Improving the Writing Quality of User Stories: A Canonical Action Research Study

Sabine Molenaar and Fabiano Dalpiaz

Dept. of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
{s.molenaar,f.dalpiaz}@uu.nl

Abstract. [Context] User Stories (USs) are a popular notation for writing requirements in Agile software development. USs are often stored in Issue Tracking Systems (ITSs) and are a starting point for defining software development tasks. [Problem] While writing high-quality requirements statements is a typical concern when authoring requirements specification documents, this is less the case when writing USs in an ITS. This may also be the attributed to the fact that practitioners are not familiar with techniques for improving the quality of their USs. [Method] As part of previous research in a large organization, we found that practitioners were eager to learn how to write better USs and asked four Agile teams to participate in a study aimed at improving that practice. We conducted canonical action research where these teams were offered a lightweight intervention in the form of guidelines for writing USs—based on the Quality User Story (QUS) framework—, which they could use to reflect upon the quality of their USs. [Findings] The share of atomic and minimal violations decreased through the use of the intervention and, for the former, the positive effects lasted even after the intervention period ended. However, practitioners did not agree with all the guidelines and argued that violating the criteria can sometimes benefit them in terms of clarity and time spent. These results call for better contextualization of research on user story quality, which we initiate by proposing revised formulations of our guidelines.

Keywords: Requirements Engineering · User stories · Quality User Story framework · Canonical Action Research.

1 Introduction

Requirements in Agile software development (ASD) are defined incrementally and iteratively [13], often through the formulation of User Stories (USs), which express a requirement in a compact manner using a simple template such as the Connextra format [3]: "*As a [role], I want to [action], so that [benefit].*".

In Scrum, the most popular ASD method [23], teams rely largely on information available in their Scrum boards, often stored in issue tracking systems, and do not routinely speak to the user [11]. This increases the importance of

the quality of USs, as they are a frequently used Agile requirements engineering (RE) practice [11], and are often the only source of information for team members.

Poor requirements can lead to software errors that require rework [5]. This remains a relevant topic, as maintaining requirements quality is considered a challenge in large-scale agile system development. The same is true for time-to-market; teams want to deliver quickly, but still need to achieve requirements with sufficient quality [15].

Several frameworks have been proposed to assess the quality of USs, such as INVEST [24] and Quality User Story (QUS) [16], but their use and effects are rarely tested in practice. In a broader sense, researchers have argued that more empirical studies are needed on the effects of Agile RE and the application of Agile RE practices [14, 4].

In a previous study [20], we analyzed the US quality of eight Agile teams by evaluating them on four QUS framework criteria [16]. We found that all four criteria were violated to various degrees. The participants expressed that they were unfamiliar with some of the criteria, but were eager to learn and improve their USs. We invited half of these teams again to participate in this canonical action research (CAR) [6] study to assess whether the quality of their USs could be improved regarding these criteria. In addition, we measured information retention by repeating the analysis after the use of the intervention.

All four teams were given guidelines to use to write their USs for six, two-week sprints. After the period, we interviewed the participating Product Owners (POs) and Scrum Masters (SMs), to get a qualitative perspective on the guidelines' usefulness. In addition, we assessed the written USs on the same four QUS framework criteria and compared the violations to those of the previous study.

We found that the number of violations can be reduced through the use of our lightweight intervention. In addition, the participants stated they would recommend the use of the guidelines to other Agile teams within the organization. However, they disagreed with some guidelines, saying they would make their processes more complex and less time-efficient. Based on their feedback, we propose a reformulation of the guidelines to make them more suitable for use in a real-world Agile development setting, by increasing their pragmatism.

The remainder of this paper is structured as follows. We discuss related literature in Section 2. The methods used are described in Section 3, followed by the results in Section 4. Finally, we present a discussion in Section 5 and provide conclusions in Section 6.

2 Related work

We discuss relevant background regarding the QUS framework, as well as studies which evaluate the use of agile requirements in industry settings.

Heck & Zaidman categorized quality criteria for agile requirements (specifications) into three main groups: completeness, uniformity, and consistency & correctness [10]. The latter category focuses on the correctness of individual

requirements and their consistency with others, and an example of this is the INVEST mnemonic [24]. INVEST suggests that USs should be Independent, Negotiable, Valuable, Estimable, Small, and Testable.

Lucassen *et al.* created the QUS framework as a response to limited methods and frameworks for determining and improving the quality of USs. At the time of their study, only INVEST was available. The QUS framework consists of thirteen criteria, which apply to either individual USs or a set of USs [16]. While the authors did experiment with US quality assessment in an industry setting, this required a training session and the use of the AQUSA tool [17], which raises the threshold for participation. Moreover, the training needs to be repeated if new members join the team. Whether the knowledge gained by the practitioners was retained by them after a longer period of time was out of scope. A longitudinal cohort study by Fucci *et al.* showed that it is possible for participants to retain information learned over a period of several months, although they specifically studied Test-Driven Development [9].

