# Does Quality of Requirements Specifications matter? Combined Results of Two Empirical Studies

Jakob Mund, Henning Femmer, Daniel Méndez Fernández, Jonas Eckhardt
Technische Universität München
Garching b. München, Germany
{mund,femmer,mendezfe,eckharjo}@in.tum.de

***Abstract*—[*Background*] Requirements Engineering is crucial for project success, and to this end, many measures for quality assurance of the software requirements specification (SRS) have been proposed. [*Goal*] However, we still need an empirical understanding on the extent to which SRS are created and used in practice, as well as the degree to which the quality of an SRS matters to subsequent development activities. [*Method*] We studied the relevance of SRS by relying on survey research and explored the impact of quality defects in SRS by relying on a controlled experiment. [*Results*] Our results suggest that the relevance of SRS quality depends both on particular project characteristics and what is considered as a quality defect; for instance, the domain of safety critical systems seems to motivate for an intense usage of SRS as a means for communication whereas defects hampering the pragmatic quality do not seem to be relevant as initially thought. [*Conclusion*] Efficient and effective quality assurance measures must be specific for carefully characterized contexts and carefully select defect classes.**

## I. Introduction

Stakeholder-appropriate requirements constitute critical determinants of project quality. Incorrect or missing requirements are supposed to lead to various problems in later phases such as effort and time overrun or an increased effort in acceptance testing [1]. In fact, a large extent of documented project failures are meant to be caused by insufficient requirements engineering (RE) [1]–[4].

It became conventional wisdom that the quality of the created RE artifacts, most prominently the software requirements specification (SRS), are a measurement for the overall process quality as weaknesses in the SRS might cause problems in subsequent development phases. In literature, we can consequently find various proposals on how to structure an SRS including content models (e.g., [5]), and broader documentation guidelines as well as best-practices (e.g., [6], [7]), and finally quality models (e.g., [8]–[12]) on what characteristics an SRS should feature, all together supposed to improve the quality in RE processes. The role and relevance of RE artifacts thereby became a frequent field of investigation (see e.g. [13]).

However, yet not completely answered is the question how much the process quality is eventually determined by the quality of the artifacts, which includes also the question how much project participants eventually rely on the created artifacts. Consequently, we are confronted with the following interesting and, at the same time, challenging questions

- under which (project) circumstances an SRS matters,
- what quality dimensions of an SRS matter, and finally
- how we can properly assure the quality of an SRS.

Since the explicit documentation of requirements and their quality assurance are labor-intensive tasks, practitioners are often confronted with a trade-off between effort and adherence to schedules on the one hand and the achievement of the necessary quantity and quality of requirement documentation on the other hand.

**Problem Statement:** So far, we lack an empirical understanding on the extent to which SRS are created and used in practice, as well as the degree to which the quality of an SRS matters to subsequent development activities that rely on the artifacts. Such an understanding would also allow for a critical reflection on the effectiveness of SRS-based quality assurance.

**Research Objectives:** We aim at contributing to a better understanding on the impact the quality of an SRS eventually has and formulate two research objectives to understand:

RO 1: To which extent and under which conditions are SRS created and used?
RO 2: Does the quality of an SRS matter to subsequent development activities?

**Contribution:** In the paper at hands, we make two contributions:

1) We conduct a survey to explore to which extent and under which conditions SRS are created and used in practical environments. This contribution shall address RO 1.
2) We design and execute a controlled experiment to analyse the degree to which the quality of an SRS matters to subsequent development activities. This contribution shall address RO 2.

The results of our survey are presented in Sect. III, the results of our experiment are presented in Sect. IV. Based on our findings, we critically reflect on the challenges introduced above regarding SRS-based quality assurance in RE in Sect. V.

## II. Related Work

Several studies address (the use of) *documentation* of requirements engineering in practice. In a qualitative study [13], Liskin investigated, among other, the suitability of specific RE artifacts for activities related with requirements specifications, by means of interviews. Lethbridge, Singer and Forward [14] reported on three empirical studies on documentation in software engineering in practice. Results identified both applications of documentation in general during software engineering and issues with documentation, in particular, outdated information.

Furthermore, there exists empirical evidence providing insights into how requirements are *communicated*. Abelein and Paech [15] conducted a series of semi-structured interviews concerning the state of the practice of user-developer communication in large-scale IT projects. The results of their study indicate that the direct user-developer communication is limited and that no common method for this communication in the design and implementation method exist. We extend the context of their work by adding multiple stakeholders (i.e. users of the SRS) and make it more concrete by using the SRS as single means for communication. Bjarnason et. al. conduct in [16] an explanatory case study in order to deepen their understanding of the cause and effects of communication gaps in a large-scale industrial setup. Their results show that that communication gaps cause failure to meet the customers' expectations, quality issues, and wasted effort. In contrast to this work, their study is of explanatory nature, and furthermore has a larger scope; They address communication gaps of requirements in general, while we refine the existing body of knowledge by relating the relevance of documentation and its use for communication, taking also into account project-specific circumstances, in order to provide a distinctive picture on the usage of SRS in practice.

## III. Relevance of SRS: An Expert Survey

To answer the question overall about the whether and how mich the quality of SRS matter, we must first clarify whether it is actually used. To this end, we conducted a survey with a broad spectrum of practitioners from one industrial partner to explore the extent to which SRS are created and used.

In the following, we first introduce the research questions and the survey design, before summarizing and discussing the results.

### A. Research Questions

In this study, we explore two facets of an SRS, namely its degree of completeness and detail in the documentation of requirements in a persistent artifact and its use as a means for communicating requirements within RE and beyond RE. To this end, for formulate two research questions described next.

*RQ 1-1: To which extent are requirements documented?:* This research questions examines the degree to which requirements are documented in SRS or comparable artefacts (e.g., product backlogs). We distinguish between two dimensions when documenting requirements. First, we want to know how comprehensive requirements are documented in terms of quantity, i.e. what is the proportion of documented requirements compared to all requirements identified during requirements engineering. Second, requirements can be specified with varying level of detail. Therefore, we want to know how detailed the requirements are documented.

*RQ 1-2: To what degree are SRS used to communicate requirements?:* SRS, or comparable artifacts, also serve the purpose to communicate requirements from stakeholders to various roles in the systems engineering process, e.g., architects, implementers or testers. RQ 2 investigates the degree to which internalization of knowledge about requirements is based on the SRS. To this end, we want to know whether and how often a SRS is used as a means for communication considering both the communication within RE and the communication of requirements to subsequent development activities. In case it is used, we further want to know as how important it is seen in comparison with other means of communicating requirements in practice.

*Is SRS usage related to specific project circumstances?:* In a secondary study, Kalus and Kuhrmann [17] identified criteria which lead to (i) an expansion resp. reduction of documentation within projects, and (ii) an orientation towards formalized resp. open communication patterns. Since we expect the SRS to not be used equally under all circumstances, we want to know in addition to both research questions if there are specific criteria which influence whether requirements are documented (*RQ 1-1*) and a SRS is used for communicating requirements (*RQ 1-2*). We therefore use the secondary study of Kalus and Kuhrmann [17] as a basis for our hypotheses against which we test our data samples.

