Applying bayesian data analysis for causal inference about requirements quality: a controlled experiment

Empirical Software Engineering (2025) 30:29
https://doi.org/10.1007/s10664-024-10582-1

Applying bayesian data analysis for causal inference about requirements quality: a controlled experiment

Julian Frattini[1] · Davide Fucci[1] · Richard Torkar[2,3] · Lloyd Montgomery[4] · Michael Unterkalmsteiner[1] · Jannik Fischbach[5,6] · Daniel Mendez[1,6]

Accepted: 24 October 2024

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 (Méndez Fernández et al. 2019), serve as input for various subsequent software engineering (SE) activities, such as deriving a software architecture, implementing features, or generating test cases (Méndez Fernández and Penzenstadler 2015). As a consequence, the quality of an SRS impacts the quality of *requirements-dependent* activities (Frattini et al. 2023; Femmer and Vogelsang 2018; Femmer et al. 2015). A quality defect in an SRS—for example, an ambiguous formulation—can cause differing interpretations

Communicated by: Irit Hadar

Extended author information available on the last page of the article

and result in the design and implementation of a solution that does not meet the stakeholders' needs (Méndez et al. 2017). The inherent complexity of natural language (NL), which is most commonly used for specifying requirements (Franch et al. 2017), aggravates this challenge further. Since quality defects are understood to scale in cost for removal (Boehm 1984), organizations are interested in identifying and removing these defects as early as possible (Montgomery et al. 2022).

Within the requirements engineering (RE) research domain, the field of requirements quality research aims to meet this challenge (Montgomery et al. 2022). Requirements quality research has already identified several attributes of requirements quality (Montgomery et al. 2022) (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 Femmer et al. 2014) as well as tools that automatically detect alleged quality defects (Femmer et al. 2017). However, existing approaches fall short in at least three regards (Frattini et al. 2023): i) only a fraction of publications provide empirical evidence that would demonstrate the impact of quality defects (Montgomery et al. 2022), ii) the few empirical studies that do so largely ignore potentially confounding context factors (Mund et al. 2015; Juergens and Deissenboeck 2010), 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*) (Femmer et al. 2014; Deissenboeck et al. 2007). These gaps have impeded the adoption of requirements quality research in practice (Franch et al. 2017).

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* (Femmer et al. 2014) and *ambiguous pronouns* (Ezzini et al. 2022). The investigation of the impact of passive voice is a conceptual replication (Baldassarre et al. 2014) of the only controlled experiment studying the impact of passive voice on domain modeling (Femmer et al. 2014) known to us. Therefore, our experiment also strengthens the robustness of their conclusions by providing diagnostic evidence (Nosek and Errington 2020). 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 (McElreath 2020). The latter has recently been popularized in SE research (Furia et al. 2019) since it generates more nuanced empirical insights. Our study is categorized as a laboratory experiment in a contrived setting (Stol and Fitzgerald 2018), 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 (Femmer et al. 2014);
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 (Frattini 2024).

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 (Femmer and Vogelsang 2018; Frattini et al. 2023). Quality defects in requirements specifications may, therefore, ultimately cause budget overrun (Philippo et al. 2013) or even project failure (Méndez et al. 2017). 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 (Franch et al. 2023; Wagner et al. 2019). Secondly, the cost of removing quality defects scales the longer they remain undetected (Boehm 1984). 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 (Montgomery et al. 2022). One main driver of this research is *requirements quality factors* (Frattini et al. 2022), 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 (Femmer et al. 2014). Automatic detection techniques using natural language processing (NLP) (Zhao et al. 2021) can automatically evaluate quality factors to detect defects in NL requirements specifications (Femmer et al. 2017).

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 (O'Grady et al. 2001). 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 (Krisch and Houdek 2015). 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 (Pohl 2016). However, while several guidelines advise against the use of passive voice based on the theoretical argument of information omission presented above (Pohl 2016; Génova et al. 2013; Gleich et al. 2010; Kof 2007; Hasso et al. 2019), 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 unproblematic as the surrounding context information compensated the omission of the semantic agent of the sentence (Krisch and Houdek 2015). 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 (Femmer et al. 2014). The limited evidence for the harmfulness of using passive voice in requirements specifications (Krisch and Houdek 2015; Femmer et al. 2014) stands in stark contrast to the amount of tools and approaches proposed to automatically detect quality defects by identifying the use of passive voice (Femmer et al. 2017; Kof 2007; Knauss et al. 2009; Femmer and Vogelsang 2018; Hasso et al. 2019; Rosadini

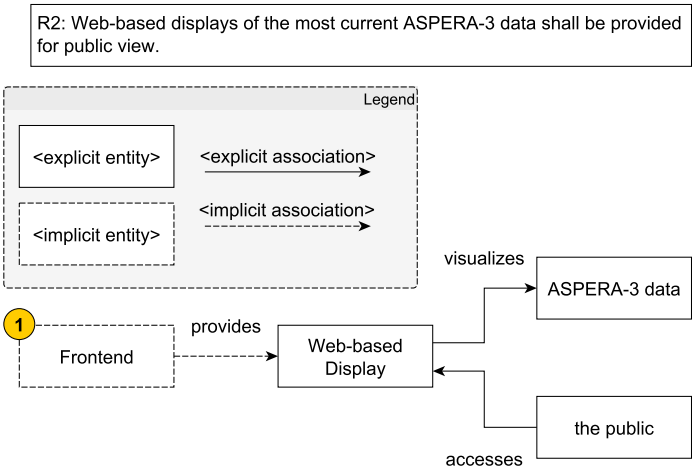

Fig. 1 Formalization of a requirements specification R2 using passive voice

et al. 2017; Soeken et al. 2014; Drechsler et al. 2014; Parra et al. 2015; Ferrari et al. 2018; Génova et al. 2013).

Ambiguous pronouns The inherent ambiguity of natural language (Poesio 1996) poses several challenges for requirements specifications using natural language (Nuseibeh and Easterbrook 2000; Bano 2015). One commonly researched requirements quality factor related to ambiguity is the use of *ambiguous pronouns*, which is a type of *referential ambiguity* (Berry and Kamsties 2004). 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" (Ezzini et al. 2022). 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 (Pohl 2016) at the expense of conciseness. However, the number of publications proposing tools and algorithms to automatically identify and resolve ambiguous pronouns (Ezzini et al. 2022; Yang et al. 2010, 2011; Kamsties and Peach 2000; Sharma et al. 2016; Ezzini et al. 2022; Shah and Jinwala 2015; Ezzini et al. 2021; Chantree et al. 2006) 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 (Kamsties et al. 2005). 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

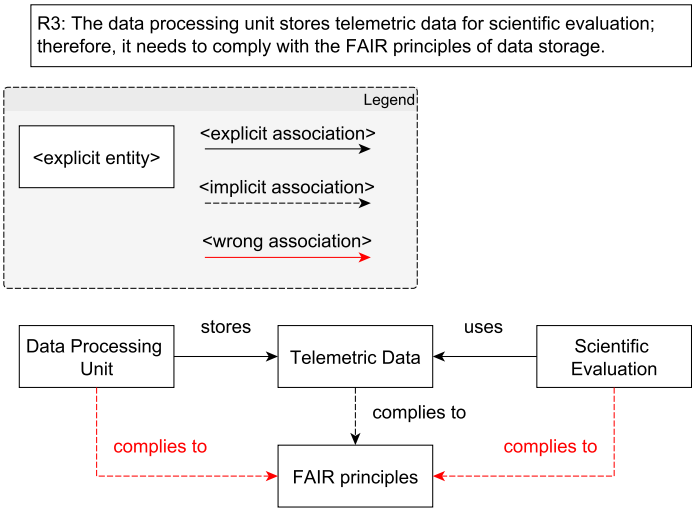

Fig. 2 Formalization of a requirements specification R3 using an ambiguous pronoun

ambiguity has a negligible effect on downstream software engineering activities (de Bruijn and Dekkers 2010; Philippo et al. 2013). Other than these empirical contributions, the aforementioned publications proposing solutions rather than investigating the relevance of the problem refer to deontic guidelines (Pohl 2016; Belev 1989), anecdotal evidence about ambiguity in general (Boyd et al. 2005; Firesmith 2007; Ferrari and Esuli 2019; Christel and Kang 1992), or—in very rare cases—cognitive science theory (Poesio 1996).