Through a survey, Wang *et al.* found that requirements analysts in agile settings discuss requirements once or twice a week with their customers, to confirm new requirements for each sprint or to capture changes to existing requirements [25]. Most respondents used a two-week iteration or sprint, suggesting that requirements need to be captured and documented quickly and often. In addition, high workloads, requirement refinement, creating and estimating USs and requirements ambiguity were among the main challenges in large-scale agile transformation, according to a systematic literature review by Dikert *et al.* [8]. This emphasizes the need for supporting tools, but also shows that these should not be time-consuming to avoid further increasing the workload. This is supported by Kasauli *et al.*, who found that quality and time-to-market are a trade-off large-scale agile systems development often struggles with [15].

Previous studies also evaluated the creation and use of agile requirements through empirical methods. Writing requirements, for example, was investigated by evaluating whether potential POs, people with limited to no experience with writing requirements but who are familiar with the context, are able to write USs [21]. They provided participants with Cohn's US template [3], as well as an example US. They evaluated the output and found that, in general, the participants adhered to the template. Berends & Dalpiaz focused on refinement of USs [2]. Through example mapping, they had team members discuss how the requirement should function in the software; what is allowed and what is not according to the US in question. Their results show that example mapping has a positive impact on the shared understanding of the team [2].

As for the use of requirements, the Requirements Specification for Developers (RSD) approach was developed to tailor the requirements to developers in order to support them in requirements validation [18]; this approach focuses more on the creation and use of acceptance criteria. Medeiros *et al.* evaluated the approach in practice and report that the RSD approach results in a more objective requirements specification, which was deemed more suitable for devel-

opers [19]. However, in some cases, multiple requirements were included in one RSD artifact, which negatively affected productivity.

A 2021 study [22] investigated both the creation and use of requirements, by studying how the quality of USs affects the development process, for instance through rework or delays. First, they assessed the quality of 3,414 USs on the well-formed, atomic, minimal, unique and uniform criteria from the QUS framework. The results were expressed as a quality score, which they tested for a correlation with the number of associated bugs, the number of times rework was done and the number of delays. Their results show that lower quality USs correlate with more bugs, increased work, and delays, while high quality USs are less likely to suffer these development problems [22].

3 Research method

We conducted a Canonical Action Research [6] study to investigate whether US quality can be improved via a not-too-intrusive intervention. Davison defines five main principles for CAR: (i) researcher-client agreement, (ii) cyclical process model, (iii) theory, (iv) change through action, and (v) learning through reflection [6]. In this section, we specify which decisions contribute to which principle, by including them in parentheses (i.e., CAR-i). We selected CAR because the participants in our previous study [20] expressed eagerness to learn and improve (CAR-i/iv). Through CAR, we aim to contribute to the research-industry balance, by making this study relevant to practitioners [12]. We intend to answer the following main research question (MRQ): *How does supporting Agile teams with lightweight guidelines affect the quality of user stories?*

We aimed for a lightweight intervention because (Agile) development teams often have a high workload and therefore cannot always afford to spend time on (additional) training (CAR-iv). Moreover, when a member leaves the team, their knowledge and experience is lost, making training less valuable in the long-term. This is another argument in favor of CAR; only end-users can evaluate whether use of the intervention is viable in the long-term (CAR-iii). Wohlin [26] states that integrating a study into the daily work of the industry is key to the interest and commitment from the industry. We describe the intervention in Section 3.2.

We interviewed the practitioners responsible for writing USs for their team to evaluate the intervention and gain insight into their experience (CAR-v). US quality is measured by assessing each US on the four quality criteria at the basis of the intervention: well-formed, atomic, minimal and full sentence. This selection of criteria is in agreement with our previous study and allowed us to compare results; the number of violations of each criterion (CAR-i). We combine qualitative and quantitative methods; the qualitative perspective helps explain patterns found in quantitative data [26].

3.1 Research design

Figure 1 illustrates our research design. First, we provided the participating

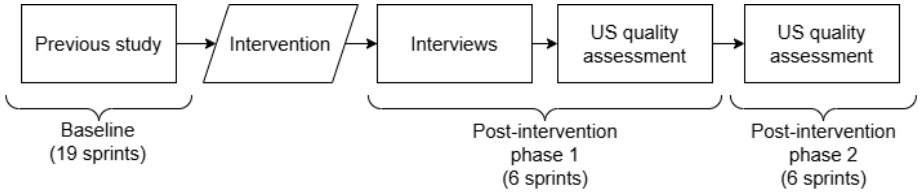

Fig. 1. Research design showing data gathering process.

teams with the intervention, the guidelines to use, in the form of a digital one-pager. The practitioners discussed the guidelines at the beginning of the first sprint, and used them to write and refine USs for six sprints, the first post-intervention phase (PI-1), during this time, the researchers and teams did not interact; this also means that we did not enforce the guidelines. After the six sprints were finished, we interviewed the POs and SMs of the participating teams to learn about their experience using the guidelines (CAR-ii). We then assessed the quality of the USs and compared the number of violations from before and after using the guidelines, in absolute numbers and as a share of the USs included. Inspired by Fucci *et al.* [9], we also assessed the quality of the USs created in the six sprints after that, which we call the second post-intervention phase (PI-2), to see whether any information learned was retained (CAR-iv).