### B. Survey Design

The survey was conducted at a large, multi-national company headquartered in Germany. Although operating in different domains, typical products are medium to large systems or engineering solutions in which software plays a significant or even crucial role.

*Participants:* We targeted participants directly or indirectly involved in requirements engineering for software-intensive systems, either in the sense of being involved when eliciting and specifying requirements, or in the sense of relying with their particular activities on requirements, e.g. architects or implementers.

*Survey Instrument:* The questionnaire consists of two main parts. Part I includes questions on the frequency of project characteristics to occur independent of individual projects, and part II refers to the most recently completed project the participant was involved in. For that particular project, the participants are asked to characterize (a) the project itself, (b) the degree to which requirements are documented in a SRS and/or (c) the use of the SRS as a means to communicate requirements. Questions of part II(b) and II(c) are only shown if the participant specified she was involved in the elicitation and specification of requirements respectively required knowledge of requirements for her tasks, in the particular project. In the questionnaire, we relied on closed questions with Likert-scales, and occasionally open questions to capture rationales or unforeseen options, e.g., means of communications. The Likert-scales were defined on an ordinal scale from 1 (e.g. *"I strongly disagree"*) to a maximum of 6 (e.g. *"I strongly agree"*) to avoid that they check the middle. Details on the instrument are available online[1].

*Data Collection:* We implemented the questionnaire as an online survey using the *Enterprise Feedback Suite 10.5* tool. Due to organizational restrictions, we only conducted the survey anonymously. We made the survey available to participants working in systems engineering project via an announcement on selected working-group mailing lists of the company. In addition, we selected participants based on former,

[1] www4.in.tum.de/~mund/srs-quality-om.zip

company-internal projects conducted in collaboration with our partner. However, the list of participants was undisclosed to us.

*Data Analysis:* For RQ 1-1, we considered only those participants who stated to be involved in requirements elicitation or specification. For both the completeness and the level of detail, we extracted the number of projects for each level of the ordinal scales. For RQ 1-2, we compared the number of projects in which an SRS (paper-based and tool-based) was used to those of meetings/workshops, personal talks, and groupware solutions. Furthermore, for those respondents who stated to use an SRS, we evaluated the participants ranking of the communication means according to specified relevance for informing about requirements.

We investigated the relation between project criteria and the SRS by means of rank-based correlation coefficients and hypothesis test based on Kendall's $\tau$. We rated $\tau \geq 0.3$ as moderate and $\tau \geq 0.5$ as strong correlation. For testing our hypotheses, i.e. those phenomena we expected to occur in evidential relation to certain project circumstances (see our baseline [17], respectively Tab. II), we used a significance level of $\alpha = .05$. Because we tested multiple hypothesis, we also calculated an adjusted p-value $p_{fdr}$ based on Benjamini and Hochberg [18] to mitigate the threat that correlations were only found by chance. In addition, we extracted reasons from open questions for complete, incomplete, shallow and detailed specifications, which we then either assigned to existing criteria or generalized as candidates for new criteria.

*Validity Procedures:* The instrument for our survey was reviewed by two additional researchers and two industrial partners who also checked no terms with different or ambiguous meanings within the company were used. To avoid both bias towards single projects and multiple answers, we verified all projects gathered in the survey are unique by manually comparing the specified project names (mandatory). Qualitative data resulting from open questions was reviewed independently to avoid misinterpretation. For identified correlations (RQ 1 & 2), results were visualized using bubble plots and checked for plausibility.

### C. Results and Interpretation

In the following, we summarize our results structured according to the research questions. For each, we conclude with a brief interpretation of the results.

*Study Population:* The survey was accessed 85 times, of which 46 participants (54%) completed the survey[2]. Four participants did specify to neither being involved during requirements specification nor that knowledge on requirements was required for their tasks. The remaining 42 participants had an average experience of $\geq$10 years and completed $6-10$ projects on average. Projects specified by the participants were balanced between products and custom solutions (23 to 19) and between new and continuous development (22 to 20). The majority of projects (71%) had release cycles between six month and two years, but short ($\leq$6 months) and long ($\geq$5 years) also occurred.

[2]61 participants (72%) partly completed the survey, mostly until the mandatory specification of the project name

TABLE I. PARTICIPANTS BY PROJECT ROLE AND RE INVOLVEMENT (DURING SPECIFICATION OR IN THE SENSE OF REQUIRED KNOWLEDGE OF REQUIREMENTS)

| Role | Specification | Required Knowledge (excl.) | Total |
|---|---|---|---|
| Product Manager | 9 | 4 (-) | 9 |
| Project Lead | 5 | 4 (-) | 5 |
| Req. Engineer | 9 | 5 (1) | 10 |
| Architect | 5 | 8 (3) | 8 |
| Implementer | - | 4 (4) | 4 |
| Tester | 1 | 1 (-) | 1 |
| Quality Manager | 1 | 2 (1) | 2 |
| Other | 1 | 3 (2) | 3 |
| All | 31 | 31 (11) | 42 |

*RQ 1: Documenting requirements in SRS:* In general, our results indicate to a rather comprehensive usage of SRS to document requirements. Considering the scope, the respondents stated for 15 out of 31 projects that all identified requirements were actually documented in a SRS, and in only four cases (13%) half or less of identified requirements were documented (Fig. 1a). The level of detail of the documented requirements was balanced between shallow and detailed specifications (17 cases, 55%). SRS were considered shallow (3 cases, 10%) and exhaustively detailed (1 case, 3%) only rarely (Fig. 1b). We found a moderate rank-correlation (Kendall-$\tau = 0.33$, $p = 0.04$) between the degree of completeness and the level of detail in SRS, depicted as a bubble chart in Fig. 1c.

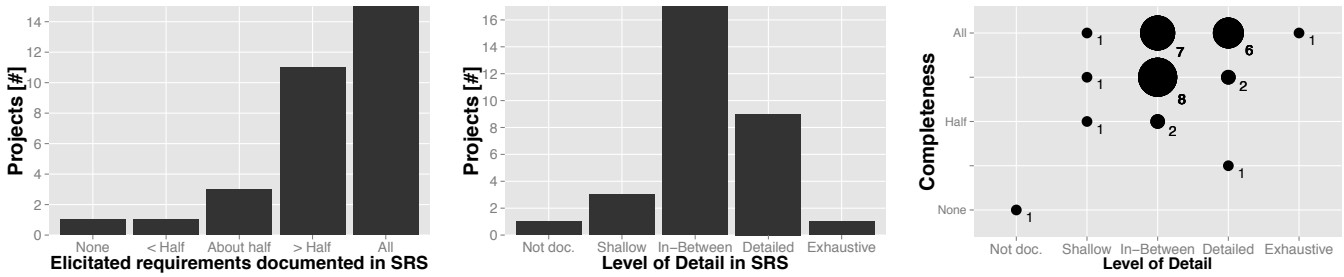
The relationship between project criteria and the documentation, i.e., the degree of completeness and the level of detail, is described by the rank-correlation coefficients and associated $p$-values in the left part of Tab. II. The data shows positive correlations between the completeness and three factors, namely that

1) safety/security concerns are relevant for the project goal
2) measurements are required (both project-specific and overall)
3) the team and stakeholders work in a good and collaborative way (project-specific only)

All but the last also positively correlate with the level of detail in the SRS, but the project-specific correlation regarding measurements was not statistically significant. In contrast, a high complexity of the system under consideration (project-specific and overall) and volatile requirements (overall only) correlate negatively with the completeness of the SRS. The circumstance that the stakeholders and the team worked together in previous cooperation negative correlations with the level of details in an SRS.