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 (Frattini et al. 2023). 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 (Frattini et al. 2023). Bano et al. found only two publications within their sample of 28 studies that empirically investigated the importance of ambiguity detection (Bano 2015). 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) (Montgomery et al. 2022). Instead, most requirements quality publications draw on anecdotal evidence and unproven hypotheses (Frattini et al. 2023). This lack of empirical evidence undermines the trust in requirements quality research and hinders its adoption in practice (Franch et al. 2017; Phalp et al. 2007; Femmer 2018).

Lack of context Second, existing research mostly ignores the influence of context factors on the effect of quality defects (Frattini et al. 2023). Context factors encompass all human and organizational factors influencing the downstream SE activities involving requirements specification (Petersen and Wohlin 2009). For example, the *domain experience* of a stakeholder or the *process model* used during development may mediate the effect of ambiguity in requirements specifications (Philippo et al. 2013).

Requirements quality research has acknowledged the relevance of context factors to requirements quality (Mund et al. 2015; Juergens and Deissenboeck 2010). 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 (Briand et al. 2017; Méndez et al. 2017). However, this initiative has shown little effect in requirements quality research so far (Frattini et al. 2023).

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 (Deissenboeck et al. 2007) (*impact* or *no impact*) or three (Femmer et al. 2015) (*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 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 (Frattini et al. 2023). 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

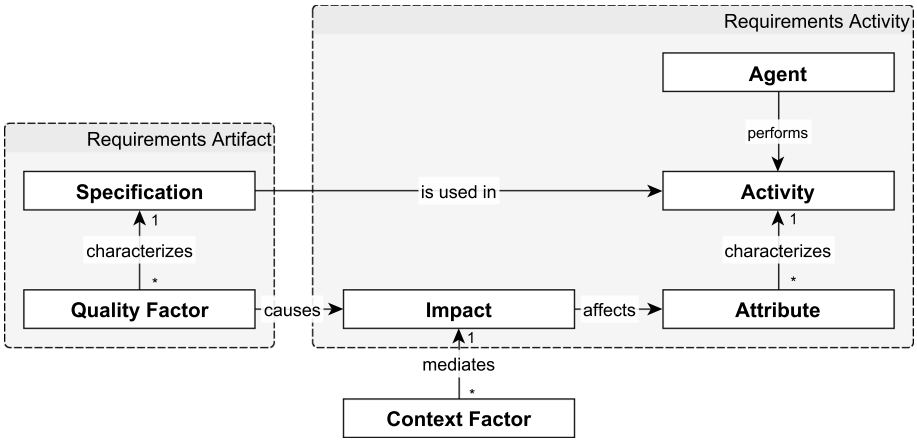

Fig. 3 Reduced version of the activity-based requirements quality theory (Frattini et al. 2023)

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 (Deissenboeck et al. 2007; Wagner et al. 2012), it has not been actively used yet. Even the predecessor of the theory (Femmer et al. 2015) was explicitly ignored in follow-up research by its authors due to the complexity of its implementation (Femmer et al. 2017). 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 (McElreath 2020). These frameworks make causal assumptions explicit (Elwert 2013) and allow reasoning about causally relevant variables (Pearl 1995; Pearl et al. 2016). Furthermore, BDA abstains from reducing complex variable distributions to binary inference (McElreath 2020). 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 (Furia et al. 2019). 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 (Brooks et al. 2011), tools like `Stan` (Carpenter et al. 2017), and libraries like `rethinking` (McElreath 2020) and `brms` (Bürkner 2017).

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 (Furia et al. 2019).
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 (Gelman et al. 2020) makes any hypothesis of causal relations explicit. Analyses become more transparent, and competing causal assumptions are easier to assess.

Furia et al. (2019), and Torkar et al. (2020) 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 (Furia et al. 2022). 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 (Vieira et al. 2022), to confirm the broken window theory in SE (Levén et al. 2022), to investigate gender differences in personality traits of software engineers (Russo and Stol 2020), and to understand data-driven decision making practices (Svensson et al. 2019). In the area of requirements engineering, BDA has been used to evaluate the effect of obsolete requirements on software estimation (Gren and Berntsson Svensson 2021) and to compare requirements prioritization criteria (Berntsson Svensson and Torkar 2024).

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 (Baldassarre et al. 2014) of the study conducted and reported by Femmer et al. (2014) and re-analyzed by us Frattini et al. (2024), as a subset of our hypotheses overlaps with theirs and our study contributes diagnostic evidence for their claims (Nosek and Errington 2020). Therefore, we report the design of the experiment with emphasis on the replication following the guidelines by Carver (2010).

3.1 Goals

We formulate our goal using the goal-question metric approach (Wohlin et al. 2012). 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 (Femmer et al. 2014), 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 (Femmer et al. 2014) 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 (Wohlin et al. 2012). 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 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.

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

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 (Frattini et al. 2024) for the following reasons:

1. In a small-scale experiment employing a parallel design, there is no measure to control subject variability (Vegas et al. 2015), 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 (Femmer et al. 2014) as a directed acyclic graph (Elwert 2013) (the syntax of which is further explained in Section 3.4.10) and Fig. 4b shows the revision in scope of the reanalysis (Frattini et al. 2024). 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 (Gómez et al. 2010), which is sometimes referred to as a test of robustness (Nosek and Errington 2020). 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, the re-analysis

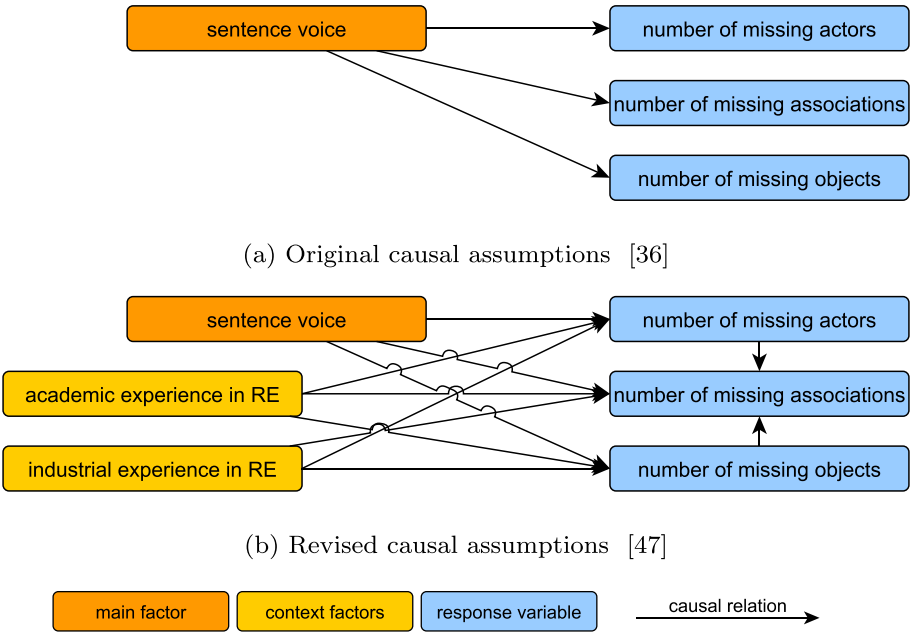

Fig. 4 Causal assumptions about the impact of passive voice

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 (Frattini et al. 2024).