3.2 Intervention: Guidelines for US writing

We formulated fourteen guidelines for the teams to follow. Each guideline was proposed by the first author and reviewed by the second. We distinguish between practices to follow, "do's", and practices to avoid, "don'ts", when writing a single US. Both Dutch and English versions of the guidelines were made available. Each guideline corresponds to a quality criterion. We summarize the guidelines and the criteria to which they are related in Table 1. The do's are indicated by a *P* for positive in the ID, while don'ts are indicated by an *N* for negative.

All guidelines were (i) created using a deductive approach; informed by quality issues encountered in our earlier study [20], in which we objectively assessed each criterion (CAR-ii), and (ii) based on the QUS criteria. We intentionally used theory to inform our guidelines, since we assumed that POs and SMs write US to the best of their ability already and we wished to bring theory and practice together.

The guidelines presented in Table 1 address both 'broad' (violating a criterion) and 'narrow' (improving an element of the US) quality issues. The former are indicated by IDs in **bold**. A US that contains a role and an action does not violate the well-formed criterion, but also does not mean that it contained a 'good' role. For example: "*As [the organization], I want to sort tasks alphabetically.*" While this US is well-formed according to QUS, the role could be improved by mentioning a specific stakeholder instead, we call this a narrow quality issue. Such issues were included in the qualitative findings of our previous study.

Table 1. Guidelines presented to participating teams, organized as positive do's (Px) and negative don'ts (Ny), including the corresponding quality criterion.

ID	Description	QUS criterion
P1	Use a specific template	Well-formed
P2	Specify at least a role and an action	Well-formed
P3	Specify a desire for exactly one functionality	Atomic
P4	Formulate a clear benefit or motivation	Minimal
P5	Use a full, grammatically correct sentence	Full sentence
P6	Write in one language (jargon excepted)	Full sentence
P7	Use a function title as a role	Well-formed
N1	Specify the (technical) solution	Minimal
N2	Use negations in actions	Minimal
N3	Add unnecessary information in brackets or at the end	Minimal
N4	Force tasks (e.g., bug fixes, maintenance) into a user story	Well-formed
N5	Specify non-functional requirements in user stories	Well-formed
N6	Refer to other user stories or documents	Minimal
N7	Use a system or application as a role	Well-formed

While the narrow quality issues are not reflected in the quantitative results, they were discussed in the interviews and part of the qualitative results. Note that the participants were only given the description from Table 1 preceded by "do" or "don't", but were unaware to which criterion a guideline is related. This was done so they could not focus on a single guideline in particular to improve on their violations from the first study.

3.3 Participants

We held our study in a large organization based in the Netherlands; we cannot disclose the identity due to the confidentiality constraints. The organization maintains many applications and over 200 Agile teams are active. The organization and the teams were chosen for convenience, but the researchers were only involved with the teams for the sake of this study (CAR-ii). Six of the teams included in [20] were asked to participate in this study, two of which declined due to high workload. All interviews were held with the POs and/or SMs of the team. A brief description of the teams can be found in Table 2.

Table 2. Demographics of the participating teams.

Team ID	Size in #employees	Type of dev.	Ext. members	Interviewees
T1	8 to 10	Low code	No	2
T2	8 to 10	Full code	Yes	2
T3	12 to 18	Full code	Yes	1
T4	8 to 10	Mix	Yes	3

3.4 Qualitative data gathering

The interviews were semi-structured: interviewees were asked questions, but could share whatever they wanted and ask questions in return. The following questions were asked in every interview:

1. Did you use the guidelines?
2. Were you able to understand the guidelines?
3. What was the most useful guideline?
4. What was the least useful guideline?
5. Was any information new to you?
6. Would you recommend the guidelines to someone else?
7. Is there anything else you would like to share?

We shared no information on the performance of their team with the interviewees before or after the receiving of the guidelines.

3.5 Quantitative data gathering

We performed the quality assessment on the USs of the participating teams again after the interviews were conducted. For each team, we gathered the USs of the six sprints that followed right after providing them with the guidelines, the first post-intervention phase. The participants were aware of which criteria their USs would be assessed on. We also gathered the USs of the six sprints that followed after that, the second post-intervention phase, to see if retainment was present; meaning if fewer violations are present after participation in this study has ended. Unfortunately, Team 3 ceased expressing requirements through USs in the second post-intervention phase, as they switched to task descriptions; therefore, for that phase, no data are available for them.