Qualitative feedback supports the visible correlations. Several participants stated that the degree of completeness in the SRS was required by the development process, with multiple participants stating that it is a consequence of the domain (*regulatory requirements (healthcare sector) enforce documentation*) and/or required for ensuring traceability (which is perceived as *mandatory for healthcare products* and *required for distributed teams*.

In agile processes, SRS documents were perceived as complete due to *only documented requirements entering the*




(a) Degree of Completeness (b) Level of Detail (c) Correlation of (a) and (b)

Fig. 1. Survey results on requirements documentation (RQ1): portion of documented requirements (a), associated level of detail (b) and correlation (c)

*backlog*. Moreover, participants stated that complete requirement specifications were obtained due to the *involvement of all (relevant) stakeholders* and *iterations between development and product management*. In contrast, bare *[existence of] too many requirements (thousands)* or the application of *traditional RE approaches on complex projects* resulted in incomplete requirements specification according. Long release cycles were also perceived as a reason either directly, since *topics [. . . ] which are relevant late were documented very coarsely* and *long project durations [imply] many changes*, or more indirectly, because *permanent changing goals, constraints, stakeholders and project teams*. Also, external documentation (e.g., *availability of legacy systems*), limited time (*restrictive timeboxing* and *not enough time [. . . ] to elicit all requirements*), and *stakeholder lacked knowledge of requirements in detail* were mentioned.

For the level of detail, participants mentioned that *good coordination of requirements in the team allowed for less detail in documentation* and a *rough direction [is] enough [because] experts clarify details during implementation*. Most predominantly, participants stated solution-orientation as a reason for a shallow level of detail including statements like (*results more important than documentation*, *urgency for technical results limits time for specifications*) and *not enough time and resources for detailed analysis*. *Development process constraints* were mentioned as an exemplary reason for detailed SRS.

*Interpretation:* In general, requirements seem to be not exhaustively documented in an SRS in every project. Our results suggest that requirements are, however, documented in nearly every project, and with substantial quantity (completeness) and a high level of detail. Since the results indicate to a higher exhaustiveness regarding the completeness than the level of specification detail (prevalence of projects above an imaginary diagonal in Fig.1c), we conclude so far that the former is more important than the latter. If we take specific project circumstances into account, we can observe that correlations obtained for individual projects are generally weaker than for the overall set of projects the respondents worked in. This may indicate that documentation is influenced to a large degree not by individual project circumstances, but by the chosen domain-specific development process.

Based on the revealed correlations and a priori hypothesis [17], we conclude that safety/security concerns and demand for measurement increase the need for documentation, and propose a novel hypothesis that a good cooperation between stakeholders (principal) and agent allows more exhaustive documentation, and vice versa. For negative correlations, we argue that volatile requirements hamper the degree of completeness of SRS. However, we are indecisive if high complexity decreases the need for documentation, because of the inherent inefficiency associated with the difficulties of documenting such requirements (*uselessness of traditional RE approaches*), or whether it simply impedes documentation without actually diminishing its need.

*RQ 2: SRS as Communication Means:* The SRS, independent of its materialization (paper-based and tool-based), was used as a means to communicate requirements in 23 out of 31 projects (74%), ranked third behind meetings/workshops (29 projects, 94%) and personal talks (27 projects, 87%, cf. Fig. 2a). Considering participants not involved during specification exclusively, the SRS is used slightly more often (82%, $+8\%$), while meetings/workshops (91%, $-3\%$) and personal talks (82%, $-5\%$) are used marginally less often. In any case, groupware solutions are used rarely (20% and 27%, respectively). To further investigate the importance of the SRS as a communication means, we asked the participants to rank the specified means by how informative they were perceived for their individual project tasks. Our results reveal that in case an SRS is used, it is the primary source for communicating requirements (55%), but only slightly more often than meetings/workshops (45%). In fact, we observed a pronounced polarisation between SRS-based and artifact-agnostic communication. If meetings are the primary source, individual personal talks were specified predominantly as the secondary source of information, with the SRS being used in only one case as the secondary means. Out of five projects using groupware solutions, it was considered the least in all but one case, where it was ranked second to last, attesting groupware solutions an inferior relevance for communication. Considering only participants not involved during the requirements specification, SRS-based communication of requirements was considered as primary source significantly more often (75%, $+20\%$), indicating its superiority for communicating requirements for development phases subsequent to requirements engineering.

Compared to the documentation of requirements, we observed a more alleviated effect of project characteristics on the communication of requirements. First and foremost, we could not reject the null hypothesis for the principal usage of an SRS, and identify only two correlations for the ranking of

communication means: a strong correlation ($\tau$=0.52, $p$=0.01) with the relevancy of safety/security for the project goal, and a moderate correlation with the length of release cycles ($\tau$=0.33, $p$=0.08). However, our results show several weak correlations (around $\tau$=0.2) which are in tune with our hypothesis. For instance, we found weak correlations for team parameters regarding size and turnover, as well as some of the factors identified as significant for the documentation (cf. Sec. III-C), such as the demand for measurements ($\tau$=0.27 for ranking) or high complexity ($\tau$= –0.18 for usage and $\tau$=– 0.12 for ranking). In addition, we investigated whether the participants' background knowledge regarding the domain or product impacts the communication. However, we could not reject the null hypothesis, and hence background knowledge may not have an impact on the communication means used[3].

*Interpretation:* Overall, we conclude so far that SRS are a well-established means to communicate requirements. However, as we cannot guarantee that our participants reflect the composition of project teams in practice, we focus on a distinction between two important communication relationships, namely a communication within RE activities, e.g. elicitation and negotiation, and a communication of RE results to subsequent development activities, e.g. testing. We rely our interpretation on the observation that participants involved during the specification of requirements exhibit significantly different communication preferences than participants who only required knowledge of the requirements for their individual project assignments. Consequently, we argue that for projects with a high degree of division of labor in terms of project activities, communication within RE and adjacent activities (e.g., high-level architecture, cf. Tab. I) non-artifact-based communication means prevail, while for the communication of SRS to subsequent development activities, an SRS seems predominately used for that communication.

