table 4 (implementations).

[9] Now publicly available at `https://doi.org/10.5281/zenodo.7484023`

Table 3: List of and reference to recovered data sets

Artifact	Link	Ref
UIC EIRENE SRS	https://doi.org/10.5281/zenodo.1414117	[215]
HWS Documentation	https://doi.org/10.5281/zenodo.80815230	[216]
3 user story SRS	(upon request)	[132]
Real estate SRS	https://doi.org/10.5281/zenodo.7499290	[34]
IMAGS II SRS	https://doi.org/10.5281/zenodo.7619051	[145]
11 SRS	https://doi.org/10.5281/zenodo.80143477	[217]
MadeByGraph SRS	https://doi.org/10.5281/zenodo.7602827	[218]
Helpdesk Support	https://doi.org/10.5281/zenodo.8183338	[219]

Table 4: List of and reference to recovered implementations

Artifact	Link	Ref
S-HTC	https://doi.org/10.5281/zenodo.7584181	[215]
CAR	https://doi.org/10.5281/zenodo.7584193	[220]
AQUSA	https://doi.org/10.5281/zenodo.7573781	[132]
Bidirectional Chatbot	https://git.uni-paderborn.de/jkers/sfb-b1-cordula-bidirectional	[221]
Cordula Desiree	https://doi.org/10.5281/zenodo.7484023	[213]
ARBIUM	https://doi.org/10.5281/zenodo.7528522	[222]
Ambiguity detector	https://doi.org/10.5281/zenodo.1476902	[223]
Near-synonymy detector	https://github.com/RELabUU/revv	[224]

Answer to RQ1: Success of the Recovery Request

Approaching owners of unavailable artifacts with a request to recover them showed significant success in the requirements quality research community. Crowd-sourcing the request in the research community further benefited the endeavor. The status of several unavailable research artifacts could be clarified this way, either by explaining their unavailability with valid reasons or by recovering the artifacts.

3.3 Threats to Validity

The answer to RQ1, as stated in Section 3.2, is subject to the following threats to validity, grouped via the categories introduced by Wohlin et al. [47]. The main type of threat affecting the validity of our claim is *external validity*. The generalizability of the claim is inhibited by the sample of primary studies involved in the recovery attempt. The sample originates from previous secondary studies [6, 41]. The representativeness of our sample is inherently limited by their cutoff date, the 27th of March 2020. However, the original secondary study [6] uses a rigorous sampling strategy, which supports the reliability of the claim at least for the respective time

frame.

Additionally, threats to *internal validity* may affect the conclusions of the study. Confounding factors could have affected the result. For example, we are unable to explain the different reasons that caused the non-responsiveness of some authors. Possible factors include negligence, distrust, or oversight. Given the setup of the study, we could not control these factors.

4 Evaluation of Reasons for Artifact Unavailability

To infer deeper insight into the factors influencing artifact (un-)availability in our sample, we analyzed the data from the two-phase recovery initiative. For our sample of research artifacts from the requirements quality research domain, we ask the following research questions:

- RQ2: Which factors influence the availability of research artifacts?
- RQ3: Which factors influence the success of the recovery initiative?

Section 4.1 states our hypotheses and summarizes the available variables, and Section 4.2 documents the analysis procedure. Section 4.3 contains the results and Section 4.4 interprets them, before Section 4.5 discusses threats to validity of this analysis.

4.1 Hypotheses and Variables

The collected data allows us to infer insights about four aspects of artifact availability. Within our sample of observed artifacts, we investigate the factors that influence

1. the **original availability** of artifacts (*orig*), i.e., the inclination of authors to disclose artifacts upon publication of their article (regardless of the longevity of this artifact),
2. the **persistence** of disclosed artifacts (*per*), i.e., the longevity of an artifact,
3. the **recoverability** of an unavailable artifact (*recov*), i.e., the ability to make it available again, and
4. the **updated availability** of all artifacts (*avail*), i.e., the accessibility of artifacts after the recovery initiative.

The original availability of artifacts is determined by whether a primary study contains a link to the disclosed artifact. In this case, it does not matter whether that link still resolves, as we assume that it at least resolved at the time of publication.

The persistence is measured by whether a link to an artifact still resolves. The recoverability is measured by whether the recovery was successful or had to be excused by the corresponding author. The updated availability is measured via the availability status of the sampled artifacts after the recovery initiative.

The original availability and persistence of research artifacts represent the main variables of interest that motivated the recovery initiative in the first place. Investigating hypotheses involving these two variables contributes insights into the adherence to open science principles and the artifact availability in requirements quality research. The recoverability and updated availability of research artifacts represent variables describing the outcome of the recovery initiative itself. Investigating hypotheses involving these two variables contributes insights about executing artifact recovery initiatives and helps decide whether the effort to implement such an initiative is worthwhile.

Given the collected data, the impact of the following independent variables on the presented dependent variables can be investigated:

1. **Recency** (rec): Relative age of the publication presenting the artifact
2. **Type** ($type$): Whether the artifact is a data set or implementation
3. **Hosting** ($host$): Whether an artifact was hosted using a public (e.g., Zenodo, Github, Sourceforge) or private (e.g., institutional or personal websites) service

The category *public* of the independent variable **hosting** could further differentiate *archival*, i.e., whether the public host is committed to a long-term retention policy, similar to what Winter et al. [199] investigated. Due to the lack of data points, we were unable to consider this differentiation.

Further independent variables could causally impact the dependent variables, like the artifact policy of venues at the time of publication or a corresponding author's previous knowledge of open science principles and practices. We did not collect additional data beyond the documentation of the artifact recovery initiative and confined our inference to the available variables. This limits our causal inference—which we further discuss in the threats to validity in Section 4.5—but still allows limited reasoning within an explicitly delineated space of variables.

The variables and their data types are summarized in Table 5. For the three of the total four dependent variables *original availability* ($orig$), *recoverability* ($recov$), and *updated availability* ($avail$), we investigate the impact of the two independent variables *recency* (rec) and *type* ($type$). Consequently, we formulate the following six hypotheses: "The {*recency* of a publication, *type* of an artifact} has no effect on the {*original availability*, *recoverability*, *updated availability*} of an artifact" ($h^{ind \in \{rec,type\}}_{dep \in \{orig,recov,avail\}}$). For the fourth dependent variable *persistence* (per), we investigate the impact of the independent variables *recency* (rec) and *hosting* ($host$) in the scope of the following hypothesis: "The {*recency* of a publication, *hosting* of an

artifact} has no effect on the *persistence* of the artifact" ($h_{per}^{ind \in \{rec,host\}}$). This results in eight hypotheses to be tested in this evaluation. We index hypotheses based on the combination of independent and dependent variables under investigation, e.g., h_{orig}^{rec}: "The *recency* of a publication has no effect on the *original availability* of an artifact." For conciseness, we keep the rest of the hypotheses in their modular format. They are spelled out in the respective analysis files contained in our replication package.

Table 5: Independent (ind) and dependent (dep) variables for empirical analysis

Variable	Name	Type	Description	Range
Recency	`rec`	ind	Relative age of an artifact	[0, 1]
Type	`type`	ind	Type of the artifact	{dataset, implementation}
Hosting	`host`	ind	Type of artifact hosting	{private, public}
Original Availability	`orig`	dep	The availability of an artifact before the recovery initiative	{available, unavailable}
Persistence	`per`	dep	Whether a once disclosed artifact was available before the recovery initiative	{persistent, non-persistent}
Recoverability	`recov`	dep	Whether the availability of an artifact could be improved	{recoverable, unrecoverable}
Updated Availability	`avail`	dep	The availability of an artifact after the recovery initiative	{available, unavailable}

4.2 Bayesian Data Analysis

We conducted a Bayesian data analysis (BDA) according to Pearl's framework for causal inference [225]. Furia et al. have popularized the use of BDA in software engineering for causal inference since it outperforms common frequentist approaches in terms of reliability and level of detail of the inference. Most importantly, BDA models unknown parameters as probability distributions instead of fixed values, allowing for more sophisticated insights than point-wise comparisons of frequentist statistical methods [48]. For brevity, we only report the most important elements of the analysis in this manuscript. We refer the reader interested in a gentler introduction to the topic to appropriate literature on frameworks for causal inference [54, 225], textbooks on BDA [62], exemplary applications of BDA in SE research [48, 226], and our replication package mentioned in the introduction.

We implemented the BDA according to an established Bayesian workflow in three major steps [54]: modeling, identification, and estimation. In the first *modeling* step, we formalized our causal assumptions as a directed, acyclic graph (DAG) as visualized in Figure 7. In the DAG, nodes represent the variables of interest, and directed edges represent assumed causal relationships between these nodes. Based on the DAG, variables to include and exclude can be determined via selection criteria [62] in the *identification* phase. Because our DAG does not involve any variables

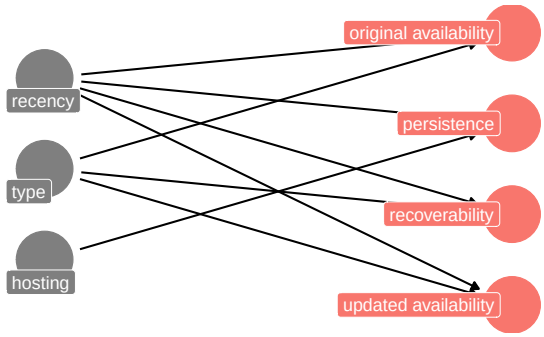


Figure 7: Directed, acyclic graph of causal assumptions between independent (grey) and dependent (red) variables

that influence both the independent and the dependent variables, all variables are eligible to be included in the next phase.

In the final *estimation* phase, we trained Bayesian models following the Bayesian workflow by Gelman et al. [227] and using the R library `brms` [228]. Each dependent variable was modeled in relationship to its independent variables, which allows to quantify the impact of each independent variable on the dependent variable after training.

We selected an appropriate likelihood distribution for each dependent variable according to the maximum entropy criterion [229]. Since all dependent variables are categorical with two categories, a single-trial Binomial or Bernoulli distribution is appropriate [62]. We opt for the latter due to simplicity. Next, we selected uninformative priors, i.e., parameter distributions that represent prior beliefs but are unspecific enough for the model to update these beliefs according to the data [227]. We confirmed the appropriateness of these priors through graphical prior predictive checks [230].

We then trained all models on the collected data and assessed the appropriateness of the trained models both through graphical posterior predictive checks [227] and assessment of the chain mixing property $\hat{R} < 1.01$ [231]. Then, we evaluated our models by plotting the conditional effects of the independent variables. The conditional effects represent the impact of the independent variables on a dependent variable as well as their interaction as learned by the model via Bayesian inference. It, therefore, quantifies the strength and direction of the relationship between independent and dependent variables.

Since these conditional effects only visualize the uncertainty of a single variable, we additionally evaluate the model by sampling from the full posterior distribution of the learned model parameters [227]. For each dependent variable, we compare the samples from the posterior when fixing each independent variable at the extreme values of its spectrum (0 and 1 in the case of *recency*, and both categorical values in the case of *type* and *hosting*). By drawing 60.000 random samples from the posterior

distribution once for each of the two extreme values, the model allows to quantify the impact of each independent variable in terms of percentages.

For more in-depth explanations of Bayesian data analysis, we refer the interested reader to established literature [48, 62, 225, 227] and our replication package containing detailed documentation of the analysis.

4.3 Results

The following sections contain the evaluation of the hypotheses (h_{orig} in Section 4.3.1, h_{per} in Section 4.3.2, h_{recov} in Section 4.3.3, and h_{avail} in Section 4.3.4). For brevity, we removed figures not contributing significantly to the results. All visualizations and evaluations are, however, available in our replication package.

4.3.1 Factors influencing original Availability

Figure 8 visualizes the distribution of the original availability of data sets and implementations over the time span of the articles. The value $recency = 0$ represents the oldest data set (1998, a requirements specification presented by Romano et al. [232]) and implementation (1997, the ARM tool by Wilson et al. [134]) respectively, $recency = 1$ represents the most recent data set (2019, several requirements specifications by Wang et al. [233]) and implementation (2019, the PASER tool by Wang et al. [233]). An artifact was considered available when its availability status code (see Table 1) was *broken* or better, as we assume that now-broken links worked upon the initial publication of the primary study.

(a) Distribution of data set availability

(b) Distribution of implementation availability

Figure 8: Visualization of original availability

Figure 9 visualizes the conditional effect of both recency and the artifact type on the original availability of artifacts as picked up by the trained model. The plot shows that data sets are, in general, more likely to be available upon publication of a primary study and corroborates the impression that the original availability improved over recent years, i.e., authors were more inclined to disclose their artifacts upon

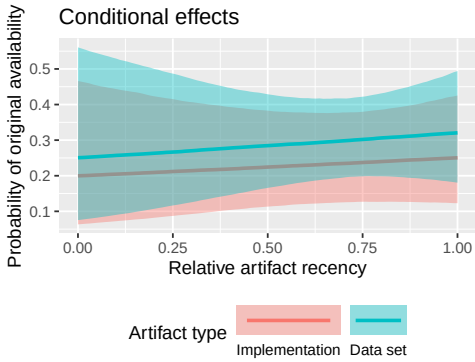


Figure 9: Conditional effect between recency and artifact type on original availability

publication.

These inferences are further strengthened by the random samples from the posterior distribution: while data sets are—on average and taking into account all uncertainty of the Bayesian model—made available in around 29%, implementations are only available in 23% of the time. The original availability of artifacts, in general, is 24.6% for the oldest and 29.2% for the most recent publications, representing an average increase of original availability of around 5%.

4.3.2 Factors influencing Artifact Persistence

The raw data already shows that all cases of artifacts not persisting occurred using private hosting services: all three non-persistent artifacts were hosted on private services. The sample of persistent artifacts is both privately and publicly hosted. All of the five artifacts hosted on public services had persisted. The conditional effect in Figure 10 shows a strongly positive influence of public hosting services on the probability of artifact persistence, i.e., publicly hosted artifacts are much more likely to persist than privately hosted artifacts. According to the sampling from the posterior, privately hosted artifacts persist on average in 59% of all cases, while publicly hosted artifacts persist in 76%. Additionally, recency has a positive influence on persistence when using a public hosting service, while the impact is negligible for private hosting.

Answer to RQ2: Factors influencing Artifact Availability

In the requirements quality research domain, more recent research articles are more likely to disclose research artifacts, and data sets are more likely to be disclosed than implementations. The use of a public hosting service has a positive impact on the persistence of these artifacts.

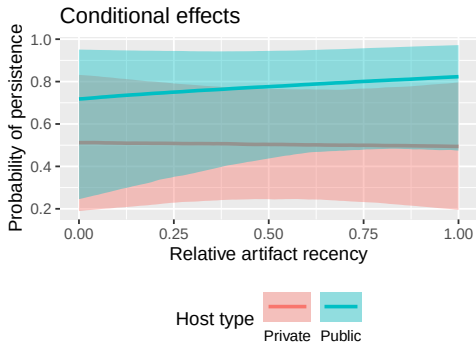
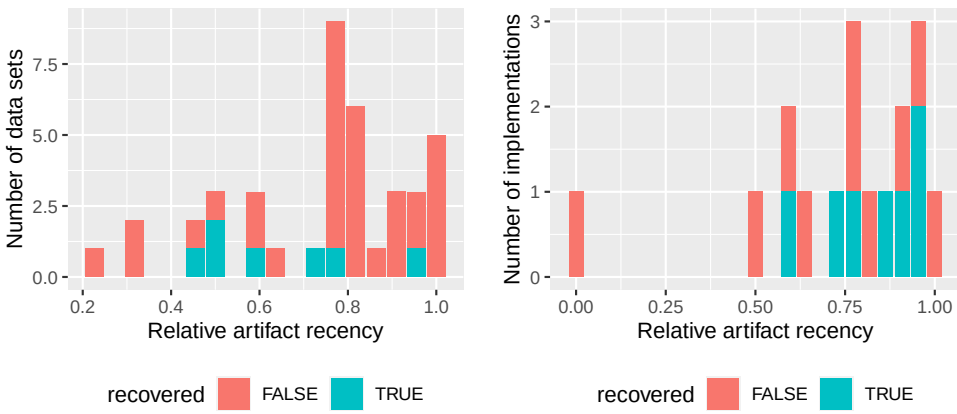


Figure 10: Conditional effect between recency and hosting type on artifact persistence

4.3.3 *Factors influencing Artifact Recoverability*

Figure 11 visualizes the distribution of recovered and non-recovered artifacts. The figure shows that the majority of artifacts addressed by their respective corresponding author during the artifact recovery initiative remained unrecoverable. The conditional effects in Figure 12 show that implementations were, in general, more likely to be recovered. According to the posterior samples, implementations have a probability of recovery of 38% while data sets only 23%. Recency shows no significant influence on the recoverability of implementations but a negative influence on the recoverability of data sets. This means that more recent data sets were less likely to be recovered.



(a) Distribution of data set recoverability

(b) Distribution of implementation recoverability

Figure 11: Visualization of Recoverability

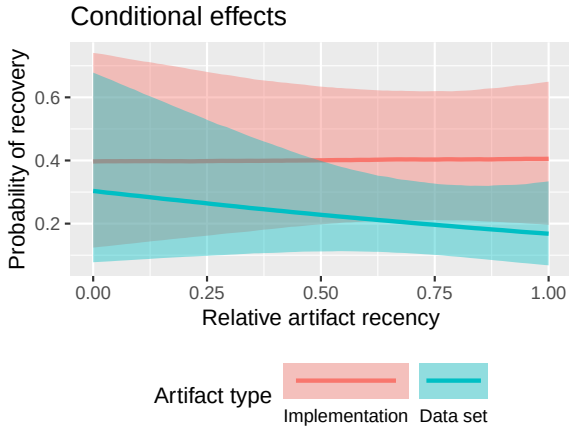
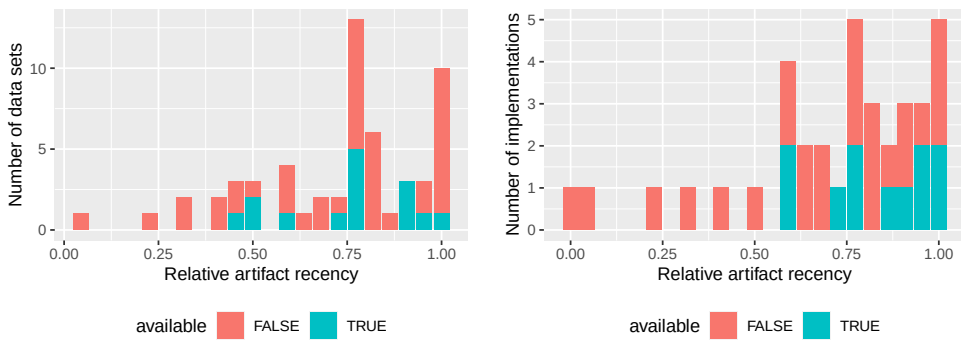


Figure 12: Conditional effect of recency and artifact type on recoverability

4.3.4 *Factors influencing overall Availability*

Figure 13 visualizes the distribution of the updated availability of artifacts in the sample, i.e., the overall availability after the artifact recovery initiative. The data shows and the conditional effects in Figure 14 confirm that recency has a positive impact on overall artifact availability, i.e., more recent artifacts are more likely to be available. According to the sampling from the posterior, the least recent artifacts are only available with a probability of 21% while the most recent with 33%. In our sample, the model perceived the difference in updated availability between implementations and data sets as negligible.



(a) Distribution of data set availability

(b) Distribution of implementation availability

Figure 13: Visualization of updated availability

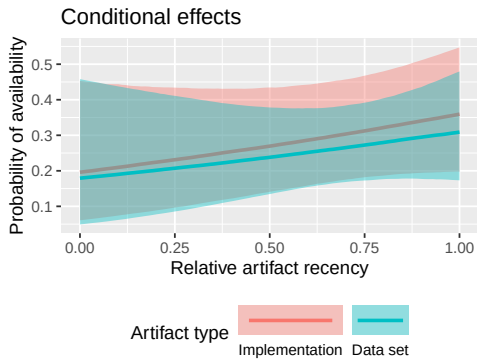


Figure 14: Conditional effect of recency and artifact type on overall availability

Answer to RQ3: Factors influencing Artifact Recovery

In the requirements quality research domain, implementations were more likely to be recovered than data sets. The recoverability of data sets decreased the more recent the publication in which they are contained was. After the two phases of the recovery initiative, data sets were equally likely to be available, and more recent artifacts were, overall, more likely to be available.

4.4 Interpretation

The data analysis on our sample of 94 artifacts allows the following inferences: the recency of artifacts benefits the original (h^{rec}_{orig}) and updated availability (h^{rec}_{avail}) while showing a negative effect on artifact recoverability (h^{rec}_{recov}) for data sets. A follow-up hypothesis is that this is due to the increased use of sensitive, company-owned data in recent publications, which does not allow the recovery of unavailable artifacts.

Data sets are overall more likely to be disclosed upon publication of a study (h^{type}_{orig}) but less likely to be recovered (h^{type}_{recov}) if they were unavailable. This raises the following concern: implementations produced by the authors of a study are likely owned by these authors, in contrast to the (potentially private, company-owned) data to evaluate an implementation. The lack of available implementations despite a high likelihood of them being owned by the corresponding authors constitutes a clear opportunity for improvement in requirements quality research.

Finally, the use of a public hosting service strongly benefits the persistence of artifacts (h^{host}_{per}), which provides support for the advocacy of preferring dedicated artifact hosting services over institutional or private services. As mentioned in Section 4.1, we cannot make a statement about the impact of long-term retention policies of public hosting services. However, recent research from Winter et al. [199] suggests that hosts committed to a long-term retention policy further improve the

artifact availability.

The data evaluation shows that—despite the moderate success of the artifact recovery initiative—artifact availability remains improvable in the requirements quality research domain. Specifically, the strong positive effect of public hosting services on artifact persistence motivates greater effort in promoting adherence to open science principles and the adoption of open science tools. We address this need with our artifact management guideline in Section 5.

4.5 Threats to Validity

The main threat to the validity of our empirical evaluation classifies as a threat to *internal validity* according to the guideline by Wohlin et al. [47], i.e., the causal link between independent and dependent variables. We selected the independent variables for the empirical evaluation based on their availability as a result of the recovery initiative. Consequently, the dependent variables may be influenced by other independent variables that were not considered in the evaluation, for example, an individual researcher's knowledge of open science principles. We address this threat by adhering to the modeling step of our analysis framework [54, 225], which makes our causal assumptions and, therefore, the considered variables explicit. The DAG allows scrutinizing and extending our causal assumptions by including additional variables in future research [48].

5 Open Science Artifact Management Guideline

The lack of adherence to open science principles in the requirements quality research domain—especially despite the evident benefit of open science platforms like Zenodo [199]—constitutes room for proactive improvement. Especially the unavailability of software artifacts, which are mostly produced and owned by authors of a publication, is unfortunate in requirements quality research and undermines a core deliverable of this research area [202].

Since open science first reached the SE community roughly 10 years ago (2011/2013) [202, 204], the understanding and use of open science principles have evolved and grown. This expansion necessitates a consolidation and communication of recommended practices in a simple, concise, and easy-to-read way. For this, we have created the Artifact Management Guideline [234] presented in this section. Section 5.1 explains how the guideline was derived and Section 5.2 its format. Section 5.3 summarizes the guideline [234] and Section 5.4 outlines its long-term vision. The guideline is archived at `https://doi.org/10.5281/zenodo.8134403`, and a collaborative version is accessible at `https://docs.google.com/document/d/1gIg3g-_zxCeiw2IJkBGbiGI9-3HeQU5FR63yAv3PhiM`.

5.1 Method

To derive this guideline, the authors combined their collective knowledge regarding open science [44, 45] with a review of recent guidelines for artifact evaluation tracks (AET) at premiere SE research conferences as well as further material regarding open science in software engineering. Most software engineering conferences have an AET [204], organizations such as ACM[10] and IEEE[11] have official open access policies, secondary-articles are being published regarding artifact availability [45, 199], meta-articles are being published with detailed descriptions and instructions on open science [44, 235], and new ideas about open science, artifacts, and reuse have begun to arise [236].

Using the work of Hermann et al. [204] as a starting point, we reviewed the AET guidelines of the International Conference on Software Engineering (ICSE), the International Conference on Requirements Engineering (RE), and the Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). We utilized the last two years (2021/2022) of both their AET guidelines and their open science policy where available. The work of Mendez et al. [44, 235] and Hermann et al. [204] guided the conceptualization and framing of some of the sections.

5.2 Chosen Format

The purpose of this guideline is to deliver information in a simple, concise, comprehensible, and collaborative manner while following a collegial tone directed at the reader. In that sense, we opted for a more pragmatic way and writing style for the guideline and, in consequence, for subsequent sections of this manuscript. We have chosen to create the guideline using GoogleDocs[12] and archive major increments via Zenodo (see [234]). The document is set to receive comments, of which its maintainers will be notified and respond to within the document to foster interaction and collaboration.

5.3 The Guideline

The objective of the guideline is to advise scientists on how to collect, document, license, archive, and share artifacts. "Artifacts" includes—but is not limited to—scientific protocols, raw and derived data (text, CSV, etc.), scripts, figures, tables, and software. The greater mission is to improve scientific rigor, encourage scientific collaboration, and increase the rate of scientific progression.

This guideline provides pragmatic support for the following five aspects of ar-

[10] `https://www.acm.org/publications/openaccess`
[11] `https://open.ieee.org/about/`
[12] `https://www.google.com/intl/en/docs/about/`

Table 6: Guideline Overview

Section	Aspect	Content
Section 5.3.1	Collect	Gathering all data relevant to the scientific work, including:
	• Open data	Raw and derived data, scientific protocols, tables, and extended findings
	• Open material	Software tools, data collection, transformation, and analysis scripts
	• Open access	Permanent, accessible, and unique identification of the manuscript and data
Section 5.3.2	Document	Creating a complete and coherent description of the artifact that allows installing, using, and evolving it
Section 5.3.3	License	Specifying the conditions under which the artifact and its constituents can be used
Section 5.3.4	Archive	Hosting the artifact in a permanent and accessible way with a unique identifier (DOI)
Section 5.3.5	Share	Disseminating the artifact to invite the scientific community to use and evolve it, therefore contributing value to the community

tifact management: collecting, documenting, licensing, archiving, and sharing. Table 6 summarizes the content of this guideline.

5.3.1 Collect

Scientific work is inherently exploratory and iterative, the result of which consists of drafted, incomplete, and often misplaced artifacts. This necessitates finding, improving, and reviewing the artifacts associated with a scientific article. The subject of this phase is the artifacts associated with open data, open material (inc. open source), and open access. The collected artifacts should be placed in one folder and organized in a logical manner, such as methodological phases.

Open data Open data covers all data that contributed to the scientific claims made in an article, as future researchers need the exact data used in your scientific work to replicate, verify, and improve on scientific work. This includes raw data, derived data, scientific protocols, but also figures, tables, and extended findings.

Raw data is used to generate or support claims in scientific work. This data tends to be untouched by the analysis, sometimes even before cleaning (since data cleaning may have introduced bias). Derived data is created as a result of scientific analysis (automatically or manually), like models trained through machine learning algorithms and data created as a result of qualitative methods, such as coding tables, schemata, etc. Scientific Protocols pertaining to the planning, execution, and adjustment of scientific work. This includes logs of decisions taken by participating scientists, protocols given to study participants, rationale for change requests, discussion notes, etc. Figures and tables used to visualize results in the manuscript are as relevant to include as the code to generate them. Furthermore, extended findings that did not fit into the manuscript fall into this category.

Open material Open material (including **open source**) covers all material that contributed to the scientific claims made in an article. Future researchers need these algorithms to replicate, verify, and improve on published work. This includes the following data collection scripts, data transformation scripts, analysis scripts, and software tools. Data collection scripts are scripts used to collect research data, e.g., a custom web scraper, a script to iteratively access an API, or an HTML-to-SQL data-writing script. Data transformation scripts are scripts used to transform data in unique ways, e.g., static analysis, machine learning, image recognition, or generative AI. Analysis scripts are scripts used to analyze the final output data and potentially produce the published results. This includes scripts that generate figures or tables. Software tools are used in the research, e.g., a custom survey tool, a new IDE, a Jira or GitLab plugin, etc.

Open access Finally, open access ensures that future users can find and access the associated article by including a permanent DOI link to an open-access article in the README, as well as the permanent DOI to the published version of the artifact. Section 5.3.2 contains details on how to document and Section 5.3.3 on open access licensing.

5.3.2 Document

Artifacts need to be documented such that they are approachable to someone unfamiliar with the employed workflow, development style, and organizational mindset. At a minimum, this requires a `README.md` file in the top-level artifact folder containing the following sections:

- **Summary of artifacts**: a concise description of the motivation and purpose of the artifact.
- **Author and article details**: a list of involved authors, including contact information such as emails and how to cite the work. This information may be subject to change as the proper citation string or the DOI of the article may not be known while preparing the artifact for submission. Provide as much information as possible at the current point in time and update the `README.md` once the information becomes available.
- **Description of artifacts**: an explanation of the folders and files, including what was not included (and why).
- **Licenses**: the chosen licenses for the artifacts (see Section 5.3.3 for more details).

If using the artifact requires more than opening PDFs, CSVs, etc., then an `INSTALL.md` file (also located in the top-level artifact folder) becomes necessary. It should include the following sections:

- **System requirements**: a generalization of the environment and programs necessary to execute any software or scripts.
- **Installation instructions**: an instruction on how to execute the software or scripts in question. If possible, a virtualized setup (e.g., via Docker or a virtual environment) shall be provided.
- **Steps to reproduce**: commands on how to reproduce the data, figures, tables, or results presented in the article. If the artifact is simple enough to not necessitate an `INSTALL.md` file, this section can be moved to the `README.md` file.

Additional visualizations of the artifacts, e.g., a UML diagram of a software system, aid their accessibility and understandability [237–239].

5.3.3 License

An essential part of sharing artifacts is an explicit statement about their (re)use as determined by licenses. Open data requires a license attached to explicitly describe how third parties can use the data. The Creative Commons (CC) licenses[13] are often employed licenses for open data and offer a variety of regulations determining what third parties are allowed to do with the data.

The license applied to open material takes on the form of an open-source license, which applies specifically to source code. Several online resources assist the selection of an open source license.[14]

The copyright license applied to an article itself is already decided through the copyright agreement with the publisher (IEEE, ACM, Elsevier, etc.). Understanding an author's rights over their work, particularly across the different versions of your article ("author-submitted article", "accepted article", "final published version", etc.), is essential in ensuring open access to research articles. Tools like Sherpa ROMEO (now just "Sherpa"),[15] assist in checking compliance with publisher copyright models.

Once appropriate licenses are determined for all artifacts of an article—including the identification of licenses of not self-owned by reused, external artifacts—a section in the `README.md` document should state and explain the chosen licenses as discussed in Section 5.3.2. An additional and encouraged norm is to obtain the licenses as text files and place them in the artifact folder.

5.3.4 Archive

A critical step to artifact availability is to upload it to a publicly available archival website. A hosting website should be selected when it meets all of the following

[13] `https://creativecommons.org/about/cclicenses/`

[14] E.g. `https://choosealicense.com` and `https://opensource.org/licenses`

[15] `https://beta.sherpa.ac.uk/`

three criteria:

1. **Hosted online for public access**: Artifacts are hosted online for anyone to access via the internet. Additionally, there is no need for registration to access the artifacts.[16]

2. **Dedicated DOIs and immutable data**: DOIs are automatically created for artifacts, and both the DOIs and data they point to are immutable. Artifacts can be updated, but each version must be maintained with its own DOI.

3. **Long-term maintainability**: The organization hosting the URL has committed to a long-term retention policy, i.e., it plans to maintain the artifact for the foreseeable future. For example, Zenodo states in its policies that "Items will be retained for the lifetime of the repository" [240] which covers "the next 20 years at least" [240].

In principle, any organization that fulfills the above requirements can be used for archiving artifacts. In reality, there are currently only a few known organizations that satisfy these requirements: ArXiv for articles, and Zenodo and FigShare[17] for all types of artifacts.

The following notable services do usually not satisfy the above requirements despite their recurring use. Institutional websites,[18] employee web pages and research group websites usually do not satisfy criterion 2 and 3, since institutions update their websites over time and do not maintain access to resources and URLs as supported by previous research by Winter et al. [199]. Employee web pages are taken offline when the employee leaves. Similar problems exist for research group websites. Cloud storage providers, such as Dropbox, Google Drive, OneDrive, and iCloud, are solutions for backing up and/or syncing data, not archiving data. For example, individuals can change the data and URLs at any time. GitHub, GitLab, and other Social Git/Code Platforms offer features for social product development, which conflict with the requirements for open science artifact archival. For example, they allow repositories to be deleted or renamed.

Nevertheless, platforms like GitHub are important for open-source software, as they offer a suite of features designed to foster an open and collaborative environment. Some of these features, however, are in direct conflict with open science principles. For example, GitHub allows deleting a repository. However, already published articles are difficult to update, such that links to research artifacts may no longer be

[16] "Free registration" is not acceptable, as it is an additional barrier to obtaining the artifacts, and "free" often involves hidden clauses.

[17] Note that FigShare is a "for profit" commercial organization, which may affect long-term maintainability.

[18] Institutional websites can conform to the three open science requirements for archival as described above, in which case they would qualify for archival. However, this is usually not the case.

resolved. This undermines the recoverability of the artifact [46]. The conflict can be avoided by utilizing both services jointly, i.e., one archival service for artifact persistence and one Git-based version control service for collaboration. Step-by-step guides to automatically archive GitHub project releases to Zenodo[19] and FigShare[20] enable this synergy.

5.3.5 Share

Science entails dissemination as much as knowledge-building. Science without communication is as empty as science without findings, and neglecting the dissemination of research artifacts reduces their potential impact. Sharing research artifacts is, hence, an integral part of managing them. Social media platforms like Twitter lowered the barrier of dissemination significantly. Announcing research articles by summarizing the motivation, approach, findings, and artifacts in a few sentences each is a valid first step to garner interest and invite interaction.

5.4 Long-Term Vision

Our vision is to maintain the online GoogleDoc version of this guideline as our community's understanding of open science evolves, and the desires of our research community for open science standards grow. This vision requires the collective knowledge and effort of our community. Just as science is conducted, we hope to have this guideline discussed, reviewed, challenged, and updated. For this reason, the GoogleDoc version has:

- **Open access** for all to share, read, and comment (via the GoogleDocs comment feature)
- **Document notifications** enabled for the primary maintainers (for quick activity reaction)
- **Document history** and version numbers for transparent and traceable evolution

In addition, we archive significant increments via Zenodo for persistence.

5.5 Projected Use

We believe that the guidelines presented in this section contribute to the availability, persistence, and usability of research artifacts in software engineering. The guidelines support the authors of scientific work in preparing, disclosing, and sharing the

[19] `https://guides.github.com/activities/citable-code/`

[20] `https://help.figshare.com/article/how-to-connect-figshare-with-your-github-account`

artifacts connected to their research. The checklist format in the guidelines [234] makes the content suitable also to junior researchers.

We further invite organizers of scientific events like conferences and workshops to disseminate the guidelines among the authors so that they can receive more systematic guidance when sharing their artifacts. Instead of migrating artifact evaluation guidelines from one event website to the next, as they are typically used for only one instance of an event, our guidelines constitute a central and maintainable artifact that can be referred to from any website. This shall ensure that any progress of the community regarding artifact availability is recorded centrally.

6 Conclusion and Future Work

The availability of research artifacts is a vital precondition for the reproducibility of scientific work [46], on which the reliability and robustness of scientific results hinges [195, 196]. In this work, we contribute both to the availability of research artifacts of previous publications in the area of requirements quality research through a two-phase, crowd-sourced recovery initiative resulting in 10 recovered data sets and 7 recovered implementations and to the availability of research artifacts of future publications through the compilation of a concise, pragmatic artifact management guideline. Additionally, we derive insights into the reasons for unavailability based on our sample of 57 primary studies from the requirements quality literature, including empirical evidence for artifact availability improving over time and public hosting services positively influencing artifact persistence.

Improving the availability of research artifacts through adherence to open science principles is a continuous effort we aim to contribute to with this study. We hope the insights derived from our data analysis and the resources provided in our replication package, including the recovery request generation script and artifact management guideline, allow the reproduction of this study in other fields of research to extend the scope of this study and shape a more accessible landscape of research artifacts for future researchers.

Paper V

A Second Look at the Impact of Passive Voice Requirements on Domain Modeling: Bayesian Reanalysis of an Experiment

Abstract

The quality of requirements specifications may impact subsequent, dependent software engineering (SE) activities. However, empirical evidence of this impact remains scarce and too often superficial as studies abstract from the phenomena under investigation too much. Two of these abstractions are caused by the lack of frameworks for causal inference and frequentist methods which reduce complex data to binary results. In this study, we aim to demonstrate (1) the use of a causal framework and (2) contrast frequentist methods with more sophisticated Bayesian statistics for causal inference. To this end, we reanalyze the only known controlled experiment investigating the impact of passive voice on the subsequent activity of domain modeling. We follow a framework for statistical causal inference and employ Bayesian data analysis methods to re-investigate the hypotheses of the original study. Our results reveal that the effects observed by the original authors turned out to be much less significant than previously assumed. This study supports the recent call to action in SE research to adopt Bayesian data analysis, including causal frameworks and Bayesian statistics, for more sophisticated causal inference.

Keywords: Requirements Engineering, Requirements Quality, Controlled Experiment, Bayesian Data Analysis

1 Introduction

Requirements specifications serve as input to several subsequent software engineering (SE) activities [21]. Consequently, the quality of requirements specifications im-

pacts the performance of these dependent activities [18]. For example, ambiguous or incomplete requirements specifications may result in incorrect or missing features when implementing the requirements. Because the cost for remediating these defects scales the longer they remain in the development process [102], organizations are interested in detecting and removing requirements quality defects as soon as possible.

The requirements quality research domain aims to meet this need [6]. However, while requirements quality research abounds with normative rules about requirements quality [41], it lacks empirical evidence that supports the relevance of these rules [6, 40]. Moreover, the few studies contributing empirical evidence are often confounded, too abstract, and their inference reduces complex, context-sensitive data to binary results, for example, through the use of frequentist methods [241]. The insufficient quantity and quality of evidence impede the adoption of requirements quality research in practice [9].

With this study, we aim to demonstrate how more sophisticated inference methods than frequentist approaches derive deeper insights from an empirical study and may even revise frequentist claims. This paper makes the following contributions:

1. A recovery of the analysis of one of the only controlled experiments on requirements quality known to us [34].

2. A reanalysis of the hypothesis of this experiment using more sophisticated statistical methods.

Data Availability

We disclose all supplementary material, including the data, figures, and analysis scripts, in our replication package.[1]

2 Related Work

2.1 Requirements Quality

Requirements quality research is a sub-domain within requirements engineering (RE) research dedicated to the assessment and improvement of requirements artifacts and processes [6]. Given the importance of RE to the software development life cycle, the quality of its artifacts and processes plays a major role in project success or failure [18, 21]. For requirements artifacts like (systematic) requirements specifications, use cases, user stories, and others [23], a popular concept to identify quality defects is the *requirements quality factor* [41]. A requirements quality factor is a normative metric that maps a requirements artifact onto some level of quality based on

[1] `https://zenodo.org/doi/10.5281/zenodo.10283010`

defined criteria [41]. One commonly researched requirements quality factor is *passive voice* [34, 171], which associates the use of passive voice in a natural language (NL) requirements sentence with bad quality since it potentially omits the semantic agent of the sentence [34]. For example, the requirements specification "If the settings *are changed*, ..." obscures the agent of the requirement. An active formulation of this specification, "If an administrator *changes* the settings, ..." makes the agent explicit.

Recent research has identified a major shortcoming of requirements quality factors, namely their relevance [40]. The requirements quality research domain abounds with publications proposing new quality factors and tools to detect violations against them but lacks empirical evidence for the implied causal relationship, i.e., that the violation causes an actual impact on subsequent SE activities [242]. A previous literature survey has revealed that among 57 primary studies proposing requirements quality factors, only 40 discuss their impact at all, and of these, only 11 provide some sort of empirical evidence [40]. Without empirical evidence of the impact of a requirements quality factor on subsequent activities, these factors do not reliably identify requirements quality defects that matter. Practitioners rightfully harbor skepticism toward requirements quality research given this lack of evidence which impedes research adoption in practice [1, 8, 9].