All USs were assessed on the following, adapted, QUS criteria [16] (CAR-iii):

1. Well-formed: A US includes at least a role and an action.
2. Atomic: A US expresses a requirement for exactly one feature.
3. Minimal: A US contains only role, action and benefit.
4. Full sentence: A US is a well-formed, full sentence.

For each US, we manually assessed whether a criterion was met or not, using the same guidelines as described in our previous study [20]. First, for the well-formed criterion, we checked whether a role and an action were included. Second, for atomic, we focused on the use of words such as “and” and “or”, indicating enumerations. Third, additional text in USs, such as after the period or included in parentheses, resulted in a minimal violation. Fourth, the full sentence criterion was considered violated if the sentence was syntactically incorrect. The manual assessment was performed by the first author, discussing edge cases with the second. In addition to confidentiality reasons, we made this choice because Wouters *et al.* [27] reported high inter-rater reliability for the well-formed and atomic criteria.

4 Results

While the quantitative data were collected last, we present these results first in Section 4.1, and then use the qualitative results in Section 4.2 for triangulation.

4.1 Quantitative results

We compare the quantitative results of three phases of the study:

1. *Base*: the baseline, the quality and violations observed prior to the intervention (obtained in [20]), which consisted of nineteen sprints;
2. *PI-1*: the first post-intervention phase, focusing on the six sprints during with the participants were asked to use the intervention;
3. *PI-2*: the second post-intervention phase, focusing on the six sprints that immediately follow PI-1.

Based on the number of violations and on the classification from our previous work [20], the USs are divided into three groups: those of high quality (no violations), medium quality (one violation), and low quality (two violations). USs with more than two violations were not observed.

In Table 3, the quality score of the USs written by the teams is shown for each phase. We also include the number of USs assessed. For the baseline, which included more sprints, we include a normalized number of USs in parentheses. Note that in the PI phases, the teams worked with the lower range in number of employees described in Table 2, which explains the decrease in number of USs per sprint per team. The quality of USs across the teams improved after the intervention; from 74% high quality USs in the baseline to 79% post-intervention. The second post-intervention phase also shows an overall improvement, but of weaker strength (76%). Only team 2 shows a decrease in quality, while teams 1 and 3 show an increase in high quality USs and team 4 only shows an improvement through a reduction in low quality USs (from 2% to 0%).

Table 3. Number and share of USs per quality score and team. The total number for the Base period, over 19 sprints, is also presented in parentheses after normalization to 6 sprints, allowing direct comparison between Base, PI-1, and PI-2.

	Low			**Medium**			**High**			**n**		
	Base	**PI-1**	**PI-2**	**Base**	**PI-1**	**PI-2**	**Base**	**PI-1**	**PI-2**	**Base**	**PI-1**	**PI-2**
T1	3	0	0	62	5	4	160	33	35	225 (71)	38	39
	1%	0%	0%	28%	13%	10%	71%	87%	90%	-	-	-
T2	0	1	0	15	14	5	106	32	22	121 (38)	47	27
	0%	2%	0%	12%	30%	19%	88%	68%	81%	-	-	-
T3	5	0	-	47	1	-	133	29	-	185 (58)	30	-
	3%	0%	-	25%	3%	-	72%	97%	-	-	-	-
T4	4	0	2	55	10	15	136	22	25	195 (62)	32	42
	2%	0%	5%	28%	31%	36%	70%	69%	60%	-	-	-
Total	12	1	2	179	30	24	535	116	82	726 (229)	147	108
	2%	1%	2%	25%	20%	22%	74%	79%	76%	-	-	-

Table 4 reports the number and share of QUS violations per criteria and team. Again, we also include the number of USs assessed and a normalized number of USs for the baseline. After the intervention, no well-formed violations were

Table 4. Number and share of QUS violations per criteria and team.

	Well-formed			Atomic			Minimal			Full sentence			n		
	Base	PI-1	PI-2	Base	PI-1	PI-2	Base	PI-1	PI-2	Base	PI-1	PI-2	Base	PI-1	PI-2
T1	1	0	0	59	5	4	7	0	0	1	0	0	225 (71)	38	39
	0%	0%	0%	26%	13%	10%	3%	0%	0%	0%	0%	0%	-	-	-
T2	0	0	0	11	10	3	4	4	2	0	2	0	121 (38)	47	27
	0%	0%	0%	9%	21%	11%	3%	9%	7%	0%	4%	0%	-	-	-
T3	2	0	-	37	1	-	16	0	-	2	0	-	185 (58)	30	-
	1%	0%	-	20%	3%	-	9%	0%	-	1%	0%	-	-	-	-
T4	0	0	0	46	5	10	15	4	9	2	1	0	195 (62)	32	42
	0%	0%	0%	24%	16%	24%	8%	13%	21%	1%	3%	0%	-	-	-
Total	3	0	0	153	21	17	42	8	11	5	3	0	726 (229)	147	108
	0%	0%	0%	21%	14%	16%	6%	5%	10%	1%	2%	0%	-	-	-