3.4 Our Experiment

The reanalysis (Frattini et al. 2024) of the study by Femmer et al. (2014) 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 (Vegas et al. 2015), 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 (Vegas et al. 2015).
- **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) (Femmer et al. 2014) 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 (Baltes and Ralph 2022).
- **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 (Baltes and Ralph 2022).
- **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 (Ferrari et al. 2017) rather than from a requirements specification written in a student project (Femmer et al. 2014) to increase the realism of the experimental task (Sjøberg et al. 2003).
- **Analysis**: The crossover design produces paired data as opposed to the unpaired data of the original experiment, which changes the appropriate hypothesis test (Vegas et al. 2015)

(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 (Benjamini and Hochberg 1995). Furthermore, we additionally analyze the data using BDA@.

Our experiment differs from the original experiment (Femmer et al. 2014) in all elements (Gómez et al. 2010). However, because a subset of our hypotheses aligns with their hypotheses, part of our study counts as a conceptual replication (Baldassarre et al. 2014) since it contributes diagnostic evidence for the original claims (Nosek and Errington 2020). In the rest of this subsection, we report the design of our experiment following the guidelines by Jedlitschka et al. (2008).

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 (Femmer et al. 2014). 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 (Pohl 2016):

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 Fig. 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 Fig. 5).

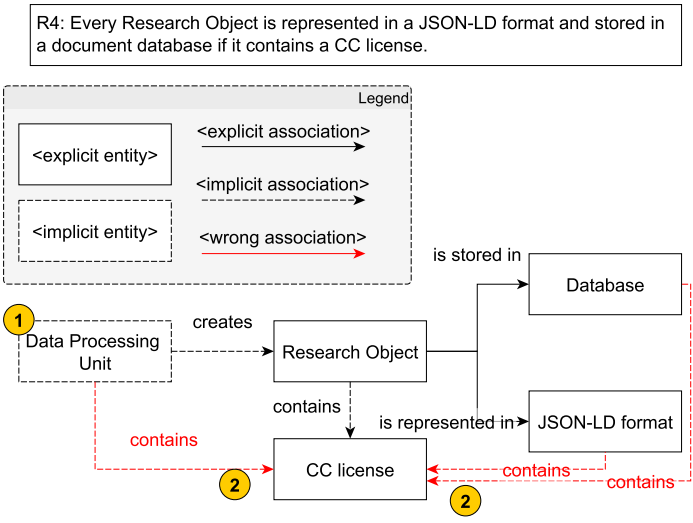

Fig. 5 Domain modeling task example for requirement 4

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 (Femmer et al. 2015; Frattini et al. 2023). 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 (Petersen and Wohlin 2009; Mund et al. 2015; Juergens and Deissenboeck 2010), 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 (Bogner et al. 2023), 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 (Salman et al. 2015). Additionally, the line between students and practitioners becomes increasingly blurred as students more commonly gather industrial experience before or during their studies (Carver et al. 2004). Consequently, we subsume the participant type variable by the causally more meaningful and fine-grained

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	$[0, E_{expected}]$

Table 1 continued

Variable	Name	Type	Description	Data type	Range
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	$[0, A_{found}]$

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 (Sjøberg et al. 2003), we anticipated a moderate sample size of participants. Consequently, we opted for a crossover design (Vegas et al. 2015) 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 (Kitchenham et al. 2003). 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 (Vegas et al. 2015); and
4. the subject variability, as software engineering tasks are highly dependent on the skill of involved individuals (Pickard et al. 1998).

However, recent adoptions of best practices from other disciplines made this design applicable to SE research without compromising the validity of the results (Vegas et al. 2015; Fucci et al. 2016). The threats to validity can be mitigated during design and analysis (Vegas et al. 2015) 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 (Wohlin et al. 2012). Secondly, it accounts for subject 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 (Kitchenham et al. 2003). 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 (Brown Jr 1980; Vegas et al. 2015). 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.

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}

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).

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 (Ferrari et al. 2017). 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 (Baltes and Ralph 2022). 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 (Vegas et al. 2015).

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

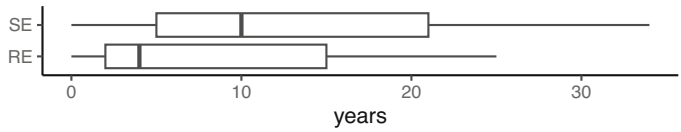

Fig. 6 Distribution of SE and RE experience

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 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 (Wohlin et al. 2012).

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 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 (Ferrari et al. 2017).

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.

[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/

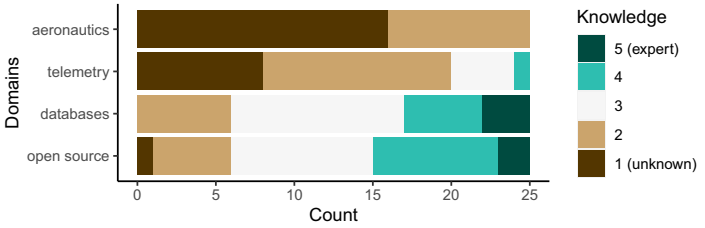

Fig. 7 Distribution of knowledge in the four relevant domains

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 Frattini 2024). 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 (Femmer et al. 2014). 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 (Hsu and Lachenbruch 2014) or the non-parametric Wilcoxon signed-rank test (Wilcoxon 1945) based on the distribution of the variables, which we evaluated using the Shapiro-Wilk test results (Shapiro and Wilk 1965).

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 (Benjamini and Hochberg 1995): 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 (Dybå et al. 2006) using Cohen's D for the paired student t-test (Cohen 1969) and the matched-pairs rank biserial correlation coefficient for Wilcoxon signed-rank test (King et al. 2018).

3.4.10 Bayesian Data Analysis

We apply Bayesian data analysis with Pearl's framework for causal inference (Pearl et al. 2016) to complement the frequentist data analysis (Furia et al. 2019). Given the limited adoption of Bayesian data analysis in software engineering research (Furia et al. 2019, 2022, 2023), 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 (Siebert 2023), which are the major steps of Pearl's original model of causal statistical inference (Pearl et al. 2016), 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 (Elwert 2013). 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 (Vegas et al. 2015).
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 (McElreath 2020). Visualizations of interaction effects in DAGs have been proposed (Nilsson et al. 2021) 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 (Poesio 1996)
2. *requirements quality * period* carryover effect (Vegas et al. 2015): the effect of a treatment may be influenced by the treatments applied in previous periods

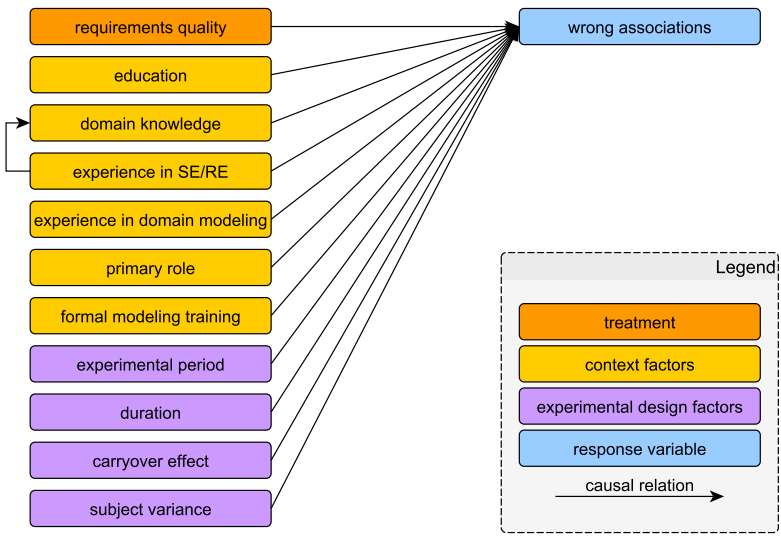

Fig. 8 DAG for the analysis of wrong associations

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 (Pearl 1995). 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 (McElreath 2020) in the *identification* step. A series of four criteria (McElreath 2020) 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 → 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 (Gelman et al. 2020). We select the distribution type based on the maximum entropy criterion (Jaynes 2003) 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).