For example, while several sources advise against the use of passive voice as described above [33, 130, 158, 171] only two publications known to the authors investigate its actual impact on subsequent activities. Krisch et al. conducted a document study in which domain experts classified active and passive requirements sentences as either problematic or unproblematic [172]. The results indicate that passive voice is generally unproblematic as adjacent text often compensates for the information omitted due to the passive voice. Femmer et al. conducted a controlled experiment with university students to assess how passive voice in requirements sentences impacts the domain modeling activity [34]. The authors conclude that passive voice requirements increase the number of missing associations with statistical significance but not the number of missing actors or domain objects.

2.2 Inferential Statistics

Most statistical methods applied in SE beyond descriptive statistics are limited to frequentist inferential statistics. These usually take the form of null hypothesis significance testing (NHST), which stratifies the distribution of a dependent response variable by one or more independent variables and compares their mean. We assume that the popularity of these methods stems from the established guidelines [47], the availability of tools to perform them, and their acceptance in the community.

However, frequentist methods like NHST have several shortcomings. From a research design perspective, they overemphasize the variables involved in an alleged,

causal relationship without a systematic approach for addressing confounders [76]. From a data analysis perspective, common issues like the multiple-hypothesis problem [243] and the unscientific practice of fishing for significant test results below an arbitrary significance level [244] are well-known, yet still occur in practice [245]. Moreover, NHST reduces complex, context-sensitive data down to binary answers (i.e., whether there is a significant difference in the distributions' mean or not), which leads to superficial and overly abstracted research results that are void of any uncertainty that the data originally encoded [48].

The recent rise of Bayesian data analysis (BDA) aims to mitigate these shortcomings [62, 241] by (1) embedding inferential statistics in causal reasoning frameworks [54, 76] and (2) applying Bayesian statistics, i.e., encoding the uncertainty of the impact that independent variables have on dependent variables in probability distributions [62]. Prior to any data analysis, involved variables and their causal relationship are made explicit. During the data analysis, explicit prior assumptions are updated in light of the observed data using Bayes' Theorem. As a result, BDA produces uncertainty-preserving statistical inferences with explicit causal assumptions. Recently, SE researchers have started to advocate for the adoption of BDA methods [48, 226, 246] but they still remain to be niche [54].

3 Method

In this study, we aim to demonstrate how frameworks for causal inference and Bayesian statistics provide more sophisticated insights which reduce issues of drawing inappropriate conclusions from empirical studies. To this end, we reanalyzed the data of a previous controlled experiment using BDA. Section 3.1.1 presents the design of the original experiment and Section 3.1.2 elaborates on the issues with the experiment. Section 3.2 then presents the reanalysis performed in the scope of this study.

3.1 Original Experiment

The original experiment by Femmer et al. aims to understand the impact of passive voice in requirements on domain modeling [34] by asking the following research questions:

- RQ1.1: Is the use of passive sentences in requirements harmful for finding actors?
- RQ1.2: Is the use of passive sentences in requirements harmful for identifying domain objects?
- RQ1.3: Is the use of passive sentences in requirements harmful for identifying associations?

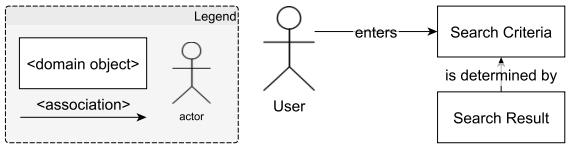


Figure 1: Domain model example

3.1.1 Design

The experimental task was to create a domain model based on a single-sentence NL requirements specification. The domain model consisted of the following three types of elements: *actors*, which represent human participants in the requirement, *domain objects*, which represent any non-human entities in the requirement, and *associations*, which connect elements that have a relationship according to the requirement. Figure 1 visualizes a domain model for the requirements specification "The system shall be capable of returning the search results latest 30 seconds after the user has entered the search criteria." [34]

The authors of the original study conducted a controlled experiment with independent measures, i.e., every participant is assigned to only one treatment [47]. The authors recruited $n_p = 15$ participants for the experiment. The participants consisted of two Bachelor students, eight Master students, four Ph.D. students, and one student with an unknown background. In addition to the participants' study program, the authors also recorded their age group as well as their industrial and academic experience in SE, RE, and programming on an ordinal scale.

To enable independent measures, seven participants were assigned to the control group (A) and eight to the treatment group (P). The control group received the requirements specifications in active formulation. The treatment group received semantically similar requirements specifications in passive formulation. For example, the authors transformed the aforementioned active requirements sentence to the following passive formulation for the treatment group: "The search results *shall be returned* no later 30 seconds after the user has entered the search criteria." [34].

After assessing the general SE and RE knowledge in a quiz, the participants conducted the experimental task. Every participant received $n_r = 7$ requirements specifications, such that the experiment produced $n_p \times n_r = 105$ observations. The authors then compared the 105 domain models with the sample solution and counted the number of missing actors, domain objects, and associations. To evaluate the hypotheses implied by the research questions, the authors summed up these numbers for all seven requirements sentences of each participant. Each participant was associated with a total number of missed actors, domain objects, and associations throughout all seven requirements. Then, the authors calculated the mean and median number of missing elements for the control and treatment groups and conducted a Mann-Whitney test with a 95% confidence interval to determine whether there was

Table 1: Results of the original study [34]. P-values indicating a statistically significant difference with $\alpha = 0.05$ are prefixed with an asterisk (*)

Element	Mean (A)	Mean (P)	Median (A)	Median (P)	P-Value	Conf. Int.	Cliff's δ
Actors	0.43	1.00	0	1	0.10	(0; ∞)	0.39
Objects	1.29	2.00	1	1	0.25	(-1; ∞)	0.25
Associations	4.14	7.88	3	8	*0.02	(1; ∞)	0.75

a statistically significant difference between the two groups.

Table 1 shows the results of the original study [34]. With a significance level of $\alpha = 0.05$, the NHST rejects only the null hypothesis implied by RQ1.3 ($p = 0.02 < \alpha$). The authors conclude that the use of passive voice does not have a statistically significant impact on the number of actors and domain objects missing from resulting domain models, but it does have an impact on the number of missing associations.

3.1.2 Issues

The original experiment by Femmer et al. [34] suffers from at least the following issues.

Issues with reproduction The authors originally disclosed their experiment data at `http://goo.gl/WlTPE5`, which was forwarded to `https://www.in.tum.de/i04/~femmer/data/passives_experiment.zip`. However, this link does no longer resolve given that institutional websites commonly discontinue hosting resources of members that change their affiliation [199, 206]. Thankfully, the authors of the original paper were able to recover the lost replication package [45] and archived it via Zenodo.[2] Still, the replication package contains only the study protocol and obtained data, but not the script to reproduce the evaluation. The lack of reproducibility impedes our goal of comparing methods of statistical inference.

Issues with drawing appropriate conclusions The employed research design and analysis risks drawing inappropriate conclusions in two regards. Firstly, the significance test investigates the isolated impact of passive voice on the three dependent variables. Possible confounders, like the experience of participants, were recorded but not considered in the evaluation. Secondly, frequentist NHSTs reduce the data to single, binary results, omitting any uncertainty [48] and comparing point estimates, which are unreasonably precise.

Issues selecting an appropriate study design The selected experimental design introduced one more potential confounder. Because the authors of the original study used an *independent measures* design [47] the evaluation does not account for between-subject variance [50]. In other words: the evaluation does not consider that the ob-

[2]Now available at `https://zenodo.org/records/7499290`

served differences in the dependent variables are caused by the treatment or by other factors like the individual skill of each participant.

3.2 Reanalysis

We address the first of the three issues by reproducing the original evaluation and disclosing it for future replication. For this, we extracted the experimental results from the original study and performed the evaluation according to the information in the manuscript [34]. The reproduced evaluation script is contained in our replication package.

To address the second and third issue, we reanalyze the data generated by the experiment using an established framework for causal inference and Bayesian instead of frequentist methods. The framework allows us to (1) revise and extend the causal assumptions of the original experiment and (2) consider potential confounders in the analysis, while the use of BDA allows us to (3) generate more sophisticated inferences that preserve the uncertainty of the causal influences.

We employ the framework for statistical causal inference that was developed by Siebert [54]. This framework is based on Pearl's original model of causal inference [76] and consists of the three major steps modeling, identification, and estimation. The following paragraphs briefly summarize each of these steps and are further elaborated in our replication package. For a gentler introduction to frameworks for statistical causal inference, we refer the interested reader to appropriate literature [54, 76]. For a gentler introduction to BDA, we refer the interested reader to appropriate textbooks [62] or descriptive demonstrations of the application of BDA in SE research [48, 226, 246, 247].

3.2.1 Modeling

In the first step, we make our causal assumptions of the phenomenon under investigation explicit [54]. These causal assumptions are specified in a directed acyclic graph (DAG), in which nodes represent variables and directed edges between them represent assumed causal effects of one variable on another [248]. In our reanalysis, the eligible variables are limited to the variables collected during the original experiment [34].

3.2.2 Identification

In the second step, we select all variables that form the so-called adjustment set. Four causal criteria inform this selection and prevent variable bias like colliders or backdoors [62], mitigating that non-causal correlations do not influence the causal relation of interest. The selection of the adjustment set mitigates the second issue mentioned in Section 3.1.2.

3.2.3 Estimation

In the third and final step, we derive a regression model from the adjustment set of eligible variables. We first select an appropriate probability distribution type to represent each of the three response variables based on the maximum entropy criterion [249] and ontological assumptions. All three variables are whole numbers bounded by the number of expected actors, domain objects, and associations. Consequently, we model all response variables with Binomial distributions.

We model the parameter p—which defines the shape of the Binomial distribution—in dependency of all eligible independent variables, called the predictors. Each predictor is multiplied with a coefficient that represents the strength and direction of the influence that the predictor has on the response variable. To begin, we assign an uninformative prior distribution to each of these coefficients, i.e., a normal distribution centered around $\mu = 0$ with a standard deviation of $\sigma = 1$. This represents our prior belief of the causal relationship between the predictors and response variables, which are yet unknown. We confirm the appropriateness of the selected priors via prior predictive checks [230].

The predictors of each response variable consist of the independent variables selected during the identification step. Further, we include the following variables as predictors:

- Intercept: The global average of missing any element of the domain model. This represents the general challenge of creating a domain model from an NL requirements specification, independent of any predictor values.

- Participant-specific intercept: The participant-specific average of missing any element of the domain model. This represents the general skill of a participant.

- Requirement-specific intercept: The requirement-specific average of missing any element of the domain model. This represents the general complexity of a requirement.

While involving a global intercept is a general best practice [62], the two group-specific intercepts retain local variance in the model [247]. The resulting hierarchical model makes use of partial pooling, which is understood to outperform purely global or local models [62, 247]. The inclusion of a participant-specific intercept mitigates the third issue mentioned in Section 3.1.2, as it represents between-subject variance in the statistical evaluation.

Given the selected probability distribution and predictors, we train one Bayesian model for each of the three response variables with the experimental data gathered during the original experiment [34]. We conduct this step using the `brms` library [228] in *R*. During the training process, Hamiltonian Monte Carlo Markov Chains update the prior distributions of the predictor coefficients to better reflect the impact of the

Table 2: Results of the strict reproduction

Element	Mean (A)	Mean (P)	Median (A)	Median (P)	P-Value	Conf. Int.	Cliff's δ
Actors	0.43	1.00	0	1	0.19	(0; 1)	0.38
Objects	1.29	2.00	1	1	0.50	(-1; 3)	0.22
Associations	4.14	7.88	3	8	*0.03	(1; 7)	0.68

predictors in light of the observed data [250]. This produces the posterior distributions of the predictor coefficients, which then represent the updated belief of the model about the strength and direction of the influence with which a predictor impacts a response variable. The standard deviation of each coefficient reflects the uncertainty of the impact of its associated predictor. This further mitigates the second issue mentioned in Section 3.1.2 by retaining the uncertainty of each impact.

We confirm that the model was trained appropriately by inspecting the Markov Chains [62] and by performing posterior predictive checks [230]. Finally, we evaluate the trained models by plotting the marginal effects of relevant predictors, mainly the use of passive voice. The marginal plots show the distribution of the response variable for all levels of the selected predictor while keeping all other predictors at representative levels. The resulting mean predictions and confidence intervals visualize the difference that the chosen predictor has on the response variable. This visualization represents the isolated effect of that predictor on the outcome.

4 Results

4.1 Reproduction of the original evaluation

Table 2 shows the strict reproduction of the experimental results using the same frequentist methods as the original study [34]. The mean and median values match exactly. The calculated p-values differ (0.10 vs. 0.19, 0.25 vs. 0.50, 0.02 vs. 0.03), but using the same significance level $\alpha = 0.05$ would result in the same hypotheses being rejected (i.e., only the hypothesis implied by RQ1.3). Similarly, the effect size calculated via Cliff's δ matches with a margin of 0.07. Only one extreme end of every confidence interval could not be reproduced. We assume this to be due to incorrect calculation or reporting in the original study.

4.2 Reanalysis of the data using BDA

Figure 2 visualizes the DAG that makes the causal assumptions of the phenomenon under investigation explicit. The DAG is populated with all variables recorded during the original experiment [34] and connected with all causal relationships that we assume based on our prior knowledge. The causal relationships between the main factor (red node) and the three dependent response variables (turquoise nodes) were already assumed in the original study [34] and are the main relationships of interest.

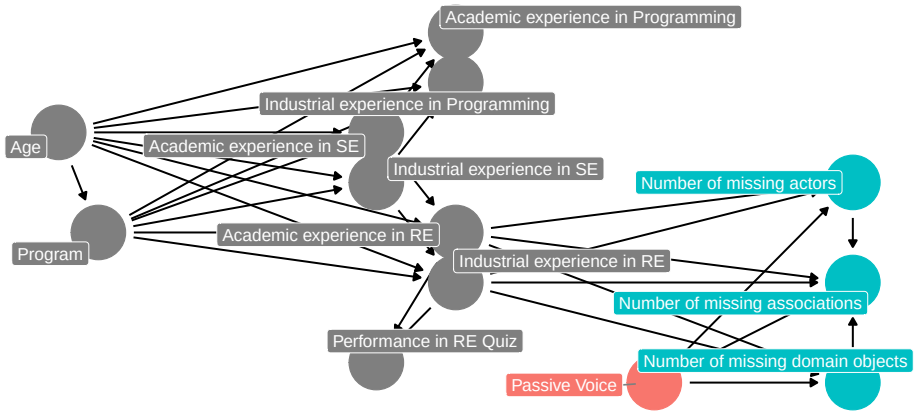


Figure 2: Full DAG visualizing the causal assumptions (red: exposure/main factor, turquoise: response/dependent variables)

We assume additional relationships, for example:

- Age → Program: The older a participant, the more likely it is that they have progressed further in their studies.
- Program → Academic experience in RE: The more advanced the study program, the higher the academic experience that a student has collected in RE.
- Academic/industrial experience in RE → number of missing actors/domain objects/associations: The higher the experience in RE, the fewer mistakes a student makes during domain modeling.
- Number of missing actors/domain objects → Number of missing associations: Missing an actor or domain object leads to missing an association, as one of the two nodes connected through an expected association is unavailable.

All other causal assumptions and their justification can be found in our replication package. Figure 3 visualizes the reduced DAG resulting from the identification step. This DAG contains only variables included in the adjustment set, i.e., all variables relevant for the causal analysis. The causal effect of all excluded variables passes through these remaining variables. Hence, they suffice to model the causal influence on the response variables.

Figure 4 visualizes the marginal effects of the main factor (passive voice) on the three response variables. All plots show that the use of passive voice slightly raises the mean of the response variable distribution, i.e., the use of passive voice increases the likelihood of missing more actors, domain objects, and associations. However, the confidence intervals of the main factor overlap in all three cases, meaning that

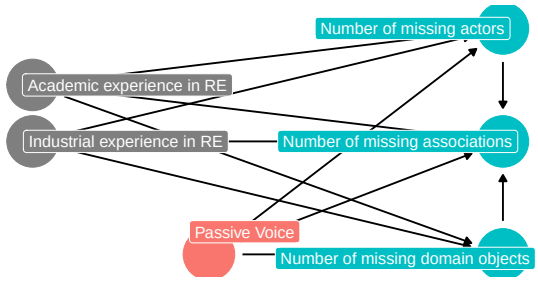


Figure 3: Reduced DAG including all variables eligible for the regression model

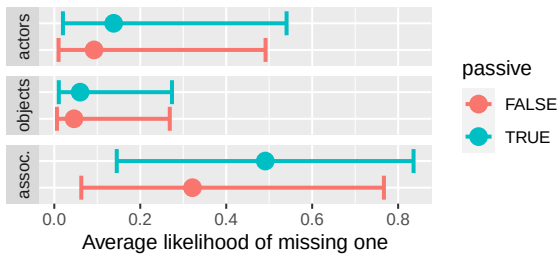


Figure 4: Isolated impact of passive voice on the likelihood of missing an actor, object, or association (``assoc.'')

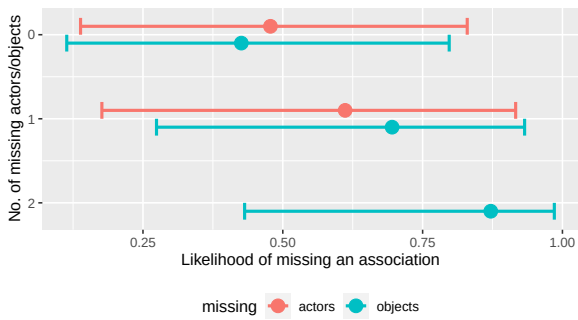


Figure 5: Impact of the number of missing actors and objects on the likelihood of missing an association

this difference is not significant. The chance that the use of passive voice results in equal or even fewer missing actors, domain objects, and even associations remains.

Figure 5 shows the marginal effects of the number of missing actors and missing domain objects on the likelihood of missing an association. The plot shows that missing an actor or domain model increases the likelihood of missing an association, which confirms the causal assumption represented in our DAG. The average and confidence interval for the number of missing actors (red in Figure 5) is only defined for 0 and 1 because the experiment data did not contain any observation with more than one missing actor per domain model.

5 Discussion

Finally, we discuss the implications of the results in Section 5.1 and address remaining threats to validity in Section 5.2.

5.1 Implications

Issues of reproduction can be overcome as long as the authors of the original work preserve their replication package. This encounter supports the observation by Gabelica et al. [206] and Winter et al. [199] that replication packages hosted on institutional websites are prone to become inaccessible over time. We strongly advise hosting replication packages via services that committed to a long-term retention policy, like Zenodo[3] or figshare.[4]

More importantly, the reanalysis presented in this study shows that the lack of a framework for causal inference as well as frequentist methods may cause issues with drawing appropriate conclusions. The results of the reanalysis revealed that the use of passive voice does not have a significant impact on the number of missing associations in resulting domain models as claimed in the original study [34]. Instead, the use of a framework for causal inference showed that this impact is confounded by the number of missing actors and domain objects, which also do not experience a significant impact by the main factor of interest. Additionally, the use of Bayesian statistics highlighted that the remaining difference in the response variables is uncertain and not significantly different.

These insights imply two recommendations for future research. For research design, the use of an explicit framework for causal inference provides a systematic approach for dealing with potential confounders [76, 226]. For data analysis, the use of Bayesian statistics retains uncertainty and allows transparent inferences from empirical data [48, 62, 246].

5.2 Threats to validity

The reanalysis continues to suffer from threats to validity. We discuss these according to the classification by Cook et al. [251].

Construct validity The construct validity suffers from *inadequate preoperational explication of constructs* for all variables concerning experience [251]. In the experiment, industrial and academic experience in RE—two of the predictors with an impact on the three response variables—are measured on an ordinal scale with four levels: no experience, up to 6 months, 6 to 12 months, and more than 12 months [34]. Whether these variables adequately reflect experience remains questionable.

[3] `https://zenodo.org/`

[4] `https://figshare.com/`

Internal validity The internal validity suffers from potential *confounders*. The reanalysis could only involve the variables recorded during the original study and was, therefore, constrained to the variables listed in Figure 2. Other variables with a potential causal impact on the response variables—like domain knowledge or prior training in domain modeling—were not available. The internal validity further suffers from an unknown *interaction with selection* due to the design of the experiment. Given the independent measures design, each participant was exposed to only one treatment [47, 50]. This produced the risk of an interaction effect between the participant and the treatment, i.e., participants of one group could excel with their respective treatment for unknown reasons.

External validity The external validity suffers from an *interaction of selection and treatment*, i.e., the experiment participants are potentially not a representative sample of the intended target population. The study only involved university students of different programs. Hence, there is no evidence that the conclusions are generalizable to SE practitioners.

6 Conclusion

This study reanalyses the only controlled experiment investigating the impact of passive voice in requirements specifications [34] by employing a framework for statistical causal inference [54] and using Bayesian in contrast to frequentist data analysis methods [48]. We could show that the results of the original study are much less significant than suggested by the frequentist analysis and that passive voice has, in consequence, a much smaller impact in the studied context than the original study had assumed.

Needless to say, our aim is not to criticize the original study [34] itself. In fact, we would like to acknowledge the authors' contributions to the requirements quality research domain, especially as controlled experiments were, and still are, rare in this domain [40]. Instead, our intention is to critically reflect upon frequentist analysis that still constitutes the prevalent choice in the empirical software engineering community with little to no attention to its limitations.

Our reanalysis continues to suffer from several threats to validity. For example, the experimental design made it impossible to identify whether some participants performed particularly well or badly given their assignment to the control or treatment group. Using a crossover design in which all treatments are applied to all subjects could mitigate this threat [50].

One hope that we associate with our study is to raise awareness of the shortcomings of frequentist analyses, especially when applied as a universal tool. We especially hope that our short demonstration, as well as our replication package, will caution fellow SE researchers to use out-of-the-box frequentist approaches and, in-

stead, encourage them to consider Bayesian data analysis approaches [62], which include (1) proper frameworks for statistical causal inference [54, 76] and (2) Bayesian statistics [48, 226]. These approaches ensure that experimental designs are informed by explicit causal assumptions, and their execution produces more sophisticated inferences preserving uncertainty, in turn enriching scientific contributions to be more reflected and insightful.

Paper VI

Crossover Designs in Software Engineering Experiments: Review of the State of Analysis

Abstract

Experimentation is an essential method for causal inference in any empirical discipline. Crossover-design experiments are common in Software Engineering (SE) research. In these, subjects apply more than one treatment in different orders. This design increases the amount of obtained data and deals with subject variability but introduces threats to internal validity like the learning and carryover effect. Vegas et al. reviewed the state of practice for crossover designs in SE research and provided guidelines on how to address its threats during data analysis while still harnessing its benefits. In this paper, we reflect on the impact of these guidelines and review the state of analysis of crossover design experiments in SE publications between 2015 and March 2024. To this end, by conducting a forward snowballing of the guidelines, we survey 136 publications reporting 67 crossover-design experiments and evaluate their data analysis against the provided guidelines. The results show that the validity of data analyses has improved compared to the original state of analysis. Still, despite the explicit guidelines, only 29.5% of all threats to validity were addressed properly. While the maturation and the optimal sequence threats are properly addressed in 35.8% and 38.8% of all studies in our sample respectively, the carryover threat is only modeled in about 3% of the observed cases. The lack of adherence to the analysis guidelines threatens the validity of the conclusions drawn from crossover design experiments.

Keywords: Experimentation, Design, Crossover, Literature Survey

1 Introduction

Experimentation is an important method to infer causal relationships in any empirical research discipline [47]. The design of an experiment, i.e., how levels of the main factor are assigned to subjects, has a critical impact on the validity of its conclusions. One possible design, the *crossover* design, has the advantage of increasing the number of data points obtained for the same amount of subjects involved in an experiment but is often critiqued for introducing threats to validity due to its difficult analysis [49]. To mitigate these threats, Vegas et al. [50] provided a thorough explanation and guidelines for the analysis of crossover-design experiments.

In this paper, we reflect on the impact of the guidelines by Vegas et al. [50] by answering the following research question (RQ): How do SE experiments that utilize a crossover design based on guidelines by Vegas et al. analyze their data? To this end, we conduct a forward snowballing of primary studies citing the guidelines [50] and assess how they deal with the threats to validity for which this experimental design is often critiqued [49].

Our contribution is two-fold:

1. We reproduce and archive the data analysis from the original guidelines [50] which were previously unavailable.

2. We show gaps in SE literature on analyzing data from crossover design experiments.

3. We reflect on the impact of the original guidelines [50] on the landscape of SE experimentation since its publication.

The rest of the paper is structured as follows: Section 2 explains different experimental designs and reviews literature studies with a similar purpose. Section 3 reports the applied method, Section 4 presents the results, and Section 5 discusses their implications. Section 6 acknowledges threats to validity and outlines future work, before we conclude the paper in Section 7.

Data Availability Statement

All data, protocols, material, figures, and scripts generated and used during this study are publicly available [252].

2 Background and Related Work

During an experiment, researchers apply one or more levels of a main factor (i.e., treatments) to experimental subjects and observe one or more dependent variables

assumed to be impacted by the factor. Observing a significant difference in the distribution of a dependent variable for different treatments allows the inference that this factor has a causal relationship with the independent variable. These significant differences are typically determined by selecting and applying an appropriate null-hypothesis significance test (NHST) [47], like the T-Test, Mann-Whitney U test, or ANOVA. Experimentation has been used in software engineering (SE) to determine the effect of different tools [253], methods [254], but also demographic factors [255] on SE tasks. Section 2.1 presents the concept and challenges of experimental design. Section 2.2 provides an overview of previous studies with a similar purpose to ours.

2.1 Experimental Design

An important decision when designing an experiment is whether each subject is administered only one or multiple treatments. The former is referred to as an independent measure (or between-subject) design, where subjects are split into groups and each group applies only one treatment, the latter is referred to as a repeated measures (or within-subject) design. The benefit of a repeated measures design is two-fold. Firstly, the same amount of experimental subjects produce more data points than in an independent measure design. Secondly, comparing relative rather than absolute values of each response variable deals with subject variability. On the other hand, such a design introduces new threats to validity. Vegas et al. identified four types of threats to validity in their guidelines [50]:

1. *Maturation/exhaustion:* Participants may perform better or worse in subsequent observations due to a learning or exhaustion effect.

2. *Optimal sequence:* Some sequences in which the treatments are administered may be favorable over others.

3. *Subject variability:* The tasks performed in SE experiments are often strongly influenced by differences between human subjects [256] that are difficult to quantify.

4. *Carryover*: The effect of an administered treatment might carry over to a subsequent period.

The crossover design, a special form of the repeated-measures design where participants receive the treatments in different sequences [50], counterbalances the threats by dispersing their impact on the response variable evenly. Still, the threats affect the observations of the response variable but are largely ignored in most SE papers [50]. In medical research, the carryover effect is often addressed via a washout period [257], i.e., a period between the experimental periods long enough for the medical compound to exit the subject's system. In SE research, a washout period is

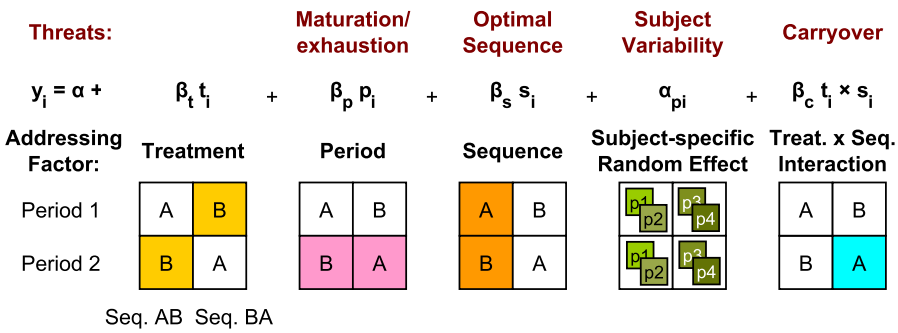


Figure 1: Relevant Factors Influencing the Response Variable in a Crossover-Design Experiment

often not feasible since it would require participants to unlearn techniques or tools they were subjected to [49]. Still, many SE papers even fail to acknowledge the carryover threat [50].

Consequently, Vegas et al. recommended abandoning simple NHSTs that only test whether different treatments change the distribution of the response variable and rather adopting a linear mixed model (LMM) to analyze the data from a crossover-design experiment. An LMM can involve additional factors with an influence on the response variable, which (1) isolates the true effect of the main factor in question and (2) provides insight into whether the threats to validity apply in a particular instance of an experiment.

Figure 1 visualizes an LMM (second row) that determines the effect of treatment t_i on a response variable y_i while also addressing the above-mentioned threats (first row) by modeling factors that address those threats (third row). The bottom row visualizes which parts of the AB/BA crossover design experiment these terms affect. For example, the treatment $\beta_t t_i$ affects period 2 in sequence 1 and period 1 in sequence 2 (marked yellow in Figure 1). For simplicity, we constrain the visualization to a crossover design experiment with one main factor containing two levels (A and B), which requires two periods and two sequences (AB and BA) and is commonly referred to as AB/BA crossover design.

2.2 State of Practice

Since the introduction of evidence-based SE by Kitchenham et al. [258], the field has been subject to reviews about the state of practice of various aspects of empirical research. For example, Kampenes et al. reviewed the state of practice of designing, conducting, and evaluating quasi-experiments [259]. They conclude terminological ambiguity and a common lack of awareness of specific biases affect the validity of drawn conclusions. Menzies et al. reviewed data analysis practices in empirical SE

research [245] and consolidated 12 “bad smells” commonly committed in publications. Hannay et al. surveyed SE experiments regarding the degree of realism of the employed experimental material and task [260] and detected a lack of awareness of the implications of realism in SE publications.

The aforementioned work by Vegas et al. also contained a secondary study of crossover-design experiments [50], which motivated the guidelines described in Section 2.1. Their review identified similar problems in the state of practice. Several publications misuse terminology, remain unclear in their design decision and apply an incorrect analysis to the data from crossover-design experiments. These results are supported by a similar study by Kitchenham et al. which focused on families of experiments [261].

The guidelines by Vegas et al. [50] have been extended in several regards. For example, Madeyski et al. investigated and demonstrated the calculation of effect sizes for crossover-design experiments [262]. Kitchenham et al. add the importance of determining and interpreting the correlation between participants’ response variable measures [263]. Cruz et al. propose the use of generalized estimating equations (GEEs) over LMMs to analyze crossover-design data [264, 265]. Still, none of these studies has reflected on the impact that the original guidelines [50] had on the SE literature, which we aim to contribute in this study.

3 Method

To answer our RQ, we conducted a literature survey in the following three steps. First, we selected an appropriate sample of primary studies (Section 3.1) to be considered in further steps. Then, we extracted relevant data from these primary studies (Section 3.2). Finally, we analyzed the extracted data (Section 3.3).

3.1 Study Inclusion

An answer to our RQ requires considering empirical SE publications analyzing crossover-design experiments. We limit this population of primary studies to publications that explicitly cite the guidelines by Vegas et al. [50] for two reasons:

1. By explicitly evaluating papers citing the guidelines we provide a reflection on the guidelines’ impact.

2. We assume that the proportion of papers correctly analyzing crossover-design experiments is larger in the subset that cites the guidelines than in the subset that does not. We assume that the former provides an upper bound for the proportion of correctly analyzed crossover-design experiments.

Table 1: Inclusion (In) and exclusion (Ex) criteria

ID	Criterion
In1	The article is related to software engineering.
In2	The article contains an empirical study as a main contribution.
In3	The empirical study is an experiment comparing at least two levels (e.g., baseline and treatment) of a main factor.
In4	The experiment utilizes a crossover design in which all subjects are administered all levels of the main factor.
In5	The subjects of the experiment are humans.
Ex1	The article is not available through the university's access program.
Ex2	The article is not written in English.
Ex3	The article is extended by or a duplicate of an already included article.
Ex4	The article is not peer-reviewed (e.g., a thesis or blog post).

We gathered studies by selecting all entries from Google Scholar citing the guidelines [50]. The obtained sample consisted of 136 entries on the 1st of March 2024. We filtered this sample using the inclusion and exclusion criteria in Table 1. We considered a study eligible if it meets all inclusion and none of the exclusion criteria.

Inclusion criteria In1-In4 ensure that the study fits our research goal. In5 further limits eligible studies to those where the specific benefit of crossover-design experiments (controlling subject variability) is relevant. Ex1 and Ex2 exclude inaccessible studies, Ex3 removes duplicates, and Ex4 serves as a quality assurance measure.

The first and second authors of this study conducted the inclusion phase. The 136 articles were distributed among the two authors based on their availability (90 for the first author, 46 for the second). For each paper, the assigned author read the abstract, introduction, and method section to decide the inclusion and exclusion criteria. Only criterion Ex3 (i.e., filtering out duplicates or extensions) was performed centrally by the first author by clustering the sample by author names and investigating similar candidates. During the inclusion phase, unclear decisions were flagged and later reviewed by the third author of this study. The third author acted as an arbiter and decided on the three unclear cases. The inclusion phase identified 48 primary studies ($48/136 = 35.3\%$) as eligible for the subsequent data extraction phase.

Before conducting the inclusion phase, we assessed the reliability of the criteria by randomly selecting a subset of 14 studies ($14/136 = 10.3\%$ of the sample) to be rated by both the first and second authors. The two authors reached a perfect agreement on all ratings, supporting the mutual understanding and reliability of the criteria before proceeding with the main inclusion phase.

Table 2: Data Extraction Attributes

Attribute	Description	Type
Subjects		
Subject number	Number of human participants	Count
Subject type	Type of participants	Enum
Analysis		
Analysis Method	Statistical method applied for inferential analysis of the treatment's effect	Enum
Test Type	(only if the analysis method is NHST) Type of statistical significance test	Enum
Threat Addressal	Way of dealing with the specific threats of validity of crossover-design experiments at analysis time	Enum
Washout	Whether a washout period was scheduled between experimental periods	Bool
Material		
Availability	Degree to which material (data set and analysis scripts) are available	Enum
Location	URL of the material if available	Text

3.2 Data Extraction

From each of the 48 eligible primary studies, we extract the attributes in Table 2 for every individual experiment reported in the study (as one study may conduct and report multiple experiments).

The attribute group **subjects** characterizes the number and types of participants involved in each experiment. The attribute group **analysis** contains the main variables of interest. The *analysis method* represents the type of statistical method applied for inference, for example, NHSTs, (generalized) linear models (GLMs), or (generalized) linear mixed models (GLMMs).[1] If a primary study reported analyzing the data from the experiment using an NHST, we additionally recorded the *test type* (e.g., paired or unpaired T-test, Mann-Whitney U test, etc.). The attribute *threat addressal* represents the main attribute of interest in the scope of this study. For each of the four types of threats to validity as mentioned in Section 2 (i.e., maturation/exhaustion, optimal sequence, subject variability, and carryover), we determined how the primary study addresses it on the categorical scale shown in Table 3.

Finally, the attributes from the **material** group record to what degree both the data obtained by the experiment and the script(s) used to perform the analysis are available. We recorded the *availability* attribute based on a previously established, categorical scale of research artifact availability [45] which includes levels like *archived*, *reachable*, and *unavailable*. If the material was available, we also recorded how to access it in the *location* attribute.

We summarized the extraction guidelines containing a definition, concise extraction rules, as well as examples in a shared document. To assess the reliability of these guidelines, the first and second authors performed an overlap of the extraction task prior to the main extraction phase. During this overlap, the two authors applied the extraction guidelines to the seven primary studies from the 14 that were already involved in the inclusion overlap and which were included according to our criteria.

[1] We do not distinguish between GLMs and LMs, nor between GLMMs and LMMs, as only their shared property of containing a random effect or not is relevant to our study.

Table 3: Types of Threat Addressal

Type	Description
Modeled	The authors address the threat to validity by modeling the factor in the analysis (e.g., as a parameter in a GLM or GLMM).
Isolated	The authors analyze the threat to validity in isolation, i.e., conduct a statistical test with the threat variable as the only independent variable.
Acknowledged	The authors do not address the threat in the analysis but acknowledge its (unaddressed) influence in the threats to validity section.
Neglected	The authors do not address the threat to validity in the analysis, but claim it is negligible due to the employed design.
Ignored	The authors neither address nor acknowledge the threat to validity.

We assessed the agreement of all attributes from the subjects, analysis, and material group except the *location* attribute as it bears no empirical value. We calculated Pearson's correlation coefficient (PCC) for the numerical attribute *subject number* and Bennett's S-score [142] between the two ratings of categorical attributes. None of the seven primary studies in our subset applied a washout period, which is why we did not calculate the inter-rater agreement for this variable. We selected Bennett's S-score over the more common Cohen's Kappa as the latter is known to be unreliable for uneven marginal distributions [141]. The two raters were in perfect agreement about the *subject number* attribute ($PCC = 1.0$) and achieved an average S-score of 85.8% over all categorical columns. After confirming the reliability of the extraction guidelines, the two first authors proceeded with applying them to the 48 primary studies originally assigned to them during the inclusion phase. The third author acted as an arbiter and clarified seven unclear instances in the data extraction.

3.3 Data Analysis

Upon completion of the data extraction phase, we generated descriptive statistics from the distribution of attribute values. We visualized categorical data using bar charts and numerical data using box plots. The main attribute of interest, the *threat addressal*, was represented as a heatmap where one axis listed the four threats to validity and the other axis the types of threat addressal.

4 Results

The 48 primary studies describe 67 experiments. Main factors of interest include, among others, testing paradigms, like test-driven development [266–268], the effect

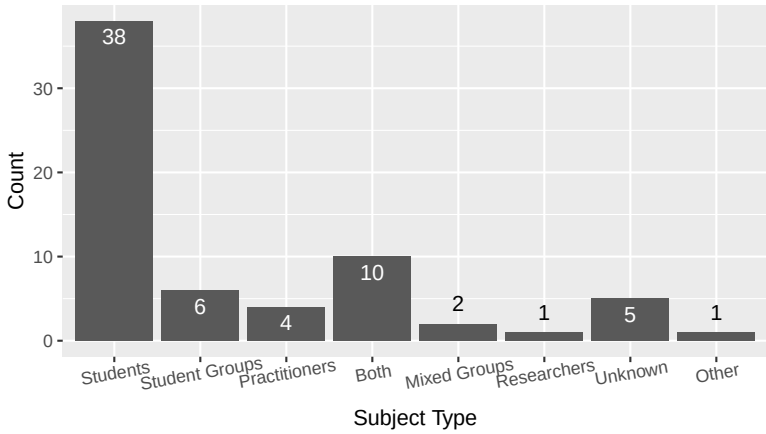


Figure 2: Types of subjects in the experiments

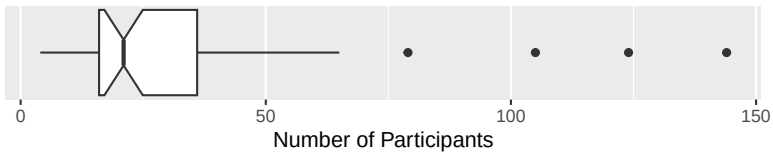


Figure 3: Number of subjects in the experiments

of API design rules [255], and noise during software development [269]. The following subsections report the obtained results grouped by the attribute groups in Table 2.

4.1 Subjects

Figure 2 visualizes the distribution of subject types among the experiments. The predominant type of participants are students, while practitioners are underrepresented. In five cases, the authors do not state the participant type at all. The case labeled as "other" sampled from app users, which were only partly students [270].

Figure 3 visualizes the distribution of subject count among the experiments (median of 21, mean of 31.2). Notable outliers are experiments with 144 [271], 124 [272], and 105 [255] participants. The former two involved students, the latter both students and practitioners. One study did not mention the number of participants involved in the experiment [254].