observed, but these numbers were low in the baseline too. The atomic violations decreased from 21% to 14% in PI-1 and are still lower than the baseline in PI-2 (16%). The minimal violations show a slight improvement in PI-1 (5%) compared to the baseline (6%), but increase to 10% in PI-2. The full sentence violations increased slightly in PI-1 to 2% (was: 1%), but disappeared in PI-2 (0%).

From both perspectives (quality score and violations per type), Team 1 presents a continuous improvement from baseline to PI-1 to PI-2. The same is true for Team 3, but here we can only make observations regarding the difference between the baseline and PI-1. Team 4 shows some improvement, but in some cases performs worse after the intervention. The same can be said for team 2, but in less severe terms.

4.2 Qualitative results

We report the results gathered in the semi-structured interviews. We discuss each interview question (see Section 3.4) and include additional feedback from the interviewees at the end. For each finding, we specify which team representatives support it by listing their IDs. For instance (T1/T4), means that the POs/SMs of teams 1 and 4 mentioned the finding. When relevant, observations from Section 4.1 are included to triangulate the findings.

1. Did you use the guidelines? All four teams stated that they used the guidelines for writing their USs during PI-1. Two teams specified that they kept the guidelines at hand while writing USs, but did not check the USs on the guidelines specifically after writing was finished (T1/T4). One of these teams discussed the guidelines with the entire team beforehand (T4), while the other adapted their template; specifically including the benefit of the US (T1). One team summarized the guidelines, keeping only what they 'needed' (T3).

2. Were you able to understand the guidelines? All four teams understood the guidelines just by reading them and had no need to clarify. Two teams were able to recognize many aspects in the guidelines from their own way of working

(T1/T4) and one specified that they appreciated this as a validation of what they were already doing (T1).

3. What was the most useful guideline? *P3* is considered one of the most useful guidelines by Teams 3 and 4. Specifying a need for only one functionality is still a challenge for Team 3, but they noticed that they split USs [7] into smaller parts more often than before. This is also reflected in the quantitative results: Team 3 went from 20% atomic violations to 3% and Team 4 from 24% to 16% (baseline to PI-1). Team 1 also shows an improvement here, from 26% in the baseline to 10% in PI-2.

Three teams (T1/T2/T4) tried to refrain from including unnecessary information (*N3*), for instance by evaluating the US after it was written to check if all information included was truly necessary, if not, it was left out (T1). Team 1 was successful in reducing the share of minimal violations (from 3% to 0%), as was team 3 (from 9% to 0%). Both Teams 2 and 4 had an increase percentage-wise, from 3% to 9% and 8% to 13%, respectively. Two teams (T2/T4) mentioned the avoidance of negations (*N2*). No requirements by negation were observed in the PI-1 and PI-2 phases for Teams 2 and 4.

Team 1 tried to avoid specifying a technical solution (*N1*), because they discuss solutions among the team and may have different opinions on what the solutions should be. To keep all team members in the know, they include implementation hints in the documents and try to keep USs as functional and problem-oriented as possible. This is especially challenging when the stakeholder requesting the feature already includes solutions in their request (T1).

Team 3 mostly prefers the 'positive' guidelines and mentioned *P7* as an important one. At first, they used to include themselves or the system as the role, since they wrote the USs for improvement of that system. Now they choose someone from the business to include in the role and this person is asked to accept the US too (T3). After the intervention, the system was used as a role only once. They also try to write in one language now (*P6*). No USs were written using more than one language after the intervention (jargon excepted). *N4* was mentioned by only one team (T2). After the intervention, all teams still recorded maintenance tasks in a US template, such as: "*I want to update [system] to [version]*".

4. What was the least useful guideline? Teams 2, 3 and 4 considered *N1* the least useful. Technical solutions are included in the US to ensure all team members are on the same page (T3/T4). They had to start this practice, because in the past some USs were so poorly written that the solution did not meet the requirement (T4). Working with third-party team members is also mentioned as a reason, since they sometimes have to work asynchronously. So, in order to work more efficiently, they include the solution rather than having to schedule another meeting; every feedback cycle and meeting costs time (T4). Team 3 explained it is "pointless" to write 'good' USs (meaning without violations), if you need to have a bunch of conversations about them to make sure everyone understands what they mean (T3). Another motivation for including solutions is the lack of

experience of team members (T2). All teams continued to specify (technical) solutions after the intervention. Examples are specifying which modules and APIs to use or specific fields to filter on, e.g.: "*As a [role], I want [module] to use [API] to support [event]*" and "*As a [role], I want [object ID] to be available on [specific page]*".