In contrast to the documentation of requirements, the use of SRS as a communication means may be less determined by the development processes but be specific to the project or even the individual. The later could also explain why it was only possible to reveal weak correlations, e.g., team size ($\tau$=0.20) and distribution ($\tau$=0.22). For moderate or strong correlations, we propose the following causal interpretations and possible explanations: the length of release cycles impacts the use of SRS for communication (e.g., because of the persistent nature of artifacts), and that safety/security concerns demands documented traceability. Also, we interpret that the degree of documentation effects the role of the SRS for communication, supported by the revealed moderate correlation between the degree of documentation and its use for communication ($\tau$=0.36 for completeness, and $\tau$=0.46 for level of detail), but limited to participants involved in specification *and* requiring knowledge of requirements for their project tasks.[4]

In summary, we therefore draw the conclusion that the SRS is created in detail and with a high degree of completeness under specific project circumstances (such as the application domain) to communicate requirements.

[3]Note that due to scoping, we did not investigate whether there is *less* communication, independent of the actual *means* to transfer knowledge

[4]Since we relied on the specified completeness and level of detail as perceived by participants during specification.

### D. Limitations

Considering the internal validity, we had to cope with limited control regarding sampling and delivery because of the particular industrial setting. Hence, we were unable to establish a random sampling or accomplish higher response rates. Therefore, statistical results have to be considered with a salt of grain and participation bias is possible. While we used the survey results to gain first insights into when and how the SRS is used, the number of participants was too low to apply statistical methods reliably when discriminating between aspects, e.g. for the subgroup of participants not involved in the specification of requirements.

Also, we cannot conclude with statistical significance that the observed correlations occurred only by chance, since most of the corrected confidence intervals $p_{fdr}$ were above the threshold. Despite our validity procedures, we suspect some terms still to be subject to misinterpretation, partly because of the heterogeneity of requirements engineering in practice.

## IV. Impact of Quality Defects in SRS: A Controlled Experiment

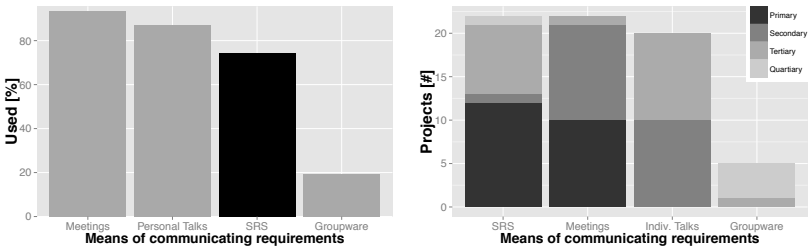
Our study in Sect. III concludes that in different project situations SRS are used to foster communication with subsequent activities. Here, we investigate the complementary question to what extend the quality of the RE artifacts, as used, actually impacts subsequent engineering activities. Since a complete analysis of all subsequent activities is infeasible in a controlled setting, our study focuses on an activity that strongly depends on the contents of SRS: system testing. We further focus on two deficiencies of SRS, an exemplary one for semantic quality and another one for pragmatic quality (relying on the terminology of Lindland et al. [19]).

*RQ 2-1: Do incorrect SRS statements impact system testing?:* According to [19], semantic quality defects can be characterized as incomplete and/or incorrect information in the SRS with respect to the stakeholder's actual demands on the system. In RQ 2-1, we investigate whether incorrect information in the SRS inevitable leads to flawed system test cases or makes the inference of system tests less efficient.

*RQ 2-2: Do negative SRS statements impact system testing?:* In contrast, RQ 2-2 focuses on pragmatic quality, i.e. the unambiguous comprehensibility of the SRS by the target audience, e.g., test engineers. Such defects (cf. ISO29148 [20] for a list) describe valid information but can nonetheless lead to flawed test cases if the SRS is misunderstood or not understood at all. In a previous investigation [21], practitioners were unsure about the validity of *negative statements*, one quality factor in this list, yet without empirical foundation. Therefore, we investigate the impact of this particular defect on system testing in terms of omissions or incorrect test cases.

*RQ 2-3: Does domain knowledge compensate for quality defects in SRS?:* The main confounding factor that prevents generalization of results from controlled settings is context knowledge. Hence, we want to know if and to what degree a-priori knowledge about the application domain, also called the problem space [22], effectively compensates the quality defects of RQ 2-1 and RQ 2-2.

Fig. 2. Usage of SRS: means for communicating requirements ranked according to the participants' perceived importance for information




(a) Usage (b) Ranking

TABLE II. IMPACT OF PROJECT PARAMETERS ON THE RELEVANCE OF THE SRS (MODERATE CORR. FOR $\tau \geq .3$, SIG. LEVEL $\alpha = .05$)