4.2 Analysis

Figure 4 visualizes the distribution of applied statistical methods. The applied methods are largely limited to NHSTs and LMMs. Exceptions (coded as "other") are papers that, for example, only compare the mean values of the response variable strat-

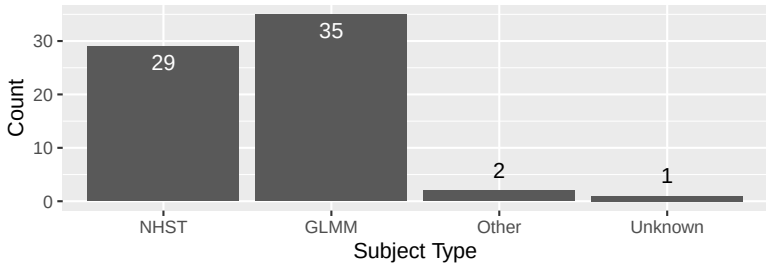


Figure 4: Applied statistical methods

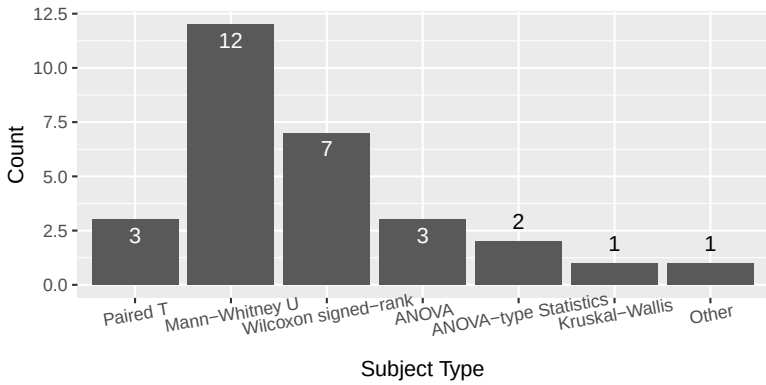


Figure 5: Applied NHSTs

ified by the treatments [273]. Figure 5 shows the distribution of test types applied to the experiments that analyzed their data using an NHST (n=29).

Figure 6 shows how authors address the threats to validity detailed by Vegas et al.[50] in their analysis, i.e., the figure visualizes the distribution of types of addressal per threat to validity. For example, the first, top-left cell indicates that for 26 experiments, the maturation/exhaustion threat was ignored. The visualization shows that the maturation/exhaustion and the optimal sequence threat are mostly either ignored completely or modeled via the period and sequence variable respectively. The subject variability and carryover threat are mostly either ignored or acknowledged. Rarely do authors analyze the threat type in isolation (4.5% of all cases). In total, 47.0% of all threats to validity ($(26 + 33 + 38 + 29)/(4 \times 67)$) were simply ignored by the primary studies in our sample. A subset of papers at least acknowledges these threats, but they either leave it at this acknowledgment (14.9% of threats are *acknowledged*) or claim that the threat is mitigated by design (9.7% of threats are *neglected*). The discouraged addressal in isolation only occurs rarely (4.5% of threats are *isolated*) while only 23.8% of the threats to validity are explicitly modeled. None of the papers in our sample contained an analysis that follows the guidelines completely, i.e., addressed all four threats to validity by modeling the respective factors. The

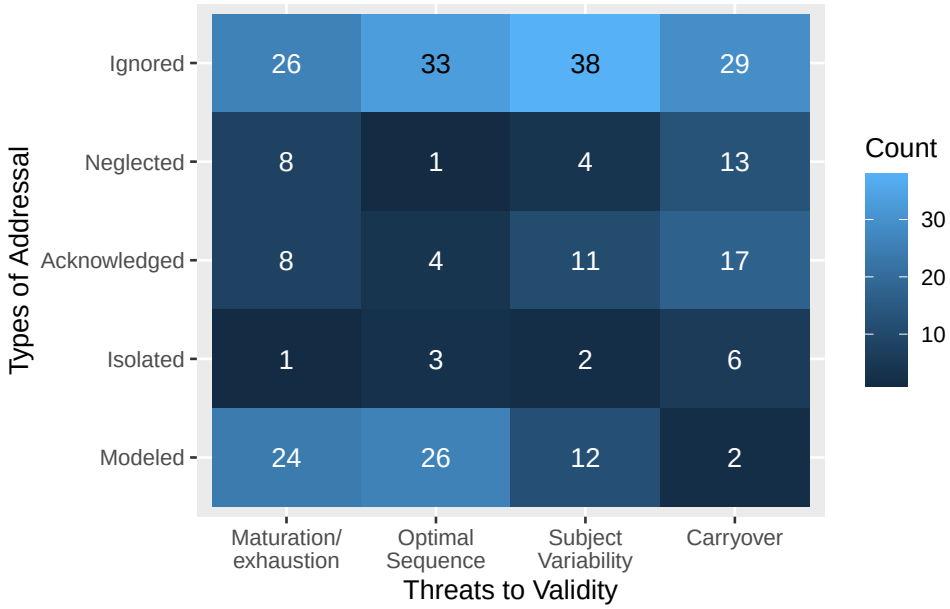


Figure 6: Addressal of the Threats to Validity

experiment where the analysis comes closest to the recommended guidelines was performed and reported by Bünder et al. [274], who *modeled* the period, sequence, and subject variability in their GLMM and *acknowledged* the carryover threat. However, this was still considered valid by the original guidelines, especially when the carryover threat is confounded with other treats [50].

Only three experiments include a washout period [266, 269, 275]. The duration of the washout periods varies between 30 minutes [269] and a day [266] or was not specified [275].

4.3 Material

Figures 7 and 8 visualize the availability of data sets and analysis script.[2] The majority of material (20 data sets and 30 analysis scripts are *unavailable*) were never available and a portion (4 data sets and 4 analysis scripts are *broken*) has become unavailable since their original publication. Among the remaining material, several data sets ($12/48 = 25\%$) and scripts ($12/48 = 22.9\%$) have been properly *archived* and, therefore, preserved for future use like reproduction.

[2]Note that these are distributions among the 48 publications, not among the 67 experiments, because, in our sample, we observed that any additional material was always associated with a publication, not an individual experiment.

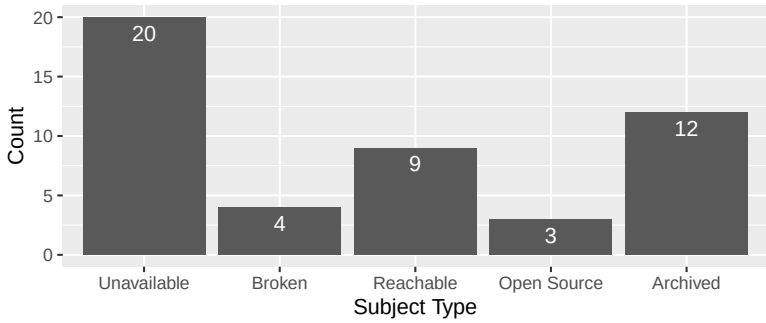


Figure 7: Availability of Data Sets

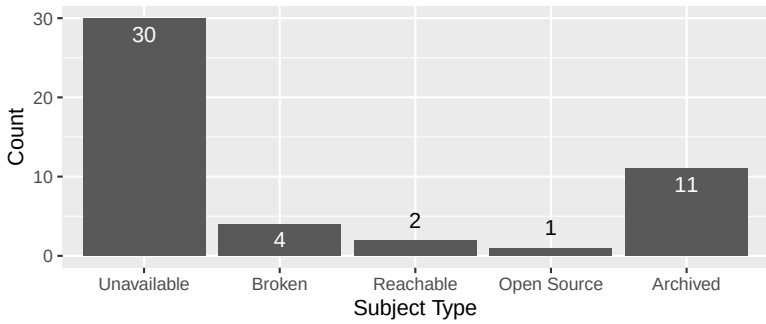


Figure 8: Availability of Analysis Scripts

5 Discussion

Despite clear guidelines [50], the majority of the subset of SE papers that indicate that they are aware of them through citation do not comply with them. This indicates that SE papers reporting the analysis of crossover-design experiments run the risk of drawing incorrect conclusions due to their incomplete analysis [49, 50]. The portion of experiments where a threat to validity was neglected shows that some authors assume that the crossover design mitigates this threat by default. However, the crossover design merely counterbalances, i.e., de-confounds the threats from the effect of the treatment, but the effect still applies to the response variable and needs to be modeled when analyzing the data.

The results imply that the guidelines [50] did not unfold the full effect that the authors hoped to achieve with their contribution. Many experiments published in SE literature claim to follow established guidelines but fail to do so. A possible reason for this is that the guidelines were not supplemented with the analysis scripts that could have provided practical guidance on how to implement them.

However, despite the remaining room for improvement, we observed positive effects of the guidelines. While the original literature survey reported that none of the 38 primary studies in its sample dealt with carryover at analysis time [50], our sample showed at least two studies (3%) that modeled the carryover effect and six

that analyzed it in isolation (9%). 17 (25.3%) at least acknowledged the carryover threat. Additionally, while the original literature survey observed only one primary study that explicitly defined its experimental periods, exactly half of our sample (24 papers) contained either a table [270, 274, 276] or figure [269, 277] visualizing the periods and sequences of their experimental design.

To improve future analyses of crossover-design experiments in SE, we recommend increasing the awareness of established guidelines [50]. Including these guidelines in textbooks [47] and standards [43] will aid authors in more rigorous analyses. Furthermore, we urge reviewers to put more emphasis on guideline adherence. This not only requires awareness of the existence of guidelines (i.e., checking that appropriate guidelines were cited) but also of their content (i.e., checking that the guidelines were properly followed) as our study results show.

6 Limitations and Future Work

6.1 Threats to Validity

Our study suffers from the following threats to validity [47]. Most prominently, our study is subject to a threat to external validity. While we aim to draw general conclusions about experimentation in SE literature, our sample is limited to SE literature that cites the investigated guidelines [50]. However, we argue that our sample is adequate for this paper for the following reasons. The guidelines [50] are—to the best of our knowledge—the only SE-specific guidelines for analyzing crossover-design experiments in SE. Hence, authors of our selected subset had access to the only guidelines explicitly advising them on how to properly analyze their experimental data. Therefore, we are confident that our sample represents an upper bound to extrapolate to the SE experimentation literature.

Additionally, our study inclusion and data extraction phases were subject to two threats to construct validity. Firstly, both phases involved subjective judgment of criteria and extraction guidelines. We mitigated this threat by quantifying the inter-rater agreement of both phases. Given the satisfactory inter-rater agreement, we are confident in the reliability of our results. Secondly, the categories of the *threat addressal* attribute were devised ad-hoc and not based on an existing taxonomy. We addressed this threat through thorough discussions among all three authors.

6.2 Future Work

In the scope of this study, we only surveyed how authors addressed the four threats to validity presented in Section 2.1 as per the original guidelines [50]. We did not consider additional threats to validity which crossover-design experiments are subject to. Additional threats we plan to investigate is the *material* threat, i.e., the influence of the used experimental material (e.g., code snippet), and the interaction effects be-

tween subjects and treatments, e.g.,, personal preference for specific treatments.

Furthermore, we plan to extend our sample by including SE studies presenting experiments with a crossover design that do not cite the guidelines by Vegas et al. [50]. By comparing this sample with our current one we aim to further characterize the impact of these guidelines.

Finally, we plan to reproduce the analysis of the surveyed experiments as far as possible. While we cannot reproduce the analysis of experiments where the raw data is not available, we plan to (1) reproduce the original analyses as described in the publication, and (2) reanalyze the data according to the data analysis guidelines [50]. Contrasting these analyses will produce more examples for an application of the guidelines, but also reveal studies that drew incorrect conclusions due to incorrect data analyses. This quantifies the effect of the guidelines in terms of the risk of drawing inappropriate conclusions and will serve as further motivation for adherence. Additionally, we plan to analyze and reproduce crossover-design experiments that do not cite the data analysis guidelines to compare them with our current sample and, therefore, further study the guidelines' impact.

7 Conclusion

In this reflection, we investigate the state of practice of analyzing crossover design experiments in SE. A sample of publications citing explicit guidelines [50] shows that the state of practice still contains several, significant gaps threatening the validity of the drawn conclusions. While the guidelines by Vegas et al. [50] supported a significant portion of authors to analyze their obtained data at least in parts, the general sample of papers shows room for improvement.

We hope that the overview of the state of practice of analyzing crossover design experiments in SE encourages authors to investigate this topic more thoroughly and that both our visualizations as well as the recovery of the data analysis script from the original guidelines [50] will support the latter in guiding authors towards a correct analysis. We encourage authors to abandon the overly simple NHST analysis [81] for more complex (G)LMMs, which enable them to adequately address threats to the validity of crossover-design experiments during analysis.

Paper VII

Applying Bayesian Data Analysis for Causal Inference about Requirements Quality: A Controlled Experiment

Abstract

It is commonly accepted that the quality of requirements specifications impacts subsequent software engineering activities. However, we still lack empirical evidence to support organizations in deciding whether their requirements are good enough or impede subsequent activities. We aim to contribute empirical evidence to the effect that requirements quality defects have on a software engineering activity that depends on this requirement. We conduct a controlled experiment in which 25 participants from industry and university generate domain models from four natural language requirements containing different quality defects. We evaluate the resulting models using both frequentist and Bayesian data analysis. Contrary to our expectations, our results show that the use of passive voice only has a minor impact on the resulting domain models. The use of ambiguous pronouns, however, shows a strong effect on various properties of the resulting domain models. Most notably, ambiguous pronouns lead to incorrect associations in domain models. Despite being equally advised against by literature and frequentist methods, the Bayesian data analysis shows that the two investigated quality defects have vastly different impacts on software engineering activities and, hence, deserve different levels of attention. Our employed method can be further utilized by researchers to improve reliable, detailed empirical evidence on requirements quality.

Keywords: Requirements Engineering, Requirements Quality, Experiment, Replication, Bayesian Data Analysis

1 Introduction

Software requirements specify the needs and constraints that stakeholders impose on a desired system. Software requirements specifications (SRS), the explicit manifestation of requirements as an artifact [24], serve as input for various subsequent software engineering (SE) activities, such as deriving a software architecture, implementing features, or generating test cases [23]. As a consequence, the quality of an SRS impacts the quality of *requirements-dependent* activities [1, 2, 40]. A quality defect in an SRS—for example, an ambiguous formulation—can cause differing interpretations and result in the design and implementation of a solution that does not meet the stakeholders' needs [18]. The inherent complexity of natural language (NL), which is most commonly used for specifying requirements [37], aggravates this challenge further. Since quality defects are understood to scale in cost for removal [3], organizations are interested in identifying and removing these defects as early as possible [6].

Within the requirements engineering (RE) research domain, the field of requirements quality research aims to meet this challenge [6]. Requirements quality research has already identified several attributes of requirements quality [6] (e.g., unambiguity, completeness, consistency) and proposes *quality factors*, i.e., requirements writing rules (e.g., the use of *passive voice* being associated with bad quality [34]) as well as tools that automatically detect alleged quality defects [130]. However, existing approaches fall short in at least three regards [40]: i) only a fraction of publications provide empirical evidence that would demonstrate the impact of quality defects [6], ii) the few empirical studies that do so largely ignore potentially confounding context factors [87, 135], and iii) the analyses conducted in existing publications do not go beyond binary insights (i.e., a quality factor *does* have an impact or it *does not*) [34, 71]. These gaps have impeded the adoption of requirements quality research in practice [37].

In this article, we aim to address the above-mentioned shortcomings by i) conducting a controlled experiment with 25 participants simulating a requirements- dependent activity (i.e., domain modeling) using four natural-language requirements as input. The experiment contributes empirical evidence on the impact of two commonly researched quality factors *passive voice* [34] and *ambiguous pronouns* [278]. The investigation of the impact of passive voice is a conceptual replication [63] of the only controlled experiment studying the impact of passive voice on domain modeling [34] known to us. Therefore, our experiment also strengthens the robustness of their conclusions by providing diagnostic evidence [85]. Further, we ii) collect data about relevant context factors such as experience in software engineering (SE) and RE, domain knowledge, and task experience, and integrate these data in our data analysis. Finally, we iii) contrast the state-of-the-art frequentist data analysis (FDA) with Bayesian data analysis (BDA), which entails both a causal framework and Bayesian modeling for statistical causal inference [62]. The latter has recently been popular-

ized in SE research [48] since it generates more nuanced empirical insights. Our study is categorized as a laboratory experiment in a contrived setting [279], isolating the effect of the selected quality factors of interest. The causal inference of their impact contributes to our long-term goal of providing an empirically grounded understanding of the impact of requirements quality. This will support organizations in assessing their requirements and detecting relevant quality defects early.

This paper makes the following contributions:

1. a controlled experiment investigating the impact of requirements quality;
2. a conceptual replication of the only controlled experiment investigating the impact of passive voice [34];
3. the application of BDA to requirements quality research, which is among the first of its kind in RE; and
4. an archived replication package containing all supplementary material, including protocols and guidelines for data collection and extraction, the raw data, analysis scripts, figures, and results [280].

The remainder of this manuscript is organized as follows. Section 2 introduces relevant related work. We present our research method in Section 3 and the results in Section 4. We discuss these results in Section 5 before concluding our manuscript in Section 6.

2 Background

Section 2.1 introduces the research domain of this work by summarizing existing research on requirements quality. Section 2.2 motivates BDA—the statistical tool employed in this work—by explaining its adoption in SE research.

2.1 Requirements Quality

Section 2.1.1 introduces the general area of requirements quality research and Section 2.1.2 presents two research directions within. Section 2.1.3 summarizes the three major shortcomings that currently challenge requirements quality research.

2.1.1 Requirements Quality Research

It is commonly accepted that the quality of requirements specifications impacts subsequent SE activities, which depend on these specifications [1, 40]. Quality defects in requirements specifications may, therefore, ultimately cause budget overrun [281] or even project failure [18]. Two further factors aggravate the effect. Firstly, natural

language (NL), which is inherently ambiguous and, hence, prone to quality defects, remains the most commonly used syntax to specify requirements [21, 25]. Secondly, the cost of removing quality defects scales the longer they remain undetected [3]. For example, clarifying an ambiguous requirements specification takes comparatively less effort than re-implementing a faulty implementation based on the ambiguous specification. However, it requires detecting the ambiguity and predicting that the ambiguity potentially causes the implementation to become faulty before it happens. These circumstances necessitate managing the quality of requirements specifications to detect and remove requirements quality defects preemptively.

Requirements quality research seeks answers to this need [6]. One main driver of this research is *requirements quality factors* [41], i.e., metrics that can be evaluated on NL requirements specifications to determine quality defects. For example, the *voice* of an NL sentence (active or passive) is considered a quality factor, as the use of *passive voice* is associated with bad requirements quality due to potential omission of information [34]. Automatic detection techniques using natural language processing (NLP) [51] can automatically evaluate quality factors to detect defects in NL requirements specifications [130].

2.1.2 Existing Research on Passive Voice and Ambiguous Pronouns

We present two examples of commonly researched requirements quality factors in the following sections.

Passive Voice One commonly researched requirements quality factor is using *passive voice* in natural language requirements specifications. A sentence in passive voice elevates the semantic patient rather than the semantic agent of the main verb to the grammatical subject [282]. For example, in the passive voice sentence "Web-based displays of the most current ASPERA-3 data shall **be provided** for public view.", the patient of the *providing* process—the "web-based displays"—becomes the grammatical subject of the sentence. Even though passive voice sentences may still contain the semantic agent (e.g., "Web-based displays of the most current ASPERA-3 data shall be provided for public view *by a front-end*."), writers often omit it intentionally or unintentionally [172]. Figure 1 visualizes the omission of the semantic agent in this exemplary requirement specification.

Omitting the semantic agent of a sentence in a passive voice formulation obscures critical information in a requirements specification. Hence, requirements quality guidelines advise against using passive voice [33]. However, while several guidelines advise against the use of passive voice based on the theoretical argument of information omission presented above [33, 158, 169, 171, 283], only two papers investigate whether passive voice has an actual impact on requirements quality: Krisch et al. let domain experts rate active and passive voice requirements as either problematic or unproblematic. They concluded that most passive voice requirements were

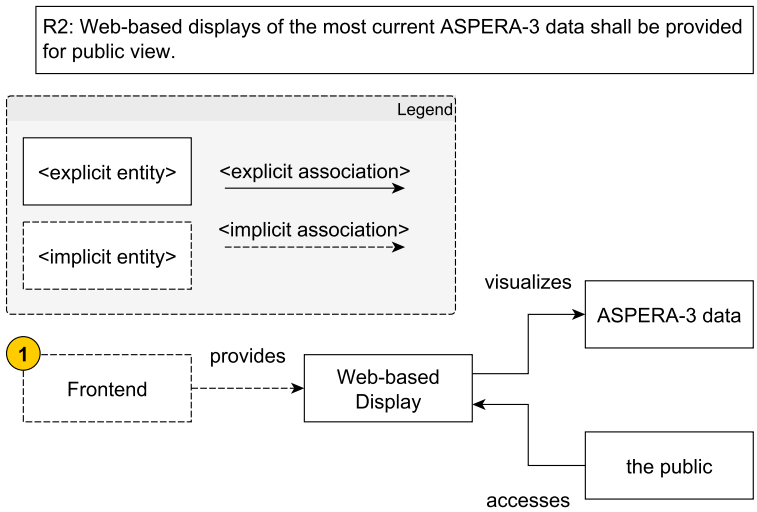


Figure 1: Formalization of a requirements specification R2 using passive voice

unproblematic as the surrounding context information compensated the omission of the semantic agent of the sentence [172]. Femmer et al. conducted an empirical investigation of the impact of the use of passive voice in requirements specification on the domain modeling activity in a controlled experiment. They concluded that passive voice only causes missing relationships from the domain model, but not missing actors or entities as initially assumed [34]. The limited evidence for the harmfulness of using passive voice in requirements specifications [34, 172] stands in stark contrast to the amount of tools and approaches proposed to automatically detect quality defects by identifying the use of passive voice [1, 90, 130, 133, 158, 169, 171, 284–287].

Ambiguous Pronouns The inherent ambiguity of natural language [288] poses several challenges for requirements specifications using natural language [22, 242]. One commonly researched requirements quality factor related to ambiguity is the use of *ambiguous pronouns*, which is a type of *referential ambiguity* [32]. An ambiguous pronoun exhibits *anaphoric ambiguity*, that "occurs when a pronoun can plausibly refer to different entities and thus be interpreted differently by different readers" [278]. For example, in the requirements specification "The *data processing unit* stores *telemetric data* for *scientific evaluation*; therefore, **it** needs to comply with the FAIR principles of data storage.", the pronoun *it* could syntactically refer to the "data processing unit", the "telemetric data", or the "scientific evaluation." Figure 2 visualizes how a reader can resolve the reference.

To avoid deviating interpretations of a requirements specification, established requirements quality guidelines advise against the use of ambiguous pronouns [33]

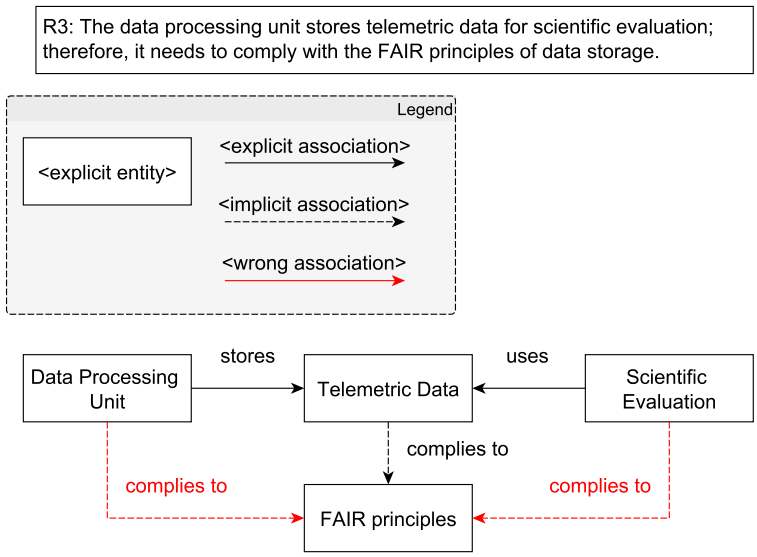


Figure 2: Formalization of a requirements specification R3 using an ambiguous pronoun

at the expense of conciseness. However, the number of publications proposing tools and algorithms to automatically identify and resolve ambiguous pronouns [36, 91, 144, 155, 278, 289–292] significantly outweighs the singular publication that actually has empirically investigated the effect of ambiguous pronouns. Kamsties et al. investigated the effects of formalizing requirements, which included evaluating the propagation of ambiguous pronouns from NL into more formal specifications [293]. Their experiment involving students revealed that 20-37% of all ambiguous pronouns were incorrectly resolved while formalizing NL requirements specifications. While Kamsties et al. concluded that requirements formalization does not sufficiently resolve ambiguities, these results also support the assumption that ambiguous pronouns propagate into subsequent artifacts depending on the requirements specifications. On the contrary, the scarce empirical work on the effect of ambiguity in general (not specifically ambiguous pronouns) agrees that ambiguity has a negligible effect on downstream software engineering activities [281, 294]. Other than these empirical contributions, the aforementioned publications proposing solutions rather than investigating the relevance of the problem refer to deontic guidelines [33, 295], anecdotal evidence about ambiguity in general [223, 296–298], or—in very rare cases—cognitive science theory [288].

2.1.3 *Shortcomings in Requirements Quality Research*

The previous examples highlight at least three shortcomings from which requirements quality research suffers.

Lack of empirical evidence First, the relevance of quality factors like passive voice or ambiguous pronouns is rarely determined empirically [40]. Scientific contributions proposing solutions (i.e., detecting or removing quality defects) outweigh those investigating the actual extent of the assumed problem. Without knowledge about this extent, it remains unclear whether a proposed solution addresses a problem that is actually relevant to practice.

Previous systematic research has come to the same conclusion. For example, in a previous systematic study, we determined that the effect of quality defects is determined empirically in only 18% of the publications included in our sample [40]. Bano et al. found only two publications within their sample of 28 studies that empirically investigated the importance of ambiguity detection [242]. Montgomery et al. systematically investigated empirical research on requirements quality research and also concluded that most studies focus on improving requirements quality (i.e., detecting and removing defects) rather than defining or evaluating it (i.e., understanding the actual effect) [6]. Instead, most requirements quality publications draw on anecdotal evidence and unproven hypotheses [40]. This lack of empirical evidence undermines the trust in requirements quality research and hinders its adoption in practice [8, 37, 173].

Lack of context Second, existing research mostly ignores the influence of context factors on the effect of quality defects [40]. Context factors encompass all human and organizational factors influencing the downstream SE activities involving requirements specification [88]. For example, the *domain experience* of a stakeholder or the *process model* used during development may mediate the effect of ambiguity in requirements specifications [281].

Requirements quality research has acknowledged the relevance of context factors to requirements quality [87, 135]. Recent propositions have advocated for a shift away from the unrealistic goal of developing a one-size-fits-all solution to requirements quality and, instead, moving towards more context-sensitive research [18, 152]. However, this initiative has shown little effect in requirements quality research so far [40].

Lack of detailed projections Third, the few empirical contributions to requirements quality research limit their insights to categorical projections, i.e., the evaluation of a quality factor on a requirements specification (e.g., *using passive voice* or *not using passive voice*) are projected on a categorical scale (e.g., *good quality* or *bad quality*). Most commonly, the categorical output space consists of two [71] (*impact* or *no impact*) or three [2] (*positive impact*, *no impact*, or *negative impact*) categories. This simplification inhibits a nuanced comparison of different quality factors. On an absolute scale, a quality factor having an impact does not automatically entail that this impact is significant and warrants resources for detection and mitigation. On

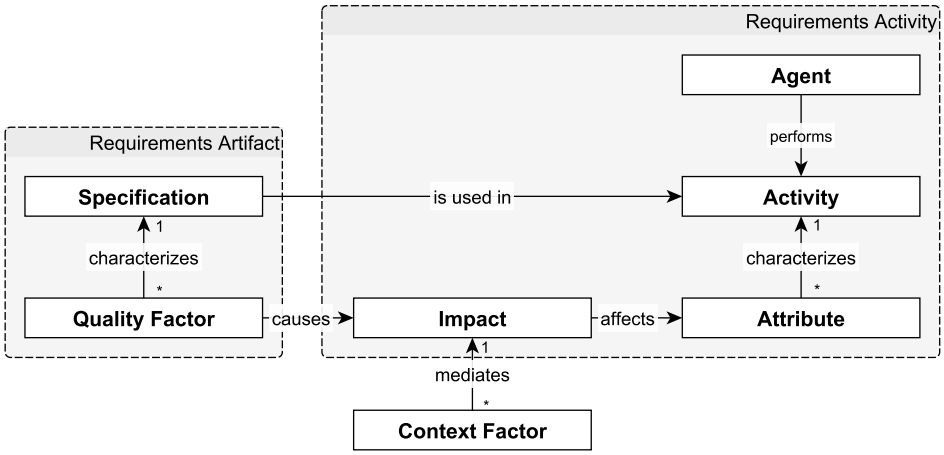


Figure 3: Reduced version of the activity-based Requirements Quality Theory [40]

a relative scale, two quality factors that have an impact are impossible to compare to allocate resources towards the more significant one. Consequently, even empirical contributions to the field of requirements quality lack sophisticated insights that would support organizations in determining and dealing with relevant quality factors to control during the RE phase.

Requirements Quality Research Gaps

Requirements quality research suffers from (1) a lack of empirical evidence about the relevance of quality factors, (2) a lack of context-sensitivity, and (3) evaluations of impact that are more fine-grained than categorical.

2.1.4 Requirements Quality Theory

Based on the identification of the above-mentioned shortcomings, we have developed a requirements quality theory in previous research [40]. This theory frames requirements quality as the impact that properties of requirements specifications (called the *quality factors*) in combination with context factors have on the properties (called *attributes*) of activities that use these specifications as input. Figure 3 visualizes the requirements quality theory.

The requirements quality theory facilitates overcoming the aforementioned shortcomings. Because the RQT makes the quality of a requirements specification dependent on its impact on subsequent activities, it demands empirical evidence about this impact before claiming that a quality factor reflects actual requirements quality. The inclusion of context factors in the definition of requirements quality mandates context-sensitivity. The abstraction of the impact concept allows for more advanced

relationships between specifications and impacted activities than just the categorical type.

However, while the requirements quality research draws on mature software quality research [71, 72], it has not been actively used yet. Even the predecessor of the theory [2] was explicitly ignored in follow-up research by its authors due to the complexity of its implementation [130]. The work presented in this manuscript constitutes the first application of the theory known to the authors.

2.2 Bayesian data analysis in software engineering

In recent years, SE research has adopted Bayesian data analysis (BDA) for statistical causal inference. BDA signifies a departure from frequentist methods like null-hypothesis significance testing (NHST), the previous state-of-the-art in terms of inferential statistics in SE research. NHST determines whether there is a "statistically significant" difference between two or more distributions. Observations of a dependent variable are stratified by an independent variable to obtain a binary answer of whether or not different values of the independent variable correlate with different distributions of the dependent variable.

Opposed to that, BDA encourages the use of causal frameworks [62]. These frameworks make causal assumptions explicit [248] and allow reasoning about causally relevant variables [76, 299]. Furthermore, BDA abstains from reducing complex variable distributions to binary inference [62]. Instead, dependent variables are expressed as a probability distribution, which preserves the natural uncertainty with which any variable is determined. Similarly, the impact of any independent variable on the dependent variable is expressed in terms of a probability distribution. Using Bayes' Theorem, these assigned prior probability distributions are updated with observed data to obtain a posterior probability distribution [48]. Given the observed data, these posterior probability distributions model the most likely impact of variable values. BDA methods are becoming widely adopted also due to the modern computational power enabling Markov Chain Monte Carlo (MCMC) randomized algorithms [250], tools like `Stan` [300], and libraries like `rethinking` [62] and `brms` [228].

While BDA is associated with a much steeper learning curve than frequentist methods, it offers several advantages.

1. BDA is not based on the unsound probabilistic extension of the *modus tollens* like frequentist hypothesis testing. The modus tollens ($P \rightarrow Q, \neg Q \therefore \neg P$, or *if P implies Q and Q is false, then P must also be false*) applies to propositional, Boolean logic, but not when inferring from probabilities [48].

2. BDA provides more complex insights than point-wise comparisons. Although BDA lacks out-of-the-box statistical methods like frequentists' t-tests that are

simple to apply, its results reflect the uncertainty of the data, the influence of context, and they can be interpreted more intuitively.

3. The causal framework entailed by BDA makes causal assumptions explicit. The Bayesian workflow [227] makes any hypothesis of causal relations explicit. Analyses become more transparent, and competing causal assumptions are easier to assess.

Furia et al. [48], and Torkar et al. [78] advocate for the adoption of BDA in software engineering research by discussing its advantages over the frequentist counterpart and mitigating its steep learning curve with extensive demonstrations [226]. SE researchers have begun to apply BDA in various evaluations. Previous studies have used BDA to model bug-fixing time in open source software projects [301], to confirm the broken window theory in SE [302], to investigate gender differences in personality traits of software engineers [303], and to understand data-driven decision making practices [304]. In the area of requirements engineering, BDA has been used to evaluate the effect of obsolete requirements on software estimation [305] and to compare requirements prioritization criteria [306].

3 Method

We conducted a controlled experiment that investigates the impact of requirements quality on a software engineering activity. Our goal is both to (1) contribute empirical evidence to the effect of quality defects and (2) compare the inferential capabilities of frequentist (FDA) with Bayesian (BDA) statistics. Part of our experiment contributes a conceptual replication [63] of the study conducted and reported by Femmer et al. [34] and re-analyzed by us [81], as a subset of our hypotheses overlaps with theirs and our study contributes diagnostic evidence for their claims [85]. Therefore, we report the design of the experiment with emphasis on the replication following the guidelines by Carver [307].

3.1 Goals

We formulate our goal using the goal-question metric approach [47]. We aim to *characterize* the impact of passive sentences and sentences using ambiguous pronouns in requirements on domain modeling *with respect to* the quality of the created domain model artifacts *from the point of view of* software engineers *in the context of* an analysis of requirements from an industrial project. In this definition, *software engineer* includes all roles that work with requirements specifications, including software developers, requirements engineers, business analysts, managers, and more. We derive the following research questions from our goal:

- RQ1: Do quality defects in NL requirements specifications harm the domain modeling activity?
  - RQ1.1: Does the use of *passive voice* in NL requirements specifications harm the *duration*, *completeness*, *conciseness*, and *correctness* of the domain modeling activity?
  - RQ1.2: Does the use of *ambiguous pronouns* in NL requirements specifications harm the *duration*, *completeness*, *conciseness*, and *correctness* of the domain modeling activity?
  - RQ1.3: Does the combined use of *passive voice* and *ambiguous pronouns* in NL requirements specifications harm the *duration*, *completeness*, *conciseness*, and *correctness* of the domain modeling activity?
- RQ2: Do context factors influence the domain modeling activity?
  - RQ2.1: Do context factors harm the domain modeling activity?
  - RQ2.2: Do context factors mediate the impact of quality defects on the domain modeling activity?

RQ1 is dedicated to the main relationship of interest between quality defects and an affected activity. RQ1.1 aligns with the research question driving the original study [34], which makes this part of our study a conceptual replication. RQ1.2 extends the scope of the investigation of quality factors with ambiguous pronouns. RQ1.3 investigates the interaction between the two quality factors. RQ2 adds a context-sensitive perspective to the relationship. RQ2.1 focuses on the direct effect that context factors have on the affected activity. RQ2.2 investigates whether context factors mediate the effect of quality defects on the activity.

3.2 Original Experiment

The original study [34] addresses the research question "Is the use of passive sentences in requirements harmful for domain modeling?" The authors involved 15 university students from different study programs (2 B.Sc., 8 M.Sc., 4 Ph.D., one unknown) in a controlled randomized experiment with a parallel design [47]. Each participant was assigned to one of two groups and received seven requirements that were formulated either using active or passive voice. The experimental task was to derive a domain model from each requirement that contains all relevant actors, domain objects, and associations between them. The study material and results are available online.[1]

For the dependent variable, the authors calculated the *number of missing domain model elements* (i.e., actors, objects, and associations). Although the authors

[1] `https://doi.org/10.5281/zenodo.7499290`

also recorded context variables, such as a categorical assessment of general knowledge in SE and RE, these were not used in the analysis. The analysis followed a frequentist approach performing a null-hypothesis significance test for each of the three domain model elements to determine whether a statistically significant difference between the experimental groups exists. The study shows a statistically significant difference in the number of identified associations but not in the number of actors or objects. The authors conclude that the commonly assumed impact of passive voice on missing domain model actors is actually negligible, but passive voice impedes the understanding of the relationships between entities in the requirements specification. However, a re-analysis of their data rectified some causal assumptions in the analysis and revealed that the effect of passive voice on the number of missing associations is much smaller than originally claimed [81]. We will consider the original study [34] together with the re-analysis of its data [81].

3.3 Reanalysis

The original study by Femmer et al. analyzed its data under simplified assumptions. Among these is the assumption that the three dependent variables (number of missing actors, objects, and associations) only depend on the main factor (use of active or passive voice). We challenged this assumption in a re-analysis of the original data [81] for the following reasons:

1. In a small-scale experiment employing a parallel design, there is no measure to control subject variability [50], such that context factors like experience or skill might affect the dependent variables.

2. Missing an actor or object in the domain model (i.e., a node) necessarily causes an association to be missed (i.e., an edge that would have connected these nodes).

Figure 4a visualizes the causal assumptions of the original experiment [34] as a directed acyclic graph [248] (the syntax of which is further explained in Section 3.4.10) and Figure 4b shows the revision in scope of the reanalysis [81]. The revision includes (1) two context factors that were already recorded but not used in the original experiment, and (2) two causal relations between the response variables.

We performed a re-analysis, i.e., an independent analysis of the same data using a different statistical model [83], which is sometimes referred to as a test of robustness [85]. During this re-analysis, we replaced the NHSTs with regression models that include context factors and the affecting response variables in the case of missing associations.

The results of this re-analysis agree with the original study in that the effect of passive voice on the number of missing actors and objects is negligible. However,

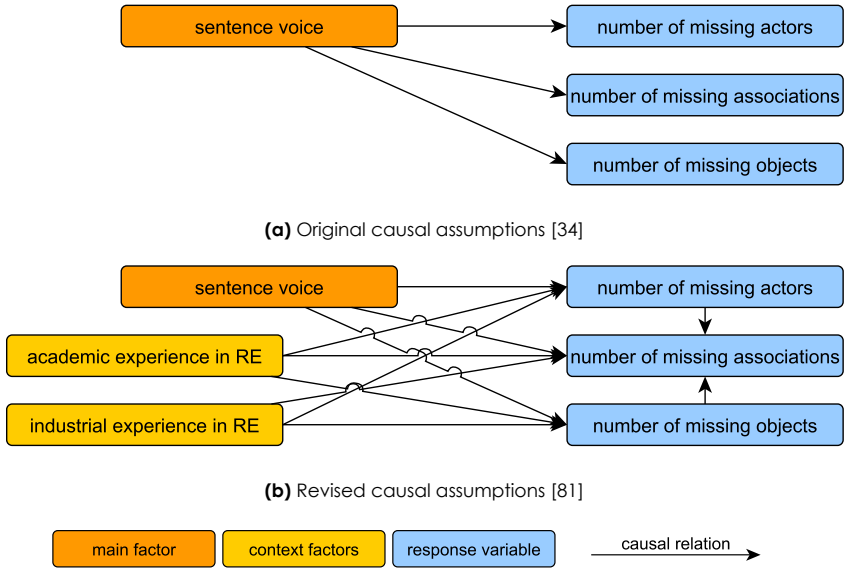


Figure 4: Causal assumptions about the impact of passive voice

the re-analysis disagrees with the original study regarding the effect of passive voice on the number of missing associations. The re-analysis determined that passive voice slightly increases the number of missing associations ($\beta_{pv} = 0.7$). Still, the confidence interval of this effect ($CI_{pv} = (-0.56, 1.90)$) intersects 0 and is, therefore, not significant. On the other hand, the effect of missing objects on missing associations was significant ($\beta_{act} = 1.12$, $CI_{act} = (0.32, 1.96)$). The re-analysis did not find a significant effect of the available context variables on the response variables. Our re-analysis concludes that the effect of passive voice on the domain modeling activity is less significant than originally assumed [81].

3.4 Our Experiment

The reanalysis [81] of the study by Femmer et al. [34] did improve the conclusion validity of the results but failed to address other shortcomings. For example, the subject variability still threatened the internal validity of the results due to the parallel design of the experiment [50], and the context factors were limited to those recorded during the original study. Hence, we used their study as inspiration for our own presented in this paper and aimed to improve upon the research design. During the preparation of our study, we conferred with the authors of the original study and made the following changes to the original study.

- **Experimental design**: We employ a factorial crossover instead of a parallel design, which minimizes the risk of confounding (i.e., each participant acts as

their own control) while requiring a smaller sample [50].