$$logit(p_i) = \alpha + \alpha_{PID} + \beta^T_{RQD} \times RQD_i + \beta_{SE} \times exp.se_i$$

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 (Demaris 1992; McElreath 2020). 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 (Vegas et al. 2015) modeled via partial pooling (Ernst 2018)
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. (2014) 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 (Wesner and Pomeranz 2021). 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

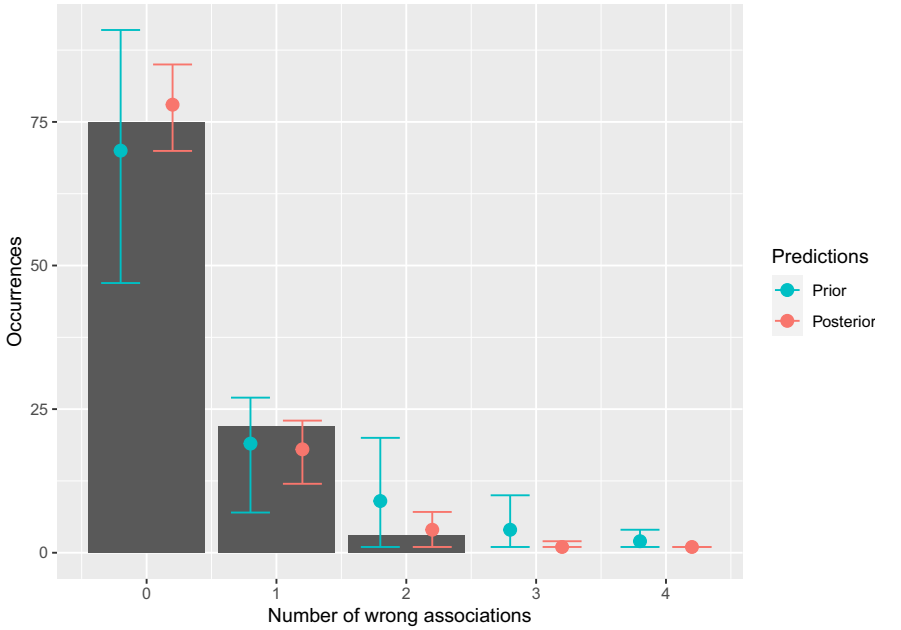

Fig. 9 Predictive checks with prior and updated posterior coefficient distributions

predictive check. The grey bars represent the actual observed distribution 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 (Bürkner 2017) in *R*. Hamiltonian Monte Carlo Markov Chains (MCMC) (Brooks et al. 2011) 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 (McElreath 2020). 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 (Furia et al. 2023). 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 (Gren and Berntsson Svensson 2021). 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. (2014). Note that the results are not directly comparable to those in the original study as both our experimental objects and treatments varied.

Table 4 lists the results of our frequentist data analysis and relates them to the results from the original study (Femmer et al. 2014).

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

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 Results of frequentist analysis including the p-value of the hypothesis test (p), confidence interval (CI), and effect size (ES)

Outcome	Treatment	Original (Femmer et al. 2014)			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

Statistically significant results in **bold** (original study $\alpha = 0.05$, this experiment $\alpha' = 0.01$)

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 (Cohen 2008) 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 Fig. 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.

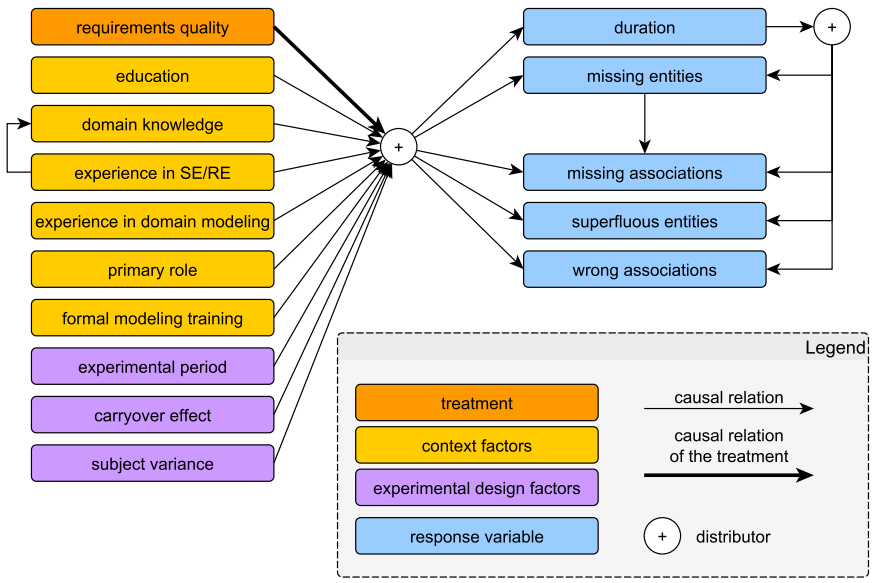

Fig. 10 Directed acyclic graph visualizing all causal assumptions

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 Frattini et al. 2024).

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 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 Fig. 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 Fig. 8 lists the variable duration as an independent variable, while the final DAG in Fig. 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 (Siebert 2023), 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 (Jaynes 2003), 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

[5] The remaining cases ($100\% - less - more$) are omitted from the table

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%

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.

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.

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 (Femmer et al. 2014). In fact, the strength of the effect of passive voice on missing associations ($\mu_{RQT}^{PV} = 0.38$) is similar to the strength of the effect

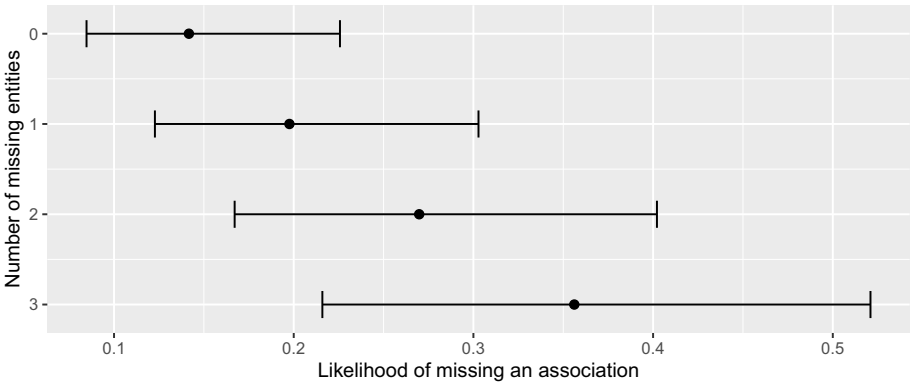

Fig. 11 Impact of missing entities on missing associations

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.