| Parameter | | Impact on Documentation in SRS | | | | | | Impact on Communication using SRS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Completeness | | | Level of Detail | | | SRS used? | | | SRS Ranking | | |
| | | $\tau$ | $p$ | $p_{fdr}$ | $\tau$ | $p$ | $p_{fdr}$ | $\tau$ | $p$ | $p_{fdr}$ | $\tau$ | $p$ | $p_{fdr}$ |
| **Team Size** | Proj. | -0.08 | 0.63 | 0.85 | 0.06 | 0.72 | 0.95 | 0.07 | 0.65 | 0.8 | 0.2 | 0.28 | 0.63 |
| | Gen. | -0.08 | 0.6 | 0.82 | 0.17 | 0.28 | 0.52 | 0.04 | 0.81 | 1 | 0.05 | 0.79 | 0.89 |
| **Team Distribution** | Proj. | 0.06 | 0.7 | 0.85 | 0.02 | 0.89 | 0.95 | 0.09 | 0.59 | 0.8 | 0.02 | 0.92 | 0.92 |
| | Gen. | -0.04 | 0.81 | 0.84 | 0.25 | 0.12 | 0.32 | -0.14 | 0.41 | 1 | 0.09 | 0.62 | 0.85 |
| **Team Turnover** | Proj. | -0.02 | 0.88 | 0.93 | 0.06 | 0.71 | 0.95 | -0.01 | 0.94 | 0.94 | 0.22 | 0.26 | 0.63 |
| | Gen. | -0.09 | 0.56 | 0.82 | 0.01 | 0.94 | 0.95 | -0.04 | 0.83 | 1 | 0.19 | 0.32 | 0.75 |
| **Management unavailable** | Proj. | 0.01 | 0.93 | 0.93 | 0.06 | 0.69 | 0.95 | 0.1 | 0.54 | 0.8 | 0.13 | 0.52 | 0.73 |
| | Gen. | 0.03 | 0.84 | 0.84 | 0.28 | 0.08 | 0.29 | 0.08 | 0.61 | 1 | 0.26 | 0.19 | 0.75 |
| **Financial controlling req.** | Proj. | 0.09 | 0.59 | 0.85 | 0.12 | 0.44 | 0.95 | 0.15 | 0.37 | 0.77 | 0.12 | 0.53 | 0.73 |
| | Gen. | -0.05 | 0.73 | 0.84 | 0.09 | 0.55 | 0.75 | 0.3 | 0.07 | 0.71 | -0.18 | 0.34 | 0.75 |
| **Measurement req.** | Proj. | 0.34 | 0.03 | 0.18 | 0.27 | 0.1 | 0.53 | 0.14 | 0.4 | 0.77 | 0.27 | 0.17 | 0.62 |
| | Gen. | 0.37 | 0.02 | 0.08 | 0.37 | 0.02 | 0.16 | 0.25 | 0.13 | 0.71 | -0.21 | 0.29 | 0.75 |
| **Many stakeholders** | Proj. | -0.1 | 0.55 | 0.85 | 0.06 | 0.71 | 0.95 | -0.05 | 0.77 | 0.85 | 0.04 | 0.84 | 0.92 |
| | Gen. | -0.11 | 0.51 | 0.82 | 0.16 | 0.33 | 0.52 | 0.14 | 0.42 | 1 | -0.15 | 0.46 | 0.8 |
| **Stakeholder unavailable** | Proj. | -0.12 | 0.46 | 0.85 | 0.03 | 0.87 | 0.95 | 0.21 | 0.22 | 0.77 | 0.08 | 0.69 | 0.84 |
| | Gen. | -0.17 | 0.29 | 0.64 | 0.17 | 0.28 | 0.52 | 0 | 1 | 1 | -0.03 | 0.89 | 0.89 |
| **High complexity** | Proj. | -0.31 | 0.06 | 0.21 | 0.01 | 0.95 | 0.95 | -0.18 | 0.31 | 0.77 | -0.12 | 0.53 | 0.73 |
| | Gen. | -0.36 | 0.02 | 0.08 | 0.02 | 0.92 | 0.95 | 0.03 | 0.86 | 1 | 0.03 | 0.89 | 0.89 |
| **Safety/Security relevant** | Proj. | 0.38 | 0.02 | 0.18 | 0.36 | 0.02 | 0.25 | 0.14 | 0.42 | 0.77 | 0.52 | 0.01 | 0.08 |
| | Gen. | 0.32 | 0.04 | 0.12 | 0.34 | 0.03 | 0.16 | 0.01 | 0.96 | 1 | 0.38 | 0.05 | 0.51 |
| **Release Cycle Length** | Proj. | -0.19 | 0.24 | 0.67 | -0.01 | 0.95 | 0.95 | 0.26 | 0.11 | 0.77 | 0.33 | 0.08 | 0.46 |
| **Previous cooperation** | Proj. | -0.03 | 0.83 | 0.83 | -0.33 | 0.04 | 0.16 | -0.11 | 0.52 | 0.81 | -0.07 | 0.71 | 0.81 |
| | Gen. | -0.01 | 0.95 | 0.95 | -0.19 | 0.22 | 0.45 | 0.03 | 0.87 | 0.96 | -0.17 | 0.38 | 0.81 |
| **Good cooperation** | Proj. | 0.44 | 0.01 | 0.04 | 0.19 | 0.25 | 0.34 | 0.07 | 0.71 | 0.81 | -0.05 | 0.81 | 0.81 |
| | Gen. | 0.26 | 0.11 | 0.23 | -0.15 | 0.36 | 0.48 | -0.07 | 0.69 | 0.96 | -0.29 | 0.15 | 0.81 |
| **Small Budget** | Proj. | 0.12 | 0.45 | 0.6 | -0.19 | 0.23 | 0.34 | 0.04 | 0.81 | 0.81 | -0.14 | 0.48 | 0.81 |
| | Gen. | 0.01 | 0.95 | 0.95 | -0.27 | 0.08 | 0.32 | 0.18 | 0.28 | 0.96 | 0.02 | 0.92 | 0.81 |
| **Volatile Requirements** | Proj. | -0.21 | 0.19 | 0.38 | -0.04 | 0.81 | 0.81 | -0.08 | 0.62 | 0.81 | -0.1 | 0.59 | 0.81 |
| | Gen. | -0.47 | 0 | 0.01 | -0.01 | 0.95 | 0.95 | -0.01 | 0.96 | 0.96 | -0.13 | 0.51 | 0.81 |
| **Prev. Domain Knowledge** | Proj. | - | - | - | - | - | - | 0.28 | 0.09 | 0.18 | 0 | 1 | 1 |
| **Prev. Product Knowledge** | Proj. | - | - | - | - | - | - | -0.03 | 0.85 | 0.85 | -0.08 | 0.68 | 1 |

### A. *Experiment Design*

We inject pre-defined defects into real-world use cases and ask experiment participants to specify system test cases that appropriately verify the stakeholders' requirements. To this end, we provide tabular templates to be filled out within a 45 minutes time slot. No questions are allowed during the experiment. Additionally, we ask the participants about how difficult they perceived the inference of test cases for each use

TABLE III. EXPERIMENT TREATMENT: PRESENCE OF DEFECTS (Y/N) INJECTED INTO USE CASES OF SRS

| Use Case | Defect | Type | Correct statement | Flawed statement |
|---|---|---|---|---|
| UC 1 | D1.1 | Incorrect | *In case the data is complete and valid, the data will be imported.* | *In case the data is complete and valid,* ***an error message occurs****.* |
| | D1.2 | Incorrect | *Data records must then be further approved by a case handler and explicitly activated.* | *Data records must then be checked by a* ***plausibility algorithm and are activated afterwards****.* |
| UC 2 | D2.1 | Negation | *The user must enter at least one character.* | *The user* ***is not allowed to enter zero characters.*** |
| | D2.2 | Negation | *The system must treat lowercase and uppercase letters the same.* | *The system* ***must not distinguish*** *between lowercase and uppercase letters.* |
| | D2.3 | Negation | *The user may only select one company at a time.* | *The user* ***cannot select more than one company*** *at a time.* |
| UC 3 | D3.1 | Negation | *The user may nominate up to three substitutes.* | ***No user must nominate more than*** *three substitutes.* |
| | D3.2 | Negation | *If the user selects herself as a substitute, an error message is shown.* | *A user* ***cannot select herself*** *as a substitute.* |

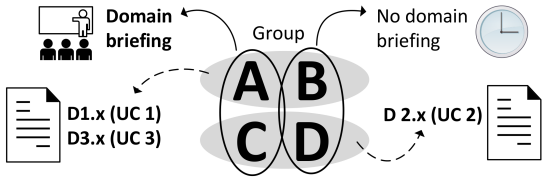



Fig. 3. Experiment Design Overview

TABLE IV. METRICS FOR EVALUATING TESTING IMPACT

| Metric | Description |
|---|---|
| $Total_{\texttt{Grp,Def}}$ | Number of rated test cases regarding defect(s) `Def`, limited to participants from `Grp` |
| $Correct_{\texttt{Grp,Def}}$ | Number of correct test cases regarding defect(s) `Def`, limited to participants from `Grp` |
| $Detected_{\texttt{Grp,Def}}$ (D 1.x only) | Number of test cases which detected `Def`, limited to participants from `Grp` |
| $Omitted_{\texttt{Grp,Def}}$ (D 2.x & 3.x only) | Number of test cases which omitted to test for the requirement specified by `Def`, limited to participants from `Grp` |
| $R\_Correct_{\texttt{Grp,Def}}$ | $Correct_{\texttt{Grp,Def}}$ / $Total_{\texttt{Grp,Def}}$ |
| $R\_Detected_{\texttt{Grp,Def}}$ | $Detected_{\texttt{Grp,Def}}$ / $Total_{\texttt{Grp,Def}}$ |
| $R\_Omit_{\texttt{Grp,Def}}$ | $Omitted_{\texttt{Grp,Def}}$ / $Total_{\texttt{Grp,Def}}$ |

case, both quantitatively (8 point Likert-scale) and qualitatively using open questions.