Including specific stakeholders as a role can be difficult (*P7*/*N7*), when the team mostly focuses on the back-end of a system (T4) or when the team recognizes it is something that "just needs to be done" (T2). In some cases, they need to search for a user and at that point, they include the system, because it is easier for the team members to understand the need that way (T4).

Bugs are often issues encountered by the business, so in order to relate their planning to their stakeholders' needs, teams write bug fixes (*N4*) in US format (T4). If possible, they would like to see a different template for different topics, such as a bug fix and performance template (T4). Team 2 somewhat agrees, describing that they understand that bug fixes do not belong in USs, but they think maintenance tasks do belong, since they are part of the US lifecycle management (T2). Non-functional requirements (NFRs) (*N5*) are included in USs if team members lack experience and are at risk of not considering them while fulfilling the USs or simply because NFRs can constrain USs (T2). T3 takes a more practical approach: if not in the USs, where do you document maintenance tasks and NFRs? They consider these artifacts necessary building blocks for a US. A solution for them could be an NFR template (T3). Teams 1, 3 and 4 included NFRs in US templates after the intervention, for instance: "*As a [role], I want the status of [module] to be clearer, so that it is easier to interpret.*"

Not referring to other USs (*N6*) seems counterintuitive to the teams, as this can often save them time (T1/T2). Other USs can sometimes include information they need for their own work or their USs are dependent on those of other teams (T1/T2). In some cases, there is a 'big' US that various teams divide into USs they can work on, but most information, such as objectives, are included in the 'big' US (T1). Both teams still employ this practice after the intervention.

5. Was any information new to you? Two teams (T3/T4) were unaware that NFRs should not be formulated as USs. Team 4 asked when non-functional becomes functional and how to address this. Team 3 did not know technical solutions should not be included.

6. Would you recommend the guidelines to anyone else? All four teams stated they would recommend the guidelines to others, and one team had already shared them with a team not included in this study (T2). Two teams explained that a standard across the organization would be beneficial, since they sometimes need to collaborate or are dependent on other teams; "*it would be nice to work with USs that you did not write, but that are still well written*" (T1/T3). Notably, both of these teams showed improvement in all four criteria, while the other two did not.

7. Additional feedback Team 1 considers the *do's* more useful for beginners and *don'ts* more useful if you already have experience with writing USs. They

can help to identify and change 'bad habits' (T1). Additional guidelines were also requested, for instance for test cases (T4) and acceptance criteria (T3/T4). How can you describe acceptance criteria well and make them 'SMART' (T3)? Team 4 thinks LLMs can save time and effort in requirements refinement, for instance by using these guidelines as restrictions for prompts and assessing the USs on the QUS criteria.

5 Discussion

We discuss the threats to validity using the five quality criteria for CAR as recommended by SIGSOFT: reflexivity, credibility, resonance, usefulness and transferability [1].

Reflexivity The four included criteria were evaluated by one researcher, discussing unclear cases with a second researcher. However, guidelines and instructions were created beforehand and applied to all USs and the selected criteria are mostly objective. Previous work has shown that well-formed and atomic violations can be reliably assessed [27] and the researchers have been familiar with QUS since its publication, with the second author being a co-author of QUS.

Credibility The guidelines served a particular goal, so it is not unreasonable to assume the participating teams predicted what they would be assessed on in this study. To get better results, they may have put in extra effort beyond what they would do in a non-study related setting or used external sources and support. However, this seems unlikely, as the interviewees did not mention using anything but their experience and the guidelines. Furthermore, they were in no way incentivized to perform better, other than their intrinsic motivation, since results were anonymized and there were no rewards. In addition, we also analyzed lasting effects by assessing their USs again, after the intervention phase ended. We triangulated results by using both quantitative and qualitative findings.

Teams may have improved over time regardless of the intervention. POs and SMs might have paid more attention to US quality, since they were made aware of it, but not necessarily due to applying the guidelines. In an attempt to mitigate this threat, we informed the participating teams that certain errors were made by teams within the organization, but did not tell them which teams made which errors. Therefore, they were unable to focus on specific quality criteria. We also held no authority over any of the participating teams and we did not enforce any of the guidelines throughout the study. The participants were also not informed of which guideline related to which criterion, so they were unable to target a specific 'weakness' in their work. In addition, by sheer chance it is easier for teams to improve on quality criteria with many violations in the baseline assessment. Changes among the team members were also not taken into account, however, the POs and SMs remained the same throughout the study.