- **Independent variables**: This study investigates—in addition to using passive voice—the impact of *ambiguous pronouns* in requirements specifications and their combined usage to extend the range of requirements quality defects.

- **Dependent variables**: We merged two types of elements in the domain model (the nodes of the model, i.e., *actors* and *objects*) into a single type *entity* because they represent the same concept in the domain model (nodes) [34] and the distribution in our experimental objects is heavily skewed (16/17 entities are objects). Furthermore, we increased the dependent variables by additionally evaluating the number of *superfluous entities*, the number of *wrong associations*, and the *duration* for creating the domain model.

- **Sampling strategy**: We sample from both students and practitioners of software engineering to more accurately represent the target population of software engineers. This change aims at increasing the external validity of our results [211].

- **Instrumentation**: The participants performed the experimental task online using a web-based application rather than offline using pen and paper. This allows for more flexibility in reaching industry participants [211].

- **Context factors**: To obtain a richer understanding of the impact of quality defects, we included seven additional context factors. This change made it necessary to extend the questionnaire used in the original study to collect demographic information from the participants.

- **Experimental object**: We sampled the objects from a data set of industrial requirements specifications [308] rather than from a requirements specification written in a student project [34] to increase the realism of the experimental task [100].

- **Analysis**: The crossover design produces paired data as opposed to the unpaired data of the original experiment, which changes the appropriate hypothesis test [50] (Mann-Whitney U test in the original study vs. Wilcoxon signed-rank test in this study). In addition, we extend the original FDA by performing a Bonferroni correction to deal with family-wise error rate when testing multiple hypotheses [243]. Furthermore, we additionally analyze the data using BDA.

Our experiment differs from the original experiment [34] in all elements [83]. However, because a subset of our hypotheses aligns with their hypotheses, part of our study counts as a conceptual replication [63] since it contributes diagnostic evidence

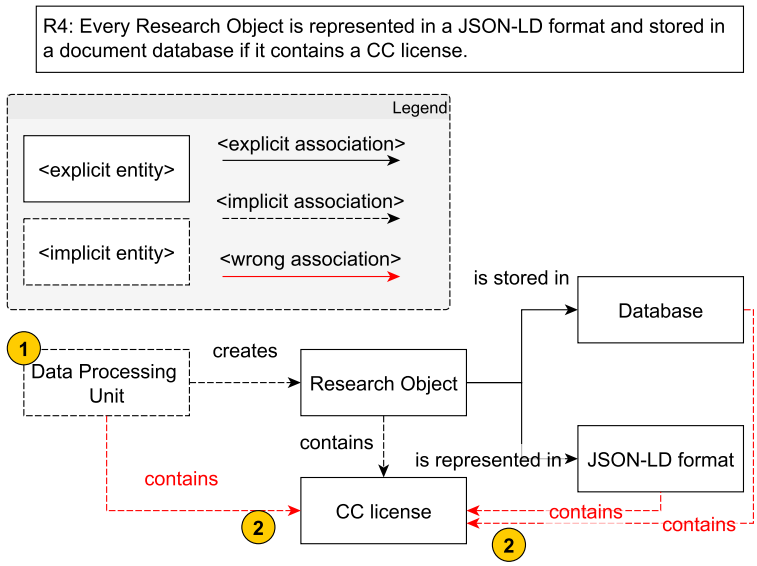


Figure 5: Domain modeling task example for requirement 4.

for the original claims [85]. In the rest of this subsection, we report the design of our experiment following the guidelines by Jedlitschka et al. [309].

3.4.1 Experimental Task

We simulate the use of a requirements specification by subjecting participants to a requirements processing activity, i.e., a common task representing the use of requirements [34]. In particular, we present four single-sentence, natural language requirements to the participants and request them to derive a domain model for each of them. Figure 5 visualizes the expected domain model for the requirement "Every research object is represented in a JSON-LD format and stored in a document database if it contains a CC license." which contains both of the two seeded quality defects. These defects result in the following challenges according to literature [33]:

1. The verb in **passive voice** omits an important entity of the requirement; i.e., that *the data processing unit* stores the research object in a document database (Label 1 in Figure 5).

2. The **ambiguous pronoun** "it" can syntactically be connected to several preceding noun phrases ("Every research object", "JSON-LD format", and "a document database" or the implicit "Data Processing Unit") by a reader but semantically only applies to the research object (Label 2 in Figure 5).

The goal of the experimental task is to derive a semantically correct domain model from the natural language requirement which includes identifying all entities

(including the implicit ones) and connecting these entities correctly (including those derived from syntactically vague associations).

The selection of dependent variables was driven by the activity-based requirements quality theory [2, 40]. Accordingly, requirements quality is measured by the effect that quality factors have on the relevant attributes of requirements-dependent activity. We selected the following dependent variables representing the relevant attributes of the domain-modeling activity with the given motivation:

- **Duration**: the longer the domain modeling task takes, the more expensive it is.
- **Number of missing entities**: entities missing from the domain model produce potential cost for failing to involve the respective actor or object.
- **Number of superfluous entities**: entities added to the domain model but not implied by the requirement unnecessarily constrain the solution space.
- **Number of missing associations**: associations missing from the domain model produce a potential cost for failing to identify a dependency between two entities.
- **Number of wrong associations**: associations connecting two entities that establish an unnecessary dependency between them while neglecting an actual dependency.

We characterize the domain modeling activity in terms of immediacy (duration), completeness (missing entities and associations), conciseness (superfluous entities), and correctness (wrong associations).

3.4.2 Hypotheses

The three independent variables $ind \in \{PV, AP, PVAP\}$ (passive voice, ambiguous pronoun, and the coexistence of passive voice and ambiguous pronoun) and the five dependent variables $dep \in \{D, E^-, E^+, A^-, A^\times\}$ (duration, missing entities, superfluous entities, missing associations, wrong associations) define our 15 null hypotheses as follows.

$$\sum_{ind \in \{PV, AP, PVAP\}} \sum_{dep \in \{D, E^-, E^+, A^-, A^\times\}} H_0^{ind \rightarrow dep}$$

"There is no difference in *{dep}* of the domain models based on requirements specifications containing no quality defect and requirements specifications containing *{ind}*."

To capture the context of the experiment, we collected factors based on related work [87, 88, 135], including the experience of a practitioner regarding software

and requirements engineering, but also in SE roles and in the modeling task itself. Additionally, we assume that the practitioners' education and domain knowledge influence the dependent variables. Table 1 summarizes the variables involved in this study.

The variables in Table 1 do not include a *participant type* that distinguishes students from practitioners. While including such a variable is common practice in SE research [255], meta-research on the eligibility of students as experiment participants suggests that the labels *student* or *practitioner* are merely a proxy for levels of more meaningful factors like domain knowledge and experience [310]. Additionally, the line between students and practitioners becomes increasingly blurred as students more commonly gather industrial experience prior to or during their studies [311]. Consequently, we subsume the participant type variable by the causally more meaningful and fine-grained variables of experience, education, domain knowledge, and formal modeling training. We compared two models—one using the binary distinction and one using the more fine-grained variables—and determined that the latter outperforms the former in predictive power, even though only slightly. This confirms to us that the variables we used are at least as expressive as the binary participant type variable.

3.4.3 Experimental Design

Our experimental design includes one factor (RQD) representing the alleged quality defect seeded in a requirements specification. This main factor contains four treatments: a control one (no defects) and three experimental ones (passive voice (PV), ambiguous pronoun (AP), and both (PVAP)).

Given our sampling strategy involving industry practitioners, which are difficult to recruit for controlled experiments [100], we anticipated a moderate sample size of participants. Consequently, we opted for a crossover design [50] instead of a parallel design, i.e., we apply every treatment to all subjects instead of distributing the subjects among the treatments. Previously, Kitchenham et al. advised against the use of crossover designs [49]. Mainly, the validity of crossover design experiments is challenged by the following confounding factors:

1. the period in which a treatment is applied to a subject, as certain periods may influence the dependent variables (e.g., participants may mature and perform increasingly better the more often they perform the experimental task subsequently);

2. the sequence in which the treatments are applied to a subject, as certain sequences may have a beneficial effect on a dependent variable (e.g., there might be an optimal sequence to apply the treatments in);

3. the effect from a previous treatment may *carry over* to the period when applying a subsequent treatment [50]; and

Table 1: Variables of the study (**ind**ependent, **con**text, and **dep**endent).

Variable	Name	Type	Description	Data type	Range
Requirements Quality Defect	RQD	ind	The use of a verb in passive voice, an ambiguous pronoun, or both	categorical	{none, PV, AP, PVAP}
Experience in SE	exp.se	con	Years of experience in software engineering	count	$\mathbb{N}$
Experience in RE	exp.re	con	Years of experience in requirements engineering	count	$\mathbb{N}$
Education	edu	con	Highest acquired degree	ordinal	{High School, B.Sc., M.Sc., Ph.D.}
Primary role	role	con	SE-related Role with the most years of professional experience	categorical	{requirements engineer, product owner, software architect, developer, tester, quality engineer, trainer, manager, other, none}
Task experience	exp.task	con	Experience with the task of domain modeling	ordinal	{never, rarely, from time to time, often}
Formal modeling training	formal	con	Formal training in domain modeling	categorical	{true,false}
Domain knowledge in {domain}	dom. {domain}	con	Knowledge of the domain $\in$ {telemetry, aeronautics, databases, open science}	ordinal	{1,2,3,4,5}
Tool experience	tool	con	Experience with using Google Docs for modeling	ordinal	{none, rarely, from time to time, often}
Duration	D	dep	Number of minutes it took the participant to complete the experimental task on one requirement	count	$\mathbb{N}$
Missing entities	E^-	dep	Number of relevant entities missing from the submitted domain model	count	$[0, E_{expected}]$
Superfluous entities	E^+	dep	Number of not relevant entities included in the submitted domain model	count	$\mathbb{N}$
Missing associations	A^-	dep	Number of associations missing from the submitted domain model	count	$[0, A_{expected}]$
Wrong associations	$A^\times$	dep	Number of associations where either the source or target of the edge is a different entity than implied by the requirement	count	$[0, A_{found}]$

Table 2: Variables of the study (experimental design factors)

Variable	Description	Data type	Range
Period	Index of the experimental period in which the data was obtained	ordinal	[1; 4]
Sequence	Order in which a participants received the treatments	categorical	{1234, 1243, ..., 4321}
Carryover effect	Interaction between the period and the treatment	categorical	$\{1 \times none, 1 \times PV, ..., 4 \times PVAP\}$
Subject variability	Index of a participant	categorical	{1, 2, ..., 25}

4. the subject variability, as software engineering tasks are highly dependent on the skill of involved individuals [256].

However, recent adoptions of best practices from other disciplines made this design applicable to SE research without compromising the validity of the results [50, 268]. The threats to validity can be mitigated during design and analysis [50] by (1) randomizing the order of treatments and (2) including the independent variables' period, sequence, the interaction between them (representing the carryover effect), and subject variability in the analysis. Consequently, we consider the *experimental design variables* listed in Table 2 in addition to the variables listed in Table 1.

When controlling the threats to validity, the crossover design provides two benefits. Firstly, it requires fewer participants, as an experiment with n_p participants and n_t treatments yields $n_p \times n_t$ observations instead of only n_p [47]. Secondly, it accounts for subject-specific variability, as the dependent variables can be measured in relation to the average response of each subject instead of the average response of each treatment group [49]. Therefore, each subject acts as its own control and mitigates within-subject variability.

Each experimental session contained four main periods in which we applied one treatment to the subject. We randomized the order of treatment application to disperse the confounding sequence and carryover effect [50, 312]. This resulted in 24 unique sequences of treatment application ($n_t! = 24$) and, consequently, 24 experimental groups. The experiment was single-blinded—i.e., the participants did not know the requirements' sequence, but the researchers did.

3.4.4 Objects

The experimental object consisted of four English, single-sentence NL requirements specifications R_1-R_4. An additional warm-up object (R_0) preceded the actual experimental objects, adding a fifth experimental period to each session. It was only used to familiarize the participants with the experimental task and tool and was not considered in the data analysis. The four experimental objects were manually seeded with defects corresponding to our four treatments: one requirement containing none of the two faults, one containing a verb in *passive voice*, one containing an *ambiguous pronoun*, and one containing both a verb in passive voice and an ambiguous pronoun. The requirements' mean length is 17.8 words (sd=4).

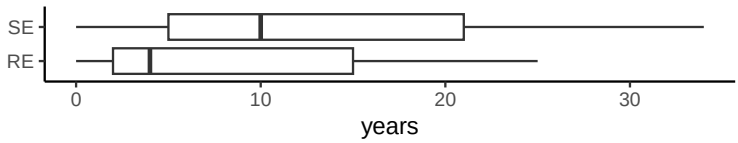


Figure 6: Distribution of SE and RE experience.

The first author derived the experimental objects from the requirements specification of the Mars Express ASPERA-3 Processing and Archiving Facility (APAF), a real-world specification from the PuRE data set [308]. From this requirements specification, the first author selected five single-sentence natural language requirements and modified them to ensure two defect-free requirements (one warm-up object R_0 and one for the defect-free baseline R_1) and three objects with the respective defects (R_2-R_4). The second author reviewed and adjusted the selected objects.

3.4.5 Subjects

The target population of interest consists of people involved in software engineering who work with requirements specifications. We used a non-probability sampling approach based on a mix of purposive and convenience sampling [211]. In particular, we wanted to select participants who are diverse in terms of the context variables as determined in Table 1, including their experience, education, and software engineering roles. We approached both students participating in RE courses at our respective institutions and practitioners in our collaborators' network to purposefully diversify the experience and education of our sample. For all other demographic context factors (e.g., SE roles) we had to rely on convenience sampling.

We approached 52 potential candidates (32 practitioners and 20 students), and 27 candidates (19 & 8) agreed to participate in the experiment (response rate of 52%). Two students did not show up to the agreed time slot. Our final sample includes 19 practitioners from six different companies and six students from two universities. Participation in the experiment was entirely voluntary and not compensated. The number of participants ($n_p = 25$) exceeded the number of experimental sequences ($n_t! = 24$) such that we had at least one subject per experimental group and evenly dispersed any confounding sequence or carryover effect [50].

Figure 6 shows the distribution of experiment subjects' experience in SE and RE in years. which is widespread in our sample. Among the 25 participants, the reported primary roles are developer (10), architect (6), requirements engineer (5), manager (1), and no prior professional role (3). Students who have not yet had a professional software engineering role constitute this last group. The distribution covers many SE-relevant roles but excludes others, such as testers or product owners.

Among the 25 participants, 5 reported a high school degree as their level of education, 8 a Bachelor's degree, and 12 a Master's degree. No participant reported a Ph.D. degree as their highest degree of education. Figure 7 visualizes the distribution

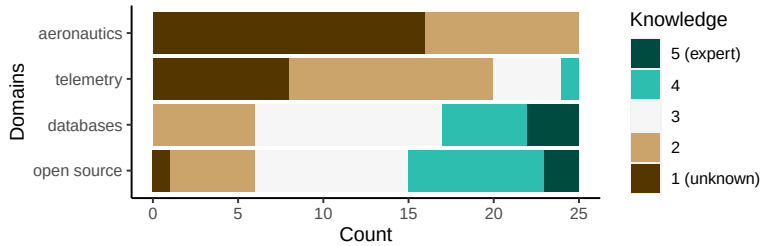


Figure 7: Distribution of knowledge in the four relevant domains.

of the participant's experience in the four domains[2] that contribute semantic knowledge to understanding the requirements: aeronautics, telemetry, databases, and open source. For the latter two, results are balanced across the different knowledge levels, while the former two confirm our assumption that all participants had a low-level knowledge of aeronautics and telemetry systems.

The experiment tool—the Google Draw plugin within the Google Document—was unknown to most (18 never used it, 6 rarely, and 1 from time to time). A total of 16 participants (64%) report having received a form of training in the modeling activity. The modeling experience (never: 1, rarely: 11, from time to time: 10, often: 3) resembles a normal distribution. We did not discard data from participants who reported having no modeling experience or formal training in modeling given that our experiment included both comprehensive instructions and a warm-up phase as described in Section 3.4.4.

Given the distribution of responses in these context variables, we disqualified the following predictors: aeronautics domain knowledge, telemetry domain knowledge, and experience with the experiment tool. These variables are not sufficiently distributed in our sample of study participants, i.e., several categories of these variables are underrepresented. Consequently, they are unable to effectively block the influence of that variable on the dependent variable [47].

3.4.6 *Instrumentation*

We used a Google Docs document[3] for the task and a Google Form questionnaire[4] to collect demographic information. The Google Docs document lent itself to the task due to its accessibility and its simple modeling tool with the embedded Google Drawings. The modeling tool represented the optimal trade-off between complexity—as neither previous knowledge nor additional software was necessary to conduct the experimental task—and suitability—as it contains all elements relevant to the domain modeling task (i.e., nodes for entities and edges for associations). This main study

[2]Domain does not exclusively mean application domain, but rather any coherent ontology related to a specific topic.

[3]`https://www.google.de/intl/en/docs/about/`

[4]`https://www.google.com/forms/about/`

document explained the experimental task, an example of the domain modeling task, and a short context description of the system from its original requirements specification [308].

We created a survey questionnaire to collect demographic information relevant to the experiment using Google Forms. All participants could answer the survey only after completing the task to avoid fatiguing effects. At the beginning of the questionnaire, participants entered their assigned participant ID (PID), such that we could connect their response to the experimental task to their response to the questionnaire without storing any personal data. The questions were designed to collect all relevant independent variables listed in Table 1.

We piloted the experiment in a session with two Ph.D. students in SE. We clarified the instruction text and task descriptions based on the collected feedback.

3.4.7 Data Collection Procedure

We scheduled a one-hour session according to the availability of the participants. Because of differing schedules and time zones, we scheduled 16 sessions with up to three participants simultaneously. We conducted the sessions between 2023-04-03 and 2023-04-17.

Each session started with the first author explaining the general procedure of the experiment and obtaining consent to evaluate and disclose the anonymized data. No participant refused this consent and all data points could be included in the data evaluation procedure. Then, participants were instructed to read the prepared document in order and complete the contained tasks. The document contained all descriptions of the task such that all participants received the same instructions. The first author oversaw all sessions to address technical difficulties and recorded the minutes each participant spent per period. Ten minutes were estimated per period, but participants were free to allocate their time. In case participants took longer than the scheduled one hour, they completed the task in as much time as they required. Once the task was complete, participants also filled in the questionnaire to provide demographic information on context variables.

3.4.8 Data Preparation

To evaluate the collected data, we created a code book that characterizes issues in domain models. We developed detection rules for each dependent variable of the resulting product—i.e., missing entity, superfluous entity, missing association, and wrong association—and summarized them in a guideline (available in our replication package [280]). Then, the first author manually evaluated the resulting domain models of each participant using this guideline and recorded all detected issues.

The result of the coding process was a table where one row represents the evaluation of one domain model. Given $n_p = 25$ participants and $n_r = 4$ requirements, we ended up with $n_p \times n_r = 100$ data points. Each data point contained the number of

issues of each of the four types that occurred in the respective domain model. Finally, we standardized numerical variables in the demographic data for easier processing.

To assess the reliability of the rating, the fourth author of the paper independently recorded issues of three randomly selected participant responses, yielding an overlap of twelve ratings. Since each domain model can contain an arbitrary number of issues of each type, but our dependent variables only model the number of times that an issue type occurred, we consider each rating of a domain model as a vector of dimensions equal to the number of issue types. We then calculated the inter-rater agreement of the same domain model using the Spearman rank correlation between the vectors. The average cosine similarity is 77.0% and represents substantial agreement. The two raters discussed the remaining disagreement and concluded that they represented acceptable variance in the interpretation of participants' responses.

3.4.9 Frequentist Data Analysis

We performed a frequentist data analysis of the experimental data as in the original experiment [34]. Since the factor is categorical, all dependent variables are continuous, and our samples are dependent, our statistical method of choice falls between the parametric paired t-test [313] or the non-parametric Wilcoxon signed-rank test [314] based on the distribution of the variables, which we evaluated using the Shapiro-Wilk test results [315].

We reject a null hypothesis if the resulting p-value of a two-tailed statistical test is lower than the significance level α. To account for type I errors when performing multiple hypotheses tests targeting the same independent variable, we apply the Bonferroni correction [243]: we considered $\alpha' = \frac{\alpha}{m}$ where $\alpha = 0.05$ and m is the number of hypotheses tested for each value of the independent variable. For our five families of hypotheses $\alpha' = \frac{0.05}{5} = 0.01$.

Additionally, we report the effect size [316] using Cohen's D for the paired student t-test [317] and the matched-pairs rank biserial correlation coefficient for Wilcoxon signed-rank test [318].

3.4.10 Bayesian Data Analysis

We apply Bayesian data analysis with Pearl's framework for causal inference [76] to complement the frequentist data analysis [48]. Given the limited adoption of Bayesian data analysis in software engineering research [48, 80, 226], we complement this method section with a running example for understandability. In this running example, we illustrate the methodological steps of Bayesian data analysis for the hypothesis that requirements quality defects influence the number of wrong associations in a resulting domain model.

Three steps [54], which are the major steps of Pearl's original model of causal statistical inference [76], comprise the analysis. We explain each step in the following paragraphs.

Modeling In the modeling step, we make our causal assumptions about the underlying effect explicit in a graphical causal model. The graphical causal model takes the form of a directed acyclic graph (DAG) in which nodes represent variables and directed edges represent causal effects [248]. Our DAG contains four groups of variables:

1. Treatment: The independent variable that represents the requirements quality defect present in the requirement.

2. Context factors: The independent variables that represent the properties of the participants.

3. Experimental design factors: The independent variables that represent all factors of the crossover experiment design influencing the response variables [50].

4. Response variables: The dependent variables.

The effect of the treatments on the response variables is the subject of the analysis. By including both context and confounding factors, their influence is factored out from the treatments' causal effect on the response variables. Consequently, the effect of interest can be isolated from any confounding factor included in the DAG. We assume causal relations—represented by edges in the DAG—between every independent (treatment, context, and confounding) and the dependent variables. Additionally, independent variables may influence other independent variables.

Figure 8 shows the DAG of the running example. The treatment, response variable, and context factors correspond to the study variables as outlined in Table 1. The experimental design variables correspond to the factors listed in Table 2. The *experimental period* blocks the learning effect, i.e., the influence on the response variable caused by repeatedly performing the task. The *duration* blocks the time effect, i.e., the influence on the response variable caused by the amount of time that a participant took for each instance of the task. Note that the factor *tool experience* listed in Table 1 is missing from the DAG since we excluded it as explained in Section 3.4.5. Note that the factor *sequence* listed in Table 2 is missing from the DAG since it is confounded with subject variability (further explained in Section 5.4.1).

The DAG does not visualize the *interaction effects* we assume between two independent variables. An interaction effect occurs when the influence of one independent variable on the dependent variable depends on the value of another independent variable [62]. Visualizations of interaction effects in DAGs have been proposed [319] but are not common practice. In the running example, we assume two interaction effects via the following hypotheses:

1. *requirements quality * domain knowledge*: domain knowledge can compensate the effect of requirements quality defects [288]

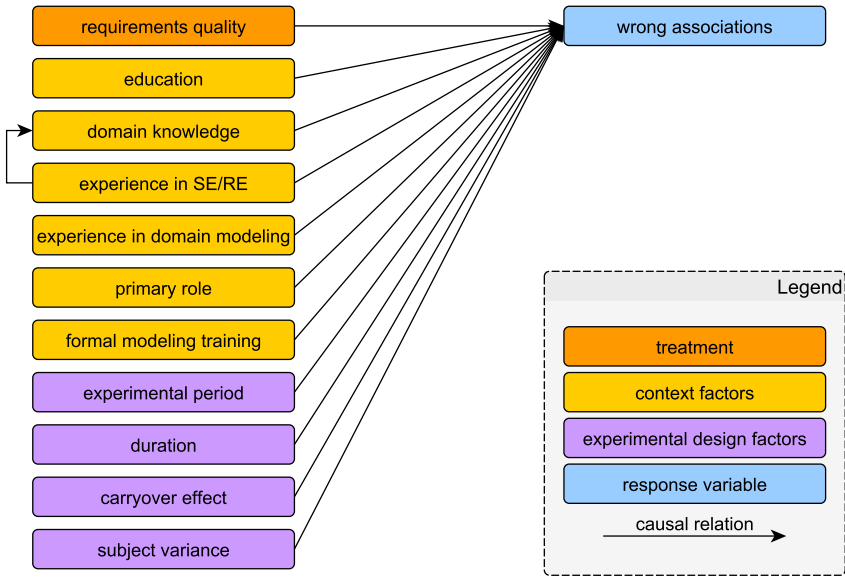


Figure 8: DAG for the analysis of wrong associations

2. *requirements quality* * *period* carryover effect [50]: the effect of a treatment may be influenced by the treatments applied in previous periods

Identification Including an independent variable Z that has an assumed causal effect on both the treatment X (i.e., $Z \rightarrow X$) and the outcome Y (i.e., $Z \rightarrow Y$) opens a non-causal path (i.e., $X \leftarrow Z \rightarrow Y$) from the treatment to the outcome [299]. This so-called backdoor path introduces spurious associations. Consequently, blindly moving forward with all variables may harm the causal analysis. Instead, the so-called adjustment set of variables needs to be selected [62] in the *identification* step. A series of four criteria [62] allows to make an informed selection of variables to include in the final estimation step. This way, we avoid variable bias like colliders which confound the causal effect between the treatment and the response variable.

In the running example, we assume the following causal relation between independent variables. The more experience a participant has in SE or RE, the more likely it is that they have acquired respective domain knowledge (experience in SE/RE $\rightarrow$ domain knowledge). We need to consider this relationship in the next step to avoid attributing impact to the wrong independent variable. For instance, in the running example, we need to distinguish whether *experience in SE/RE* has a direct influence on *wrong association* or whether it just influences *domain knowledge*, which influences the response variable.

Because we employ an experiment as our research method and fully control the treatment, there is no influence of any other independent variable on the treatment

variable.

Estimation In the estimation step, we perform a regression analysis. The regression analysis results in estimates of the response variable depending on the values of the independent variables. The result of the regression analysis is a Bayesian model trained with empirical data. The model provides the magnitude and sign of the effect that each independent variable has on the dependent response variable.

The estimation step begins by selecting a distribution type (likelihood) that represents the dependent response variable [227]. We select the distribution type based on the maximum entropy criterion [249] and ontological assumptions. This means we select the least restrictive distribution that fulfills all ontological assumptions about the variables' properties.

In our running example, the response variable is a count of wrong associations in a domain model. Consequently, the distribution must be discrete and only allow positive numbers or zero. Additionally, the response variable is bounded by the number of *expected associations* of the domain model, i.e., the number of associations in the sample solution, since a participant can only connect as many associations wrongly in the model as there were associations expected. Any associations added beyond the expected associations count as *superfluous associations*, a different response variable. Consequently, we represent the response variable with a *Binomial* distribution. The following formula encodes that the number of wrong associations in one domain model i ($E_i^{\times}$) is distributed as a Binomial distribution with the number of trials equal to the number of expected associations (E) and a probability $p_i \in [0, 1]$ of getting one association wrong.

$$E_i^{\times} \sim Binomial(E, p_i)$$

This formula assumes that the event—connecting one association wrong—is independent, i.e., one wrong association does not influence the success of any other association.

In the next step, we define the parameter that determines the response variable distribution (in the running example: p_i) in relation to the predictors selected in the identification step. The following formula shows a simplified version of this parameter definition (excluding most of the previously mentioned predictors in Table 1 for brevity).

$$logit(p_i) = \alpha + \alpha_{PID} + \beta_{RQD}^{T} \times RQD_i + \beta_{SE} \times exp.se_i$$

The `logit` operator scales the parameter p_i to a range of $[0, 1]$ since the probability parameter of the Binomial distribution only accepts this range of values [62, 320]. The parameter p_i is, in this example, determined by the following predictors:

1. Intercept (α): the grand mean of connecting an association wrongly, i.e., the baseline challenge of getting an association wrong.

2. Group-level intercept (α_{PID}, where the results of one participant represent one group): the participant-specific mean of connecting an association wrongly, i.e., the within-subject variability of response variables [50] modeled via partial pooling [247]

3. Treatment (RQD_i): the influence of a requirements quality defect on the probability of connecting an association wrong (as an offset from the grand mean).

4. Software Engineering Experience ($exp.se_i$): the influence of the subject's software engineering experience on the probability of connecting an association wrong (as an offset from the grand mean).

The variables RQD_i and $exp.se_i$ contain the values recorded during instance i of conducting the experimental task and are each prefixed with coefficients β^T_{RQD} and β_{SE}. These coefficients are Gaussian probability distributions that represent the magnitude and direction of the influence that the variable values have on the parameter p_i and, therefore, on the distribution of the response variable. The mean μ of the coefficient represents the average effect of the variable on the parameter p_i, and the standard deviation σ encodes the variation around this average effect. A standard deviation of $\sigma = 0$ would mean that the variable has a deterministic effect of strength μ on p_i and, therefore, the distribution of the response variable. In reality, this is highly unrealistic. Hence, the standard deviation captures the uncertainty of the effect of a variable on p_i.

Special cases of variables are the intercepts α and α_{PID}, which are probability distributions without any variable and, hence, represent the predictor-independent general and participant-specific probability of connecting an association wrong. In the beginning, we assign probability distributions spread around $\mu = 0$ ($\beta^T_{RQD} \sim Normal(0, 0.5)$), so-called *uninformative priors*, to these coefficients. These distributions encode our prior beliefs about the influence of the respective predictor, i.e., that it is yet unknown whether the predictor has a positive ($\mu > 0$) or negative ($\mu < 0$) influence on p_i. Only where previous evidence for the impact of a predictor on the response variable exists, we select more informative priors. For example, the experiment by Femmer et al. [34] indicates that *missing entities* and *missing associations* are in general rare, which we represent in our priors by selecting $\alpha \sim Normal(-1, 0.5)$ for the intercept.

We assess the feasibility of the selected prior distributions via prior predictive checks [230]. During this check, we sample only from the priors, i.e., we predict the response variable given the recorded data of independent variables and the prior probability distributions of the predictor coefficients. Figure 9 visualizes the result of the prior predictive check. The grey bars represent the actual observed distribution

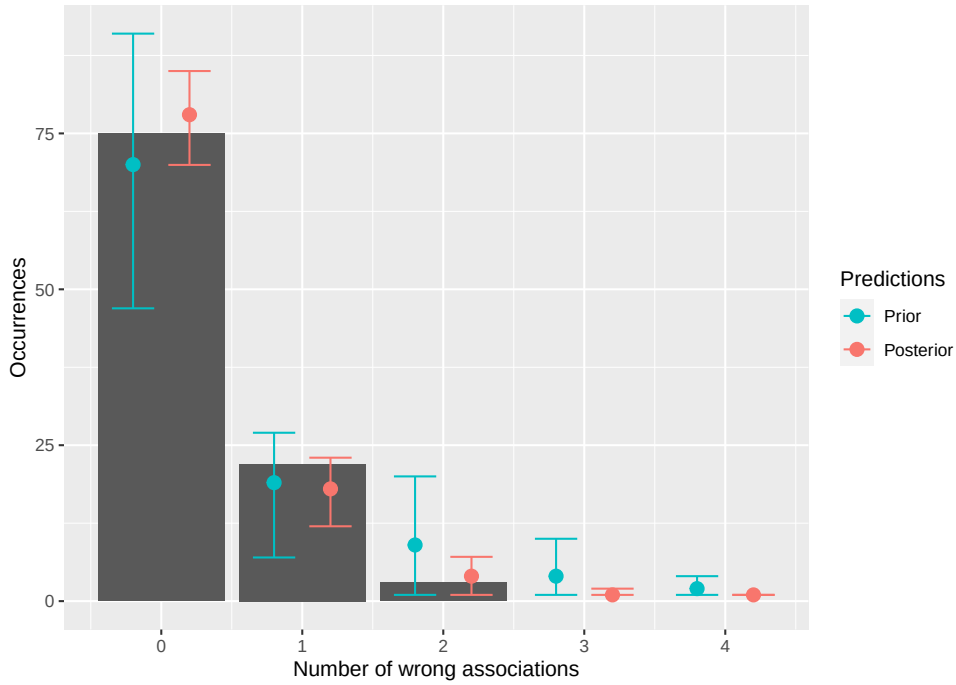


Figure 9: Predictive checks with prior and updated posterior coefficient distributions

of the response variable. For example, 75 domain models contained zero wrong associations. The distribution of predicted values for the response variable (cyan whisker plots) encompasses the actual observed distribution of the response variable. This confirms that the actually observed distribution is approximately determined by the uninformative prior distributions.

Upon confirmation of the priors' feasibility, we train the Bayesian models with the data recorded during the experiment. We conducted the analysis using the `brms` library [228] in *R*. Hamiltonian Monte Carlo Markov Chains (MCMC) [250] update the coefficient distributions based on the empirical data. During this process, the parameters of the coefficient distributions are adjusted to better reflect the response variable based on the predictor variable values.

After the training process, we perform posterior predictive checks, which work similarly to the prior predictive check but use the updated posterior coefficient distributions instead of the prior distributions. Figure 9 also visualizes the posterior predictive check for the running example. The distribution of the predicted values (red whisker plots) still encompasses the actually observed distribution of the response variable but has narrowed around these values. This indicates that the posterior distributions encode the influence of the predictor variables more accurately than the prior distributions, i.e., that the model has successfully gained predictive power during the training process.

To overcome the problem mentioned in the identification step, i.e., attributing impact to the wrong predictor, we train additional models per response variable to test for conditional independence [62]. For example, to determine the correct causal relationship between the two independent variables *experience in SE*, *domain knowledge*, and the dependent variable, we train two additional models where each one misses one of the two variables [80]. After training, we compare the posterior distributions of the remaining parameter coefficients. If a posterior distribution significantly moves from $|\mu| > 0$ towards $\mu = 0$ when including a variable, then the response variable is independent of that variable when conditioning on the included variable. The model does not gain any further information from the variable with $\mu \simeq 0$, and its causal relation is disputed. If the posterior distribution does not deviate significantly when including another variable, its causal impact is confirmed.

Finally, we perform a stratified posterior prediction to answer our research questions. To this end, we construct a synthetic data set with four data points, one for each value of the main factor variable (i.e., baseline, PV, AP, PVAP). We fix all other independent variables at representative values—i.e., the mean for continuous and the mode for discrete variables. Then, we sampled $6,000$ predictions for each of the four data points. This isolates the effect of the treatment but maintains the uncertainty of the influence of every independent variable encoded in the standard deviation of every predictor coefficient and, hence, more accurately describes the causal relationship between the treatment and the outcome. We compare the $6,000$ predictions of each of the three treatments (PV, AP, PVAP) with the $6,000$ predictions from the baseline (no defect) and count how often the treatment causes a higher, equal, or lower outcome variable. We scale these values to percentages to summarize the effect of the treatment on the outcome variable. This evaluation avoids a point-wise reduction of the results and comparison to an arbitrary significance level as customary in frequentist analyses [305]. Rather than providing a binary answer to the hypotheses, we present the more informative distribution of results. However, for the sake of reporting, we consider the distribution of the duration variable *skewed* if the two percentages differ from the mean (50%) by 10% each and consider the other distributions skewed if the two percentages differ by 10% from each other.

Additionally, we plot the marginal effect of selected independent variables to visualize their isolated impact on the response variable. The isolated impact reveals how context and confounding factors influence the response variable. This includes visualizing the carryover effect, i.e., the interaction between the treatment and the period.

4 Results

Section 4.1 shows the results of the frequentist and Section 4.2 the results of the Bayesian data analysis. Section 4.3 compares the part of our results that contributes

a conceptual replication to the original study. Section 4.4 compares the results from our FDA to the results from our BDA.

4.1 Frequentist Data Analysis

Table 3 shows the mean and median values of the response variables similar to how they are reported by Femmer et al. [34]. Note that the results are not directly comparable to those in the original study as both our experimental objects and treatments varied.

Table 3: Mean and median response variable values (reported as mean/median in each cell)

Defect	Duration D	Missing Entities E^-	Superfluous Entities E^+	Missing Associations A^-	Wrong Associations $A^\times$
none	7.38/6.5	0.23/0	0.5/0	0.38/0	0.08/0
PV	6.88/7	0.81/1	0.42/0	0.81/1	0.62/0
AP	7.12/6	1.23/1	0.96/0.5	1.12/1	0.54/0.5
PVAP	7.96/7	1.27/1	0.46/0	1.38/1	0.38/0

Table 4 lists the results of our frequentist data analysis and relates them to the results from the original study [34].

Table 4: Results of frequentist analysis including the p-value of the hypothesis test (p), confidence interval (CI), and effect size (ES). Statistically significant results in **bold** (original study $\alpha = 0.05$, this experiment $\alpha' = 0.01$).

Outcome	Treatment	Original [34]			Replication		
		p	CI	ES	p	CI	ES
Duration	PV				0.67	$(-0.4, 0.7)$	-0.13
	AP				0.86	$(-0.7, 0.86)$	0.01
	PVAP				0.49	$(-0.5, 0.25)$	0.14
Missing Actors	PV	0.10	$(0, \infty)$	0.39			
Missing Objects	PV	0.25	$(-1, \infty)$	0.25			
Missing Entities	PV				$\ll$ **0.01**	$(-1, 0)$	-0.79
	AP				$\ll$ **0.01**	$(-1.5, 0)$	-0.93
	PVAP				$\ll$ **0.01**	$(-2, 0)$	-0.81
Superfluous Entities	PV				0.64	$(-0.5, 0)$	0.14
	AP				0.19	$(-2.5, 0)$	-0.41
	PVAP				0.62	$(-1, 1)$	0.15
Missing Associations	PV	**0.02**	$(1, \infty)$	0.75	0.025	$(-1, 0)$	-0.58
	AP				$\ll$ **0.01**	$(-2, -1.5)$	-0.87
	PVAP				$\ll$ **0.01**	$(-2, -1)$	-0.84
Wrong Associations	PV				1.0	$(0, 0)$	0.0
	AP				$\ll$ **0.01**	$(-1, 0)$	-0.85
	PVAP				0.052	$(-1.5, 0)$	-0.67

The frequentist data analysis suggests rejecting the following hypotheses and, therefore, proposes the following effects as statistically significant (with $\alpha' = 0.01$):

1. $H_0^{PV \to E^-}$: passive voice impacts the number of missing entities
2. $H_0^{AP \to E^-}$: ambiguous pronouns impacts the number of missing entities

3. $H_0^{PVAP \rightarrow E^-}$: the co-occurrence of passive voice and ambiguous pronouns impacts the number of missing entities
4. $H_0^{AP \rightarrow A^-}$: ambiguous pronouns impact the number of missing associations
5. $H_0^{PVAP \rightarrow A^-}$: the co-occurrence of passive voice and ambiguous pronouns impacts the number of missing associations
6. $H_0^{AP \rightarrow A^\times}$: ambiguous pronouns impact the number of wrong associations

The associated effect size is considered large [321] in all cases.

4.2 Bayesian Data Analysis

This section follows the methodology described in Section 3.4.10 by presenting the DAG in Section 4.2.1, posterior predictions in Section 4.2.2, and marginal plots in Section 4.2.3.

4.2.1 Causal Model and Adjustment Set

Figure 10 visualizes the DAG that graphically models our causal assumptions. It is an extension of Figure 8, the running example, including all five dependent variables. To preserve the readability of the DAG, we introduce a *distributor* node. This node substitutes the connections from the source of every incoming edge to the target of every outgoing edge.

The edges represent the same causal reasoning as presented in the running example in Section 3.4.10. We assume that all independent variables (treatments, context factors, and experimental design factors) have an impact on all dependent variables. The impact of the requirements quality defect (the three treatments) is the relationship of interest in our analyses. In addition, to the already described impact of experience in SE on domain knowledge, we assume the following causal relationships:

1. Duration impacts all other dependent variables: Since we did not constrain the time for each period, different amounts of minutes taken for each object may influence the results. Taking a longer time for one domain model may reduce the amount of defects in the final model.
2. Missing entities impact missing associations: If an entity is missing from the domain model, any association involving that entity will also be missing (as already supported by our re-analysis [81]).