To evaluate the research questions, we randomly assign participants to one of four Groups A-D (see Tbl. III):

*For RQ 2-1*, we inject two defects into UC1: An obviously incorrect defect D1.1, which requires to show an error message in case of success, and a more subtle defect D1.2, which suggests that a certain strictly manual check is executed automatically. Groups A & B are faced with this flawed use case, whereas Groups C & D serve as control group.

*For RQ 2-2*, we convert five positively stated requirements in UC2 and UC3 into their negative versions (D2.x and D3.x), carefully preserving the meaning of the each statement. Here, Groups A & B receive the negated version of UC3 and serve as control group for UC2, and Groups C & D receive the negated version of UC2 and serve as control group for UC3.

*For RQ 2-3*, we provide the participants of Groups A & C with certain knowledge about the domain, directly before assigning the task to them: This briefing includes the purpose of the overall system, the relevant business processes, important rationales and necessary constraints from the perspective of a long-term employee of the company. In particular, the briefing includes the intended behavior for the defects D1.1 and D1.2. At the end of the presentation, questions are allowed to further foster the participants' understanding. For this RQ, Groups B & D serve as the control group without domain knowledge.

### B. Study Objects

In order to keep the setting close to reality, we reuse a real-world SRS from an industrial partner. The original requirements specification was 21 pages long and written in natural language. For the experiment, we selected three out of 18 use cases, together with the original overview description and problem statement. All company-specific terms and acronyms were either removed or renamed due to legal reasons.

### C. Data Collection and Analysis Methodology

We evaluate the obtained test cases for each defect by manual inspection: We assign `correct` to a test case if and only if the test explicitly covers the stakeholder's intended requirements (correct versions in Tab. III), `flawed` if unintended requirements are tested, and `omit` if the test does not cover the requirement at all. For semantic defects (RQ 2-1), we furthermore inspected whether the participants actually `detect` the defects (i.e. are aware of it). We evaluate this based on whether we encountered remarks on D1.1 or D1.2 in the test cases, use cases or open questions. All metrics used for analysis are listed in Tab. IV, and we apply statistical tests for the following (alternative) hypotheses:

- $\mathbf{H}_{A,C}$ The presence of a defect in the use case and the correctness of the test cases (regarding this defect) are not independent.
- $\mathbf{H}_{A,O}$ The presence of a defect in the use case the omission of the associated requirement in the test cases are not independent.
- $\mathbf{H}_{A,D}$ The perceived difficulty is different for use cases with defects present.

For the impact on the test quality, we test $\mathbf{H}_{A,C}$ and $\mathbf{H}_{A,O}$ (D 2.x and 3.x only) using Pearson's $\chi^2$ test for each defect. To evaluate the impact on efficiency, we apply the Mann-Whitney test to $\mathbf{H}_{A,D}$ in order to evaluate the impact on efficiency. Both times, we demand a significance level of $\alpha$=0.05. Furthermore, to validate the direction of the impact, i.e. whether it is indeed negatively for quality defects, we consult *R_Correct* and *R_Omit* for correct and flawed use

TABLE V. EXPERIMENT RESULTS (SIG. LEVEL $\alpha = .05$)

| Defect | | Independence | | R_Correct$_{\mathrm{ID}}$ | | R_Det./R_Omit | |
|---|---|---|---|---|---|---|---|
| ID | Grp | Omit | Corr. | Correct | Flawed | Correct | Flawed |
| | All | - | 0.01 | 1.00 | 0.47 | - | 0.80 |
| D 1.1 | A&C | - | 0.18 | 1.00 | 0.50 | - | 0.83 |
| | B&D | - | 0.07 | 1.00 | 0.44 | - | 0.78 |
| | All | - | 0.00 | 1.00 | 0.00 | - | 0.00 |
| D 1.2 | A&C | - | 0.01 | 1.00 | 0.00 | - | 0.00 |
| | B&D | - | 0.00 | 1.00 | 0.00 | - | 0.00 |
| D 2.1 | All | 0.54 | 0.88 | 0.83 | 1.00 | 0.68 | 0.53 |
| D 2.2 | All | 0.20 | - | 1.00 | 1.00 | 0.26 | 0.53 |
| D 2.3 | All | 0.28 | 0.71 | 1.00 | 0.80 | 0.47 | 0.71 |
| D 3.1 | All | 0.61 | 0.34 | 0.56 | 0.80 | 0.25 | 0.42 |
| D 3.2 | All | 0.22 | - | 1.00 | 1.00 | 0.08 | 0.35 |
| D 2.x | All | 0.15 | 0.91 | 0.90 | 0.93 | 0.38 | 0.51 |
| & | A&C | 0.19 | 0.83 | 0.84 | 0.90 | 0.34 | 0.51 |
| D 3.x | B&D | 0.59 | 1.00 | 0.96 | 0.95 | 0.42 | 0.50 |

cases. For the relevance of domain knowledge (RQ 2-3), we compare the aforementioned metrics, based on whether they received the domain knowledge briefing before the experiment.

*Validity Procedures:* To ensure the reliability of the manual inspection of test cases and hence the assignment of results, a second researcher independently rated 11 test cases ($\tilde{25}\%$). The obtained inter-rater reliability measures attested a very high level of agreement: we agreed on 93% of all cases, and Cohen's $\kappa = 0.90$ is typically interpreted as (almost) perfect agreement. Furthermore, to avoid unintentional influence of groups A and C beyond explaining the business process and employee experiences, the presentation was voice-recorded and checked later on, e.g., for accidental hints about the test cases.

### D. Results and Interpretation

We conducted the experiment as part of our RE lecture at the Technical University Munich (TUM). Participants were mostly advanced undergraduate or early graduate students of computer science or information systems, and the experiment was conducted late in the term so that students had a fundamental understanding of the contents and applications of SRS.

Overall, 41 students participated in the experiment, with about ten students in every group. Tab. IV presents the obtained metrics and Fig. 4 illustrates the results of the self-evaluation of the participants.

*RQ 2-1: Impact of Incorrect Statements:* The correctness of the test cases obtained from flawed specifications differed substantially between the defects D1.1 and D1.2: While for D 1.2 no participant detected the defect and hence no correct test case was inferred, about half of all participants ($47\%$) explicitly corrected the defect D 1.1 although 80% of the participants actually recognized it. Hence, in 20% of the cases, the defect was not detected at all. In contrast, for both D1.1 and D1.2, in the control group, a correct specification also led to correct test cases. Therefore, we were able to reject the null hypothesis in favor of $\mathbf{H}_{A,C}$ with statistical significance ($p \leq \alpha$ = 0.05 for both, D 1.1 and D 1.2). Also, participants perceived the task as more difficult for UC 1 (D 1.1 and 1.2) (in favor of $\mathbf{H}_{A,D}$, p=0.02).

