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

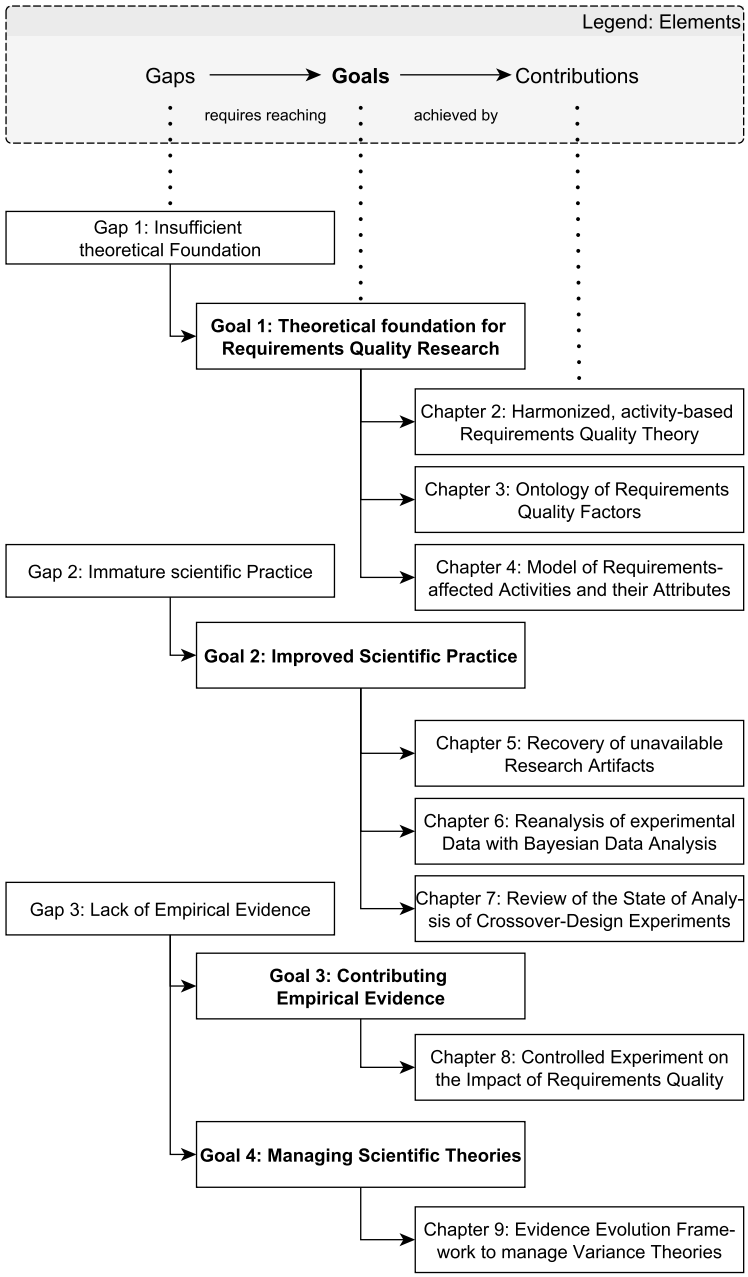

Figure 1.1: Gaps, goals, and contributions of this thesis

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-

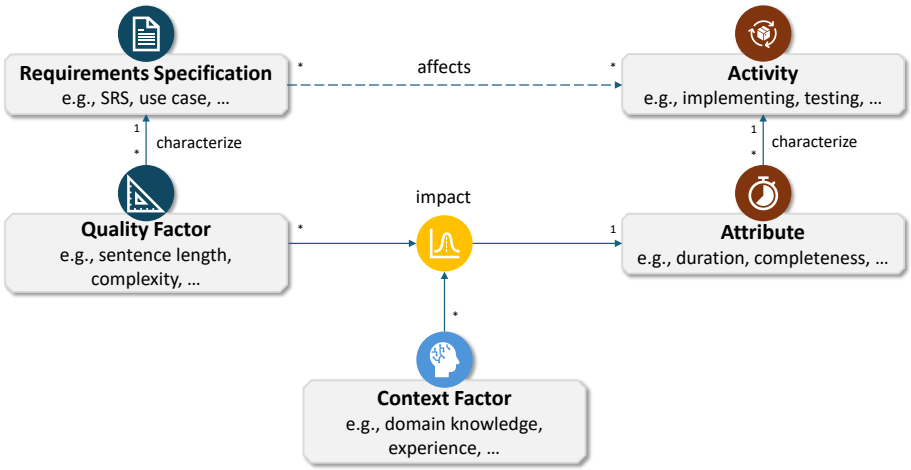

Figure 1.2: Core Concepts of the Activity-based Requirements Quality Theory

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.

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.