Equally notable are the non-existing associations between nodes, especially between context factors. In our DAG, we only assume an impact of experience in SE/RE on domain knowledge (as explained in Section 3.4.10). We do not assume, for example, a causal relation between education and experience in SE/RE as higher

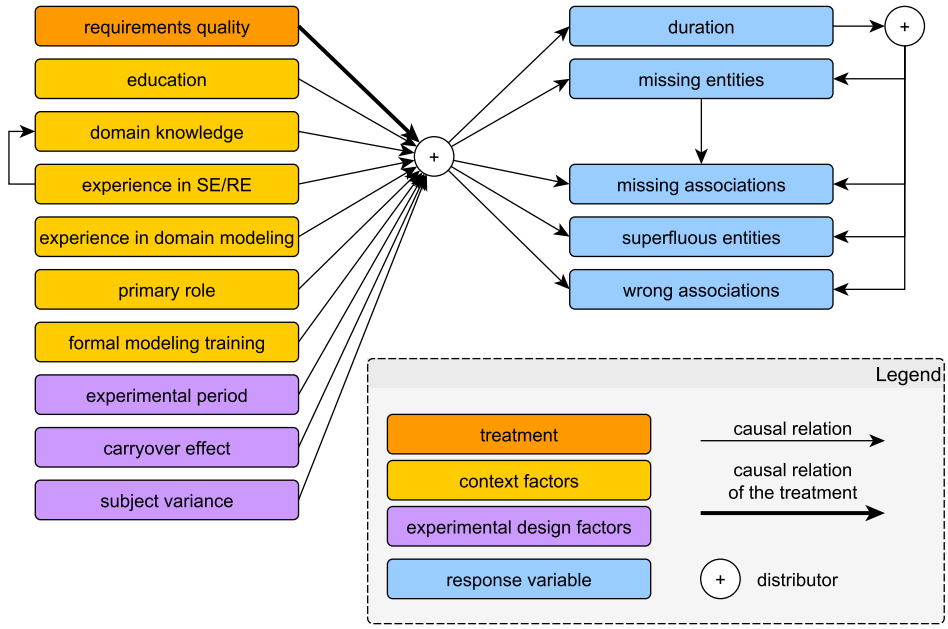


Figure 10: Directed acyclic graph visualizing all causal assumptions

levels of education do not entail more industrial experience or vice versa. Similarly, we assume education to be independent of domain knowledge as most educational programs known to us are domain-independent. The resulting set of associations visualized in Figure 10 corresponds to the authors' shared beliefs that warrant assuming causal relationships between two variables. While we do not expect every reader to share these beliefs, we hope that the explicit and transparent documentation of our assumptions invites constructive, iterative improvements by challenging them via empirical investigations.

The DAG from the running example in Figure 8 lists the variable duration as an independent variable, while the final DAG in Figure 10 lists it as a response variable. The variable duration takes on two distinct roles depending on the current analysis. In the case where the analysis targets the effects on the duration, it is the sole response variable. In all other cases, it is an independent variable.

In the second step in the statistical causal inference [54], the identification step, we found the adjustment set to include all variables for prediction. To discern the impact of independent variables with causal relations among them, we developed comparison models in which certain variables were excluded. The comparison showed that the exclusion did not change the estimations of the coefficient distributions. Hence, we assume all causal relationships are feasible and evaluate the full model, including all eligible variables as predictors.

4.2.2 Posterior Predictions

Based on maximum entropy [249], we model the five response variables using the following probability distributions. The duration is centered around the global mean, therefore, we model it with a Gaussian distribution around $\mu = 0$. The number of superfluous entities is an unbounded count with an index of dispersion of about 1.5, hence, we model it as a negative binomial distribution. Missing entities, missing associations, and wrong associations are bounded counts and, hence, modeled as Binomial distributions. Table 5 contains the result of the predictions from the posterior distributions. Each cell contains the resulting likelihood that the occurrence of a factor causes fewer or more issues of the respective outcome compared to the baseline of no quality defects[5]. The larger the difference between the likelihood of more $(+)$ than fewer $(-)$ issues, the stronger the effect of that factor on the outcome. If the likelihood of more $(+)$ issues outweighs the likelihood of fewer $(-)$ issues, the factor has a negative effect. If the two values are similar, then the factor has no clear effect on the outcome.

Table 5: Likelihood that a treatment produces fewer $(-)$ or more $(+)$ occurrences of the respective outcome variable.

Outcome	PV		AP		PVAP	
	-	+	-	+	-	+
Duration	57.2%	42.8%	51.7%	48.3%	44.8%	55.2%
Missing entities	29.6%	32.5%	24.7%	40.1%	27.4%	35.6%
Superfluous entities	26.6%	22.5%	22.5%	33.3%	26.4%	24.6%
Missing associations	25.0%	45.0%	20.2%	51.2%	22.2%	49.4%
Wrong associations	10.5%	11.5%	5.2%	44.6%	6.9%	31.5%

For example, the first two cells in the second row of Table 5 state that using passive voice causes *fewer* missing entities in 29.6% and *more* missing entities in 32.5% of all cases. In the remaining 37.9% of all cases, passive voice causes neither more nor fewer missing entities. Given this balance, the effect of passive voice on missing entities is unclear, and there is not enough evidence to reject $H_0^{PV \to E^-}$.

Result of Posterior Predictions

The following effects are likely given the skewed distribution of posterior predictions: passive voice, ambiguous pronouns, and their co-occurrence cause an increasing number of missing associations. Ambiguous pronouns cause an increasing number of wrong associations. Ambiguous pronouns cause an increasing number of missing and superfluous.

We use an arbitrary threshold of 10% to report notable results in textual form. Refer to Table 5 for the actual, more fine-grained results.

[5]The remaining cases ($100\% - less - more$) are omitted from the table

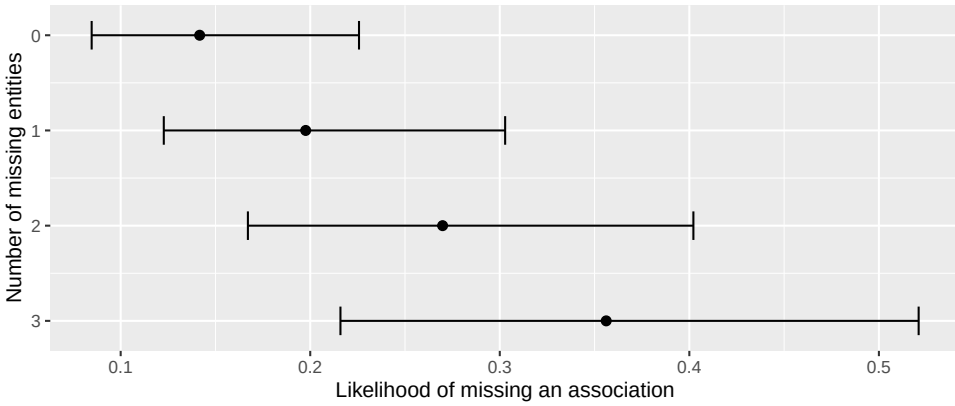


Figure 11: Impact of missing entities on missing associations.

4.2.3 Marginal and Conditional Effects

Marginal plots visualize the isolated effect of specific predictors when fixing all other predictors to representative values. In the following, we present selected marginal plots that show the effects of interest. The remaining plots can be found in our replication package.

Missing entities impact missing associations Figure 11 visualizes the effect of the number of missing entities on the number of missing associations. The y-axis represents the expected value of missing entities over multiple attempts with a trial size of one. Hence, it corresponds to the likelihood of missing one entity.

The plot supports the assumption that missing an entity promotes missing an association, which the original experiment did not consider and instead attributed the missing associations fully to the use of passive voice [34]. In fact, the strength of the effect of passive voice on missing associations ($\mu^{PV}_{RQT} = 0.38$) is similar to the strength of the effect of missing entities on missing associations ($\mu_{E^-} = 0.40$). However, the uncertainty of the impact of passive voice ($\sigma^{PV}_{RQT} = 0.32$) is higher than that of missing entities ($\sigma_{E^-} = 0.09$). This means that the effect of missing entities on missing associations is much more reliable than the effect of passive voice on missing associations.

Impact of duration Figure 12 visualizes the impact of relative duration (i.e., deviation in duration from the overall average time of creating a domain model in minutes) on the two response variables superfluous entities and wrong associations. The red estimate shows that the longer a participant took to generate a domain model (relative duration > 0), the more likely they were to introduce superfluous entities. The cyan estimate shows that the shorter time a participant took to generate a domain model (relative duration < 0), the more likely they were to connect an association wrongly.

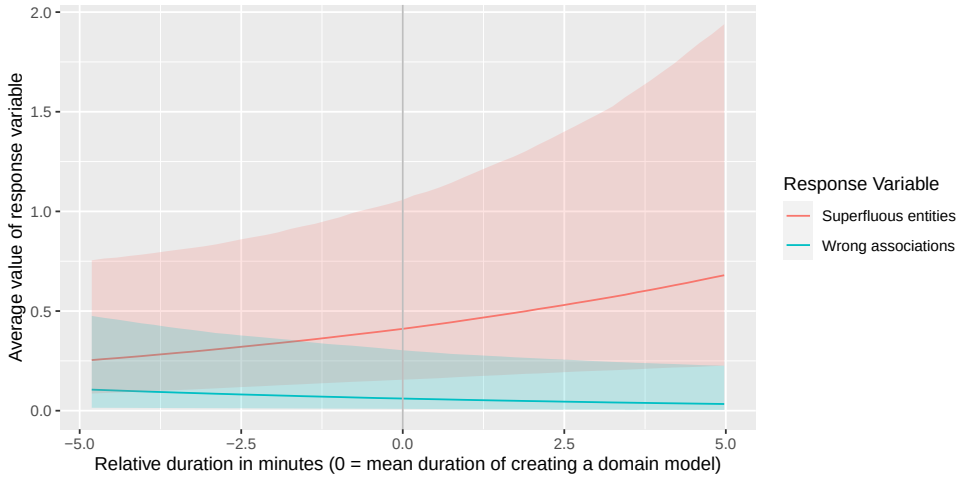


Figure 12: Impact of relative duration on the number of superfluous entities and wrong associations

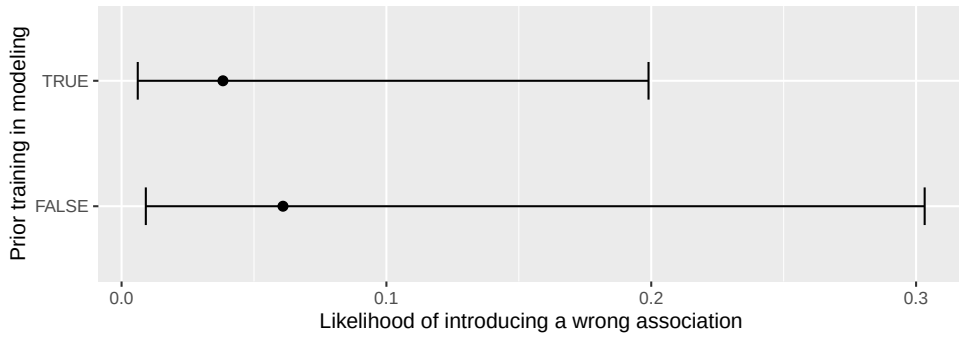


Figure 13: Marginal effect of prior formal training in modeling on the number of wrong associations

Impact of previous training in modeling Figure 13 visualizes the impact of prior formal training in modeling on the number of wrong associations in a domain model. A participant with prior formal training ($formal = TRUE$) shows a slightly lower likelihood of connecting associations wrongly. The overlapping confidence intervals do, however, indicate a strong variance of the effect.

Impact of remaining context factors None of the remaining context factors has a stronger effect on any of the response variables than the previously mentioned impact visualized in Figure 13. This means that the other context factors are neither notable ($\mu > 0.4$) nor significant ($\sigma < \mu$). The replication package contains a detailed summary of all coefficients.

Interaction between domain knowledge and the treatment Conditional plots visualize interaction effects between two predictors. Figure 14 visualizes the interac-

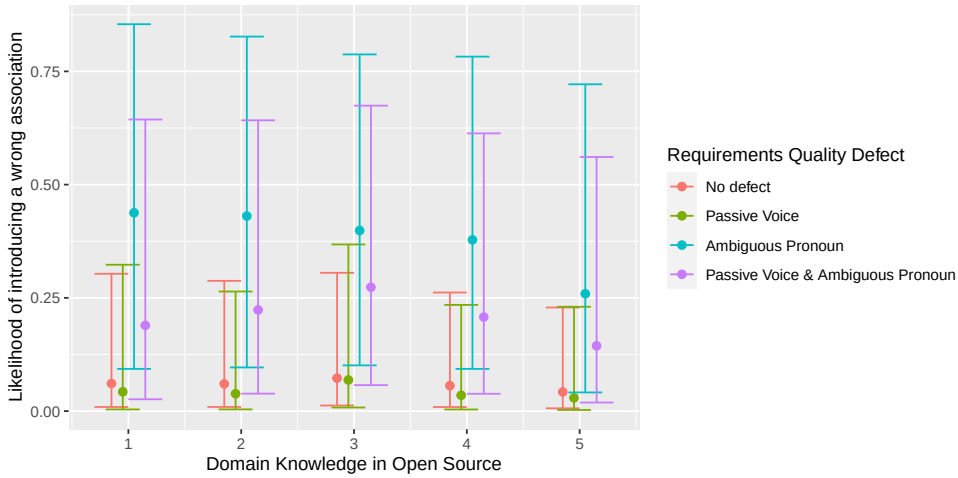


Figure 14: Interaction effect between domain knowledge and the treatment on the number of wrong associations

tion effect between domain knowledge about open source and the treatment on the number of wrong associations. The figure shows that the impact of ambiguous pronouns (cyan whisker plots) on the response variable *number of wrong associations* diminishes the greater the domain knowledge about open source. For the co-occurrence of ambiguous pronouns and passive voice (purple whisker plots), the effect is less pronounced but symmetrical, i.e., the factor has the strongest impact on the response variable when the domain knowledge is medium. However, the effect contains high uncertainty when the treatment involves ambiguous pronouns, represented by the large and overlapping confidence intervals (cyan and purple whiskers in Figure 14). The collected data does not suffice to support the significance of this effect.

Result of Marginal and Conditional Effects

Most context factors do not show a significant impact on either the response variables directly or mediate the effect of quality defects. The few context factors that do show an impact are not significant.

4.3 Comparison of original with our Study Results

For the part of our study that serves as a conceptual replication, we compare the results of the original study [34] and its re-analysis [81] with our results [307]. Regarding $H_0^{PV \to E^-}$, we obtain conflicting results from the FDA as we reject the null hypothesis while the original study does not, but consistent results from the BDA, as the distribution of the posterior prediction of missing entities is balanced. We obtain conflicting results for $H_0^{PV \to A^-}$ from the FDA as we cannot reject it as in the origi-

nal study. Our BDA suggests that passive voice has a slight impact on the number of missing associations (25.5% less and 45.0% more likely to miss an association). The result of the BDA does not suppose an effect as strong as the original study, but it does agree with the re-analysis of the original study [81] on a slight impact. Overall, the results of our BDA agree with the reanalyzed results of the original study. The variation of study elements (e.g., experimental subjects and objects) [83] increases the replicability space within the generalizability space [85] and identifies those elements as non-influential [84] to the original claim.

Comparison of Studies

The results of the frequentist analyses of the original and our study differ. However, the more thorough Bayesian data analysis agrees with the properly re-analyzed original results. Due to the variation of several elements of our study from the original study, the conceptual replication extends the external validity of the original claim that passive voice has only a slight impact on the domain modeling activity.

4.4 Comparison of FDA with BDA Results

Secondly, we compare the results of our frequentist data analysis with the results of our Bayesian data analysis. We obtain *consistent results* [307] for $H_0^{PV \to D}$, $H_0^{AP \to D}$, and $H_0^{PVAP \to D}$. Neither the frequentist nor Bayesian analysis suggests an impact of the treatment on the relative duration to create a domain model.

The frequentist analysis rejects $H_0^{RQD \in \{PV,AP,PVAP\} \to E^-}$, while the Bayesian analysis remains more cautious. The posterior predictions in Table 5 show a tendency towards the treatment having an impact, but with large uncertainty. Additionally, marginal plots of the Bayesian analysis reveal that the primary role, experience with domain modeling, and education impact the dependent variable. Both analyses agree that $H_0^{RQD \in \{PV,AP,PVAP\} \to E^+}$ cannot be rejected, though the Bayesian analysis attributes a tendency of causing superfluous entities to ambiguous pronouns.

The frequentist analysis suggests to reject $H_0^{RQD \in \{AP,PVAP\} \to A^-}$, i.e., ambiguous pronouns and their coexistence with passive voice influence the number of missing associations. The Bayesian analysis again shows a tendency towards an impact but retains its uncertainty about the effect. Marginal plots instead emphasize the influence of missing entities on the response variable.

The analyses agree on the impact of ambiguous pronouns on the number of wrong associations and suggest to reject $H_0^{RQD \in \{AP,PVAP\} \to A^\times}$. Both the large effect size and the skewed distribution of posterior predictions support the existence of a causal effect of ambiguous pronouns on wrong associations in a domain model.

Comparison of Analysis Methods

The results of our frequentist analysis differ from our Bayesian analysis: the Bayesian data analysis remains more cautious about several effects suggested by the frequentist analysis. The extended casual model attributes part of the effect on the response variable on other independent variables than the treatment.

5 Discussion

Section 5.1 answers the research questions. Section 5.2 discusses implications for requirements quality practice and Section 5.3 for requirements quality research. Section 5.4 presents the threats to validity.

5.1 Answers to research questions

5.1.1 Answer to RQ1

RQ1.1: Impact of passive voice. Using passive voice in natural language requirements specifications has a slightly negative effect on the domain modeling activity regarding missing associations. This finding aligns with the conclusions drawn by the original study by Femmer et al. [34, 81]. However, the Bayesian data analysis emphasizes that both context factors, and especially the number of missing entities, have a significant impact on the number of missing associations as well. Overall, these results support the claim that passive voice can have a negative impact in specific cases but is overall not a significant factor in subsequent activities depending on the requirement [34, 172].

RQ1.2: Impact of ambiguous pronouns. The use of ambiguous pronouns has a strong effect on the number of wrong associations in the resulting domain model. Additionally, using ambiguous pronouns has a slight negative effect on the number of missing and superfluous entities and missing associations. This confirms the risk of using ambiguous pronouns that have been mainly hypothesized in previous research [278] and explains the focus on ambiguity in requirements quality research [6]. An ambiguous pronoun in a requirements specification has a 44.6% chance of causing a wrongly connected association in the domain model, limiting the model's correctness and propagating risk to further activities.

RQ1.3: Combined impact. The co-occurrence of passive voice and ambiguous pronouns has a strong effect on the number of wrong associations. Additionally, it has a slight effect on the number of missing entities and associations. The impact

correlates with but never exceeds the effect of pure, ambiguous pronouns. This supports the assumption that passive voice does not create any further impact in addition to the effect of ambiguous pronouns.

5.1.2 Answer to RQ2

RQ2.1: Impact of context factors. Only a small number of context factors included in the study show a notable effect on the response variables. The duration of the domain modeling activity confirms assumed patterns: taking shorter than average increases the chance of missing elements or connecting associations wrongly, taking longer time than average increases the chance of adding superfluous entities. Prior formal training in modeling shows a slight yet not significant positive effect.

RQ2.2: Mediation of context factors. The interaction effect between domain knowledge and the treatment shows that higher domain knowledge can mitigate the negative effect of quality defects on response variables. In particular: higher domain knowledge reduces the chance of connecting associations incorrectly. While the effect still exhibits a large variance, this hints at the possibility of compensating quality defects with domain knowledge.

5.2 Implications for Requirements Quality Practice

The presented results indicate that the negative impact of two requirements quality defects can differ significantly. When allocating resources toward detecting and removing specific quality defects from requirements specifications, organizations can make informed decisions based on the calculated impact of the respective quality factor. In our case, we recommend explicitly detecting and resolving ambiguous pronouns, while passive voice is not critical enough to deserve dedicated attention. This aligns with the common perception in requirements quality research that ambiguity receives the most attention [6] while using passive voice rarely has a tangible impact [172]. By filtering requirements writing guidelines for quality factors that have a measurable effect, we expect greater acceptance of requirements quality assessment tools in practice [8, 9].

Additionally, measuring the effect of a quality defect on the relevant attributes of activities that use these requirements allows quantifying it economically [40]. While the cost of a quality defect is hard to determine, a company can quantify the cost of activities' attributes like increased duration. This economic perspective provides additional decision support for companies when assessing whether it is worth detecting and removing a specific quality defect [135].

Finally, the potential influence of context factors on the impact of quality defects on affected activities may incentivize organizations to invest in developing these factors. For example, improving domain knowledge and providing formal modeling

training may compensate for quality defects.

5.3 Implications for Requirements Quality Research

Employing Bayesian data analysis to investigate the impact of requirements quality defects provides sophisticated and sensitive insights necessary to propel requirements quality research [40]. The result of the analysis models both the direction and strength of an impacting factor while retaining information about its certainty. These insights go beyond the point-wise comparison and binary result of frequentist analyses [48]. The frequentist analysis fails to compare the impact of quality defects on the response variables, as even the calculated effect sizes are similar ($0.79 < |ES| < 0.93$). The Bayesian data analysis, on the other hand, clearly shows that some effects are much stronger (e.g., $H_0^{AP \to A^\times}$) than others (e.g., $H_0^{PV \to A^-}$). Still, the BDA relies on the causal model expressed in a DAG, statistical assumptions about variable types and their independence, and the validity of constructs. Therefore, the results obtained via BDA cannot be seen as more valid by design. However, the BDA is more transparent and allows critical debate—e.g., about the causal assumptions underlying our analysis in Figure 10—which facilitates the incremental improvement of empirical studies.

Including context factors in the prediction allows comparing the impact of requirements quality with the impact of human and process factors, revealing which causes changes in the response variable. These context factors can also represent the properties of non-human agents like generative artificial intelligence (GenAI) models which are increasingly employed for RE tasks. Involving context factors like the version number of a GenAI model, its parameters, its context window, and other factors in empirical studies resembling our approach will allow to investigate which configurations of these models excel at performing their RE task.

Abandoning simple NHSTs for identifying relevant factors of requirements quality and instead opting for a proper framework for causal inference like BDA will increase the likelihood of solving problems with practical relevance [9] that justify subsequent tool development [52]. Empirical studies with explicit causal assumptions (e.g., visualized as DAGs) and sophisticated analyses will produce context-sensitive evidence that can be synthesized in the common framework of the requirements quality theory [40]. The continuous synthesis of evidence from individual studies in this common framework will produce more reliable and generalizable conclusions [42, 322] and effectively address the lack of empirical insights in requirements quality research [6].

Using a controlled experiment benefits the investigation of the quality factor [34]. The DAG shown in Figure 10 visualizes this control, as no other factor influences the treatment in question. This eliminates spurious associations that could confuse the results [62]. On the other hand, the cost of conducting a controlled experiment—

especially with participants from industry—cannot be neglected [100]. Luckily, statistical causal inference via Bayesian data analysis works equally well with observational data, as shown by Furia et al. [80].

Finally, Bayesian data analysis allows for incremental improvement of empirical inquiry regarding requirements quality. The causal assumptions that the DAG makes explicit can be reviewed, discussed, and updated to inform future empirical methods. Insights derived from Bayesian data analysis can be used as prior knowledge in subsequent analyses, just as we sensibly used previous results [34] to inform our priors.

Worth noting is that our comparison between FDA and BDA conflates the use of causal frameworks with advanced Bayesian statistics. An FDA can also employ causal frameworks that mitigate parts of the shortcomings mentioned in Section 2.2, as previously shown by Furia et al. [80]. However, frequentist approaches tend to limit their analyses to the treatment and the response variable, disregarding potential context [87] or experimental design factors [50]. BDA, on the other hand, entails the use of an explicit causal framework [62, 299], which is why we support the recommendation of abandoning FDA for BDA in SE research [48, 78].

Implications

Quality defects in requirements specifications have a varying impact on affected activities that depend on them. Context factors may compensate for this impact but require better metrics to quantify them. Bayesian data analysis provides more fine-grained insights into these effects than frequentist methods.

5.4 Threats to Validity

We present and discuss threats that could affect our study based on the guidelines by Wohlin et al. [47] and extended by the guideline by Vegas et al. [50] for the specific threats caused by the use of a crossover design. The threats to validity are prioritized, considering our work focuses on replicating the first study testing a causal theory predicting the impact of requirements quality factors on downstream development tasks.

5.4.1 Internal validity

Our design and the blind nature of the experiment avoid the threat to *selection-maturation interaction*. Nevertheless, the new settings (i.e., online asynchronous experiment) may have caused a *diffusion or imitation of treatments*—i.e., information may have been exchanged among the participants. The experiment supervisor monitored the participants to prevent their communication with each other and asked them not to distribute the experimental task and materials. We acknowledge that *selection* can

bias our sample as volunteers are generally more motivated to perform in an experimental task [211].

The crossover design emits additional threats to validity [50]. We mitigate the *learning by practice* effect—i.e., participants getting better when repeating the experimental task—in three ways: Firstly, we disperse the learning effect evenly at design time by randomizing the sequences of treatments. Secondly, we include a warm-up object to get participants used to the task and tool but exclude that data from the analysis. Thirdly, we include the *period* variable as a predictor to factor out the learning effect during the analysis. We avoid the threat of *copying* by prohibiting communication among participants and using experimental objects where solutions cannot be copied from one task to another.

The threat of *optimal sequence* describes the risk that there is a sequence in which the treatment is applied, which optimizes the participants' performance in deriving domain models. We cannot block this threat at analysis time as the sequence and participant IDs are highly correlated. This is because we could—in all but one case—assign only one participant ($n_p = 25$) to each sequence ($n_s = n_r! = 4! = 24$). Because of this strong correlation, the Bayesian model is incapable of distinguishing between the impact of the sequence (β_{seq}) from the within-participant variance (α_{PID}) [50]. More participants per sequence would have been necessary to block the threat of an optimal sequence, but these were unavailable to us.

Finally, we address the threat of *carryover*—i.e., the change of the impact caused by the period in which the treatment was applied—at analysis time by including the term $period * treatment$ in the predictors. This way, the carryover effect is factored out from the impact of the treatment and analyzable from the posterior distributions.

5.4.2 Conclusion validity

We addressed the *reliability of measures* threat by creating and disclosing evaluation guidelines and peer-reviewing the extraction of the dependent variables from the collected domain models. Despite the acceptable inter-rater agreement score, an in-depth qualitative evaluation of the remaining disagreements may be beneficial to further improve the evaluation instrument and, therefore, the reliability of the results. We addressed the *random heterogeneity of the subjects* by a design in which each participant acts as their own control group.

Moreover, we focused on including and analyzing context variables related to the participants' experience. Our sample of participants is not representative of all context factors. Consequently, our Bayesian data analysis cannot identify all causal effects of some context factors. However, by including them in the causal considerations, the effect of the factors is isolated from the potential confounding variables [302].

The conclusion validity of our study is strengthened by applying two different data analysis approaches and comparing their results. The data analysis suffers from

the threat of *low statistical power* when it comes to evaluating interaction effects, as reliably identifying them requires a larger sample size [79]. We limit the number of interaction effects considered in our models and discuss the uncertainty around the coefficient estimates to minimize this threat.

The analysis can suffer from *violated assumptions of statistical tests*. Modeling the number of missing entities and associations as binomial distributions implies the independence of each event, i.e., that each missing entity and association is independent of all other missing entities and associations. While we did not observe any cascading, i.e., dependent, defects, their independence remains only assumed.

5.4.3 Construct validity

Our study can suffer from *mono-operation bias* as we focus only on a subset of quality factors that can potentially exist [6, 41]. Nevertheless, our goal with this replication is to extend the initial quality factor of passive voice reported by Femmer et al. [34] to a second one—ambiguous pronouns—which is widespread as indicated by the literature [6, 41].

Similarly, a *confounding of constructs and level of constructs* could influence the outcomes of our study. For example, the presence of several ambiguous pronouns or passive voice sentences rather than their binary presence or absence from a specification. Further replications, focusing on improving construct validity, should include several levels of each treatment.

Mono-method bias is a potential threat to construct validity—i.e., we measured the dependent variables using a single type of measurement, inspired by the original study. However, the measurements were based on a pre-defined protocol and peer-reviewed. Our study may result in a *restricted generalizability across constructs* since the presence or absence of the different quality factors could result in side effects for other interesting outcomes we did not measure (e.g., comprehensibility or maintainability of the specification).

Among the social threats to construct validity, we acknowledge that *hypothesis guessing* may have taken place since the participants could try to guess the concrete goal of the experimental task based on the invitation text and material provided during the sessions. Nevertheless, we used the same text and phrasing to invite all participants and the same material during the experimental task. *Evaluation apprehension* could have played a role since some participants are students at the authors' institution. However, students did not receive rewards (e.g., extra grade points) for participating in the experiment, and they received their course grades before the start of the experiment.

Moreover, our study can suffer from an *inadequate preoperational explication of constructs* as we did not validate our context factors. For example, we are unable to provide any proof that the self-reported number of years spent in RE adequately represents the latent variable of experience in RE beyond educated guesses and rely-

ing on comparable practices in our scientific community [255]. To improve the construct validity, separate studies investigating the adequacy of these measurements in representing their constructs are necessary. This particularly impacts our decision to replace the binary distinction of participants by type (students versus practitioners) with more fine-grained variables like experience and domain knowledge. While our study supports the feasibility of this step on an analytical level, we cannot prove its validity on a conceptual level. We encourage investigating the feasibility of variables to represent individual skills to improve the construct validity of studies considering this impact [310].

Finally, a variable of the selected population that may interact with the treatment that we did not analyze is the *language skill* of participants. Arguably, skills in the English language influence the ability to comprehend and process the experimental objects and, therefore, may impact the response variables. We were unable to measure this variable properly given that all participants scored the same on the Common European Framework of Reference for Languages (CEFR) [323] (i.e., non-native, fluent English speakers). While the threat is minimized in our study due to the comparable language level of participants, future studies should develop measurement instruments for this construct and involve this variable in such causal queries.

5.4.4 External validity

The main threat to the external validity of this study is the *interaction of setting and treatment* as the size and the complexity of the selected specifications, despite being sampled from a real-world data set, might not be representative of the industrial practice. Using Google Docs as the modeling tool is not fully representative of real-world practices. Given that it was appropriate and sufficient for the experimental task, however, renders this as an opportunity for improving the realism of the experiment in future studies rather than a threat to validity.

There may be the threat of *interaction of selection and treatment*, as some participants reported no modeling experience or training. These deficiencies might influence the results and render a subset of the participants as non-representative of our target population. We attempted to mitigate this threat via comprehensive instructions and including a warm-up phase in the experiment.

6 Conclusion

Requirements quality research lacks empirical evidence and research strategies to advance beyond proposing and following normative rules with unclear impact [41] to better understanding and solving problems relevant to practice [9, 37]. In the scope of our study, we conducted a controlled experiment on the impact of requirements quality defects on subsequent activities. We demonstrated a method of evaluating data collected through a controlled experiment using a crossover design with Bayesian

data analysis. We showed the impact (1) of requirements quality defects varies and (2) may be mediated by context and confounding factors. The part of our study that serves as a conceptual replication strengthens the claims of the re-analyzed original study [34, 81] that passive voice only has a slight impact on missing associations from domain models.

We can confidently support the recommendation of SE researchers to adopt Bayesian data analysis to improve causal reasoning and inference [48, 78, 226], which will propel requirements quality research. This shift requires focusing on problems such as scrutinizing the explicit causal assumptions of a DAG, visualizing requirements quality impact, evolving prior knowledge about their impact, and comparing models concerning their predictive power.

We envision that adopting sophisticated statistical tools like Bayesian data analysis and the focus of empirical studies on investigating the impact of requirements quality defects will steer requirements quality research in a relevant and effective direction. Explicit causal assumptions and sophisticated data analyses will produce empirical evidence which can be more easily synthesized to more reliable and generalizable conclusions [42] in a common framework [40]. We hope that the documentation of this study inspires fellow researchers to adopt our method and tools for replication.

Paper VIII

Replications, Revisions, and Reanalyses: Managing Variance Theories in Software Engineering

Abstract

Variance theories quantify the variance that one or more independent variables cause in a dependent variable. In software engineering (SE), variance theories are used to quantify—among others—the impact of tools, techniques, and other treatments on software development outcomes. To acquire variance theories, evidence from individual empirical studies needs to be synthesized to more generally valid conclusions. However, research synthesis in SE is mostly limited to meta-analysis, which requires homogeneity of the synthesized studies to infer generalizable variance. In this paper, we aim to extend the practice of research synthesis beyond meta-analysis. To this end, we derive a conceptual framework for the evolution of variance theories and demonstrate its use by applying it to an active research field in SE. The resulting framework allows researchers to put new evidence in a clear relation to an existing body of knowledge and systematically expand the scientific frontier of a studied phenomenon.

Keywords: Research Synthesis, Causal Inference, Variance Theories, Theory Evolution

1 Introduction

Software engineering (SE) research aims to support SE practice in a process referred to as knowledge translation [94]. It consists of knowledge *creation*, which includes gathering empirical evidence in primary studies, and knowledge *application*, i.e., providing this evidence to its target audience. However, primary studies do not provide convincing decision support to practitioners on their own [9, 84, 324, 325]. Hence, an imperative step between the creation and application of knowledge is its *synthesis*.

Research synthesis is a "collective term for a family of methods that are used to summarize, integrate, combine, and compare the findings" [326] of individual pieces of evidence and aims to infer more generally valid conclusions.

One product of synthesizing quantitative research is a *variance theory*. Variance theories estimate the variance of a dependent variable in relation to one or more independent variables [89]. In SE research, variance theories provide decision support by quantifying the strength of the effect of, for example, new tools, different technologies, or human factors on key performance indicators of the SE process. For example, the synthesis of 27 primary studies about the effect of test-driven development (TDD) on code quality and developer productivity by Rafique and Mišić determined that "TDD has a small positive effect on quality but little to no discernible effect on productivity" [327]. However, research synthesis in SE is primarily limited to meta-analysis [94]. While certain forms of meta-analysis excel at synthesizing evidence from quantitative studies, they only produce usable results under certain conditions like homogeneity of the pieces of evidence [328]. Current research synthesis practices fail to accommodate more complex relationships between individual pieces of evidence, like deviating hypotheses or the usage of different analysis methods. Consequently, the validity of variance theories produced by these research synthesis practices is limited, and they may not offer the intended decision support to practitioners.

In this work, we propose a framework for managing variance theories in SE that extends beyond current meta-analysis practices. We formally define quantitative, empirical evidence and an evolution framework that specifies how two pieces of evidence relate to each other. We demonstrate the framework by applying it to an active field of SE research to show how it can guide SE research toward more coherent and productive research agendas.

We aim to support two scientific use cases. First, our framework helps researchers to position new pieces of evidence in relation to an existing body of knowledge. As such, the framework provides a terminology to frame how new studies advance the body of knowledge with respect to existing ones. New evidence can be classified as either a replication, revision, or reanalysis, and the framework supports deciding whether this new evidence strengthens or challenges the body of knowledge. Second, our detailed application demonstrates how to apply the framework to systematically review literature containing quantitative, empirical evidence. By framing all empirical, quantitative studies investigating one phenomenon using the framework, the evolution of a variance theory about that phenomenon becomes tangible. This reveals the current scientific frontier of quantitative studies on a phenomenon and can inform future study design. Overall, our initiative aims to broaden the perspective on research synthesis to obtain more valid variance theories from SE research.

Data Availability All study data is publicly available in our replication package [329].

2 Related Work

2.1 Research Synthesis

The purpose of any endeavor in SE research is to support SE practitioners [330]. This requires translating knowledge created in research to practice [331, 332]. However, previous research has shown that singular empirical studies do not provide convincing evidence to practitioners [324]. A single study cannot compete with the beliefs of practitioners, as the findings are limited to the context of the respective primary study [325]. Consequently, Miller advocated that the field of SE research "needs to move to a portfolio of empirical studies (on a single research hypothesis) being the norm rather than the currently unconvincing 'one-off', normally laboratory-based, studies that currently dominate the research literature" [333]. These portfolios of replications (also called *families of studies or experiments* [334]) strengthen the validity of research findings and generate more reliable and general conclusions [85].

While portfolios of replications strengthen the *knowledge creation* phase of knowledge translation [94] by increasing the validity of drawn conclusions, they do not necessarily serve the *knowledge application* phase. Indeed, portfolios of replications complicate maintaining an overview of all relevant primary studies, and contradicting results are difficult to harmonize [335]. This necessitates the aggregation and integration of results from primary studies [42], commonly referred to as *research synthesis*. Research synthesis describes "methods that are used to summarize, integrate, combine, and compare the findings" [326] of primary studies with similar goals. Shepperd summarized the synthesis process in five steps [336].

1. **Problem formulation**: specifying a research question, usually about the influence of one independent on one dependent variable
2. **Locating evidence**: searching literature that contains evidence about the research question
3. **Appraising evidence quality**: applying quality inclusion criteria
4. **Evidence synthesis and interpretation**: extracting relevant data and performing the research synthesis that infers about the initial research question based on the accumulated evidence
5. **Reporting**: disseminating the results in a research report

For step four, SE research has adopted several research synthesis methods from more mature experimental disciplines [94, 256, 336–339]. Among the most popular is the narrative synthesis [340], a textual summary of research findings, often criticized for its lack of a systematic approach [341]. The more systematic vote-counting classifies the effect of an independent variable on a dependent variable on ordinal scales, e.g., depending on its sign (i.e., positive, neutral, or negative effect) [256] or

its strength of evidence (e.g., third party claims, circumstantial evidence, and strong evidence) [342]. A histogram of classified findings from primary studies indicates the tendency of the effect observed by a portfolio of replications.

A more quantitative approach to synthesize results from controlled experiments is often called meta-analysis [335]. Various forms of meta-analysis exist and are adopted in SE research [340]. The two approaches considered state-of-the-art are aggregate data (AD) meta-analysis and stratified independent participant data (IPD-S) meta-analysis [340]. AD meta-analysis pools the calculated effect sizes of primary studies together and calculates an overall effect size of the independent on the dependent variable [335], often represented in a Forest plot [343]. IPD-S meta-analysis pools together the raw data from all experiments of the primary studies and analyzes this data directly [344], but retains information about the belonging of each data point to the experiment it originated from to account for between-study variance [340]. IPD-S meta-analysis is commonly regarded as the gold standard for research synthesis [345] but is rarely applied in SE research [340]. Narrative synthesis and AD meta-analysis dominate SE research [340], though meta-analysis remains uncommon in general [346]. Some examples include synthesizing evidence about defect prediction [347, 348] and test-driven development [327].

Research syntheses of quantitative evidence produce *variance theories*, one of three commonly discussed types of theories [89], about a phenomenon under study. While *theories for understanding* organize entities into meaningful categories and *process theories* explain how something is happening [89], variance theories quantify the strength of the effect of an independent on a dependent variable [89]. Because they are a product of synthesis, variance theories have greater validity than a single piece of evidence [66]. The quantification of an effect strength offers decision support to practitioners, e.g., when deciding whether to adopt a technique like TDD.

2.2 Shortcomings of the State-of-the-Art in Research Synthesis

Literature has acknowledged several shortcomings of the state-of-the-art of research synthesis in SE [326, 336, 346].

Dealing with heterogeneity Primary studies involved in research synthesis are subject to heterogeneity, i.e., differences between studies investigating the same phenomenon [341]. An unclear study design, incomplete sample selection protocol, or unreflected operationalization of latent concepts in SE research often obscure critical factors [256, 336] like prior knowledge of participants, their experience, or intrinsic motivation. If these factors have a significant influence on the phenomenon under investigation, traditional meta-analysis techniques will produce inconclusive results.