CAR often prescribes multiple process cycles and while a single cycle is not unsound, it is rare [6]. We decided to perform one cycle for two reasons. First, it would have been difficult to mitigate the maturity effect; the USs contained

fewer violations after the intervention phase, in most cases. Second, participants explicitly stated they did not agree with some of the guidelines, so they will not be using them in the future. Arguments include that they think it takes too much time or makes their process more complex than needed. It would be unethical to 'force' them to continue using guidelines they do not perceive as beneficial. Especially when the intervention is aimed at supporting these industry participants (CAR-ii).

Resonance In order to obtain a genuine account of their experience, we did not share the quantitative results with the teams in the post-intervention interviews. Nevertheless, our revised guidelines (see next section) are based on their feedback.

Usefulness In our future research directions (Section 6.2), we provide recommendations to both researchers and practitioners based on our findings.

Transferability The participating teams were selected through convenience sampling; only teams that participated in our previous case study were asked to participate in this study. While we cannot be sure whether these results are generalizable to other teams and organizations, the participating teams used popular methods (Scrum) and requirements practices (USs).

6 Conclusion

We draw conclusions on the effectiveness of the intervention per QUS criterion assessed in this study, combining qualitative and quantitative findings. We then present a reformulation of three guidelines based on the feedback provided by the participants, general conclusions, and end with directions for the future.

Well-formed One team specifically mentioned focusing on using function titles as roles (*P7*), rather than the system or application they work on (*N7*). After the intervention, the system was used as a role only once, while before this was the rule and using function titles was the exception. The number of violations decreased after the intervention; however, there were few well-formed violations to begin with.

Atomic Two teams considered "*specify a desire for exactly one functionality*" (*P3*) one of the most useful guidelines and both these teams reduced their number of atomic violations during the use of the intervention. In general, it seems that the intervention was effective in reducing these violations, as the share of atomic violations decreased by seven percent points during the intervention and by five percent points after the intervention.

Minimal Three teams found not including unnecessary information (*N3*) one of the most useful guidelines, but only one of them was successful in reducing their minimal violations. The number of minimal violations dropped by one percent point during the use of the intervention, but increased after. A possible

explanation is that the participants disagreed with the guidelines that stated not to specify technical solutions and refer to other USs or documents. During the interviews, the interviewees said that the former ensures all the team members are on the same page, especially when working with external team members that may work asynchronously. The latter is counterintuitive for them, since referring to other USs makes dependencies explicit. In other words, they may have a different view of what is 'unnecessary'.

Full sentence This criterion was rarely violated, but the share of full sentence violations increased during the use of the intervention and decreased afterward. It is unsure how effective the guidelines were in this case. During the interviews, one team stated that they worked on writing in one language (*P6*) and were successful in this endeavor.

Expressing maintenance tasks, (technical) solutions and non-functional requirements in USs are points of contention between research and industry. First, the practitioners argue that maintenance tasks are part of the product's lifecycle and should therefore be included in fn USs, as they express functional requirements. Second, including (technical) solutions is meant to save time, as this ensures all team members know what to do at any time. Third, non-functional requirements are mentioned in USs, because they should be considered while fulfilling other USs (e.g., they can constrain other (functional) requirements) and because practitioners are unsure how else they can be specified.

All in all, the participating teams would recommend using the guidelines to other teams. They would also benefit from this, as USs they are dependent on are sometimes of poor quality and difficult to work with.

6.1 Reformulation of guidelines

The CAR principles [6] recommend reflecting on the results and taking these into account for continuation of the project at hand. Therefore, we propose a reformulation of three guidelines. In summary, participants mainly provided compelling arguments against the formulation of guidelines N1, N4 and N6. Based on their feedback, we have reformulated these guidelines as presented in Table 1 below; to make them less restrictive and more pragmatic (CAR-v). The changes are italicized (note that the guidelines are still concerned with the writing of a single US):

- N1: Specify the (technical) solution *in the user story template*;
- N4: Force tasks (e.g. bug fixes, *administrative work*) in user stories;
- N6: Refer to other user stories*, unless including their IDs,* or documents.

6.2 Future directions

The participants stated that they believe other teams could benefit from using the intervention, therefore an obvious next step would be to share the guidelines

with other Agile teams within the organization as well. However, since participants also criticized some of the guidelines, the intervention could benefit from a second iteration; making improvements based on empirical findings. We have made a first proposal by reformulating the contentious guidelines in Section 6.1. During the interviews, participants also stated they would appreciate support regarding the definition of NFRs (possibly through a template), how to document maintenance tasks, as well as guidelines for writing high quality acceptance criteria and test cases.