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

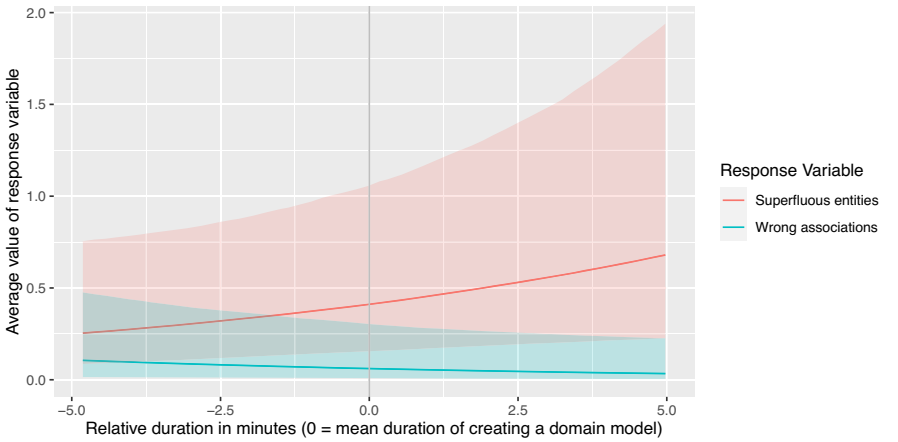

Fig. 12 Impact of relative duration on the number of superfluous entities and wrong associations

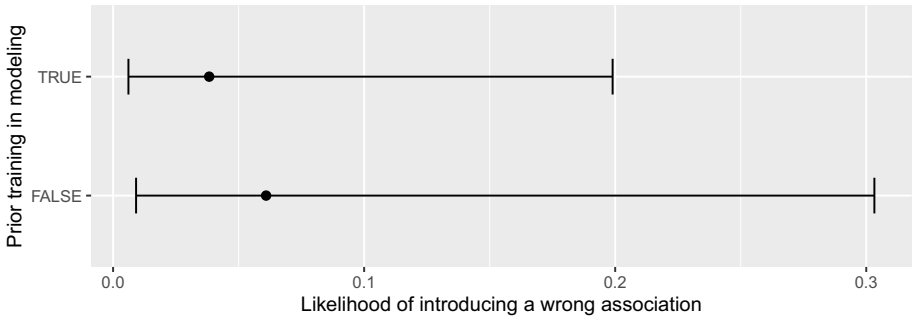

Fig. 13 Marginal effect of prior formal training in modeling on the number of wrong associations

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 Fig. 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 interaction 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

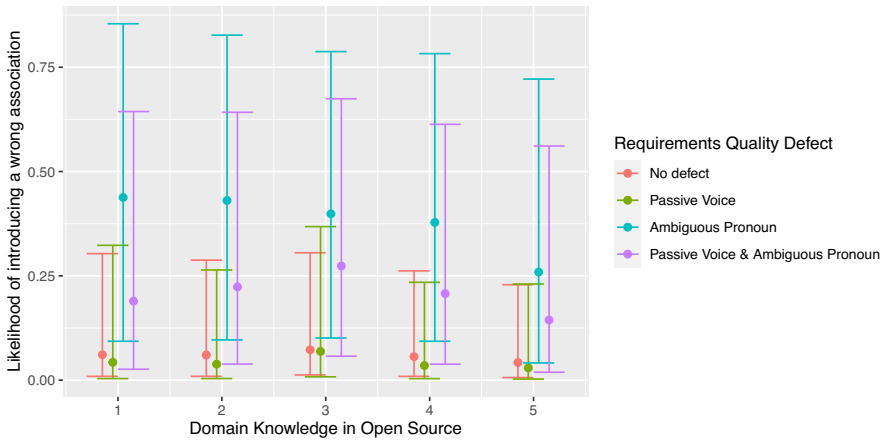

Fig. 14 Interaction effect between domain knowledge and the treatment on the number of wrong associations

and overlapping confidence intervals (cyan and purple whiskers in Fig. 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 (Femmer et al. 2014) and its re-analysis (Frattini et al. 2024) with our results (Carver 2010). 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 original 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 (Frattini et al. 2024) 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) (Gómez et al. 2010) increases the replicability space within the generalizability space (Nosek and Errington 2020) and identifies those elements as non-influential (Juristo and Vegas 2011) 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* (Carver 2010) for $H_0^{RQD \in \{PV, AP, PVAP\} \to D}$. 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 $H_0^{RQD \in \{AP, PVAP\} \rightarrow A^-}$, 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 $H_0^{RQD \in \{AP, PVAP\} \rightarrow A^\times}$. 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. (2014); Frattini et al. (2024). 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 (Femmer et al. 2014; Krisch and Houdek 2015).

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 (Ezzini et al. 2022) and explains the focus on ambiguity in requirements quality research (Montgomery et al. 2022). 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 (Montgomery et al. 2022) while using passive voice rarely has a tangible impact (Krisch and Houdek 2015). By filtering requirements writing guidelines for quality factors that have a measurable effect, we expect greater acceptance of requirements quality assessment tools in practice (Franch et al. 2020; Phalp et al. 2007).

Additionally, measuring the effect of a quality defect on the relevant attributes of activities that use these requirements allows quantifying it economically (Frattini et al. 2023). 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 (Juergens and Deissenboeck 2010).

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 (Frattini et al. 2023). 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 (Furia et al. 2019). 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 Fig. 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 (Franch et al. 2020) that justify subsequent tool development (Frattini et al. 2024). 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 (Frattini et al. 2023). The continuous synthesis of evidence from individual studies in this common framework will produce more reliable and generalizable conclusions (Badampudi et al. 2019; Ciolkowski and Münch 2005) and effectively address the lack of empirical insights in requirements quality research (Montgomery et al. 2022).

Using a controlled experiment benefits the investigation of the quality factor (Femmer et al. 2014). The DAG shown in Fig. 10 visualizes this control, as no other factor influences the treatment in question. This eliminates spurious associations that could confuse the results (McElreath 2020). On the other hand, the cost of conducting a controlled experiment—especially with participants from industry—cannot be neglected (Sjøberg et al. 2003). Luckily, statistical causal inference via Bayesian data analysis works equally well with observational data, as shown by Furia et al. (2023).

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 (Femmer et al. 2014) 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. (2023). However, frequentist approaches tend to limit their analyses to the treatment and the response variable, disregarding potential context (Mund et al. 2015) or experimental design factors (Vegas et al. 2015). BDA, on the other hand, entails the use of an explicit causal framework (McElreath 2020; Pearl 1995), which is why we support the recommendation of abandoning FDA for BDA in SE research (Torkar et al. 2020; Furia et al. 2019).

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. (2012) and extended by the guideline by Vegas et al. (2015) 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 (Baltes and Ralph 2022).

The crossover design emits additional threats to validity (Vegas et al. 2015). 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}) (Vegas et al. 2015). 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 (Levén et al. 2022).

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 (Gelman 2018). 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 (Montgomery et al. 2022; Frattini et al. 2022). Nevertheless, our goal with this replication is to extend the initial quality factor of passive voice reported by Femmer et al. (2014) to a second one—ambiguous pronouns—which is widespread as indicated by the literature (Frattini et al. 2022; Montgomery et al. 2022).

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 relying on comparable practices in our scientific community (Bogner et al. 2023). 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 (Salman et al. 2015).

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) (Martyniuk 2006) (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 (Frattini et al. 2022) to better understanding and solving problems relevant to practice (Franch et al. 2017, 2020). 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 (Femmer et al. 2014; Frattini et al. 2024) 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 (Furia et al. 2019, 2022; Torkar et al. 2020), 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 (Ciolkowski and Münch 2005) in a common framework (Frattini et al. 2023). We hope that the documentation of this study inspires fellow researchers to adopt our method and tools for replication.

Acknowledgements This work was supported by the KKS foundation through the S.E.R.T. Research Profile project at Blekinge Institute of Technology. We are deeply grateful to Parisa Yousefi from Ericsson AB for recruiting practitioners to the experiment. We further thank the reviewers for their tremendous effort that significantly improved this manuscript.

Funding Open access funding provided by Blekinge Institute of Technology.