When the factors causing these differences are well understood and recorded in the data collection process, an appropriate synthesis method can account for them [256,

328], which benefits the validity of the conclusion [85]. However, traditional meta-analysis techniques are limited to studies investigating the relationship between exactly two variables (i.e., the effect of one independent on one dependent variable) [340, 349]. Yet, phenomena in SE research can rarely be isolated into a two-variable relationship, as human [256] and other context factors [350] usually interact with the phenomenon. If the body of primary studies is very large, these context factors may be analyzed by conducting a meta-analysis of meta-analyses, as shown by Harris et al. in a study with 136 primary studies clustered into 31 meta-analyses [351]. Similarly, Rafique and Mišić conducted a meta-analysis of 27 primary studies about the effect of TDD on developer productivity and code quality, where the amount of evidence allowed a subgroup analysis [327]. This subgroup analysis revealed that the population from which study participants were drawn (practitioners vs. students) mediated some of the observed effects. However, meta-meta-analyses or subgroup analyses are no reliable tool to address the presented shortcoming for the following reasons. Firstly, the required amount of empirical evidence is mostly unavailable in SE research [334]. Secondly, such analyses are only eligible if the synthesized primary studies report additional information, such as the sample demographics. Finally, these analyses only add a hierarchical complexity to the research synthesis endeavor but do not systematically deal with more complex relationships between variables [352]. Consequently, classical research synthesis methods like meta-analysis only apply to a set of homogeneous primary studies. They fail to incorporate that causal assumptions may evolve and become more complex than simple two-variable relationships.

Limited to experimental studies Additionally, many synthesis methods are limited in the types of primary studies they can integrate. Traditional meta-analyses from medical research are constrained to controlled experiments [335]. This excludes, by design, quantitative evidence from observational studies [352] and all qualitative studies [353]. However, this discards evidence about complex phenomena that might not be studied using controlled experiments alone [353]. On the one side, observational studies are less invasive to the actual software development context but can still yield reliable, causal inferences using appropriate methods [55, 226]. Conversely, qualitative studies have shown to capture richer context information, especially in medical and social sciences [337], which has also been acknowledged in SE research [326].

Static and retrospective Finally, research syntheses receive critique for being static when they "should be updated on completion of a study to place their result in context" [345]. A continuous approach to meta-analyses [341] would avoid that included studies become outdated by the time of synthesis [354]. Furthermore, research synthesis is predominantly conducted retrospectively, i.e., it aggregates publications published prior but rarely guides the design of future ones [355]. Instead of opportunis-

tically conducted meta-analysis [340], SE research requires prospective synthesis initiatives as also called for in other disciplines like medical research [344].

3 Goal and Method

We aim to address the shortcomings outlined in Section 2.2 with a strategy to facilitate more coherent, systematic research. To this end, we composed a framework from the scientific practices of three branches of research evolution. First, we surveyed the existing body of knowledge on replication studies in SE [63, 83] and their synthesis in the form of meta-analysis [328, 336, 338]. Second, we reviewed techniques from statistical causal inference [54], particularly model comparison for causal inference from observational studies [53, 77, 226]. Finally, we surveyed references about the evolution of statistical practices [48, 55, 246]

A regular validation of the framework would require applying it in practice, i.e., comparing the evolution of variance theories with and without the framework. However, this kind of validation is not feasible at this point as it would require its prior adoption. Instead of developing the framework, applying it in a controlled manner, and validating it with collected data, we sought to involve the SE research community at an early stage of proposing this framework already to allow for critical discussions and contributions. Hence, we opted for a constructive validation of the proposed framework in a focus group setup. We presented the framework at the annual meeting of the International Software Engineering Research Network (ISERN). The ISERN community[1] consists of experts on empirical software engineering methodologies and their applications and meets as part of the Empirical Software Engineering International Week.[2] In our focus group session, we discussed the eligibility of the framework to guide future empirical research and allow effective aggregation of evidence. Based on the discussion, we revised the framework and the boundary conditions of its applicability.

4 Conceptual Framework

Our framework for managing variance theories consists of a definition of evidence in Section 4.1 and a flowchart describing the evolution of evidence in Section 4.2. Section 4.3 describes the implications of the framework on research synthesis practices in SE.

[1] https://isern.iese.de/

[2] https://conf.researchr.org/series/esem

4.1 Evidence

We define a piece of empirical, quantitative evidence e as a tuple $e := E(h, d, m)$ consisting of three components.

- **Hypothesis** h: A hypothesis consisting of variables and (assumed) causal relationships between those variables. For example, $h_1 := x \rightarrow y$ defines hypothesis h_1 as variable x causally influencing variable y.
- **Data** d: A record of observations of all variables contained in h. An eligible dataset d_1 for h_1 requires observation for both variables x and y.
- **Method** m: An analysis method that processes the data d under the hypothesis h to produce a conclusion.

Hypotheses are networks of variables and relationships among them. As such, they can be visualized via directed, acyclic graphs (DAGs), as shown in Figure 1. In these DAGs, nodes represent variables, and directed edges represent assumed causal relationships. Figure 1a shows a graphical representation of a simple, two-variable hypothesis. More often, though, manuscripts present such simple hypotheses textually. This often takes the form of a verbose null hypothesis, e.g., "There is no significant difference in values of y for different values of x." More complex hypotheses involving more variables and relationships like Figures 1b and 1c require graphical representation but are rare in SE research [54].

The two colored nodes in the four DAGs represent the main *phenomenon* of interest, i.e., the independent variable (colored red) and the outcome or response variable (colored cyan). The goal of a piece of empirical, quantitative evidence is to estimate the average causal effect (ACE) of the main independent variable(s) on the dependent outcome variable. Additional variables (colored grey) may be relevant to the hypothesis but not part of the main phenomenon under study. Therefore, the same phenomenon of interest can be involved in multiple hypotheses.

All analysis methods m require deriving a *statistical model* from the causal model, i.e., the hypothesis h [55]. A statistical model typically consists of a regression model, i.e., a specified, often linear relationship between one or more predictors and the outcome variable. In the case of simple, two-variable hypotheses, this boils down to regressing the outcome on the only predictor. For example, the statistical model derived from h_1 in Figure 1a would be $y \sim x$. In the case of more complex hypotheses, one must select the subset of independent variables. This subset should maximize the precision of the estimation of the effect of x on y and, on the other hand, ensure that the causal effect is not confounded. For example, the statistical model derived from h_3 shown in Figure 1c would be $y\ x+z$ where z is included to de-confound the effect of x on y. The statistical model of h_4 shown in Figure 1d would be $y\ x$ since including z, called a *collider*, would confound x on y [55]. The subset that deconfounds the causal effect is commonly called the *adjustment set* [77] and can be

(a) Basic hypothesis h_1

(b) Revision h_2 increasing the effect estimate precision

(c) Revision h_3 to deconfound the effect estimate

(d) Revision h_4 specifying a collider

Figure 1: DAGs representing a hypothesis and three revisions

determined by applying a systematic procedure called the *backdoor adjustment* [53]. Explaining this procedure in detail goes beyond the scope of this manuscript but is well-explained in existing literature [53, 55, 77].

The eligibility of analysis methods depends on the complexity of the hypothesis and the properties of the variables. For simple, two-variable hypotheses consisting of one independent and one dependent variable, most scholars resort to null hypothesis significance tests (NHSTs) like the Student's t-test or its variants. Here, the choice depends on the normality of the dependent variable and whether the data is paired or not. For more complex hypotheses involving more than one independent variable, scholars tend to apply linear regression models with multiple predictors.

Applying the analysis method m to the data set d based on the causal model implied by the hypothesis h produces the piece of evidence e that offers a conclusion. The nature of the conclusion depends on the analysis method. For example, NHSTs propose a p-value that scholars commonly compare with an arbitrary significance level α to determine whether the independent variable evokes a statistically significant difference in the dependent variable. For linear regression models, the conclusion takes the form of coefficients (e.g., β_x in Figure 1a, representing the strength of the impact of x on y). From these coefficients, one can additionally calculate confidence intervals for each independent variable. If the confidence interval of a variable is not consistent with 0, i.e., it does not intersect 0, then the variable is considered to have a significant impact on the dependent variable.

4.2 Evolution of Evidence

Variance theories emerge from the synthesis of multiple pieces of empirical, quantitative evidence, which increases their validity and abstracts from passing trends [66]. To accommodate research synthesis that goes beyond the meta-analysis of a homogeneous set of primary studies [256], the relationship between two pieces of evidence needs to be clear. Figure 2 visualizes the types of evolution of empirical evidence, i.e., the possible relationship between pieces of evidence. Starting from an initial piece of evidence $e_1 = E(h_1, d_1, m_1)$, we consider three types visualized as paths in Figure 2 and explained next.

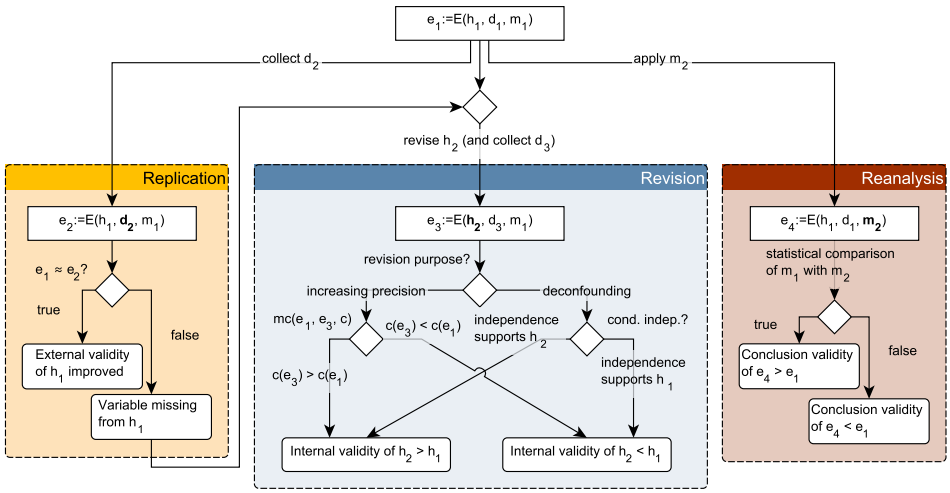


Figure 2: Framework describing the evolution of quantitative, empirical evidence

4.2.1 Replication

The most commonly known evolution type of empirical evidence in SE research is through *replication* (left branch colored yellow in Figure 2). A replication is a type of study that offers diagnostic evidence about a previous empirical study [85]. As such, a replication subscribes to the same causal hypothesis h_1 and uses the same analysis method m_1 but collects a different data set d_2 to produce a new piece of evidence $e_2 := E(h_1, d_2, m_1)$. The conclusion derived from the replication e_2 is compared with the conclusion of the original piece of evidence e_1 to check for agreement. Checking for agreement depends on the nature of the conclusion that the analysis method m_1 produces. If m_1 is a type of hypothesis test that produces a p-value, this check is referred to as an *aggregation of p-values* [340] via Fisher's or Stouffer's method [356]. If m_1 is a type of regression model that produces confidence intervals of coefficients, then the check boils down to assessing whether the confidence intervals overlap in a Forest plot [343], or AD or IPD-S meta-analysis techniques [340]. If the conclusions agree, the external validity of the causal claim h_1 is improved as the replication shows that the conclusion of e_1 also holds in a different context d_2. However, in case the conclusions disagree, SE literature offers little advice on how to relate these results. The disagreeing conclusions indicate that at least one variable that would explain the difference between e_1 and e_2 is missing from h_1, requiring a *revision* of the original hypothesis.

4.2.2 Revision

A less commonly discussed evolution of empirical evidence in SE research is through *revision* of a hypothesis (middle branch colored blue in Figure 2). Revising hy-

pothesis h_1 means proposing a competing network of variables, hypothesis h_2, that supposedly explains the phenomenon under study—which produced the data d_1 and d_2—better and, therefore, has greater internal validity. The competing network can include new or discard existing variables or may—alternatively or additionally—propose different causal relationships between variables. Only the variables belonging to the phenomenon under study need to remain included. Otherwise, the new hypothesis pertains to a different phenomenon. Figures 1b to 1d visualize revisions of Figure 1a as they contain the variables of the main phenomenon under study (x and y) but include an additional variable (z) and different relationships.

Revisions can serve two different purposes [77]. The first purpose is to increase the precision of estimating the effect β_x of x on y. For example, involving an additional, independent variable z with an assumed causal relationship $z \rightarrow y$ may increase the precision of the estimate of the average causal effect [77]. Figure 1b visualizes such a revised hypothesis h_2. The second possible purpose of a revision is to de-confound the estimation of the effect β_x of x on y [55]. This is particularly relevant to phenomena studied in observational, not experimental, settings where the independent variable of interest can be influenced by factors other than random assignment. A confounder could be a common cause as visualized in Figure 1c where variable z impacts both x and y, therefore biasing the direct effect of x on y. The adjustment set of this hypothesis includes z as a predictor of y to de-confound the effect of x on y [55].

The disagreeing conclusions from a replication but also emerging qualitative evidence (represented by the direct arrow from e_1 to e_3 in Figure 2) may trigger a revision. For example, a qualitative study might suggest that the variable z also influences y even before observing disagreeing conclusions from replications. Proposing a new hypothesis h_2 may also require collecting a new data set d_3 if h_2 contains variables not recorded in d_1. In the abstract example, a new data set d_3 that records both z in addition to x and y is necessary.

Once appropriate data are available, the competing hypotheses are evaluated by model comparison. The type of comparison depends on the purpose of the revision. To evaluate a revision aiming at increasing the precision, the out-of-sample predictive power of the two models is compared (abbreviated as `mc` in Figure 2) via an appropriate criterion (abbr. as `c`). Metrics like the Akaike Information Criterion (AIC) [357] or leave-one-out cross-validation (LOO) may be applied depending on the analysis method [358, 359]. These metrics assign scores to competing hypotheses and infer which predicts the observed data best. The model with the greatest predictive power is assumed to be more internally valid.

To evaluate a revision aiming at de-confounding, testable implications in the form of *independencies* and *conditional independencies* are derived from the hypotheses [55]. For example, according to h_3 in Figure 1c, both x and y depend on z. Conversely, h_1 in Figure 1a does not include this claim and implies that x and y are independent of z. Additionally, h_3 implies that the strength of the ACE of x on y

changes when conditioning on z via deconfounding, which h_1 does not imply. Correlational analyses on the available data set d_3 can confirm or refute these assumed independencies [55]. Comparing the statistical model $y \sim x$ (derived from h_1) with $y \sim x + z$ (derived from h_3) produces two estimates of the ACE of x on y. If the ACE is the same, then the effect of x on y is independent of z and h_1 represents the causal relations of the phenomenon under study better. If the ACE is different, then the effect depends on z, and h_3 is more valid. Consequently, the internal validity of h_3 exceeds the one of h_1 and can be considered the currently superior causal model to explain the phenomenon under investigation.

While both purposes of revisions aim to strengthen the internal validity of a hypothesis, the respective evaluation that decides the comparison is not interchangeable. Model comparison used to determine the hypothesis with the greater out-of-sample predictive power is not fit when aiming to deconfound a hypothesis, as a confounded hypothesis may very well exhibit a greater predictive power than a deconfounded one [55]. Consequently, the distinction of purpose when conducting a revision is imperative for the choice of evaluation method.

The hypothesis that currently shows the greatest internal validity should be the one that future studies should subscribe to. This means that all future studies investigating the phenomenon of x and y should record all variables involved in the hypothesis that are part of the adjustment set.

4.2.3 Reanalysis

The least commonly discussed evolution of empirical evidence in SE research is the *reanalysis* of existing data (right branch colored red in Figure 2). Reanalysis—sometimes also referred to as a "test for robustness" [85]—describes the application of a different analysis method m_2 to the same data d under the same causal assumptions h [83]. In special cases, however, reanalysis may also be necessitated by a revision. For example, extending a hypothesis to include two instead of one independent variable will make analysis methods that only operate with one independent variable (e.g., a t-test) ineligible and necessitate more complex ones (e.g., a linear model).

Reanalyses are mostly driven by adapting more advanced methods from other disciplines (e.g., statistics, or medical research). One instance of this type of evolution is the ongoing endeavor to abandon simple NHSTs for more advanced Bayesian data analysis [48]. Reanalyses increase the conclusion validity of the evidence by revising statistical assumptions [47]. The decision of which analysis method to prefer over another is often based on intricate statistical comparisons [55], which SE researchers usually adapt and do not conduct themselves.

4.3 Implications on Research Synthesis

The framework for systematic evolution of variance theories has the following properties that address two of the three shortcomings in research synthesis described in Section 2.2.

1. **Causal**: The framework takes a causal perspective to research synthesis by adding *revisions* as a type of evolution next to *replications*. This allows systematically addressing the heterogeneity of evidence conclusions by evolving hypotheses beyond two-variable relations.

2. **Open to different study types**: By abandoning the constraint of simple, two-variable relations, the framework for research synthesis also opens up to other study types. More complex networks of variables can adequately represent causal assumptions from an observational study [226]. Additionally, the framework allows qualitative studies to contribute to the evolution of quantitative variance theories by triggering revisions.

How to address the remaining, third shortcoming in research synthesis will be discussed in Section 6.3.

5 Application

We apply our framework to the research field of requirements quality. Section 5.1 introduces the relevant background about the research field. Section 5.2 demonstrates the application of the framework to a set of primary studies from this field. Finally, Section 5.3 retraces omitted steps between existing pieces of evidence to show how the framework aids in systematically evolving evidence. For readability, Sections 5.2 and 5.3 omit details on statistical operations to focus on the evolution on a conceptual level. Details on the statistical operations can be found in our replication package [329].

5.1 Requirements Quality Research

Requirements quality research [6] is concerned with identifying how properties of requirements artifacts [41] (e.g., passive voice, sentence length) impact properties of subsequent software development activities (e.g., correctness of implementing, completeness of testing) that use these requirements artifacts as part of their input [92]. Variance theories in this field quantify the strength of the effect that certain properties of requirements artifacts have—e.g., whether longer requirements sentences reduce the correctness when implementing source code [90]. Variance theories inform requirements writing guidelines by indicating to practitioners whether addressing these properties is worth it [40]—e.g., whether reducing the sentence length of

requirements should be enforced. One such property of requirements artifacts commonly discussed in requirements quality literature is the use of *passive voice* in natural language (NL) requirements. The following two versions of the same requirement illustrate the difference.

- **Active**: The system shall obtain all transaction details from the Statement Database.
- **Passive**: All transaction details shall *be obtained* from the Statement Database.

The passive formulation omits the actor ("the system") from the requirements specification, obscuring who is allowed to perform the action. The observed drop in informativeness caused textbooks to advise against using passive voice in NL requirements artifacts [33]. However, the lack of empirical evidence for this claim, and even evidence against it [172] attracted experimental studies investigating its impact.

5.2 Evolution of Evidence

We are aware of three studies that empirically investigate the impact that the use of passive voice in NL requirements has on the domain modeling activity [34, 81, 82]. The three studies arrive at the variance theory that passive voice has a slight negative impact on the completeness of domain models.

These three studies contribute four pieces of evidence as one of the studies produces two separate pieces of evidence [82]. Table 1 lists these four pieces of evidence indexed as e_1-e_4. The figure in the leftmost column of the table visualizes the relationship between the pieces of evidence as a version control graph (VCG). The color of each node in this graph reflects the evolution type that this piece of evidence represents in relation to its predecessor based on the colors in Figure 2. For example, e_3 is a replication of e_1, hence the box is yellow.

Table 1: Evolution of the variance theory on the impact of passive voice on domain modeling

VCG	Evolution Type	h	d	Analysis Method m	Conclusion
	Original Study [34]	h_1	d_1	m_1: Mann-Whitney U test	$p = 0.001$
	Revision & Reanalysis [81]	h_2	d_1	m_2: Bayesian model	$[-0.17, \sim 0.49, +0.34]$
	Replication [82]	h_1	d_2	m_1: Wilcoxon signed-rank t.	$p = 0.025$
	Revision & Replication [82]	h_3	d_2	m_2: Bayesian model	$[-0.25, \sim 0.30, +0.45]$

5.2.1 Original study

To the best of our knowledge, Femmer et al. contributed the first piece of empirical, quantitative evidence e_1 about the impact of passive voice on domain modeling. In their study [34], they investigated the research question "Is the use of passive sentences in requirements harmful for domain modelling?" In particular, they studied whether the use of passive voice in NL requirements sentences changed the number

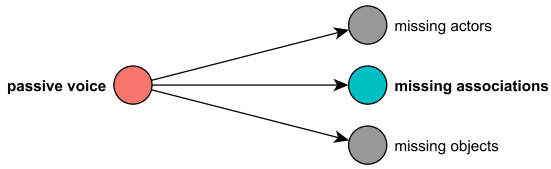


Figure 3: Hypothesis h_1 investigated by Femmer et al. [34]

of missing actors (Act^-), associations (Asc^-), and domain objects (Obj^-) from domain models derived from them. Figure 3 visualizes their three hypotheses in one DAG. In this demonstration, we focus on the impact of passive voice on the number of missing associations, i.e., $h_1 : passive \rightarrow Asc^-$.

Femmer et al. conducted a parallel-design controlled experiment with 15 university students as participants. These participants were randomly divided into either the control or the treatment group. Each participant received seven natural language requirements sentences that were either written in active voice (for the control group) or passive voice (for the treatment group). The experimental task was to generate a domain model from each requirement. Then, the authors of the study counted the number of missing actors, associations, and domain objects from the resulting domain models via comparison to a gold standard that they produced. The resulting data set d_1 consequently consisted of 105 ($= 15 \times 7$) data points recording the group (active or passive), the number of missing actors, associations, and domain objects, as well as several demographic factors like program and experience.

To produce evidence e_1, the authors applied a Mann-Whitney U test to evaluate the three hypotheses.[3] The conclusion of evidence $e_1 = E(h_1, d_1, m_1)$ is $p = 0.001$, i.e., the NHST suggests that the use of passive voice has a statistically significant impact on the number of missing associations from a domain model. The authors report an effect size calculated via Cliff's δ of 0.75 [34], which suggests a strong effect on per-participant aggregate level. Our re-calculation on domain model level amounts to an effect size of only 0.35, which is considered of medium strength.

5.2.2 Follow-up Study 1

Frattini et al. performed a follow-up study, taking a second look at the drawn conclusions [81] to produce evidence e_2. In this study, they reused the collected data d_1, but both propose a new causal hypothesis h_2 and also applied a different analysis method m_2. The revised causal hypothesis h_2 contained the following additional assumptions which are also visualized in Figure 4:

1. If either actors or domain objects are missing, associations are more likely to be missing as well as one of the nodes involved in the edge is not present.

[3] In the original study [34], the authors use the per-participant aggregate of missing entities as a response variable, which we do not to stay true to the hypothesis. The conclusion remains the same.

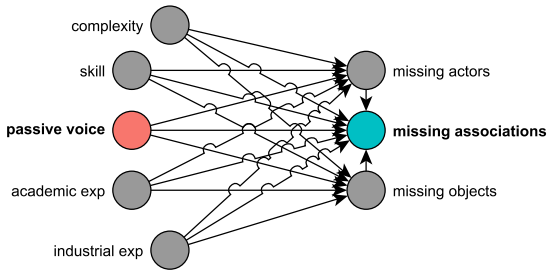


Figure 4: Hypothesis h_2 revised by Frattini et al. [81]

2. The general skill of a participant may influence their performance.

3. The complexity of a requirement may affect how easy it is to miss an actor, association, or domain object.

4. The academic and industrial experience of a participant may influence their performance.

The first additional assumption added two relationships to the DAG, and the second to fourth added new variables. Data set d_1 already recorded values for these additional variables. In a survey prior to the experiment, participants reported their academic and industrial experience on an ordinal scale with four categories (i.e., no experience, up to 6 months, 6 to 12 months, and more than 12 months).

The number of predictors involved in a statistical model derived from the causal model h_2 made the analysis method m_1 ineligible for producing a conclusion, as the Mann-Whitney U test only operates with one predictor but the statistical model requires several. Instead, the authors followed the advice to adopt Bayesian data analysis [48, 55] as their method m_2. The resulting statistical model used the available demographic factors as predictors and models the requirements' complexity and participants' skill as random effects via the IDs of the requirements and participants.

Applying a Bayesian data analysis m_2 under the causal assumptions encoded in h_2 to the existing data d_1 produced a marginal probability distribution of the impact of passive voice on the number of missing associations. Evidence $e_2 = E(h_2, d_1, m_2)$ concludes that the use of passive voice in NL requirements leads to more missing associations in about 34%, to fewer in 17%, and to an equal amount in 48% of all cases on average. Hence, e_2 agrees with e_1 regarding effect strength [34], but the Bayesian data analysis m_2 cautions about the significance of the effect suggested by m_1. The authors of the follow-up study referred to literature for the superiority of m_2 over m_1 but did not validate whether h_2 was more valid than h_1.

5.2.3 Follow-up Study 2

Frattini et al. performed a second follow-up study where they conducted their own experiment [82]. In this crossover-design experiment involving 25 participants, mostly from industry, the experimental task was similar, but the material differed (i.e., it used four different NL requirements). Additionally, due to the crossover design, participants were not divided into a treatment and control group but received all levels of the treatment (just in different orders). Also, the experiment involved another treatment (the use of ambiguous pronouns), which represents a different phenomenon and, hence, is irrelevant in this context. The resulting data set d_2 consisted of 100 ($= 25 \times 4$) data points.

This second follow-up study [82] produced two pieces of evidence. First, the authors performed a replication of e_1 by applying the same analysis method m_1 under the same causal assumptions h_1 to the new data d_2. Since the data is paired due to the crossover design, the configuration of m_1 changed slightly (making it necessary to use a Wilcoxon signed-rank test instead of a Mann-Whitney U test), but the inferential process remains comparable. Evidence $e_3 = E(h_1, d_2, m_1)$ concludes that passive voice has a statistically significant impact on the number of missing associations with $p = 0.025$. Although the p-values of e_1 and e_3 differ, both reject the null hypothesis of no impact under the common level of significance $\alpha = 0.05$ and, therefore, suggest the same conclusion.

Second, the authors produced another piece of evidence e_4 that both revises the causal assumptions of e_2 (i.e., replaced h_2 with h_3) and performs a Bayesian data analysis m_2 on the new data d_2. The revised hypothesis h_3, visualized in Figure 5, made several changes:

1. Missing actors and domain objects behave the same and, hence, can be aggregated to the number of missing *entities* in the domain model.

2. The number of missing associations may further be impacted by the education, task experience, and domain knowledge of a participant.

3. The amount of time that a participant took to generate the domain model may influence its completeness.

Evidence $e_4 = E(h_3, d_2, m_2)$ concludes that the use of passive voice in NL requirements leads to more missing associations in about 45%, to fewer in 25%, and to an equal amount in 30% of all cases on average. e_4 agrees with e_2 in that the impact of passive voice on the number of missing associations from domain models is not strictly negative, as the likelihood of missing an association remains below 50%. However, e_4 suggests that the negative impact is more likely than assumed by e_2 (45% instead of 34%). Since both pieces of evidence were produced under different causal assumptions (h_2 in e_2 and h_3 in e_4), these numbers are not directly

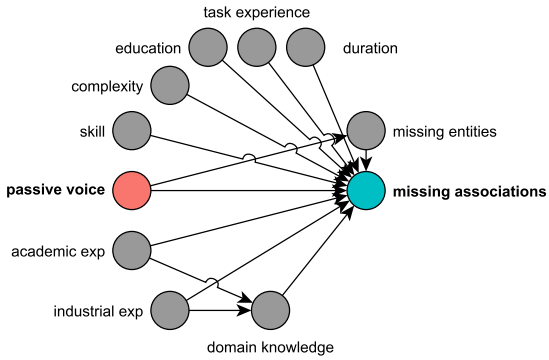


Figure 5: Hypothesis h_3 revised by Frattini et al. [82]

comparable. Again, the authors of this follow-up study did not validate whether h_3 was more valid than h_2.

Insight 1

The research synthesis produced a variance theory suggesting that the use of passive voice in NL requirements has a moderate effect on the number of associations missing from domain models. However, the application of the framework reveals that several transitions (i.e., replacing hypotheses) were not validated. Hence, their contribution to the validity of the variance theory remains questionable.

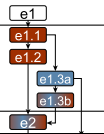
5.3 In-depth Research Synthesis

The four pieces of evidence e_1-e_4 from the three studies [34, 81, 82] were produced without an explicit framework for managing variance theories. This caused several steps in the evolution of the variance theory about the impact of passive voice on domain modeling to be of unclear validity. Evidence e_2 conflates a revision with a reanalysis, i.e., it replaces both the hypothesis (h_2 for h_1) and the analysis method (m_2 for m_1). The differing conclusions obtained from e_2 can, therefore, not be traced clearly to either of these two changes.

In the following in-depth analysis, we will zoom in on the step of evolution between the original evidence e_1 and the evidence from the first follow-up study e_2. We disentangle the sub-steps according to the proposed framework, which allows us to retrospectively assess the evolution performed by Frattini et al. [81] but also guide future contributions to this variance theory. Table 2 visualizes the deconstructed pieces of evidence between e_1 and e_2. The rows between e_1 and e_2 show the disentangled sub-steps between the two pieces of evidence. The rows after e_2 show additional

sub-steps that would have been the more valid path to pursue had the authors of the follow-up study [81] disentangled the sub-steps.

Table 2: Decomposed steps between e_1 and e_2

VCG	Evolution Type	Ref.	Hyp.	Analysis Method	Conclusion
	Original Study	[34]	h_1	m_1: Mann-Whitney U test	$p = 0.001$
	Reanalysis		h_1	$m_{1.1}$: linear model	$ci = [0.23, 0.84]$
	Reanalysis		h_1	m_2: Bayesian model	$[-0.15, \sim 0.34, +0.51]$
	Revision		h_{2a}	$m_{1.1}$: linear model	$ci = [0.17, 0.77]$
	Revision		h_2	$m_{1.2}$: linear mixed model	$ci = [-0.17, 0.82]$
	Revision/Reanalysis	[81]	h_2	m_2: Bayesian model	$[-0.17, \sim 0.49, +0.34]$
	Revision		h_{2c}	$m_{1.2}$: linear mixed model	$ci = [0.03, 0.92]$
	Reanalysis/Revision		h_{2c}	m_2: Bayesian model	$[-0.14, \sim 0.47, +0.39]$

5.3.1 *Reanalysis* $e_{1.1}$

The frequentist NHST m_1 is not directly comparable to a Bayesian data analysis m_2. Hence, an intermediate step is necessary. A simple reanalysis is to replace the Mann Whitney U test m_1 with a linear regression model $m_{1.1}$. At the core, a linear model that regresses the rank-transformed outcome variable on a single predictor is equivalent to the Mann Whitney U test [360]. Fitting a linear model $Asc^- \sim passive$ to the data d_1 produces a coefficient of $\beta_{passive} = 0.53$ and a 95% confidence interval of $ci_{e_{1.1}}(passive) = [0.23, 0.84]$. The confidence interval is not consistent with 0, i.e., it does not contain 0. Therefore, the conclusion of $e_{1.1}$ agrees with the conclusion of e_1 in its suggestion that the use of passive voice has a statistically significant impact on the number of missing associations from domain models.

5.3.2 *Reanalysis* $e_{1.2}$

Replacing the linear regression model $m_{1.1}$ with a Bayesian data analysis m_2 produces evidence $e_{1.2}$, a strict reanalysis of e_1 and $e_{1.1}$ as it only changes the analysis method but retains hypothesis h_1 and data d_1. The statistical comparison between m_2 and $m_{1.1}$ relies on existing literature that explains, at length, how Bayesian methods have a higher conclusion validity. They preserve uncertainty [246], do not make use of the invalid probabilistic extension of the modus tollens [48], and allow modeling the response variable with other distributions than the normal distribution [55, 305]. In the case of $e_{1.2}$, the response variable can be modeled with a binomial distribution $Asc^- \sim B(n, p)$ where n represents the number of expected associations and p the likelihood of missing one association. This distribution encodes the ontological assumption that the number of potentially missing associations is bounded by the number of expected associations in the gold standard of the expected domain model, i.e., a participant in the experiment cannot miss more associations than the gold standard contained. This assumption could not be implemented in the frequentist analysis methods that assumed the (rank-transformed) outcome variable to be normal. Therefore, m_2 exceeds $m_{1.1}$ in conclusion validity. Evidence $e_{1.2}$ concludes that the use of passive voice leads to more missing associations in 51%, fewer in 15%, and an

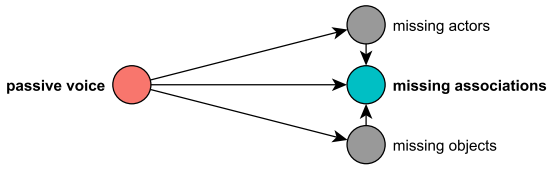


Figure 6: Hypothesis h_{2a} with the purpose of debiasing

equal amount in 34% of all cases. This conclusion still supports that the use of passive voice has an impact on the number of missing associations, but remains more cautious.

5.3.3 Revision $e_{1.3a}$

Since $e_{1.1}$ replaces the simple NHST with a linear model, we can systematically revise the hypothesis h_1 by adding additional predictors. However, the revision of h_1 to h_2 in e_2 performed two separate revisions with different purposes. Firstly, the authors added assumed causal relationships of the number of missing actors and domain objects on the number of missing associations (additional assumption 1 in Section 5.2.2). Figure 6 visualizes the hypothesis h_{2a} resulting from adding just this assumption to h_1. h_{2a} is a sub-graph of the previously introduced h_2 (Figure 4), only missing the additional variables.

To determine which of the two hypotheses h_1 and h_{2a} has greater internal validity, we need to assess the testable implications of the models. In particular, h_1 implies that the outcome variable Asc^- is independent of the number of missing actors Act^- and domain objects Obj^- when conditioning on the use of passive voice. Comparing the coefficients of the two fit linear models $e_{1.1}$ and $e_{1.3a}$ shows that $\beta_{passive}$ only shifts slightly but remains inconsistent with 0. On the other hand, β_{Obj^-} is also inconsistent with 0, confirming that the outcome variable is not independent of the mediator Obj^-. This warrants their inclusion in the hypothesis and confirms that h_{2a} is more internally valid than h_1.

5.3.4 Revision and Reanalysis $e_{1.3b}$

Secondly, the authors of the follow-up study [81] performed a revision with the purpose of increasing the precision of the estimate. To this end, they included additional variables that were recorded in d_1 to the hypothesis, resulting in h_2 as visualized in Figure 4. Two of the newly included predictors—the participants' skill and the requirements' complexity—are modeled as random effects, which requires extending the linear model $m_{1.1}$ to a linear mixed model $m_{1.2}$. Consequently, this step constitutes both a revision and a reanalysis. Both of these evolutions need to be assessed individually.

To determine which of the two analysis methods has a greater conclusion valid-

ity, we can assess the statistical properties of the pieces of evidence. For instance, the residuals of a linear model should be independent and identically distributed (iid) [361]. This property can be determined graphically by inspecting a histogram of the residuals, a QQ-plot, by running tests like the Durbin-Watson test, or other diagnostics. Both graphical analyses (to be found in our replication package [329]) and the statistical tests suggests that the residuals of $e_{1.1}$ are not iid: The Shapiro-Wilk test suggests a significant deviation from a normal distribution of the residuals ($p = 3.26e - 05$), the Durbin-Watson test suggests an autocorrelation greater than 0 ($p = 0.07$), only the Breusch-Pagan test does not suggest heteroscedasticity ($p = 0.11$). Consequently, applying a linear model $m_{1.1}$ may lead to invalid conclusions, and the conclusion validity of a linear mixed model $m_{1.2}$ is greater [361].

To determine which of the two hypotheses has the greater internal validity, we evaluate their predictive power. Since the analysis methods differ, conventional metrics like R^2 and its variants are ineligible, as they apply only to one of the two methods. Instead, we calculate the AIC, which applies to both [362]. The two pieces of evidence achieve scores of $AIC(e_{1.3a}) = 249.1$ and $AIC(e_{1.3b}) = 251.2$. The score differential of about 2 points is considered negligible when interpreting the AIC values [363]. Hence, there is no strong evidence that h_2 is more internally valid than h_{2a}. However, the confidence interval of $\beta_{passive}$ concluded by $e_{1.3b}$ is $ci_{e_{1.3b}}(passive) = [-0.17, 0.82]$. $e_{1.3b}$ is the first piece of evidence suggesting that the use of passive voice does not have a statistically significant impact on the number of missing associations in domain models.

5.3.5 Revision/Reanalysis e_2

This leads to the target evidence e_2, which can now be considered a strict revision of $e_{1.2}$ and a strict reanalysis of $e_{1.3b}$. As such, the eligibility of these evolution steps can be assessed individually. The reanalysis of $e_{1.3b}$ to e_2 again relies on literature explaining the advantage of Bayesian over frequentist methods [48, 55, 246, 305]. The revision of $e_{1.2}$ to e_2 again requires two steps and needs to determine that the inclusion of assumptions both de-biases the estimate and increases its precision.

The conclusion of e_2 is, as presented in Section 5.2.2, that using passive voice is less impactful than originally assumed [34]. The decomposed sub-steps make this more evident, as e_2 claims that passive voice causes more missing associations in only 34% of all cases, other than $e_{1.2}$, which claimed it to be 51%. However, the target evidence e_2 is subject to several shortcomings due to the conflation of the revision with the reanalysis. Firstly, the follow-up study [81] attributes the differing conclusion mainly to the use of Bayesian methods (i.e., the reanalysis) rather than the changed hypothesis (i.e., the revision). The detailed analysis, though, clearly shows that only $e_{1.3b}$, i.e., the revision to h_2, made the conclusion more cautious. Secondly, the follow-up study never assesses whether h_2 has greater internal validity than h_1 and deserves to be subscribed to. The model comparison in scope of the

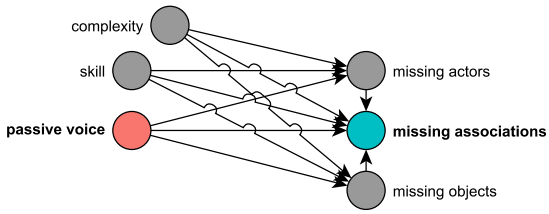


Figure 7: Hypothesis h_{2c}

revision $e_{1.3a}$ shows that the support for h_2a over h_1 is actually minimal and, hence, more debatable than the follow-up study makes it seem. Adherence to the proposed framework for managing variance theories revealed these shortcomings and made the reliability of each step transparent.

5.3.6 Revision $e_{1.3c}$

The stepwise evolution according to the proposed framework revealed that h_2 has little support over h_{2a} as the AIC scores are close enough together to consider both hypotheses of equal predictive power [363]. However, the diagnostics of $e_{1.3b}$ reveal that the inclusion of random effects had a strong impact: While $e_{1.3b}$ only has a marginal R^2 value of 0.184, it has a conditional R^2 value of 0.561. This indicates that the random effects explain a lot more of the variance of the outcome variable than the fixed effects [364]. The benefit of the random effects in h_2 may be offset by the number of predictors, as the AIC metric penalizes an increased number of predictors to avoid overfitting [363]. Hence, the authors could have formulated the competing hypothesis h_{2c} (shown in Figure 7), which retains the provenly effective random effects for requirements' complexity and participants' skill but discards the fixed effects of academic and industrial experience. The operationalization of both of these fixed effects is questionable, as they were simply measured on an ordinal scale with four levels. Because the construct validity of this operationalization is questionable, the inclusion of these fixed effects might not benefit the estimation and rather overfit the estimation.

For model comparison to assess the predictive power of the new piece of evidence, we can use the Akaike information criterion. The resulting $AIC(e_{1.3c}) = 241.4$ is significantly—i.e., more than 2 units [363]—lower than $AIC(e_{1.3a}) = 249.1$ and $AIC(e_{1.3b}) = 251.2$. Consequently, h_{2c} shows the greatest internal validity and should have been used subsequently instead of h_2. Evidence $e_{1.3c}$ concludes that passive voice has an impact of $ci_{e_{1.3c}}(passive) = [0.03, 0.92]$, which is still not consistent with 0 but broader than $ci_{e_{1.3b}}(passive) = [0.17, 0.77]$. This suggests that passive voice does have an impact on the number of missing associations, though the strength of the impact varies more.

5.3.7 Reanalysis/Revision $e_{1.4}$

The more rigorous target evidence of the follow-up study [81] would have been $e_{1.4}$, which applies the analysis method with the highest conclusion validity—Bayesian data analysis m_2—to the data d_1 under the hypothesis with the highest internal validity—hypothesis h_{2c}, not h_2. This piece of evidence classifies as a reanalysis of $e_{1.3c}$ (as it only substitutes $m_{1.2}$ with m_2) and a revision of e_2 (as it substitutes h_2 with h_{2c}). The validity of the reanalysis is again based on previous literature [48, 55, 246, 305] and the revision on the model comparison regarding predictive power. In the case of comparing two Bayesian models, we can use the leave-one-out (LOO) cross-validation [358]. The LOO comparison favors $e_{1.4}$ over e_2 as expected based on the previous comparison of frequentist models using the AIC metric. This piece of evidence $e_{1.4}$ concludes that the use of passive voice leads to more missing associations in 39%, fewer in 14%, and an equal amount in 47% of all cases. Considering this the most valid piece of evidence about the impact of passive voice at the time of the follow-up study [81], the variance theory would suggest that passive voice does have an impact on the number of missing associations from domain models, though not strictly and only in about 40% of all cases.