On more than one occasion, the participants explicitly disagreed with a guideline and did not consider using it beneficial. These guidelines, however, are all based on quality criteria from the QUS framework, which has remained largely unchallenged by industry due to its limited testing in practice. Our study shows that situational guidelines are needed, as QUS (and other frameworks) are not one-size-fits-all solutions. CAR and other empirical studies are important for identifying the needs for such more specific approaches for evolving frameworks like QUS. In short, research would benefit from evaluating new tools, methods and applications with their envisioned end-users in a real-world setting, to ensure they are suitable for use by practitioners.

References

1. ACM SIGSOFT: Empirical standards for action research (accessed 10-06-2025), `https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=ActionResearch#`
2. Berends, J., Dalpiaz, F.: Refining user stories via example mapping: an empirical investigation. In: 2021 IEEE 29th International Requirements Engineering Conference (RE). pp. 345–355. IEEE (2021)
3. Cohn, M.: User stories applied: For agile software development. Addison-Wesley Professional (2004)
4. Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.: Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software **139**, 32–50 (2018)
5. Davis, A.M.: Software requirements: objects, functions, and states. Prentice-Hall, Inc. (1993)
6. Davison, R., Martinsons, M.G., Kock, N.: Principles of canonical action research. Information systems journal **14**(1), 65–86 (2004)
7. Dellsén, E., Westgårdh, K., Horkoff, J.: Invest in splitting: User story splitting within the software industry. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 115–130. Springer (2022)
8. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software **119**, 87–108 (2016)
9. Fucci, D., Romano, S., Baldassarre, M.T., Caivano, D., Scanniello, G., Turhan, B., Juristo, N.: A longitudinal cohort study on the retainment of test-driven development. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–10 (2018)
10. Heck, P., Zaidman, A.: A systematic literature review on quality criteria for agile requirements specifications. Software Quality Journal **26**, 127–160 (2018)

11. Hess, A., Diebold, P., Seyff, N.: Understanding information needs of agile teams to improve requirements communication. Journal of Industrial Information Integration **14**, 3–15 (2019)
12. Hoda, R., Salleh, N., Grundy, J., Tee, H.M.: Systematic literature reviews in agile software development: A tertiary study. Information and software technology **85**, 60–70 (2017)
13. Hruschka, P., Lauenroth, K., Meuten, M., Rogers, G., Gärtner, S., Steffe, H.J.: RE@Agile Handbook. Handbook, International Requirements Engineering Board (May 2024)
14. Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. Computers in human behavior **51**, 915–929 (2015)
15. Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., de Oliveira Neto, F.G.: Requirements engineering challenges and practices in large-scale agile system development. Journal of Systems and Software **172**, 110851 (2021)
16. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S.: Improving agile requirements: the quality user story framework and tool. Requirements engineering **21**, 383–403 (2016)
17. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S.: Improving user story practice with the grimm method: A multiple case study in the software industry. pp. 235–252. Springer (2017)
18. Medeiros, J., Vasconcelos, A., Goulão, M., Silva, C., Araújo, J.: An approach based on design practices to specify requirements in agile projects. In: Proceedings of the Symposium on Applied Computing. pp. 1114–1121 (2017)
19. Medeiros, J., Vasconcelos, A., Silva, C., Goulão, M.: Requirements specification for developers in agile projects: Evaluation by two industrial case studies. Information and Software Technology **117** (2020)
20. Molenaar, S., Dalpiaz, F.: The impact of requirements artifacts on efficiency in agile development: A case study. In: Accepted for publication at the IEEE International Requirements Engineering conference (2025)
21. Rocha Silva, T., Winckler, M., Bach, C.: Evaluating the usage of predefined interactive behaviors for writing user stories: an empirical study with potential product owners. Cognition, Technology & Work **22**(3), 437–457 (2020)
22. Scott, E., Tõemets, T., Pfahl, D.: An empirical study of user story quality and its impact on open source project performance. In: International Conference on Software Quality. pp. 119–138. Springer (2021)
23. Verwijs, C., Russo, D.: A theory of scrum team effectiveness. ACM Transactions on Software Engineering and Methodology **32**(3), 1–51 (2023)
24. Wake, B.: INVEST in Good Stories, and SMART Tasks. `http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/`, accessed 12-03-2025 (2003)
25. Wang, X., Zhao, L., Wang, Y., Sun, J.: The role of requirements engineering practices in agile development: an empirical study. In: Requirements Engineering: First Asia Pacific Requirements Engineering Symposium, APRES 2014, Auckland, New Zealand, April 28-29, 2014. Proceedings. pp. 195–209. Springer (2014)
26. Wohlin, C.: Empirical software engineering research with industry: Top 10 challenges. In: 2013 1st international workshop on conducting empirical studies in industry (CESI). pp. 43–46. IEEE (2013)
27. Wouters, J., Menkveld, A., Brinkkemper, S., Dalpiaz, F.: Crowd-based requirements elicitation via pull feedback: method and case studies. Requirements Engineering **27**(4), 429–455 (2022)