Data Availability All supplementary material, including protocols and guidelines for data collection and extraction, the raw data, analysis scripts, figures, and results, are available in our replication package (Frattini 2024). The replication package is available on GitHub at https://github.com/JulianFrattini/rqi-proto and archived on Zenodo at https://doi.org/10.5281/zenodo.10423665.

Declarations

Conflicts of Interest The authors declare that they have no conflict of interest.

References

Badampudi D, Wohlin C, Gorschek T (2019) Contextualizing research evidence through knowledge translation in software engineering. In: Proceedings of the 23rd international conference on evaluation and assessment in software engineering, pp 306–311

Baldassarre MT, Carver J, Dieste O, Juristo, N (2014) Replication types: towards a shared taxonomy. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering, pp 1–4

Baltes S, Ralph P (2022) Sampling in software engineering research: a critical review and guidelines. Empir Softw Eng 27(4):94

Bano M (2015) Addressing the challenges of requirements ambiguity: a review of empirical literature. In: 2015 IEEE fifth international workshop on empirical requirements engineering (EmpiRE), pp 21–24. IEEE

Belev G (1989) Guidelines for specification development. In: Proceedings., annual reliability and maintainability symposium, pp 15–21. IEEE

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc: Ser B (Methodological) 57(1):289–300

Berntsson Svensson R, Torkar R (2024) Not all requirements prioritization criteria are equal at all times: a quantitative analysis. J Syst Softw 209:111909. https://doi.org/10.1016/j.jss.2023.111909

Berry DM, Kamsties E (2004) Ambiguity in requirements specification. In: Perspectives on software requirements, pp 7–44. Springer

Boehm BW (1984) Software engineering economics. IEEE Trans Softw Eng SE–10(1):4–21

Bogner J, Kotstein S, Pfaff T (2023) Do restful api design rules have an impact on the understandability of web apis? Empir Softw Eng 28(6):132

Boyd S, Zowghi D, Farroukh A (2005) Measuring the expressiveness of a constrained natural language: an empirical study. In: 13th IEEE international conference on requirements engineering (RE'05), pp 339–349. IEEE

Briand L, Bianculli D, Nejati S, Pastore F, Sabetzadeh M (2017) The case for context-driven software engineering research: generalizability is overrated. IEEE Softw 34(5):72–75

Brooks S, Gelman A, Jones G, Meng XL (2011) Handbook of Markov Chain Monte Carlo. CRC press

Brown Jr BW (1980) The crossover experiment for clinical trials. Biometrics 69–79

Bürkner PC (2017) brms: An R package for Bayesian multilevel models using Stan. J Stat Softw 80:1–28

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: a probabilistic programming language. J Stat Softw 76(1)

Carver JC (2010) Towards reporting guidelines for experimental replications: a proposal. In: 1st International workshop on replication in empirical software engineering, vol 1, pp 1–4

Carver J, Jaccheri L, Morasca S, Shull F (2004) 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), pp 239–249. IEEE

Chantree F, Nuseibeh B, De Roeck A, Willis A (2006) Identifying nocuous ambiguities in natural language requirements. In: 14th IEEE international requirements engineering conference (RE'06), pp 59–68. IEEE

Christel MG, Kang KC (1992) Issues in requirements elicitation

Ciolkowski M, Münch J (2005) Accumulation and presentation of empirical evidence: problems and challenges. ACM SIGSOFT Softw Eng Notes 30(4):1–3

Cohen BH (2008) Explaining psychological statistics. John Wiley & Sons

Cohen J (1969) Statistical power analysis for the behavioral sciences. Academic press

de Bruijn F, Dekkers HL (2010) 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, pp 233–247. Springer

Deissenboeck F, Wagner S, Pizka M, Teuchert S, Girard JF (2007) An activity-based quality model for maintainability. In: 2007 IEEE international conference on software maintenance, pp 184–193. IEEE

Demaris A (1992) Logit modeling: practical applications. 86. Sage

Drechsler R, Soeken M, Wille R (2014) Automated and quality-driven requirements engineering. In: 2014 IEEE/ACM international conference on computer-aided design (ICCAD), pp 586–590. IEEE

Dybå T, Kampenes VB, Sjøberg DI (2006) A systematic review of statistical power in software engineering experiments. Inf Softw Technol 48(8):745–755

Elwert F (2013) Graphical causal models. In: Handbook of causal analysis for social research, pp 245–273. Springer

Ernst NA (2018) Bayesian hierarchical modelling for tailoring metric thresholds. In: Proceedings of the 15th international conference on mining software repositories, pp 587–591. https://doi.org/10.1145/3196398.3196443

Ezzini S, Abualhaija S, Arora C, Sabetzadeh M (2022) Automated handling of anaphoric ambiguity in requirements: a multi-solution study. In: Proceedings of the 44th international conference on software engineering, pp 187–199

Ezzini S, Abualhaija S, Arora C, Sabetzadeh M (2022) 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, pp 1677–1681

Ezzini S, Abualhaija S, Arora C, Sabetzadeh M, Briand LC (2021) Using domain-specific corpora for improved handling of ambiguity in requirements. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp 1485–1497. IEEE

Femmer H (2018) Requirements quality defect detection with the qualicen requirements scout. In: REFSQ Workshops

Femmer H, Vogelsang A (2018) Requirements quality is quality in use. IEEE Softw 36(3):83–91

Femmer H, Fernández DM, Wagner S, Eder S (2017) Rapid quality assurance with requirements smells. J Syst Softw 123:190–213

Femmer H, Kučera J, Vetrò A (2014) 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, pp 1–4

Femmer H, Mund J, Fernández DM (2015) It's the activities, stupid! A new perspective on re quality. In: 2015 IEEE/ACM 2nd international workshop on requirements engineering and testing, pp 13–19. IEEE

Ferrari A, Esuli A (2019) An NLP approach for cross-domain ambiguity detection in requirements engineering. Autom Softw Eng 26(3):559–598

Ferrari A, Gori G, Rosadini B, Trotta I, Bacherini S, Fantechi A, Gnesi S (2018) Detecting requirements defects with NLP patterns: an industrial experience in the railway domain. Empir Softw Eng 23:3684–3733

Ferrari A, Spagnolo GO, Gnesi S (2017) Pure: a dataset of public requirements documents. In: 2017 IEEE 25th international requirements engineering conference (RE), pp 502–505. IEEE

Firesmith D (2007) Common requirements problems, their negative consequences, and the industry best practices to help solve them. J Object Technol 6(1):17–33

Franch X, Fernández DM, Oriol M, Vogelsang A, Heldal R, Knauss E, Travassos GH, Carver JC, Dieste O, Zimmermann T (2017) How do practitioners perceive the relevance of requirements engineering research? An ongoing study. In: 2017 IEEE 25th international requirements engineering conference (RE), pp 382–387. IEEE

Franch X, Mendez D, Vogelsang A, Heldal R, Knauss E, Oriol M, Travassos G, Carver JC, Zimmermann T (2020) How do practitioners perceive the relevance of requirements engineering research? IEEE Trans Softw Eng

Franch X, Palomares C, Quer C, Chatzipetrou P, Gorschek T (2023) The state-of-practice in requirements specification: an extended interview study at 12 companies. Requir Eng 1–33. https://doi.org/10.1007/s00766-023-00399-7

Frattini J (2024) 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: 21-June-2024

Frattini J, Fucci D, Torkar R, Mendez D (2024) 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)

Frattini J, Montgomery L, Fischbach J, Mendez D, Fucci D, Unterkalmsteiner M (2023) Requirements quality research: a harmonized theory, evaluation, and roadmap. Requir Eng 1–14

Frattini J, Montgomery L, Fischbach J, Unterkalmsteiner M, Mendez D, Fucci D (2022) A live extensible ontology of quality factors for textual requirements. In: 2022 IEEE 30th international requirements engineering conference (RE), pp 274–280. IEEE