Insight 2

The application of the framework to disentangle the omitted sub-steps between e_1 and e_2 revealed that h_2 was not the optimal improvement over h_1, but instead, h_{2c} would have been. Additionally, the random effects modeling participants' skill improved the precision of the ACE estimation far greater than variables like experience (measured on an ordinal scale). This indicates that years of experience is an insufficient operationalization to represent the latent context variable of modeling skill. Finally, the adjusted conclusions can be attributed more to the revision than to the reanalysis.

6 Discussion

The proposed framework enables researchers to systematically manage variance theories. The framework offers a definition of empirical, quantitative evidence, a clear terminology of the relationship between two pieces of evidence, and guidance on how to determine which piece of evidence has the greater validity. The discussion at the ISERN workshop agreed on three implications for research practice (Section 6.1) while also acknowledging several limitations (Section 6.2) that necessitate future work (Section 6.3).

6.1 Implications

First, our terminology of evolution types allows primary studies to clearly position themselves in relation to the existing body of knowledge. Authors contributing a replication, revision, or reanalysis of a phenomenon with at least one prior study, can label their follow-up study with the respective evolution type to indicate how they advance the scientific frontier of an existing body of knowledge. Additionally, the framework specifies how authors can determine whether their follow-up study is of greater validity or not, e.g., via model comparison in the case of a revision for de-confounding. We hope this helps researchers to publish also negative results, as these indicate possible "dead ends" in a research topic.

Second, the proposed framework allows researchers to relate pieces of evidence in secondary studies more clearly. We hope to inspire more rigorous literature reviews about phenomena that synthesize empirical, quantitative evidence to variance theories. Visualizing the evolution of a variance theory in the form of a version control graph as demonstrated in Sections 5.2 and 5.3 helps to communicate the progress within a research field.

Finally, the application of the proposed framework prospectively shapes future research. With the scientific frontier of a research field determined and analytically supported, an application of the framework can inform the design of future studies. For example, the hypothesis with the highest internal validity informs new research about the factors that need to be measured. Similarly, the analysis method with the highest conclusion validity informs how to perform the data analysis on collected data. This way, researchers can coordinate research agendas working towards a shared variance theory about a quantitative phenomenon.

6.2 Limitations

We differentiate the limitations of the framework itself from the limitations to the adoption of the framework in SE research. One significant limitation of the framework is that it depends on reliable operationalizations of the concepts involved in the phenomena under study. The dimension of validity that the proposed framework does not systematically address is construct validity. A natural option would have been to define hypotheses on the concept, not on the indicator level. For example, the phenomenon discussed in Section 5 could have been abstracted to the impact of requirements quality on domain modeling performance instead of passive voice on the number of missing associations. This would extend the types of evolution in Figure 2 by one that challenges the operationalization of concepts, i.e., that improves the construct validity of the measurements. In the example, this would mean challenging whether passive voice is a valid indicator of requirements quality or whether the number of missing associations adequately represents domain modeling performance. However, we opted against this as we could not find a consensus on the systematic

comparison of construct validity in SE research. Instead, we assume all variables involved in hypotheses to be on an indicator level and delegate their abstraction to concepts outside of the framework. Should a competing piece of evidence aim to improve the construct validity and propose a hypothesis in which at least one concept is operationalized differently, then these new pieces of evidence are incommensurable.

Additionally, the proposed framework depends on the rigor of the applied research methods. SE research has been shown to be subject to researcher bias [365, 366], which implies that the conclusions drawn from evidence differ not only depending on the hypotheses, data, and methods involved but also based on the people that produced the evidence. This necessitates manually ensuring a sufficient level of rigor in pieces of evidence that shall be included in a body of knowledge.

Furthermore, the usefulness of the proposed framework depends on researchers' adoption of it. One limitation to this is the complexity of the framework itself. While methodologies for replications are well established in SE research [63, 83, 336], approaches for systematic comparison of causal hypotheses via model comparison are not yet commonplace [54] and the selection of analysis methods often follows conventions. The framework offers a principled approach that places revisions and reanalyses into a relationship with the existing body of knowledge but requires that scholars familiarize themselves with model comparison and challenge statistical conventions. We envision that the proposed framework will initially be most useful to researchers or research groups willing to immerse themselves in these methods by putting their own pieces of evidence about a shared phenomenon into relation. Later, we hope that larger applications like systematic literature reviews or adoption by a whole community will become possible.

6.3 Future Work

The primary goal of our future work will be to communicate this framework and offer support in adopting it. This means not just presenting the framework as is but connecting the elements of the framework as shown in Figure 2 with literature that helps scholars to apply the presented relationships. For example, the replication branch can be supported with definitions [63, 83], philosophical stances [85], and advice on performing meta-analyses [336]. We envision that this communication support will take the form of a web platform that makes both the framework and the recommended resources, like further reading and demonstrations, accessible.

The second goal of future work is to further support the application of the framework by evolving the aforementioned platform under the umbrella of the ISERN community. We aim to extend the platform into a system where pieces of evidence can be submitted and that visualizes the evolution of a body of knowledge about a phenomenon. This feature of the platform will resemble version control systems like git for empirical evidence. Such a platform will host the current body of knowledge

about SE phenomena. It will support researchers both in finding the most recent contributions to a phenomenon and inform future study designs by indicating the hypotheses with the highest validity. This platform will address the final shortcoming mentioned in Section 2.2 and replace static, retrospective research synthesis with a dynamic, continuous process. Approaches to automate model comparison and meta-analysis may pave the way towards an even more dynamic process. Additionally, such a platform may support the initially mentioned knowledge translation by offering an interface to practitioners to obtain synthesized—and, therefore, more valid—conclusions from variance theories. These quantitative conclusions serve as decision support, for example when determining whether to adopt TDD practices [327] or whether to remove passive voice from requirements documents [82].

Finally, we aim to demonstrate the application of the framework to additional fields of SE research. The field of requirements quality was chosen due to the authors' familiarity with it and since it contains coherent yet manageable pieces of evidence. Fields that aim to produce variance theories of phenomena, like the impact of TDD on code quality and developer effectiveness [327], are eligible for such an application. We envision a special type of literature review emerging from the framework, focusing on quantitative primary studies about a particular phenomenon.

7 Conclusion

To effectively progress the development of variance theories from quantitative, empirical evidence, SE research needs a research synthesis approach that extends beyond the meta-analysis of homogeneous replications. In this article, we define quantitative, empirical evidence, propose a typology of relationships between two pieces of evidence, and offer guidance on determining which piece of evidence has greater validity. Expressing research agendas through this framework enables the systematic management of variance theories and guides SE research toward producing more rigorous conclusions.

Bibliography

[1] H. Femmer. “Requirements Quality Defect Detection with the Qualicen Requirements Scout.” In: *REFSQ Workshops*. 2018.

[2] H. Femmer, J. Mund, and D. M. Fernández. “It’s the activities, stupid! a new perspective on RE quality”. In: *RET*. 2015. DOI: 10.1109/RET.2015.11.

[3] B. W. Boehm. “Software engineering economics”. In: *IEEE transactions on Software Engineering* 1 (1984), pp. 4–21.

[4] L.-O. Damm, L. Lundberg, and C. Wohlin. “Faults-slip-through—a concept for measuring the efficiency of the test process”. In: *Software Process: Improvement and Practice* 11.1 (2006), pp. 47–59. DOI: 10.1002/spip.253.

[5] S. Fricker, T. Gorschek, C. Byman, and A. Schmidle. “Handshaking with implementation proposals: Negotiating requirements understanding”. In: *IEEE software* 27.2 (2010), pp. 72–80. DOI: 10.1109/MS.2009.195.

[6] L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej. “Empirical research on requirements quality: a systematic mapping study”. In: *Requirements Engineering* 27.2 (2022), pp. 183–209. DOI: 10.1007/s00766-021-00367-z.

[7] D. Berry, R. Gacitua, P. Sawyer, and S. F. Tjong. “The case for dumb requirements engineering tools”. In: *International working conference on requirements engineering: Foundation for software quality*. Springer. 2012, pp. 211–217.

[8] K. T. Phalp, J. Vincent, and K. Cox. “Assessing the quality of use case descriptions”. In: *Software Quality Journal* 15.1 (2007), pp. 69–97.

[9] X. Franch, D. Mendez, A. Vogelsang, R. Heldal, E. Knauss, M. Oriol, G. Travassos, J. C. Carver, and T. Zimmermann. “How do Practitioners Perceive the Relevance of Requirements Engineering Research?” In: *IEEE Transactions on Software Engineering* (2020).

[10] M. Glinz. “A glossary of requirements engineering terminology”. In: *Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version* 1 (2011), p. 56.

[11] D. T. Ross. “Reflections on Requirements”. In: *IEEE Transactions on Software Engineering* 3 (1977).

[12] K. Matyokurehwa, N. Mavetera, and O. Jokonya. “Requirements engineering techniques: A systematic literature review”. In: *International Journal of Soft Computing and Engineering* 7.1 (2017), pp. 14–20.

[13] D. M. Fernandez, S. Wagner, K. Lochmann, A. Baumann, and H. de Carne. “Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies”. In: *Information and Software Technology* 54.2 (2012), pp. 162–178. DOI: 10.1016/j.infsof.2011.09.001.

[14] A. M. Hickey and A. M. Davis. “A unified model of requirements elicitation”. In: *Journal of management information systems* 20.4 (2004), pp. 65–84.

[15] I. Sommerville. *Software Engineering*. 9th. Addison-Wesley, 2011.

[16] K. Wiegers and J. Beatty. *Software requirements*. Pearson Education, 2013.

[17] D. M. Fernández, W. Böhm, A. Vogelsang, J. Mund, M. Broy, M. Kuhrmann, and T. Weyer. “Artefacts in software engineering: a fundamental positioning”. In: *Software & Systems Modeling* 18.5 (2019), pp. 2777–2786.

[18] D. Méndez Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, et al. “Naming the pain in requirements engineering: Contemporary Problems, Causes, and Effects in Practice”. In: *Empirical software engineering* 22.5 (2017), pp. 2298–2338.

[19] A. Wassyng, E. Simmons, R. Hall, D. Gause, A. Finkelstein, D. Damian, and D. M. Berry. “To do or not to do: If the requirements engineering payoff is so good, why aren’t more companies doing it?” In: *13th IEEE International Conference on Requirements Engineering (RE’05)*. IEEE Computer Society. 2005, pp. 447–447.

[20] O. Hoehne. “I Don’t Need Requirements–I Know What I’m Doing! Usability as a Critical Human Factor in Requirements Management”. In: *INCOSE International Symposium*. Vol. 27. 1. Wiley Online Library. 2017, pp. 1026–1039.

[21] S. Wagner, D. Méndez Fernández, M. Felderer, A. Vetrò, M. Kalinowski, R. Wieringa, D. Pfahl, T. Conte, M.-T. Christiansson, D. Greer, et al. “Status quo in requirements engineering: A theory and a global family of surveys”. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 28.2 (2019), pp. 1–48.

[22] B. Nuseibeh and S. Easterbrook. “Requirements engineering: a roadmap”. In: *Proceedings of the Conference on the Future of Software Engineering*. 2000, pp. 35–46.

[23] D. Méndez Fernández and B. Penzenstadler. “Artefact-based requirements engineering: the AMDiRE approach”. In: *Requirements Engineering* 20.4 (2015), pp. 405–434.

[24] D. Méndez Fernández, W. Böhm, A. Vogelsang, J. Mund, M. Broy, M. Kuhrmann, and T. Weyer. “Artefacts in software engineering: a fundamental positioning”. In: *Software & Systems Modeling* 18.5 (2019), pp. 2777–2786.

[25] X. Franch, C. Palomares, C. Quer, P. Chatzipetrou, and T. Gorschek. “The state-of-practice in requirements specification: an extended interview study at 12 companies”. In: *Requirements Engineering* (2023), pp. 1–33. DOI: 10.1007/s00766-023-00399-7.

[26] B. Nuseibeh, J. Kramer, and A. Finkelstein. “A framework for expressing the relationships between multiple views in requirements specification”. In: *IEEE Transactions on software engineering* 20.10 (1994), pp. 760–773.

[27] C. L. Heitmeyer, R. D. Jeffords, and B. G. Labaw. “Automated consistency checking of requirements specifications”. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 5.3 (1996), pp. 231–261.

[28] D. Popescu, S. Rugaber, N. Medvidovic, and D. M. Berry. “Reducing ambiguities in requirements specifications via automatically created object-oriented models”. In: *Monterey Workshop*. Springer. 2007, pp. 103–124.

[29] O. Karras, S. Kiesling, and K. Schneider. “Supporting requirements elicitation by tool-supported video analysis”. In: *2016 IEEE 24th International Requirements Engineering Conference (RE)*. IEEE. 2016, pp. 146–155.

[30] H. Femmer, M. Unterkalmsteiner, and T. Gorschek. “Which requirements artifact quality defects are automatically detectable? A case study”. In: *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2017, pp. 400–406.

[31] M. I. Kamata and T. Tamai. “How does requirements quality relate to project success or failure?” In: *15th IEEE International Requirements Engineering Conference (RE 2007)*. IEEE. 2007, pp. 69–78.

[32] D. M. Berry and E. Kamsties. “Ambiguity in requirements specification”. In: *Perspectives on software requirements*. Springer, 2004, pp. 7–44.

[33] K. Pohl. *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*. Rocky Nook, Inc., 2016.

[34] H. Femmer, J. Kučera, and A. Vetrò. “On the impact of passive voice requirements on domain modelling”. In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2014, pp. 1–4.

[35] B. W. Boehm, J. R. Brown, and M. Lipow. “Quantitative evaluation of software quality”. In: *Proceedings of the 2nd international conference on Software engineering*. 1976, pp. 592–605.

[36] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh. “Analysing anaphoric ambiguity in natural language requirements”. In: *Requirements engineering* 16.3 (2011), pp. 163–189.

[37] X. Franch, D. M. Fernández, M. Oriol, A. Vogelsang, R. Heldal, E. Knauss, G. H. Travassos, J. C. Carver, O. Dieste, and T. Zimmermann. “How do practitioners perceive the relevance of requirements engineering research? An ongoing study”. In: *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE. 2017, pp. 382–387.

[38] D. I. Sjøberg, T. Dybå, B. C. Anda, and J. E. Hannay. “Building theories in software engineering”. In: *Guide to advanced empirical software engineering*. Springer, 2008, pp. 312–336.

[39] S. Gregor. “The nature of theory in information systems”. In: *MIS quarterly* (2006), pp. 611–642.

[40] J. Frattini, L. Montgomery, J. Fischbach, D. Mendez, D. Fucci, and M. Unterkalmsteiner. “Requirements quality research: a harmonized theory, evaluation, and roadmap”. In: *Requirements Engineering* (2023), pp. 1–14. DOI: 10.1007/s00766-023-00405-y.

[41] J. Frattini, L. Montgomery, J. Fischbach, M. Unterkalmsteiner, D. Mendez, and D. Fucci. “A live extensible ontology of quality factors for textual requirements”. In: *2022 IEEE 30th International Requirements Engineering Conference (RE)*. IEEE. 2022, pp. 274–280. DOI: 10.1109/RE54965.2022.00041.

[42] M. Ciolkowski and J. Münch. “Accumulation and presentation of empirical evidence: problems and challenges”. In: *ACM SIGSOFT Software Engineering Notes* 30.4 (2005), pp. 1–3.

[43] P. Ralph, N. b. Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, et al. “Empirical standards for software engineering research”. arXiv preprint arXiv:2010.03525. 2020.

[44] D. Mendez, D. Graziotin, S. Wagner, and H. Seibold. “Open science in software engineering”. In: *Contemporary empirical methods in software engineering* (2020), pp. 477–501.

[45] J. Frattini, L. Montgomery, D. Fucci, J. Fischbach, M. Unterkalmsteiner, and D. Mendez. “Let’s Stop Building at the Feet of Giants: Recovering unavailable Requirements Quality Artifacts”. In: *Joint Proceedings of REFSQ-2023 Workshops, Doctoral Symposium, Posters & Tools Track and Journal Early Feedback co-located with the 28th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2023)*. Vol. 3378. CEUR-WS. 2023. DOI: 10.48550/arXiv.2304.04670.

[46] R. Minocher, S. Atmaca, C. Bavero, R. McElreath, and B. Beheim. “Reproducibility improves exponentially over 63 years of social learning research”. In: (2020).

[47] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Heidelberg: Springer Science & Business Media, 2012.

[48] C. A. Furia, R. Feldt, and R. Torkar. “Bayesian data analysis in empirical software engineering research”. In: *IEEE Transactions on Software Engineering* 47.9 (2019), pp. 1786–1810.

[49] B. Kitchenham, J. Fry, and S. Linkman. “The case against cross-over designs in software engineering”. In: *Eleventh annual international workshop on software technology and engineering practice*. IEEE. 2003, pp. 65–67. DOI: 10.1109/STEP.2003.32.

[50] S. Vegas, C. Apa, and N. Juristo. “Crossover designs in software engineering experiments: Benefits and perils”. In: *IEEE Transactions on Software Engineering* 42.2 (2015), pp. 120–135. DOI: 10.1109/TSE.2015.2467378.

[51] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro. “Natural language processing for requirements engineering: A systematic mapping study”. In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–41.

[52] J. Frattini, M. Unterkalmsteiner, D. Fucci, and D. Mendez. “NLP4RE Tools: Classification, Overview, and Management”. In: *Handbook of Natural Language Processing for Requirements Engineering*. Springer International Publishing, 2024.

[53] J. Pearl. “Causal inference”. In: *Causality: objectives and assessment* (2010), pp. 39–58.

[54] J. Siebert. “Applications of statistical causal inference in software engineering”. In: *Information and Software Technology* (2023), p. 107198.

[55] R. McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018.

[56] S. Z. Schiller and M. Mandviwalla. “Virtual team research: An analysis of theory use and a framework for theory appropriation”. In: *Small group research* 38.1 (2007), pp. 12–59.

[57] J. S. Molléri, K. Petersen, and E. Mendes. “An empirically evaluated checklist for surveys in software engineering”. In: *Information and Software Technology* 119 (2020), p. 106240.

[58] R. C. Nickerson, U. Varshney, and J. Muntermann. “A method for taxonomy development and its application in information systems”. In: *European Journal of Information Systems* 22.3 (2013).

[59] D. I. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. “A survey of controlled experiments in software engineering”. In: *IEEE transactions on software engineering* 31.9 (2005), pp. 733–753.

[60] P. Runeson, M. Host, A. Rainer, and B. Regnell. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[61] D. S. Cruzes and T. Dyba. “Recommended steps for thematic synthesis in software engineering”. In: *2011 international symposium on empirical software engineering and measurement*. IEEE. 2011, pp. 275–284.

[62] R. McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Boca Raton, FL: Chapman and Hall/CRC, 2020.

[63] M. T. Baldassarre, J. Carver, O. Dieste, and N. Juristo. “Replication types: Towards a shared taxonomy”. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 2014, pp. 1–4.

[64] C. Wohlin. “Guidelines for snowballing in systematic literature studies and a replication in software engineering”. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10.

[65] T. S. Kuhn. *The structure of scientific revolutions*. Vol. 111. Chicago University of Chicago Press, 1970.

[66] J. E. Hannay, D. I. Sjoberg, and T. Dyba. “A systematic review of theory use in software engineering experiments”. In: *IEEE transactions on Software Engineering* 33.2 (2007), pp. 87–107.

[67] P. Johnson, M. Ekstedt, and I. Jacobson. “Where’s the theory for software engineering?” In: *IEEE software* 29.5 (2012), pp. 96–96.

[68] K. Schmid. “If you want better empirical research, value your theory: On the importance of strong theories for progress in empirical software engineering research”. In: *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*. 2021, pp. 359–364.

[69] K.-J. Stol, P. Ralph, and B. Fitzgerald. “Grounded theory in software engineering research: a critical review and guidelines”. In: *Proceedings of the 38th International conference on software engineering*. 2016, pp. 120–131.

[70] M. Broy, F. Deissenboeck, and M. Pizka. “Demystifying maintainability”. In: *Proceedings of the 2006 international workshop on Software quality*. 2006, pp. 21–26.

[71] F. Deissenboeck, S. Wagner, M. Pizka, S. Teuchert, and J.-F. Girard. “An activity-based quality model for maintainability”. In: *2007 IEEE International Conference on Software Maintenance*. IEEE. 2007, pp. 184–193.

[72] S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidi, A. Goeb, and J. Streit. “The Quamoco product quality modelling and assessment approach”. In: *2012 34th International Conference on Software Engineering (ICSE)*. IEEE. 2012, pp. 1133–1142.

[73] F. Deissenboeck, L. Heinemann, M. Herrmannsdoerfer, K. Lochmann, and S. Wagner. “The quamoco tool chain for quality modeling and assessment”. In: *2011 33rd International Conference on Software Engineering (ICSE)*. IEEE. 2011, pp. 1007–1009.

[74] J. Frattini, L. Montgomery, D. Fucci, M. Unterkalmsteiner, D. Mendez, and J. Fischbach. “Requirements quality research artifacts: Recovery, analysis, and management guideline”. In: *Journal of Systems and Software* (2024), p. 112120. DOI: 10.1016/j.jss.2024.112120.

[75] S. Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. Tech. rep. Technical report, ver. 2.3 ebse technical report. ebse, 2007.

[76] J. Pearl, M. Glymour, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[77] C. Cinelli, A. Forney, and J. Pearl. “A crash course in good and bad controls”. In: *Sociological Methods & Research* 53.3 (2024), pp. 1071–1104. DOI: 10.1177/00491241221099552.

[78] R. Torkar, R. Feldt, and C. A. Furia. “Bayesian data analysis in empirical software engineering: The case of missing data”. In: *Contemporary empirical methods in software engineering*. Cham: Springer International Publishing, 2020, pp. 289–324. ISBN: 978-3-030-32489-6. DOI: 10.1007/978-3-030-32489-6_11.

[79] A. Gelman. *You need 16 times the sample size to estimate an interaction than to estimate a main effect*. https://statmodeling.stat.columbia.edu/2018/03/15/need16/. Accessed: 2023-11-24.

[80] C. A. Furia, R. Torkar, and R. Feldt. “Towards causal analysis of empirical software engineering data: The impact of programming languages on coding competitions”. In: *ACM Transactions on Software Engineering and Methodology* 33.1 (Nov. 2023). ISSN: 1049-331X. DOI: 10.1145/3611667.

[81] J. Frattini, D. Fucci, R. Torkar, and D. Mendez. “A Second Look at the Impact of Passive Voice Requirements on Domain Modeling: Bayesian Reanalysis of an Experiment”. In: *International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE’24)*. 2024. DOI: 10.1145/3643664.3648211.

[82] J. Frattini, D. Fucci, R. Torkar, L. Montgomery, M. Unterkalmsteiner, J. Fischbach, and D. Mendez. “Applying Bayesian Data Analysis for Causal Inference about Requirements Quality: A Controlled Experiment”. Under Revision at EMSE Journal.

[83] O. S. Gómez, N. Juristo, and S. Vegas. “Replications types in experimental disciplines”. In: *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*. 2010, pp. 1–10.

[84] N. Juristo and S. Vegas. “The role of non-exact replications in software engineering experiments”. In: *Empirical Software Engineering* 16 (2011), pp. 295–324.

[85] B. A. Nosek and T. M. Errington. “What is replication?” In: *PLoS biology* 18.3 (2020), e3000691.

[86] F. Deissenboeck and M. Pizka. “The economic impact of software process variations”. In: *International Conference on Software Process*. Springer. 2007, pp. 259–271.

[87] J. Mund, D. M. Fernandez, H. Femmer, and J. Eckhardt. “Does quality of requirements specifications matter? combined results of two empirical studies”. In: *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. 2015, pp. 1–10.

[88] K. Petersen and C. Wohlin. “Context in industrial software engineering research”. In: *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE. 2009, pp. 401–404.

[89] P. Ralph. “Toward methodological guidelines for process theories and taxonomies in software engineering”. In: *IEEE Transactions on Software Engineering* 45.7 (2018), pp. 712–735.

[90] A. Ferrari, G. Gori, B. Rosadini, I. Trotta, S. Bacherini, A. Fantechi, and S. Gnesi. “Detecting requirements defects with NLP patterns: an industrial experience in the railway domain”. In: *Empirical Software Engineering* 23.6 (2018), pp. 3684–3733.

[91] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, and L. C. Briand. “Using domain-specific corpora for improved handling of ambiguity in requirements”. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE. 2021, pp. 1485–1497.

[92] J. Frattini, J. Fischbach, D. Fucci, M. Unterkalmsteiner, and D. Mendez. “Measuring the Fitness-for-Purpose of Requirements: An initial Model of Activities and Attributes”. In: *2024 IEEE 30th International Requirements Engineering Conference (RE)*. IEEE. 2024. DOI: 10.1109/RE59067.2024.00047.

[93] J. Frattini, D. Fucci, and S. Vegas. “Crossover Designs in Software Engineering Experiments: Review of the State of Analysis”. In: *2024 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM. 2024.

[94] P. S. M. d. Santos and G. H. Travassos. “Research synthesis in software engineering”. In: *Contemporary Empirical Methods in Software Engineering*. Springer, 2020, pp. 443–474.

[95] T. Kuhn. “The structure of scientific revolutions”. In: *International Encyclopedia of Unified Science* 2.2 (1962).

[96] D. Dell’Anna, F. B. Aydemir, and F. Dalpiaz. “Evaluating classifiers in SE research: the ECSER pipeline and two replication studies”. In: *Empirical Software Engineering* 28.1 (2023), p. 3. DOI: 10.1007/s10664-022-10243-1.

[97] H. Femmer and A. Vogelsang. “Requirements quality is quality in use”. In: *IEEE Software* 36.3 (2018), pp. 83–91.

[98] I. Lakatos. “Falsification and the Methodology of Scientific Research Programmes’ in I. Lakatos and A. Musgrave (eds.) Criticism and the Growth of Knowledge”. In: *Proceedings of the International Colloquium in the Philosophy of Science*. Vol. 4. 91. 1970, p. 196.

[99] J. Frattini, J. Fischbach, D. Fucci, M. Unterkalmsteiner, and D. Mendez. “Replications, Revisions, and Reanalyses: Managing Variance Theories in Software Engineering”. Submitted to the TSE Journal.

[100] D. I. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanović, and M. Vokáč. “Challenges and recommendations when increasing the realism of controlled software engineering experiments”. In: *Empirical Methods and Studies in Software Engineering: Experiences from ESERNET*. Springer, 2003, pp. 24–38. DOI: 10.1007/978-3-540-45143-3_3.

[101] D. Damian and J. Chisan. “An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management”. In: *IEEE Transactions on Software Engineering* 32.7 (2006), pp. 433–453.

[102] B. W. Boehm and P. N. Papaccio. “Understanding and controlling software costs”. In: *IEEE transactions on software engineering* 14.10 (1988), pp. 1462–1477.

[103] O. I. Lindland, G. Sindre, and A. Solvberg. “Understanding quality in conceptual modeling”. In: *IEEE software* 11.2 (1994), pp. 42–49.

[104] K. Pohl. “The three dimensions of requirements engineering”. In: *International Conference on Advanced Information Systems Engineering*. Springer. 1993, pp. 275–292.

[105] M. Broy, F. Deißenböck, and M. Pizka. “A holistic approach to software quality at work”. In: *Proc. 3rd world congress for software quality (3WCSQ)*. 2005.

[106] P. King, P. Naughton, M. DeMoney, J. Kanerva, K. Walrath, and S. Hommel. *Code Conventions for the Java Programming Language*. Mountain View, CA, USA: Sun Microsystems, Inc., 2021.

[107] A. J. Albrecht and J. E. Gaffney. “Software function, source lines of code, and development effort prediction: a software science validation”. In: *IEEE transactions on software engineering* 6 (1983), pp. 639–648.

[108] T. J. McCabe. “A complexity measure”. In: *IEEE Transactions on software Engineering* 4 (1976), pp. 308–320.

[109] J. Rosenberg. “Some misconceptions about lines of code”. In: *Proceedings fourth international software metrics symposium*. IEEE. 1997, pp. 137–142.

[110] T. M. Khoshgoftaar and J. C. Munson. “The lines of code metric as a predictor of program faults: A critical analysis”. In: *Proceedings Fourteenth Annual International Computer Software and Applications Conference*. IEEE Computer Society. 1990, pp. 408–409.

[111] M. Shepperd. “A critique of cyclomatic complexity as a software metric”. In: *Software Engineering Journal* 3.2 (1988), pp. 30–36.

[112] J. A. McCall. “Factors in software quality”. In: *US Rome Air development center reports* (1977).

[113] B. Kitchenham, S. Linkman, A. Pasquini, and V. Nanni. “The SQUID approach to defining a quality model”. In: *Software Quality Journal* 6.3 (1997), pp. 211–233.

[114] V. R. Basili, G. Caldiera, and H. D. Rombach. “The goal question metric approach”. In: *Encyclopedia of software engineering* (1994), pp. 528–532.

[115] R. Marinescu and D. Ratiu. “Quantifying the quality of object-oriented design: The factor-strategy model”. In: *11th Working Conference on Reverse Engineering*. IEEE. 2004, pp. 192–201.

[116] F. Deissenboeck, E. Juergens, K. Lochmann, and S. Wagner. “Software quality models: Purposes, usage scenarios and requirements”. In: *2009 ICSE workshop on software quality*. IEEE. 2009, pp. 9–14.

[117] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, and G. MacLeod. *Merritt.: Characteristics of Software Quality*. 1978.

[118] S. Winter, S. Wagner, and F. Deissenboeck. "A comprehensive model of usability". In: *IFIP International Conference on Engineering for Human-Computer Interaction*. Springer. 2007, pp. 106–122.

[119] S. Wagner, D. M. Fernandez, S. Islam, and K. Lochmann. "A security requirements approach for web systems". In: *Workshop Quality Assessment in Web (QAW 2009)*. 2009.

[120] A. Goeb and K. Lochmann. "A software quality model for SOA". In: *Proceedings of the 8th international workshop on Software quality*. 2011, pp. 18–25.

[121] S. Wagner, K. Lochmann, S. Winter, F. Deissenboeck, E. Juergens, M. Herrmannsdoerfer, L. Heinemann, M. Kläs, A. Trendowicz, J. Heidrich, et al. "The Quamoco quality meta-model". In: (2012).

[122] F. Deissenboeck, E. Juergens, B. Hummel, S. Wagner, B. M. y Parareda, and M. Pizka. "Tool support for continuous quality control". In: *IEEE software* 25.5 (2008), pp. 60–67.

[123] D. Steidl, F. Deissenboeck, M. Poehlmann, R. Heinke, and B. Uhink-Mergenthaler. "Continuous software quality control in practice". In: *2014 IEEE International Conference on Software Maintenance and Evolution*. IEEE. 2014, pp. 561–564.

[124] K. Lochmann, J. Ramadani, and S. Wagner. "Are comprehensive quality models necessary for evaluating software quality?" In: *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*. 2013, pp. 1–9.

[125] D. A. Garvin. "What does product quality really mean". In: *Sloan management review* 25 (1984), pp. 25–43.

[126] M. Kläs, K. Lochmann, and L. Heinemann. "Evaluating a quality model for software product assessments-a case study". In: *Proc. of SQMB* 11 (2011).

[127] S. Wagner, A. Goeb, L. Heinemann, M. Kläs, C. Lampasona, K. Lochmann, A. Mayr, R. Plösch, A. Seidl, J. Streit, et al. "Operationalised product quality models and assessment: The Quamoco approach". In: *Information and Software Technology* 62 (2015), pp. 101–123.

[128] S. Wagner. "A Bayesian network approach to assess and predict software quality using activity-based quality models". In: *Information and Software Technology* 52.11 (2010), pp. 1230–1241.

[129] M. K. Habib, S. Wagner, and D. Graziotin. “Detecting Requirements Smells With Deep Learning: Experiences, Challenges and Future Work”. In: *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2021, pp. 153–156.

[130] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder. “Rapid quality assurance with requirements smells”. In: *Journal of Systems and Software* 123 (2017), pp. 190–213.

[131] D. M. Berry, A. Bucchiarone, S. Gnesi, G. Lami, and G. Trentanni. “A new quality model for natural language requirements specifications”. In: *Proceedings of the international workshop on requirements engineering: foundation of software quality (REFSQ)*. 2006.

[132] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper. “Improving user story practice with the Grimm Method: A multiple case study in the software industry”. In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer. 2017, pp. 235–252.

[133] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga. “A methodology for the classification of quality of requirements using machine learning techniques”. In: *Information and Software Technology* 67 (2015), pp. 180–195.

[134] W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt. “Automated analysis of requirement specifications”. In: *Proceedings of the 19th international conference on Software engineering*. 1997, pp. 161–171.

[135] E. Juergens and F. Deissenboeck. “How much is a clone”. In: *Proceedings of the 4th International Workshop on Software Quality and Maintainability*. 2010, pp. 79–88.

[136] V. Antinyan, M. Staron, A. Sandberg, and J. Hansson. “A complexity measure for textual requirements”. In: *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*. IEEE. 2016, pp. 148–158.

[137] I. Sommerville. “Integrated requirements engineering: A tutorial”. In: *IEEE software* 22.1 (2005), pp. 16–23.

[138] J. Fischbach, J. Frattini, A. Vogelsang, D. Mendez, M. Unterkalmsteiner, A. Wehrle, P. R. Henao, P. Yousefi, T. Juricic, J. Radduenz, et al. “Automatic creation of acceptance tests by extracting conditionals from requirements: Nlp approach and case study”. In: *Journal of Systems and Software* 197 (2023), p. 111549.

[139] M. Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[140] O. R. Holsti. “Content analysis for the social sciences and humanities”. In: *Reading. MA: Addison-Wesley (content analysis)* (1969).

[141] G. C. Feng. “Mistakes and how to avoid mistakes in using intercoder reliability indices.” In: *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences* 11.1 (2015), p. 13.

[142] E. M. Bennett, R. Alpert, and A. Goldstein. “Communications through limited-response questioning”. In: *Public Opinion Quarterly* 18.3 (1954), pp. 303–308.

[143] J. S. Sinpang, S. Sulaiman, and N. Idris. “Detecting ambiguity in requirements analysis using Mamdani fuzzy inference”. In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 9.3-4 (2017), pp. 157–162.

[144] F. Chantree, B. Nuseibeh, A. De Roeck, and A. Willis. “Identifying nocuous ambiguities in natural language requirements”. In: *14th IEEE International Requirements Engineering Conference (RE’06)*. IEEE. 2006, pp. 59–68.

[145] C. Y. Din and D. Rine. “Requirements content goodness and complexity measurement based on NP chunks”. In: *Journal of Systemics, Cybernetics and Informatics* 7.3 (2009), pp. 12–18.

[146] M. Landhaußer, S. J. Korner, W. F. Tichy, J. Keim, and J. Krisch. “DeNom: a tool to find problematic nominalizations using NLP”. In: *2015 IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*. IEEE. 2015, pp. 1–8.

[147] J. L. Mackenzie. “Nominalization and valency reduction”. In: *Predicates and terms in Functional Grammar. Dordrecht/Cinnaminson: Foris* (1985), pp. 31–51.

[148] D. Méndez Fernández, B. Penzenstadler, M. Kuhrmann, and M. Broy. “A meta model for artefact-orientation: fundamentals and lessons learned in requirements engineering”. In: *International Conference on Model Driven Engineering Languages and Systems*. Springer. 2010, pp. 183–197.

[149] D. Méndez Fernández and R. Wieringa. “Improving requirements engineering by artefact orientation”. In: *International Conference on Product Focused Software Process Improvement*. Springer. 2013, pp. 108–122.

[150] H. Sharp, A. Finkelstein, and G. Galal. “Stakeholder identification in the requirements engineering process”. In: *Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*. Ieee. 1999, pp. 387–391.

[151] T. Dybå, D. I. Sjøberg, and D. 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*. 2012, pp. 19–28.

[152] L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh. “The case for context-driven software engineering research: generalizability is overrated”. In: *IEEE Software* 34.5 (2017), pp. 72–75.

[153] L. Montgomery, C. Lüders, and W. Maalej. “An alternative issue tracking dataset of public jira repositories”. In: *Proceedings of the 19th International Conference on Mining Software Repositories*. 2022, pp. 73–77.

[154] H. Femmer. “Requirements engineering artifact quality: definition and control”. PhD thesis. Technische Universität München, 2017.

[155] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh. “Extending nocuous ambiguity analysis for anaphora in natural language requirements”. In: *RE*. 2010.

[156] O. Ormandjieva, I. Hussain, and L. Kosseim. “Toward a text classification system for the quality assessment of software requirements written in natural language”. In: *SOQUA*. 2007.

[157] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper. “Improving agile requirements: the quality user story framework and tool”. In: *Requirements engineering* 21.3 (2016), pp. 383–403.

[158] G. Génova, J. M. Fuentes, J. Llorens, O. Hurtado, and V. Moreno. “A framework to measure and improve the quality of textual requirements”. In: *Requirements engineering* 18.1 (2013), pp. 25–41.

[159] R. Saavedra, L. C. Ballejos, and M. A. Ale. “Software Requirements Quality Evaluation: State of the art and research challenges”. In: *ASSE-JAIIO*. 2013.

[160] L. M. Garshol. “Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all”. In: *Journal of information science* 30.4 (2004).

[161] D. Kundisch, J. Muntermann, A. M. Oberländer, D. Rau, M. Röglinger, T. Schoormann, and D. Szopinski. “An Update for Taxonomy Designers”. In: *Business & Information Systems Engineering* (2021), pp. 1–19.

[162] M. Unterkalmsteiner and T. Gorschek. “Requirements quality assurance in industry: why, what and how?” In: *REFSQ*. Springer. 2017.

[163] F. Fabbrini, M. Fusani, V. Gervasi, S. Gnesi, and S. Ruggieri. “Achieving quality in natural language requirements”. In: *QW*. 1998.

[164] G. A. Miller. “The magical number seven, plus or minus two: Some limits on our capacity for processing information.” In: *Psychological review* 63.2 (1956), p. 81.

[165] R. E. Schneider. *A process for building a more effective set of requirement goodness properties*. George Mason University, 2002.

[166] *Systems and software engineering □ Life cycle processes □ Requirements engineering*. Standard. Geneva, CH: International Organization for Standardization, Nov. 2018.

[167] C. P. Usdadiya. “Assessing quality of use case specifications”. PhD thesis. Dhirubhai Ambani Institute of Information and Communication Technology, 2018.

[168] E. Juergens, F. Deissenboeck, M. Feilkas, B. Hummel, B. Schaetz, S. Wagner, C. Domann, and J. Streit. “Can clone detection support quality assessments of requirements specifications?” In: *ICSE*. 2010, pp. 79–88.

[169] H. Hasso, M. Dembach, H. Geppert, and D. Toews. “Detection of Defective Requirements using Rule-based Scripts.” In: *REFSQ Workshops*. 2019.

[170] J. Frattini. “Identifying relevant factors of requirements quality: an industrial case study”. In: *Requirements Engineering: Foundation for Software Quality: 30th International Working Conference, REFSQ 2024, Winterthur, Switzerland, April 8–11, 2024, Proceedings 30*. Springer. 2024.

[171] L. Kof. “Treatment of passive voice and conjunctions in use case documents”. In: *Natural Language Processing and Information Systems: 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, Paris, France, June 27-29, 2007. Proceedings 12*. Springer. 2007, pp. 181–192.

[172] J. Krisch and F. Houdek. “The myth of bad passive voice and weak words an empirical investigation in the automotive industry”. In: *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE. 2015, pp. 344–351.

[173] H. Femmer. “Requirements Quality Defect Detection with the Qualicen Requirements Scout.” In: *REFSQ Workshops*. 2018.

[174] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano. “On the effectiveness of screen mockups in requirements engineering: results from an internal replication”. In: *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*. 2010, pp. 1–10.

[175] D. Zowghi and N. Nurmuliani. “A study of the impact of requirements volatility on software project performance”. In: *Ninth Asia-Pacific Software Engineering Conference, 2002.* IEEE. 2002, pp. 3–11.