*Interpretation:* The presence of incorrect statements in the SRS does indeed impact system testing. This is true regarding the quality of the obtained test cases, for obvious defects (D 1.1) and even more for less obvious ones (D 1.2). In addition, the (perceived) difficulty indicates an increase in required efforts, and hence also impacts efficiency of testing.

*RQ 2-2: Impact of Negated Statements:* In contrast to incorrect use cases (RQ 2-1), the use of negative statements did not impact correctness of test cases substantially. For any negative statement but D 2.3, correct test cases were obtained at least as often as for the non-negative statement. Consequently, we were unable to reject the null hypothesis in favor of $\mathbf{H}_{A,C}$ for any negated statement. However, test cases did omit the specified requirement considerably more often ($+13\%$) but not significantly ($\mathbf{H}_{A,O}$, p=0.15). Participants perceived the task equally difficult ($\mathbf{H}_{A,E}$, p=0.46 (UC 2) and p=0.95 (UC 3)). Although not directly related to negative statements, participants expressed their dissatisfaction with the pragmatic quality of the specification as qualitative feedback, e.g., complaining about *wording of requirements*, *lack of chronological structure* and the *use of passive sentences*.

*Interpretation:* Our results suggest that negative statements do not impact testing in the sense that faults in terms of incorrect information are introduced into test cases, nor does it make the process of inferring these test cases more difficult. However, we observed an interesting but statistically insignificant increase in omission of requirements expressed using negative statements, which might potentially lead to lesser quality in test cases due to untested requirements. Future work should investigate this phenomena.

*RQ 2-3: Relevance of Domain Knowledge:* For both incorrect as well as negative statements, results of participants introduced to the underlying business process (groups A & C) did not vary considerably compared to the control group B & D. Notably, participants were unable to detect or correct D 1.2 (mandatory manual checks by a case handler).

*Interpretation:* We were surprised that, although we explicitly mentioned the correct behavior in D 1.2 in the briefing, participants were unable to compensate the defect. Within the experiment, the test engineers' knowledge of the system under consideration does neither compensate for incorrect requirements in specifications nor affects the quality of test cases inferred from specifications extensively using negative statements. Therefore, we conclude that, to a large extent, defects can propagate through the engineering process, even when people are briefed about the correct requirements.

### E. Threats to Validity

We see three threats as limitations of this experiment: First, we relied on RE students as participants. Therefore, the obtained test cases were of rather poor quality in general, and certainly not at the level of experts in the field of system testing. Second, a briefing cannot lead to the same depth of domain knowledge compared to own experiences and observations over a prolonged amount of time, which we

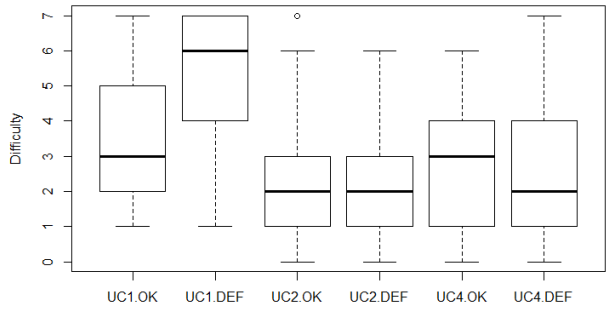



Fig. 4. Difficulty per use case as perceived by participants

expect to be superior, e.g., in terms of recognition and trust. We thereby need to extend the experiment with industry experts in the future. Last, we only investigated selected defects and, thus, we have to be careful to generalize results to classes of SRS quality [19], especially concerning pragmatic quality factors.

## V. Discussion: Relevance of SRS Quality

In this section, we discuss our study results considering our introductory stated questions.

### A. Under which circumstances does SRS quality matter?

A generic answer to this question is provided by the two studies: the survey indicates to a (context-specific) extensive use of the SRS in only about half of all projects, and the experiment suggests that certain quality factors do and certain quality factors do not spread to subsequent results. Our survey provided first indicators to circumstances which correlate with (i) a more extensive use of the SRS, e.g. the application domain of safety critical systems, whereas we could not confirm other circumstances proposed in literature. So far, our results indicate to two new challenges.

The first challenge is to determine the degree to which project circumstances may impact the *needs* to document an SRS to a certain degree. If following the taxonomy of Gorschek and Davis [23], we can see criteria of other dimensions beyond RE[5], which all simply may not permit to identify stronger correlations in practice today. Moreover, some circumstances might not be equally important to others and there might be circumstances that dominate others. Our data set already indicates, for example, that safety/security-relevant systems can be considered as a *dominant* circumstance in the sense that its presence dominates secondary circumstances such as the team-size. One potential explanation is that legal regulations enforce a rigor documentation no matter of secondary effects. However, although we could observe stronger correlations when considering non-safety-critical projects only, the number of projects was too small to draw meaningful conclusions.

The second challenge is to determine the degree to which project circumstances may impact the *possibilities* to document an SRS to a certain degree. For instance, the experiment simulated the unavailability of stakeholders by not permitting questions during the assignment. Yet, we noticed that students, working on UC 1 (incorrect statements, defects D 1 and 2) in particular, tried to ask questions in the beginning nevertheless. Also, one participant explicitly stated *[she] would like to be able to ask questions [to the customer].*

Both the needs to document an SRS to a certain degree and the possibilities that both arise from the characteristics of a project ecosystem need further investigations.

[5]For instance, criteria resulting from multi-project environments or from the socio-economic context of a customer including also cultural, psychological and even political facets.

### B. What dimensions of SRS quality matter?

In general, one may argue that the dimensions of semantic and pragmatic quality of SRS, as proposed by Lindland et al. [19], become more important if requirements are documented to a larger extent, respectively used more extensively for communicating requirements. Therefore, the revealed correlations between criteria and SRS-based documentation/communication (cf. Sec. III) are indicators for the relative importance for the semantic and the pragmatic quality as well. However, our experiment yields first insights into the absolute impact of quality: The defect D 1.1, i.e. an incorrect statement is easily recognizable, leads to flawed test cases for about every second participant, and no correct test case could be derived from the less obvious defect D 1.2. Although not part of the experiment, we do not expect better results for semantic defects in terms of missing information in the SRS. Therefore, we advocate a generalization for the semantic quality. That is, the semantic quality of the SRS is generally essential for subsequent engineering activities.

However, the impact of the pragmatic quality appears to be more diffuse. While we could show indicators that negated statements do not impact engineering activities, we refrain from generalizing our results to pragmatic quality in general. We may even assume that negated statements are rather easily correctable compared to other pragmatic quality issues, e.g., the use of passive voice. In fact, qualitative feedback suggested that pragmatic quality was at least perceived as an obstacle, and our participants omitted requirements expressed in negated statements considerably more often. We believe that the pragmatic quality is rich on facets we do not yet properly understand with some factors having more severe impacts than others. Consequently, we strongly postulate to question and rigorosly investigate factors that best practice norms on (pragmatic) quality propagate.