Frattini J, Unterkalmsteiner M, Fucci D, Mendez D (2024) NLP4RE Tools: classification, overview, and management. Springer International Publishing

Fucci D, Scanniello G, Romano S, Shepperd M, Sigweni B, Uyaguari F, Turhan B, Juristo N, Oivo M (2016) 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, ESEM '16. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2961111.2962592

Furia CA, Torkar R, Feldt R (2023) Towards causal analysis of empirical software engineering data: the impact of programming languages on coding competitions. ACM Trans Softw Eng Methodol 33(1). https://doi.org/10.1145/3611667

Furia CA, Feldt R, Torkar R (2019) Bayesian data analysis in empirical software engineering research. IEEE Trans Softw Eng 47(9):1786–1810

Furia CA, Torkar R, Feldt R (2022) Applying Bayesian analysis guidelines to empirical software engineering data: The case of programming languages and code quality. ACM Trans Softw Eng Methodol (TOSEM) 31(3):1–38

Gelman A (2018) 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 24-Nov-2023

Gelman A, Vehtari A, Simpson D, Margossian CC, Carpenter B, Yao Y, Kennedy L, Gabry J, Bürkner PC, Modrák M (2020) Bayesian workflow. arXiv:2011.01808

Génova G, Fuentes JM, Llorens J, Hurtado O, Moreno V (2013) A framework to measure and improve the quality of textual requirements. Requir Eng 18:25–41

Gleich B, Creighton O, Kof L (2010) 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, pp 218–232. Springer

Gómez OS, Juristo N, Vegas S (2010) Replications types in experimental disciplines. In: Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement, pp 1–10

Gren L, Berntsson Svensson R (2021) Is it possible to disregard obsolete requirements? a family of experiments in software effort estimation. Requir Eng 26(3):459–480

Hasso H, Dembach M, Geppert H, Toews D (2019) Detection of defective requirements using rule-based scripts. In: REFSQ workshops

Hsu H, Lachenbruch PA (2014) Paired *t* test. Wiley StatsRef: statistics reference online

Jaynes ET (2003) Probability theory: the logic of science. Cambridge University Press, Cambridge

Jedlitschka A, Ciolkowski M, Pfahl D (2008) Reporting experiments in software engineering. Guide Adv Empir Softw Eng 201–228

Juergens E, Deissenboeck F (2010) How much is a clone. In: Proceedings of the 4th international workshop on software quality and maintainability, pp 79–88

Juristo N, Vegas S (2011) The role of non-exact replications in software engineering experiments. Empir Softw Eng 16:295–324

Kamsties E, Peach B (2000) Taming ambiguity in natural language requirements. In: Proceedings of the thirteenth international conference on software and systems engineering and applications, vol 1315

Kamsties E, von Knethen A, Philipps J (2005) An empirical investigation of requirements specification languages: detecting defects while formalizing requirements. In: Information modeling methods and methodologies: advanced topics in database research, pp 125–147. IGI Global

King BM, Rosopa PJ, Minium EW (2018) Statistical reasoning in the behavioral sciences. John Wiley & Sons

Kitchenham B, Fry J, Linkman S (2003) The case against cross-over designs in software engineering. In: Eleventh annual international workshop on software technology and engineering practice, pp 65–67. IEEE

Knauss E, Schneider K, Stapel K (2009) Learning to write better requirements through heuristic critiques. In: 2009 17th IEEE international requirements engineering conference, pp 387–388. IEEE

Kof L (2007) 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, pp 181–192. Springer

Krisch J, Houdek F (2015) 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), pp 344–351. IEEE

Levén W, Broman H, Besker T, Torkar R (2022) The broken windows theory applies to technical debt. arXiv:2209.01549

Martyniuk W (2006) 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

McElreath R (2020) Statistical rethinking: a Bayesian course with examples in R and Stan. CRC press

Méndez Fernández D, Penzenstadler B (2015) Artefact-based requirements engineering: the AMDiRE approach. Requir Eng 20:405–434

Méndez Fernández D, Böhm W, Vogelsang A, Mund J, Broy M, Kuhrmann M, Weyer T (2019) Artefacts in software engineering: a fundamental positioning. Softw Syst Model 18:2777–2786

Méndez D, Wagner S, Kalinowski M, Felderer M, Mafra P, Vetrò A, Conte T, Christiansson MT, Greer D, Lassenius C et al (2017) Naming the pain in requirements engineering: contemporary problems, causes, and effects in practice. Empir Softw Eng 22:2298–2338

Montgomery L, Fucci D, Bouraffa A, Scholz L, Maalej W (2022) Empirical research on requirements quality: a systematic mapping study. Requir Eng 27(2):183–209

Mund J, Fernandez DM, Femmer H, Eckhardt J (2015) 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), pp 1–10. IEEE

Nilsson A, Bonander C, Strömberg U, Björk J (2021) A directed acyclic graph for interactions. Int J Epidemiol 50(2):613–619

Nosek BA, Errington TM (2020) What is replication? PLoS Biol 18(3):e3000691

Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. In: Proceedings of the conference on the future of software engineering, pp 35–46

O'Grady W, Archibald J, Aronoff M, Rees-Miller J (2001) Contemporary linguistics: an introduction. Bedford/St. Martin's, Boston

Parra E, Dimou C, Llorens J, Moreno V, Fraga A (2015) A methodology for the classification of quality of requirements using machine learning techniques. Inf Softw Technol 67:180–195

Pearl J (1995) From Bayesian networks to causal networks. In: Mathematical models for handling partial knowledge in artificial intelligence, pp 157–182. Springer

Pearl J, Glymour M, Jewell NP (2016) Causal inference in statistics: a primer. John Wiley & Sons

Petersen K, Wohlin C (2009) Context in industrial software engineering research. In: 2009 3rd International symposium on empirical software engineering and measurement, pp 401–404. IEEE

Phalp KT, Vincent J, Cox K (2007) Assessing the quality of use case descriptions. Software Qual J 15(1):69–97

Philippo EJ, Heijstek W, Kruiswijk B, Chaudron MR, Berry DM (2013) 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, pp 65–79. Springer

Pickard LM, Kitchenham BA, Jones PW (1998) Combining empirical results in software engineering. Inf Softw Technol 40(14):811–821

Poesio M (1996) Semantic ambiguity and perceived ambiguity. In: van Deemter K, Peters S (eds) Semantic ambiguity and underspecification. Center for the study of language and Inf, United Kingdom. https://doi.org/10.48550/arXiv.cmp-lg/9505034

Pohl K (2016) Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant. Rocky Nook, Inc

Rosadini B, Ferrari A, Gori G, Fantechi A, Gnesi S, Trotta I, Bacherini S (2017) 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, pp 344–360. Springer

Russo D, Stol KJ (2020) Gender differences in personality traits of software engineers. IEEE Trans Softw Eng 48(3):819–834

Salman I, Misirli AT, Juristo N (2015) Are students representatives of professionals in software engineering experiments? In: 2015 IEEE/ACM 37th IEEE international conference on software engineering, vol 1, pp 666–676. IEEE

Shah US, Jinwala DC (2015) Resolving ambiguity in natural language specification to generate UML diagrams for requirements specification. Int J Softw Eng Technol Appl 1(2–4):308–334

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611

Sharma R, Sharma N, Biswas K (2016) 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), pp 177–182. IEEE

Siebert J (2023) Applications of statistical causal inference in software engineering. Inf Softw Technol 107198

Sjøberg DI, Anda B, Arisholm E, Dybå T, Jørgensen M, Karahasanović A, Vokáč M (2003) Challenges and recommendations when increasing the realism of controlled software engineering experiments. In: Empirical methods and studies in software engineering: Experiences from ESERNET, pp. 24–38. Springer. https://doi.org/10.1007/978-3-540-45143-3_3