[176] E. Knauss, C. El Boustani, and T. Flohr. “Investigating the impact of software requirements specification quality on project success”. In: *Product-Focused Software Process Improvement: 10th International Conference, PROFES 2009, Oulu, Finland, June 15-17, 2009. Proceedings 10*. Springer. 2009, pp. 28–42.

[177] K. Chari and M. Agrawal. “Impact of incorrect and new requirements on waterfall software project outcomes”. In: *Empirical Software Engineering* 23 (2018), pp. 165–185.

[178] M. Borg, P. Runeson, and A. Ardö. “Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability”. In: *Empirical Software Engineering* 19 (2014), pp. 1565–1616.

[179] S. Charalampidou, A. Ampatzoglou, E. Karountzos, and P. Avgeriou. “Empirical studies on software traceability: A mapping study”. In: *Journal of Software: Evolution and Process* 33.2 (2021), e2294.

[180] K. Moløkken and M. Jørgensen. “Expert estimation of web-development projects: are software professionals in technical roles more optimistic than those in non-technical roles?” In: *Empirical Software Engineering* 10 (2005), pp. 7–30.

[181] J. Cohen. “A coefficient of agreement for nominal scales”. In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[182] I. Sommerville. “Software process models”. In: *ACM computing surveys (CSUR)* 28.1 (1996), pp. 269–271. DOI: 10.1145/234313.234420.

[183] J. Münch, O. Armbrust, M. Kowalczyk, and M. Sotó. *Software process definition and management*. Springer, 2012.

[184] W. W. Royce. “Managing the development of large software systems: concepts and techniques”. In: *Proceedings of the 9th international conference on Software Engineering*. 1987, pp. 328–338.

[185] B. W. Boehm. “A spiral model of software development and enhancement”. In: *Computer* 21.5 (1988), pp. 61–72.

[186] B. Boehm and J. A. Lane. “Guide for using the Incremental Commitment Model (ICM) for systems engineering of DoD projects”. In: *usc-csse-2009-500* (2008).

[187] I. Jacobson, G. Booch, and J. Rumbaugh. *The unified software development process*. 1999.

[188] H. D. Mills, M. Dyer, and R. C. Linger. “Cleanroom software engineering”. In: (1987).

[189] P. Bourque and R. Fairley. “Swebok”. In: *Nd: IEEE Computer society* (2004).

[190] Y. Murakami, M. Tsunoda, and H. Uwano. “WAP: Does reviewer age affect code review performance?” In: *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE. 2017, pp. 164–169.

[191] J. Natt och Dag, T. Thelin, and B. Regnell. “An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development”. In: *Empirical Software Engineering* 11 (2006), pp. 303–329.

[192] K. Wnuk, M. Höst, and B. Regnell. “Replication of an experiment on linguistic tool support for consolidation of requirements from multiple sources”. In: *Empirical Software Engineering* 17 (2012), pp. 305–344.

[193] K. Großer, M. Rukavitsyna, and J. Jürjens. “A Comparative Evaluation of Requirement Template Systems”. In: *2023 IEEE 31st International Requirements Engineering Conference (RE)*. IEEE. 2023, pp. 41–52.

[194] D. G. Feitelson. “From repeatability to reproducibility and corroboration”. In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pp. 3–11.

[195] J. T. Cacioppo, R. M. Kaplan, J. A. Krosnick, J. L. Olds, and H. Dean. “Social, behavioral, and economic sciences perspectives on robust and reliable science”. In: *Report of the Subcommittee on Replicability in Science Advisory Committee to the National Science Foundation Directorate for Social, Behavioral, and Economic Sciences* 1 (2015).

[196] B. C. Anda, D. I. Sjøberg, and A. Mockus. “Variability and reproducibility in software engineering: A study of four companies that developed the same system”. In: *TSE* 35.3 (2008), pp. 407–429.

[197] M. McNutt. *Reproducibility*. 2014.

[198] J. Tennant, J. Beamer, J. Bosman, B. Brembs, N. C. Chung, G. Clement, T. Crick, J. Dugan, A. Dunning, et al. “Foundations for open scholarship strategy development”. In: (2019).

[199] S. Winter, C. S. Timperley, B. Hermann, J. Cito, J. Bell, M. Hilton, and D. Beyer. “A retrospective study of one decade of artifact evaluations”. In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2022, pp. 145–156.

[200] A. Landi, M. Thompson, V. Giannuzzi, F. Bonifazi, I. Labastida, L. O. B. da Silva Santos, and M. Roos. “The □A□ of FAIR □ As Open as Possible, as Closed as Necessary”. In: *Data Intelligence* 2.1-2 (Jan. 2020), pp. 47–55. ISSN: 2641-435X. DOI: 10.1162/dint_a_00027. URL: https://doi.org/10.1162/dint%5C_a%5C_00027.

[201] M. Shepperd, N. Ajienka, and S. Counsell. “The role and value of replication in empirical software engineering results”. In: *Information and Software Technology* 99 (2018), pp. 120–132.

[202] S. Krishnamurthi and J. Vitek. “The real software crisis: Repeatability as a core value”. In: *Communications of the ACM* 58.3 (2015), pp. 34–36. DOI: 10.1145/2658987.

[203] M. C. Kidwell, L. B. Lazarević, E. Baranski, T. E. Hardwicke, S. Piechowski, L.-S. Falkenberg, C. Kennett, A. Slowik, et al. “Badges to acknowledge open practices: A simple, low-cost, effective method for increasing transparency”. In: *PLoS biology* 14.5 (2016), e1002456.

[204] B. Hermann, S. Winter, and J. Siegmund. “Community expectations for research artifacts and evaluation processes”. In: *Proceedings of the 28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 2020, pp. 469–480.

[205] B. A. Nosek, C. R. Ebersole, A. C. DeHaven, and D. T. Mellor. “The preregistration revolution”. In: *Proceedings of the National Academy of Sciences* 115.11 (2018), pp. 2600–2606.

[206] M. Gabelica, R. Bojčić, and L. Puljak. “Many researchers were not compliant with their published data sharing statement: mixed-methods study”. In: *Journal of Clinical Epidemiology* (2022).

[207] C. Wacharamanotham, L. Eisenring, S. Haroz, and F. Echtler. “Transparency of CHI research artifacts: Results of a self-reported survey”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14.

[208] M.-A. Sicilia, E. García-Barriocanal, and S. Sánchez-Alonso. “Community curation in open dataset repositories: insights from Zenodo”. In: *Procedia Computer Science* 106 (2017), pp. 54–60.

[209] D. Mendez and S. Wagner. “Naming the Pain in Requirements Engineering: Design of a global Family of Surveys and first Results from Germany”. In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. 2013, pp. 183–194.

[210] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin. “A model for technology transfer in practice”. In: *IEEE software* 23.6 (2006), pp. 88–95.

[211] S. Baltes and P. Ralph. “Sampling in software engineering research: A critical review and guidelines”. In: *Empirical Software Engineering* 27.4 (2022), p. 94.

[212] L. Rosen. *Open source licensing - Software Freedom and Intellectual Property Law*. Prentice Hall, 2004.

[213] F.-L. Li, J. Horkoff, L. Liu, A. Borgida, G. Guizzardi, and J. Mylopoulos. “Engineering requirements with desiree: An empirical evaluation”. In: *International Conference on Advanced Information Systems Engineering*. Springer. 2016, pp. 221–238.

[214] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer. “Automated checking of conformance to requirements templates using natural language processing”. In: *IEEE transactions on Software Engineering* 41.10 (2015), pp. 944–968.

[215] A. Ferrari, S. Gnesi, and G. Tolomei. “Using clustering to improve the structure of natural language requirements documents”. In: *Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings 19*. Springer. 2013, pp. 34–49.

[216] A. M. Rago, P. Frade, M. Ruival, and C. Marcos. “An Approach for Automating Use Case Refactoring”. In: (2014).

[217] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh. “Speculative requirements: Automatic detection of uncertainty in natural language requirements”. In: *2012 20th IEEE International Requirements Engineering Conference (RE)*. IEEE. 2012, pp. 11–20.

[218] F. Mokammel, E. Coatanéa, J. Coatanéa, V. Nenchev, E. Blanco, and M. Pietola. “Automatic requirements extraction, analysis, and graph representation using an approach derived from computational linguistics”. In: *Systems Engineering* 21.6 (2018), pp. 555–575.

[219] T. Al Balushi, O. Khod, P. R. F. Sampaio, M. Patel, B. S. W. Manchester, O. Corcho, and P. Loucopoulos. “Identifying NFRs conflicts using quality ontologies”. In: *SEKE 2008* (2008), p. 929.

[220] A. Ferrari, F. dell’Orletta, G. O. Spagnolo, and S. Gnesi. “Measuring and improving the completeness of natural language requirements”. In: *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings 20*. Springer. 2014, pp. 23–38.

[221] F. S. Bäumer and M. Geierhos. “Flexible ambiguity resolution and incompleteness detection in requirements descriptions via an indicator-based configuration of text analysis pipelines”. In: (2018).

[222] M. El-Attar and J. Miller. “Improving the quality of use case models using antipatterns”. In: *Software & systems modeling* 9 (2010), pp. 141–160.

[223] A. Ferrari and A. Esuli. “An NLP approach for cross-domain ambiguity detection in requirements engineering”. In: *Automated Software Engineering* 26.3 (2019), pp. 559–598.

[224] F. Dalpiaz, I. Van der Schalk, and G. Lucassen. “Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and NLP”. In: *Requirements Engineering: Foundation for Software Quality: 24th International Working Conference, REFSQ 2018, Utrecht, The Netherlands, March 19-22, 2018, Proceedings 24*. Springer. 2018, pp. 119–135.

[225] M. Glymour, J. Pearl, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[226] C. A. Furia, R. Torkar, and R. Feldt. "Applying Bayesian analysis guidelines to empirical software engineering data: The case of programming languages and code quality". In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31.3 (2022), pp. 1–38.

[227] A. Gelman, A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, L. Kennedy, J. Gabry, P.-C. Bürkner, and M. Modrák. "Bayesian workflow". In: *arXiv preprint arXiv:2011.01808* (2020).

[228] P.-C. Bürkner. "brms: An R package for Bayesian multilevel models using Stan". In: *Journal of statistical software* 80 (2017), pp. 1–28.

[229] E. T. Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[230] J. S. Wesner and J. P. Pomeranz. "Choosing priors in Bayesian ecological models by simulating from the prior predictive distribution". In: *Ecosphere* 12.9 (2021), e03739.

[231] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner. "Rank-normalization, folding, and localization: An improved R ^ for assessing convergence of MCMC (with discussion)". In: *Bayesian analysis* 16.2 (2021), pp. 667–718.

[232] J. J. Romano and J. D. Palmer. "TBRIM: decision support for validation/verification of requirements". In: *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*. Vol. 3. IEEE. 1998, pp. 2489–2494.

[233] Y. Wang, T. Wang, and J. Sun. "PASER: a pattern-based approach to service requirements analysis". In: *International Journal of Software Engineering and Knowledge Engineering* 29.04 (2019), pp. 547–576.

[234] L. Montgomery, J. Frattini, D. Mendez, D. Fucci, M. Unterkalmsteiner, and J. Fischbach. *Open Science - Artefact Management Guideline*. Date Last Accessed: 2023.07.12. 2023. DOI: 10.5281/zenodo.8134403. URL: https://docs.google.com/document/d/1gIg3g-_zxCeiw2IJkBGbiGI9-3HeQU5FR63yAv3PhiM.

[235] D. Méndez Fernández, W. Böhm, A. Vogelsang, J. Mund, M. Broy, M. Kuhrmann, and T. Weyer. "Artefacts in software engineering: a fundamental positioning". In: *Software & Systems Modeling* 18 (2019), pp. 2777–2786.

[236] M. T. Baldassarre, N. Ernst, B. Hermann, T. Menzies, and R. Yedida. "(Re) Use of Research Results (Is Rampant)". In: *Communications of the ACM* 66.2 (2023), pp. 75–81.

[237] S. Abrahao, C. Gravino, E. Insfran, G. Scanniello, and G. Tortora. “Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments”. In: *IEEE transactions on software engineering* 39.3 (2012), pp. 327–342.

[238] S. Ducasse and M. Lanza. “The class blueprint: visually supporting the understanding of glasses”. In: *IEEE Transactions on Software Engineering* 31.1 (2005), pp. 75–90.

[239] G. Scanniello, C. Gravino, M. Genero, J. A. Cruz-Lemus, G. Tortora, M. Risi, and G. Dodero. “Do software models based on the UML aid in source-code comprehensibility? Aggregating evidence from 12 controlled experiments”. In: *Empirical Software Engineering* 23 (2018), pp. 2695–2733.

[240] European Organization For Nuclear Research and OpenAIRE. *Zenodo General Policies v1.0*. `https://about.zenodo.org/policies/`. Accessed 2024-01-15. 2013.

[241] J. J. Lee. *Demystify statistical significance□time to move on from the p value to Bayesian analysis*. 2011.

[242] M. Bano. “Addressing the challenges of requirements ambiguity: A review of empirical literature”. In: *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*. IEEE. 2015, pp. 21–24.

[243] Y. Benjamini and Y. Hochberg. “Controlling the false discovery rate: A practical and powerful approach to multiple testing”. In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), pp. 289–300.

[244] J. Barnes and S. J. Linning. “Statistical Power, P-Values, and the Positive Predictive Value”. In: *The Encyclopedia of Research Methods in Criminology and Criminal Justice* 1 (2021), pp. 337–343.

[245] T. Menzies and M. Shepperd. “□Bad smells□ in software analytics papers”. In: *Information and software technology* 112 (2019), pp. 35–47. DOI: `10.1016/j.infsof.2019.04.005`.

[246] R. Torkar, R. Feldt, and C. A. Furia. “Bayesian data analysis in empirical software engineering: The case of missing data”. In: *Contemporary Empirical Methods in Software Engineering* (2020), pp. 289–324.

[247] N. A. Ernst. “Bayesian hierarchical modelling for tailoring metric thresholds”. In: *Proceedings of the 15th international conference on mining software repositories*. 2018, pp. 587–591. DOI: `10.1145/3196398.3196443`.

[248] F. Elwert. “Graphical causal models”. In: *Handbook of causal analysis for social research*. Springer, 2013, pp. 245–273.

[249] E. T. Jaynes. *Probability theory: The logic of science*. Cambridge: Cambridge University Press, 2003.

[250] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[251] T. D. Cook, D. T. Campbell, and A. Day. *Quasi-experimentation: Design & analysis issues for field settings*. Vol. 351. Houghton Mifflin Boston, 1979.

[252] J. Frattini, D. Fucci, and S. Vegas. *Replication Package for: Crossover Designs in Software Engineering Experiments: Review of the State of Analysis*. https://zenodo.org/doi/10.5281/zenodo.13312638. 2024.

[253] J. Sandobalin, E. Insfran, and S. Abrahao. "On the effectiveness of tools to support infrastructure as code: Model-driven versus code-centric". In: *IEEE Access* 8 (2020), pp. 17734–17761. DOI: 10.1109/ACCESS.2020.2966597.

[254] M. Trkman, J. Mendling, P. Trkman, and M. Krisper. "Impact of the conceptual model's representation format on identifying and understanding user stories". In: *Information and software technology* 116 (2019), p. 106169. DOI: 10.1016/j.infsof.2019.08.001.

[255] J. Bogner, S. Kotstein, and T. Pfaff. "Do RESTful API design rules have an impact on the understandability of Web APIs?" In: *Empirical software engineering* 28.6 (2023), p. 132. DOI: 10.1007/s10664-023-10367-y.

[256] L. M. Pickard, B. A. Kitchenham, and P. W. Jones. "Combining empirical results in software engineering". In: *Information and software technology* 40.14 (1998), pp. 811–821. DOI: 10.1016/S0950-5849(98)00101-3.

[257] J. R. Woods, J. G. Williams, and M. Tavel. "The two-period crossover design in medical research". In: *Annals of internal medicine* 110.7 (1989), pp. 560–566. DOI: 10.7326/0003-4819-110-7-560.

[258] B. A. Kitchenham, T. Dyba, and M. Jorgensen. "Evidence-based software engineering". In: *Proceedings. 26th International Conference on Software Engineering*. IEEE. 2004, pp. 273–281. DOI: 10.1109/ICSE.2004.1317449.

[259] V. B. Kampenes, T. Dybå, J. E. Hannay, and D. I. Sjøberg. "A systematic review of quasi-experiments in software engineering". In: *Information and Software Technology* 51.1 (2009), pp. 71–82. DOI: 10.1016/j.infsof.2008.04.006.

[260] J. Hannay and M. Jørgensen. "The Role of Deliberate Artificial Design Elements in Software Engineering Experiments". In: *IEEE Transactions on Software Engineering* 34.2 (2008), pp. 242–259. DOI: 10.1109/TSE.2008.13.

[261] B. Kitchenham, L. Madeyski, and P. Brereton. "Problems with statistical practice in human-centric software engineering experiments". In: *Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering*. 2019, pp. 134–143. DOI: 10.1145/3319008.3319009.

[262] L. Madeyski and B. Kitchenham. “Effect sizes and their variance for AB/BA crossover design studies”. In: *Proceedings of the 40th International Conference on Software Engineering*. 2018, pp. 420–420. DOI: 10.1007/s10664-017-9574-5.

[263] B. Kitchenham, L. Madeyski, G. Scanniello, and C. Gravino. “The Importance of the Correlation in Crossover Experiments”. In: *IEEE Transactions on Software Engineering* 48.8 (2021), pp. 2802–2813. DOI: 10.1109/TSE.2021.3070480.

[264] N. A. Cruz Gutierrez, O. O. Melo, and C. A. Martinez. “Semiparametric generalized estimating equations for repeated measurements in cross-over designs”. In: *Statistical Methods in Medical Research* 32.5 (2023), pp. 1033–1050. DOI: 10.1177/09622802231158736.

[265] N. Cruz, O. Melo, and C. Martinez. “A correlation structure for the analysis of Gaussian and non-Gaussian responses in crossover experimental designs with repeated measures”. In: *Statistical Papers* 65.1 (2024), pp. 263–290. DOI: 10.1007/s00362-022-01391-z.

[266] M. T. Baldassarre, D. Caivano, D. Fucci, S. Romano, and G. Scanniello. “Affective reactions and test-driven development: Results from three experiments and a survey”. In: *Journal of Systems and Software* 185 (2022), p. 111154. DOI: 10.1016/j.jss.2021.111154.

[267] M. Esposito, S. Romano, and G. Scanniello. “Test-Driven Development and Embedded Systems: An Exploratory Investigation”. In: *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE. 2023, pp. 239–246. DOI: 10.1109/SEAA60479.2023.00045.

[268] D. Fucci, G. Scanniello, S. Romano, M. Shepperd, B. Sigweni, F. Uyaguari, B. Turhan, N. Juristo, and M. Oivo. “An external replication on the effects of test-driven development using a multi-site blind analysis approach”. In: *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2016, pp. 1–10. DOI: 10.1145/2961111.2962592.

[269] S. Romano, G. Scanniello, M. T. Baldassarre, and D. Fucci. “On the Effect of Noise on Software Engineers’ Performance: Results from Two Replicated Experiments”. In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE. 2020, pp. 334–341. DOI: 10.1109/SEAA51224.2020.00062.

[270] R. Francese, M. Risi, and G. Tortora. “miniJava: Automatic Miniaturization of Java Applications”. In: *Proceedings of the International Conference on Advanced Visual Interfaces*. 2020, pp. 1–8. DOI: 10.1145/3399715.3399847.

[271] R. Coppola, T. Fulcini, L. Ardito, M. Torchiano, and E. Alègroth. “On Effectiveness and Efficiency of Gamified Exploratory GUI Testing”. In: *IEEE Transactions on Software Engineering* (2023). DOI: 10.1109/TSE.2023.3348036.

[272] M. U. Khan, H. Sartaj, M. Z. Iqbal, M. Usman, and N. Arshad. “Aspectocl: using aspects to ease maintenance of evolving constraint specification”. In: *Empirical Software Engineering* 24 (2019), pp. 2674–2724. DOI: 10.1007/s10664-019-09717-6.

[273] K. Schneid, S. Thöne, and H. Kuchen. “Semi-automated test migration for BPMN-based process-driven applications”. In: *International Conference on Enterprise Design, Operations, and Computing*. Springer. 2022, pp. 237–254. DOI: 10.1007/978-3-031-17604-3_14.

[274] H. Bünder and H. Kuchen. “Towards behavior-driven graphical user interface testing”. In: *ACM SIGAPP Applied Computing Review* 19.2 (2019), pp. 5–17. DOI: 10.1145/3357385.3357386.

[275] D. Moreno-Lumbreras, G. Robles, D. Izquierdo-Cortázar, and J. M. Gonzalez-Barahona. “Software development metrics: to VR or not to VR”. In: *Empirical Software Engineering* 29.2 (2024), pp. 1–49. DOI: 10.1007/s10664-023-10435-3.

[276] C. Iñiguez-Jarrín, J. I. Panach, and O. P. López. “Improvement of usability in user interfaces for massive data analysis: an empirical study”. In: *Multimedia Tools and Applications* 79.17 (2020), pp. 12257–12288. DOI: 10.1007/s11042-019-08456-6.

[277] D. Fucci, S. Romano, M. T. Baldassarre, D. Caivano, G. Scanniello, B. Turhan, and N. Juristo. “A longitudinal cohort study on the retainment of test-driven development”. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2018, pp. 1–10. DOI: 10.1145/3239235.3240502.

[278] S. Ezzini, S. Abualhaija, C. Arora, and M. Sabetzadeh. “Automated handling of anaphoric ambiguity in requirements: A multi-solution study”. In: *Proceedings of the 44th International Conference on Software Engineering*. 2022, pp. 187–199.

[279] K.-J. Stol and B. Fitzgerald. “The ABC of software engineering research”. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 27.3 (2018), pp. 1–51.

[280] J. Frattini. *Replication Package for the ”Applying Bayesian Data Analysis for Causal Inference about Requirements Quality: a Controlled Experiment”*. https://zenodo.org/doi/10.5281/zenodo.10423665. Accessed: 2024-06-21. 2024.

[281] E. J. Philippo, W. Heijstek, B. Kruiswijk, M. R. Chaudron, and D. M. Berry. “Requirement ambiguity not as important as expected□results of an empirical evaluation”. In: *Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings 19*. Springer. 2013, pp. 65–79.

[282] W. O’Grady, J. Archibald, M. Aronoff, and J. Rees-Miller. *Contemporary Linguistics: An Introduction*. Boston: Bedford/St. Martin’s, 2001. ISBN: 978-0-312-24738-6.

[283] B. Gleich, O. Creighton, and L. Kof. “Ambiguity detection: Towards a tool explaining ambiguity sources”. In: *Requirements Engineering: Foundation for Software Quality: 16th International Working Conference, REFSQ 2010, Essen, Germany, June 30–July 2, 2010. Proceedings 16*. Springer. 2010, pp. 218–232.

[284] E. Knauss, K. Schneider, and K. Stapel. “Learning to write better requirements through heuristic critiques”. In: *2009 17th IEEE International Requirements Engineering Conference*. IEEE. 2009, pp. 387–388.

[285] B. Rosadini, A. Ferrari, G. Gori, A. Fantechi, S. Gnesi, I. Trotta, and S. Bacherini. “Using NLP to detect requirements defects: An industrial experience in the railway domain”. In: *Requirements Engineering: Foundation for Software Quality: 23rd International Working Conference, REFSQ 2017, Essen, Germany, February 27–March 2, 2017, Proceedings 23*. Springer. 2017, pp. 344–360.

[286] M. Soeken, N. Abdessaied, A. Allahyari-Abhari, A. Buzo, L. Musat, G. Pelz, and R. Drechsler. “Quality assessment for requirements based on natural language processing”. In: *Forum on Specification and Design Languages. Proceedings*. Citeseer. 2014.

[287] R. Drechsler, M. Soeken, and R. Wille. “Automated and quality-driven requirements engineering”. In: *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE. 2014, pp. 586–590.

[288] M. Poesio. “Semantic Ambiguity and Perceived Ambiguity”. In: *Semantic Ambiguity and Underspecification*. Ed. by K. van Deemter and S. Peters. United Kingdom: Center for the Study of Language and Inf, 1996. DOI: 10.48550/arXiv.cmp-lg/9505034.

[289] E. Kamsties and B. Peach. “Taming ambiguity in natural language requirements”. In: *Proceedings of the Thirteenth international conference on Software and Systems Engineering and Applications*. Vol. 1315. 2000.

[290] R. Sharma, N. Sharma, and K. Biswas. “Machine learning for detecting pronominal anaphora ambiguity in NL requirements”. In: *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*. IEEE. 2016, pp. 177–182.

[291] S. Ezzini, S. Abualhaija, C. Arora, and M. Sabetzadeh. “TAPHSIR: towards AnaPHoric ambiguity detection and ReSolution in requirements”. In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2022, pp. 1677–1681.

[292] U. S. Shah and D. C. Jinwala. “Resolving ambiguity in natural language specification to generate UML diagrams for requirements specification”. In: *International Journal of Software Engineering, Technology and Applications* 1.2-4 (2015), pp. 308–334.

[293] E. Kamsties, A. von Knethen, and J. Philipps. “An empirical investigation of requirements specification languages: Detecting defects while formalizing requirements”. In: *Information Modeling Methods and Methodologies: Advanced Topics in Database Research*. IGI Global, 2005, pp. 125–147.

[294] F. de Bruijn and H. L. Dekkers. “Ambiguity in natural language software requirements: A case study”. In: *Requirements Engineering: Foundation for Software Quality: 16th International Working Conference, REFSQ 2010, Essen, Germany, June 30–July 2, 2010. Proceedings 16*. Springer. 2010, pp. 233–247.

[295] G. Belev. “Guidelines for specification development”. In: *Proceedings., Annual Reliability and Maintainability Symposium*. IEEE. 1989, pp. 15–21.

[296] S. Boyd, D. Zowghi, and A. Farroukh. “Measuring the expressiveness of a constrained natural language: An empirical study”. In: *13th IEEE International Conference on Requirements Engineering (RE’05)*. IEEE. 2005, pp. 339–349.

[297] D. Firesmith. “Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them.” In: *J. Object Technol.* 6.1 (2007), pp. 17–33.

[298] M. G. Christel and K. C. Kang. *Issues in requirements elicitation*. 1992.

[299] J. Pearl. “From Bayesian networks to causal networks”. In: *Mathematical models for handling partial knowledge in artificial intelligence*. Springer, 1995, pp. 157–182.

[300] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1 (2017).

[301] R. Vieira, D. Mesquita, C. L. Mattos, R. Britto, L. Rocha, and J. Gomes. “Bayesian Analysis of Bug-Fixing Time using Report Data”. In: *Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2022, pp. 57–68.

[302] W. Levén, H. Broman, T. Besker, and R. Torkar. “The Broken Windows Theory Applies to Technical Debt”. In: *arXiv preprint arXiv:2209.01549* (2022).

[303] D. Russo and K.-J. Stol. “Gender differences in personality traits of software engineers”. In: *IEEE Transactions on Software Engineering* 48.3 (2020), pp. 819–834.

[304] R. B. Svensson, R. Feldt, and R. Torkar. “The unfulfilled potential of data-driven decision making in agile software development”. In: *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings 20*. Springer. 2019, pp. 69–85.

[305] L. Gren and R. Berntsson Svensson. “Is it possible to disregard obsolete requirements? a family of experiments in software effort estimation”. In: *Requirements Engineering* 26.3 (2021), pp. 459–480.

[306] R. Berntsson Svensson and R. Torkar. “Not all requirements prioritization criteria are equal at all times: A quantitative analysis”. In: *Journal of Systems and Software* 209 (2024), p. 111909. ISSN: 0164-1212. DOI: 10.1016/j.jss.2023.111909.

[307] J. C. Carver. “Towards reporting guidelines for experimental replications: A proposal”. In: *1st international workshop on replication in empirical software engineering*. Vol. 1. 2010, pp. 1–4.

[308] A. Ferrari, G. O. Spagnolo, and S. Gnesi. “Pure: A dataset of public requirements documents”. In: *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE. 2017, pp. 502–505.

[309] A. Jedlitschka, M. Ciolkowski, and D. Pfahl. “Reporting experiments in software engineering”. In: *Guide to advanced empirical software engineering* (2008), pp. 201–228.

[310] I. Salman, A. T. Misirli, and N. Juristo. “Are students representatives of professionals in software engineering experiments?” In: *2015 IEEE/ACM 37th IEEE international conference on software engineering*. Vol. 1. IEEE. 2015, pp. 666–676.

[311] J. Carver, L. Jaccheri, S. Morasca, and F. Shull. “Issues in using students in empirical studies in software engineering education”. In: *Proceedings. 5th international workshop on enterprise networking and computing in healthcare industry (IEEE Cat. No. 03EX717)*. IEEE. 2004, pp. 239–249.

[312] B. W. Brown Jr. “The crossover experiment for clinical trials”. In: *Biometrics* (1980), pp. 69–79.

[313] H. Hsu and P. A. Lachenbruch. “Paired t test”. In: *Wiley StatsRef: statistics reference online* (2014).

[314] F. Wilcoxon. *Individual comparisons by ranking methods. Biom Bull 1 (6): 80–83*. 1945.

[315] S. S. Shapiro and M. B. Wilk. “An analysis of variance test for normality (complete samples)”. In: *Biometrika* 52.3/4 (1965), pp. 591–611.

[316] T. Dybå, V. B. Kampenes, and D. I. Sjøberg. “A systematic review of statistical power in software engineering experiments”. In: *Information and Software Technology* 48.8 (2006), pp. 745–755.

[317] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 1969.

[318] B. M. King, P. J. Rosopa, and E. W. Minium. *Statistical reasoning in the behavioral sciences*. John Wiley & Sons, 2018.

[319] A. Nilsson, C. Bonander, U. Strömberg, and J. Björk. “A directed acyclic graph for interactions”. In: *International Journal of Epidemiology* 50.2 (2021), pp. 613–619.

[320] A. Demaris. *Logit modeling: Practical applications*. 86. Sage, 1992.

[321] B. H. Cohen. *Explaining psychological statistics*. John Wiley & Sons, 2008.

[322] D. Badampudi, C. Wohlin, and T. Gorschek. “Contextualizing research evidence through knowledge translation in software engineering”. In: *Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering*. 2019, pp. 306–311.

[323] W. Martyniuk. “Common European Framework of Reference for Languages: Learning, teaching, assessment (CEFR)–a synopsis”. In: *Annual meeting of the consortium for language teaching and learning cornell university. Concil of Europe, Language policy division. https://rm. coe. int/16802fc1bf*. 2006.

[324] D. Lo, N. Nagappan, and T. Zimmermann. “How practitioners perceive the relevance of software engineering research”. In: *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 2015, pp. 415–425. DOI: 10.1145/2786805.2786809.

[325] P. Devanbu, T. Zimmermann, and C. Bird. “Belief & evidence in empirical software engineering”. In: *Proceedings of the 38th international conference on software engineering*. 2016, pp. 108–119. DOI: 10.1145/2884781.2884812.

[326] D. S. Cruzes and T. Dybå. “Research synthesis in software engineering: A tertiary study”. In: *Information and Software Technology* 53.5 (2011), pp. 440–455. DOI: 10.1016/j.infsof.2011.01.004.

[327] Y. Rafique and V. B. Mišić. “The effects of test-driven development on external quality and productivity: A meta-analysis”. In: *IEEE Transactions on Software Engineering* 39.6 (2012), pp. 835–856. DOI: 10.1109/TSE.2012.28.

[328] W. Hayes. “Research synthesis in software engineering: a case for meta-analysis”. In: *Proceedings Sixth International Software Metrics Symposium (Cat. No. PR00403)*. IEEE. 1999, pp. 143–151.

[329] J. Frattini. *Replication Package for the ”Replications, Revisions, and Reanalyses: Managing Variance Theories in Software Engineering”*. https://doi.org/10.5281/zenodo.14288346. Accessed: 2024-12-06. 2024.

[330] D. Budgen, B. Kitchenham, and P. Brereton. “The case for knowledge translation”. In: *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE. 2013, pp. 263–266. DOI: 10.1109/ESEM.2013.41.

[331] S. Beecham, P. O’Leary, S. Baker, I. Richardson, and J. Noll. “Making software engineering research relevant”. In: *Computer* 47.4 (2014), pp. 80–83. DOI: 10.1109/MC.2014.92.

[332] B. Cartaxo, G. Pinto, E. Vieira, and S. Soares. “Evidence briefings: Towards a medium to transfer knowledge from systematic reviews to practitioners”. In: *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement*. 2016, pp. 1–10. DOI: 10.1145/2961111.2962603.

[333] J. Miller. “Replicating software engineering experiments: a poisoned chalice or the holy grail”. In: *Information and Software Technology* 47.4 (2005), pp. 233–244. DOI: 10.1016/j.infsof.2004.08.005.

[334] V. R. Basili, F. Shull, and F. Lanubile. “Building knowledge through families of experiments”. In: *IEEE Transactions on Software Engineering* 25.4 (1999), pp. 456–473. DOI: 10.1109/32.799939.

[335] R. Rosenthal and M. R. DiMatteo. “Meta-analysis: Recent developments in quantitative methods for literature reviews”. In: *Annual review of psychology* 52.1 (2001), pp. 59–82. DOI: 10.1146/annurev.psych.52.1.59.

[336] M. Shepperd. “Combining evidence and meta-analysis in software engineering”. In: *Software Engineering: International Summer Schools, ISSSE 2009-2011, Salerno, Italy. Revised Tutorial Lectures* (2013), pp. 46–70. DOI: 10.1007/978-3-642-36054-1_2.

[337] M. Dixon-Woods, S. Agarwal, D. Jones, B. Young, and A. Sutton. “Synthesising qualitative and quantitative evidence: a review of possible methods”. In: *Journal of health services research & policy* 10.1 (2005), pp. 45–53. DOI: 10.1177/135581960501000110.

[338] O. Dieste, E. Fernández, R. G. Martínez, and N. Juristo. “Comparative analysis of meta-analysis methods: when to use which?” In: *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*. IET. 2011, pp. 36–45. DOI: 10.1049/ic.2011.0005.

[339] A. Santos, S. Vegas, M. Oivo, and N. Juristo. “A procedure and guidelines for analyzing groups of software engineering replications”. In: *IEEE Transactions on Software Engineering* 47.9 (2019), pp. 1742–1763. DOI: 10.1109/TSE.2019.2935720.

[340] A. Santos, O. Gómez, and N. Juristo. “Analyzing families of experiments in SE: A systematic mapping study”. In: *IEEE Transactions on Software Engineering* 46.5 (2018), pp. 566–583. DOI: 10.1109/TSE.2018.2864633.

[341] M. Ciolkowski. “What do we know about perspective-based reading? An approach for quantitative aggregation in software engineering”. In: *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE. 2009, pp. 133–144. DOI: 10.1109/ESEM.2009.5316026.

[342] C. Wohlin. “An evidence profile for software engineering research and practice”. In: *Perspectives on the Future of Software Engineering: Essays in Honor of Dieter Rombach* (2013), pp. 145–157. DOI: 10.1007/978-3-642-37395-4_10.

[343] S. Lewis and M. Clarke. “Forest plots: trying to see the wood and the trees”. In: *Bmj* 322.7300 (2001), pp. 1479–1480. DOI: 10.1136/bmj.322.7300.1479.

[344] R. D. Riley, P. C. Lambert, and G. Abo-Zaid. “Meta-analysis of individual participant data: rationale, conduct, and reporting”. In: *Bmj* 340 (2010). DOI: 10.1136/bmj.c221.

[345] A. J. Sutton and J. P. Higgins. “Recent developments in meta-analysis”. In: *Statistics in medicine* 27.5 (2008), pp. 625–650. DOI: 10.1002/sim.2934.

[346] B. Kitchenham, L. Madeyski, and P. Brereton. “Meta-analysis for families of experiments in software engineering: a systematic review and reproducibility and validity assessment”. In: *Empirical Software Engineering* 25 (2020), pp. 353–401. DOI: 10.1007/s10664-019-09747-0.

[347] S. Hosseini, B. Turhan, and D. Gunarathna. “A systematic literature review and meta-analysis on cross project defect prediction”. In: *IEEE Transactions on Software Engineering* 45.2 (2017), pp. 111–147. DOI: 10.1109/TSE.2017.2770124.

[348] Z. M. Zain, S. Sakri, and N. H. A. Ismail. “Application of deep learning in software defect prediction: systematic literature review and meta-analysis”. In: *Information and Software Technology* 158 (2023), p. 107175. DOI: 10.1016/j.infsof.2023.107175.

[349] J. Miller. “Applying meta-analytical procedures to software engineering experiments”. In: *Journal of Systems and Software* 54.1 (2000), pp. 29–39. DOI: 10.1016/S0164-1212(00)00024-8.

[350] S. L. Pfleeger. “Albert Einstein and empirical software engineering”. In: *Computer* 32.10 (1999), pp. 32–38. DOI: 10.1109/2.796106.

[351] M. J. Harris and R. Rosenthal. “Mediation of interpersonal expectancy effects: 31 meta-analyses.” In: *Psychological bulletin* 97.3 (1985), p. 363. DOI: 10.1037/0033-2909.97.3.363.

[352] A. J. Sutton and K. R. Abrams. “Bayesian methods in meta-analysis and evidence synthesis”. In: *Statistical methods in medical research* 10.4 (2001), pp. 277–303. DOI: 10.1177/096228020101000404.

[353] K. A. Roberts, M. Dixon-Woods, R. Fitzpatrick, K. R. Abrams, and D. R. Jones. “Factors affecting uptake of childhood immunisation: a Bayesian synthesis of qualitative and quantitative evidence”. In: *The Lancet* 360.9345 (2002), pp. 1596–1599. DOI: 10.1016/S0140-6736(02)11560-1.

[354] L. Jaccheri, Z. Kholmatova, and G. Succi. “Systematizing the Meta-Analytical Process in Software Engineering”. In: *Proceedings of the 2021 European Symposium on Software Engineering*. 2021, pp. 1–5. DOI: 10.1145/3501774.3501775.

[355] T. Dyba, T. Dingsoyr, and G. K. Hanssen. “Applying systematic reviews to diverse study types: An experience report”. In: *First international symposium on empirical software engineering and measurement (ESEM 2007)*. IEEE. 2007, pp. 225–234. DOI: 10.1109/ESEM.2007.59.

[356] M. Borenstein, L. V. Hedges, J. P. Higgins, and H. R. Rothstein. *Introduction to meta-analysis*. John Wiley & Sons, 2021.

[357] H. Akaike. “Akaike’s information criterion”. In: *International encyclopedia of statistical science* (2011), pp. 25–25.

[358] A. Vehtari, A. Gelman, and J. Gabry. “Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC”. In: *Statistics and computing* 27 (2017), pp. 1413–1432. DOI: 10.1007/s11222-016-9696-4.

[359] M. Magnusson, A. Vehtari, J. Jonasson, and M. Andersen. “Leave-one-out cross-validation for Bayesian model comparison in large data”. In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 341–351.

[360] J. Lindeløv. *Common statistical tests are linear models (or: how to teach stats)*. `https://lindeloev.github.io/tests-as-linear/`. Accessed: 2024-11-26. 2019.

[361] B. T. West, K. B. Welch, and A. T. Galecki. *Linear mixed models: a practical guide using statistical software*. Chapman and Hall/CRC, 2022.

[362] F. Vaida and S. Blanchard. “Conditional Akaike information for mixed effects models”. In: *Corrado Lagazio, Marco Marchi (Eds)* 101 (2005).

[363] M. C. Greenwood. *Intermediate statistics with R*. Montana State University, 2021.

[364] S. Nakagawa and H. Schielzeth. “A general and simple method for obtaining R2 from generalized linear mixed-effects models”. In: *Methods in ecology and evolution* 4.2 (2013), pp. 133–142.

[365] M. Shepperd, D. Bowes, and T. Hall. “Researcher bias: The use of machine learning in software defect prediction”. In: *IEEE Transactions on Software Engineering* 40.6 (2014), pp. 603–616. DOI: `10.1109/TSE.2014.2322358`.

[366] S. Romano, D. Fucci, G. Scanniello, M. T. Baldassarre, B. Turhan, and N. Juristo. “On researcher bias in software engineering experiments”. In: *Journal of Systems and Software* 182 (2021), p. 111068.