### C. How can we assure the quality of an SRS?

Since quality assurance always comes at a certain cost, an immediate conclusion of our result is that SRS-based approaches applied independent of contextual circumstances are inherently inefficient. Based on the results and discussion, we advocate that quality assurance is not applied in general, but only were actually required. This can be achieved by different means, e.g., by introducing tailoring mechanisms or by using context-specific inductive approaches. However, independent of the actual mean, an efficient SRS-based quality assurance must include a decision procedure which specifies when the quality of an SRS needs *not* to be assured. To this end, we hope the identified correlations of RQ 2-1 and RQ 2-2 will provide valuable first insights.

Finally an effective quality assurance must be able to assess[6] the quality of the SRS in a way that is meaningful for the engineering endeavor. To this end, our results provide first indicators. On the one hand, the semantic quality of the SRS strongly impacts the outcome of subsequent activities, and consequently, SRS-based quality assurance can be very effective and must be considered during quality assurance. However, certain pragmatic quality factors that are proposed by best practices were not as influential as initially thought. Since semantic quality is difficult to assess, in particular for the predominant form of natural-language specifications, approaches providing reliable indicators for semantic quality based on syntactic properties (e.g., [24]) seem promising.

## VI. Conclusion

In this paper, we presented an investigation that showed how the relevance of SRS quality may depend on both project characteristics and what is considered as a quality defect. Therefore, efficient and effective quality assurance measures should consider applicability for specific contexts, not neglect semantic quality, and carefully select defects regarding SRS understandability.

*Relation to Existing Evidence:* In [14], Lethbridge et al. observed that documentation is frequently out-dated. In RQ 1-1, we already discussed this might be one reason for the negative correlation between length of release cycles and SRS completeness. Interestingly, Lethbridge et al. also observed a moderate correlation ($0.43$, $p \leq .05$) between perceived accuracy and consultation frequency for requirements documentation, and, thus, one would suppose the SRS is used less for communication. However, the opposite appears in our result set ($\tau$=0.33, $p$=0.08). It is quite possible that in our study, we incidentally discovered a factor more important than perceived accuracy in long projects: the persistent nature of artifacts. In a previous experiment, we could show that passive voice as another pragmatic quality factor [25] leads to difficulties in understanding sentences. This strengthens our confidence on the need to carefully evaluate quality factors, since some have and some don't have an impact on subsequent activities.

*Future Work:* As future work, we will investigate both the needs to document an SRS to a certain degree and the possibilities that both arise from the characteristics of a project ecosystem. Furthermore, we believe that the pragmatic quality is rich on facets we do not yet properly understand with factors having more sever impacts than others, and we postulate the need to further investigate those factors propagated so far by best practice norms on (pragmatic) quality.

## References


[1] Méndez Fernández, D. and Wagner, S., "Naming the Pain in Requirments Enginering: A Design for a global Family of Surveys and First Results from Germany," *IST*, 2014.

[2] M. I. Kamata and T. Tamai, "How does requirements quality relate to project success or failure?" in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*. IEEE, 2007, pp. 69–78.

[3] S. Kujala, M. Kauppinen, L. Lehtola, and T. Kojo, "The role of user involvement in requirements quality and project success," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. IEEE, 2005, pp. 75–84.

[4] H. F. Hofmann and F. Lehner, "Requirements engineering as a success factor in software projects," *IEEE software*, vol. 18, no. 4, pp. 58–66, 2001.

[5] D. Méndez Fernández and B. Penzenstadler, "Artefact-based requirements engineering: The AMDiRE approach," *Requirements Engineering*, pp. 1–30, 2014.

[6] K. Wiegers, "Writing quality requirements," *Software Development*, vol. 7, no. 5, pp. 44–48, 1999.

[7] "Ieee guide to software requirements specifications," *IEEE Std 830-1998*, 1998.

[8] D. M. Berry, A. Bucchiarone, S. Gnesi, G. Lami, and G. Trentanni, "A new quality model for natural language requirements specifications," in *Proceedings of the international workshop on requirements engineering: foundation of software quality (REFSQ)*, 2006.

[9] D. Ott, "Defects in natural language requirement specifications at mercedes-benz: An investigation using a combination of legacy data and expert opinion," in *Requirements Engineering Conference (RE), 2012 20th IEEE International*. IEEE, 2012, pp. 291–296.

[10] J. Krogstie, O. I. Lindland, and G. Sindre, "Towards a deeper understanding of quality in requirements engineering," in *Advanced Information Systems Engineering*. Springer, 1995, pp. 82–95.

[11] K. Pohl, "The three dimensions of requirements engineering: a framework and its applications," *Information systems*, vol. 19, no. 3, pp. 243–258, 1994.

[12] H. Femmer, J. Mund, and D. Méndez Fernández, "Its the Activities, Stupid! A New Perspective on RE Quality," in *Proc. of the 37th International Conference on Software Engineering (ICSE'15)*, 2015.

[13] O. Liskin, "How artifacts support and impede requirements communication," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2015, pp. 132–147.

[14] T. C. Lethbridge, J. Singer, and A. Forward, "How software engineers use documentation: The state of the practice," *Software, IEEE*, vol. 20, no. 6, pp. 35–39, 2003.

[15] U. Abelein and B. Paech, "State of practice of user-developer communication in large-scale it projects," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2014, pp. 95–111.

[16] E. Bjarnason, K. Wnuk, and B. Regnell, "Requirements are slipping through the gaps?a case study on causes & effects of communication gaps in large-scale software development," in *Requirements Engineering Conference (RE), 2011 19th IEEE International*. IEEE, 2011, pp. 37–46.

[17] G. Kalus and M. Kuhrmann, "Criteria for software process tailoring: a systematic review," in *Proceedings of the 2013 International Conference on Software and System Process*. ACM, 2013, pp. 171–180.

[18] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300, 1995.

[19] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *Software, IEEE*, vol. 11, no. 2, pp. 42–49, 1994.

[20] ISO, IEC, and IEEE, "ISO/IEC/IEEE 29148:2011-Systems and software engineering - Life cycle processes - Requirements engineering," ISO IEEE IEC, Tech. Rep., 2011.

[21] H. Femmer, D. Méndez Fernández, E. Juergens, M. Klose, I. Zimmer, and J. Zimmer, "Rapid Requirements Checks with Requirements Smells: Two Case Studies," in *Proc. of the 36th International Conference on Software Engineering (ICSE'14)*, 2014.

[22] M. Jackson, *Problem frames: analysing and structuring software development problems*. Addison-Wesley, 2001.

[23] T. Gorschek and A. M. Davis, "Requirements engineering: In search of the dependent variables," *Information and Software Technology*, vol. 50, no. 1, pp. 67–75, 2008.

[24] B. Bernárdez, A. Durán, and M. Genero, "Empirical evaluation and review of a metrics-based approach for use case verification," *Journal of Research and Practice in Information Technology*, vol. 36, no. 4, pp. 247–258, 2004.

[25] H. Femmer, J. Kucera, and A. Vetrò, "On The Impact of Passive Voice Requirements on Domain Modelling," in *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2014.


[6] or establish means to improve, in terms of constructive quality assurance