Soeken M, Abdessaied N, Allahyari-Abhari A, Buzo A, Musat L, Pelz G, Drechsler R (2014) Quality assessment for requirements based on natural language processing. In: Forum on specification and design languages. Proceedings. Citeseer

Stol KJ, Fitzgerald B (2018) The ABC of software engineering research. ACM Trans Softw Eng Methodol (TOSEM) 27(3):1–51

Svensson RB, Feldt R, Torkar R (2019) 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, pp 69–85. Springer

Torkar R, Feldt R, Furia CA (2020) Bayesian data analysis in empirical software engineering: the case of missing data, pp 289–324. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-32489-6_11

Vegas S, Apa C, Juristo N (2015) Crossover designs in software engineering experiments: benefits and perils. IEEE Trans Softw Eng 42(2):120–135. https://doi.org/10.1109/TSE.2015.2467378

Vieira R, Mesquita D, Mattos CL, Britto R, Rocha L, Gomes J (2022) Bayesian analysis of bug-fixing time using report data. In: Proceedings of the 16th ACM/IEEE international symposium on empirical software engineering and measurement, pp 57–68

Wagner S, Fernández DM, Felderer M, Vetrò A, Kalinowski M, Wieringa R, Pfahl D, Conte T, Christiansson MT, Greer D et al (2019) Status quo in requirements engineering: a theory and a global family of surveys. ACM Trans Softw Eng Methodol (TOSEM) 28(2):1–48

Wagner S, Lochmann K, Heinemann L, Kläs M, Trendowicz A, Plösch R, Seidi A, Goeb A, Streit J (2012) The Quamoco product quality modelling and assessment approach. In: 2012 34th International Conference on Software Engineering (ICSE), pp 1133–1142. IEEE

Wesner JS, Pomeranz JP (2021) Choosing priors in Bayesian ecological models by simulating from the prior predictive distribution. Ecosphere 12(9):e03739

Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science & Business Media

Yang H, De Roeck A, Gervasi V, Willis A, Nuseibeh B (2011) Analysing anaphoric ambiguity in natural language requirements. Requir Eng 16:163–189

Yang H, De Roeck A, Gervasi V, Willis A, Nuseibeh B (2010) Extending nocuous ambiguity analysis for anaphora in natural language requirements. In: 2010 18th IEEE international requirements engineering conference, pp 25–34. IEEE

Zhao L, Alhoshan W, Ferrari A, Letsholo KJ, Ajagbe MA, Chioasca EV, Batista-Navarro RT (2021) Natural language processing for requirements engineering: a systematic mapping study. ACM Comput Surv (CSUR) 54(3):1–41

Julian Frattini is a Ph.D. student at the Blekinge Institute of Technology (BTH) in Karlskrona, Sweden, under the supervision of Professor Daniel Mendez. His research revolves around requirements quality and targets understanding the impact of requirements quality defects through empirical studies. In addition, he studies the use and improvement of empirical research methods in software engineering. He served on the organizing committees of RE'24, REFSQ'24, ESEM'24 and REFSQ'25 as well as AIRE'24 and CrowdRE'24, is part of the program committees of RE, REFSQ, NLP4RE, MO2RE, and QUATIC, and regularly reviews for TSE, TOSEM, IST, JSS, and REJ.

Davide Fucci is Docent at Blekinge Institute of Technology (Sweden). His research interests lie in data-driven requirements engineering, test automation, secure software engineering, and human aspects of software development. His research is performed in close collaboration with industry. Dr. Fucci has published over 70 articles in international journals and conferences and received several best paper awards. He has served on the program committees of over 20+ academic conferences, on the editorial or review boards of several top-tier software engineering journals. He is a member of ACM, ACM SIGSOFT, IEEE and IEEE Computer Society.

Richard Torkar is a professor of software engineering currently appointed head of department at the Department of Computer Science and Engineering at Chalmers University of Technology and University of Gothenburg, Sweden. Richard's main interest lies in empirical studies in software engineering, where he can apply rigorous data analysis approaches.

Lloyd Montgomery is a PhD student at the University of Hamburg, working under Prof. Dr. Walid Maalej. Currently, his primary research area is the quality of issue tracking systems, with other interests such as empirical software engineering, non-technical factors in software engineering, and science communication. He won the RE'17 best paper award for his machine learning design science research with IBM customer support. Lloyd's academic service record includes serving as the IST Publicity Chair, RE'23 Publicity Chair, NLP4RE'22 Workshop Co-Chair, and RE'21 Artefact Co-Chair. He is on the program committees of REFSQ, NLP4RE, and the RE@Next! track at RE, and regularly reviews for the journals REJ, IST, and JSS, in addition to reviews for SQ Journal and TOSEM. Lloyd also has a particular passion for artefact tracks, having served as a PC member for seven artefact tracks at RE, ICSE, and FSE. Contact him at lloyd.montgomery@uni-hamburg.de.

Michael Unterkalmsteiner has been researching Software Engineering since 2009, focusing in particular on the coordination between requirements engineering and software testing. His research work with many industry partners is shaped by empirical problem identification, in-depth analysis of the state-of-art and practice, and collaborative solution development. His current research focuses on designing and implementing automated decision support systems for engineers, using natural language processing and machine learning technologies. Michael is a senior lecturer at the Blekinge Institute of Technology.

Jannik Fischbach works as a consultant at Netlight and as a postdoctoral researcher at fortiss. His research focuses on modern Natural Language Processing techniques and their potential to support stakeholders in the software engineering process.

Daniel Mendez is Professor for Empirical Software Engineering at the Blekinge Institute of Technology, Sweden, and Lead Researcher heading the research division Requirements Engineering at fortiss, Germany. His research is on Empirical Software Engineering with a particular focus on Requirements Engineering and its quality improvement – all conducted in close collaboration with the relevant industries. He is editorial board member at the Empirical Software Engineering Journal (EMSE), the Journal of Systems and Software (JSS), and the Information and Software Technology Journal (IST), and he co-chairs the special tracks Reproducibility & Open Science at EMSE and In Practice at JSS. He is further a member of the ACM, the IEEE Computer Society, and the German association of university professors and lecturers, and he serves as the University representative to ISERN, the International Empirical Software Engineering Research Network. Further information are available at http://www.mendezfe.org.

Authors and Affiliations

Julian Frattini[1] · Davide Fucci[1] · Richard Torkar[2,3] · Lloyd Montgomery[4] · Michael Unterkalmsteiner[1] · Jannik Fischbach[5,6] · Daniel Mendez[1,6]

✉ Julian Frattini
julian.frattini@bth.se
https://julianfrattini.github.io/

Davide Fucci
davide.fucci@bth.se
https://dfucci.github.io/

Richard Torkar
richard.torkar@gu.se
https://torkar.github.io/

Lloyd Montgomery
lloyd.montgomery@uni-hamburg.de
https://lloydm.io/

Michael Unterkalmsteiner
michael.unterkalmsteiner@bth.se
https://www.lmsteiner.com/

Jannik Fischbach
jannik.fischbach@netlight.com

Daniel Mendez
daniel.mendez@bth.se
https://www.mendezfe.org/

1 Blekinge Institute of Technology, Valhallavägen 1, 37140 Karlskrona, Sweden

2 Chalmers University of Technology and University of Gothenburg, 41756 Göteborg, Sweden

3 Stellenbosch Institute for Advanced Study (STIAS), Stellenbosch, South Africa

4 University of Hamburg, Mittelweg 177, Hamburg 20148, Germany

5 Netlight Consulting GmbH, Prannerstraße 4, München 80333, Germany

6 fortiss GmbH, Guerickestraße 25, 80805 München, Germany