

RESEARCH ARTICLE

Quality Requirement Documentation Guidelines for Agile Software Development

WOUBSHET BEHUTIYE¹, PILAR RODRÍGUEZ², AND MARKKU OIVO¹¹M3S Software Engineering Research Unit, University of Oulu, 90570 Oulu, Finland²Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Woubshet Behutiye (woubshet.behutiye@oulu.fi)

This work was supported in part by the Q-Rapids Project, European Union's Horizon 2020 Research and Innovation Funded Program; and in part by the Nokia Foundation Scholarship.

ABSTRACT In agile software development (ASD), where minimal documentation and rapid delivery are the focus, Quality requirements (QRs) are often underspecified and not documented. Guidelines supporting QR documentation task are scarce. The study developed the Agile QR-Doc QR documentation guidelines, which aim to support QR documentation in ASD. We applied a design science research methodology (DSRM) to build the Agile QR-Doc. We used a survey questionnaire and open discussion with ten software practitioners, from two ASD companies to validate the Agile QR-Doc. The practitioners evaluated the guidelines in terms of usefulness, relevance, understandability, and coverage of important aspects for supporting QR documentation and their impact on the agility of the software development process. Agile QR-Doc list 12 recommendations that are grouped into two categories. The first category introduces three recommendations that focus on raising awareness about the significance of QRs, their documentation and related challenges. The second category lists nine recommendations that introduce artifacts, practices and important aspects for documenting QRs. The validation reveals the relevance, understandability and usefulness of the guidelines to support QR documentation in ASD. It also indicates that the guidelines consider important aspects for documenting QRs and that they do not negatively affect the agility of the software development process. Practitioners can utilize the practices, artifacts and knowledge from the guidelines to support QR documentation in ASD. Researchers can benefit from the knowledge on QR documentation in ASD, and application of DSRM in building artifacts.

INDEX TERMS Agile software development, design science, documentation, quality requirement.

I. INTRODUCTION

Quality requirements (QRs), also known as nonfunctional requirements, explain the desired quality properties of a system or software product such as reliability, security and maintainability [1]. QRs are of great importance in software projects [2]. ASD teams often prioritize functional requirements (FRs) over QRs [3], [4]. This leads to QRs not being documented early enough and improperly managed [5]. At times, practitioners depend on tacit knowledge and do

The associate editor coordinating the review of this manuscript and approving it for publication was Mahmoud Elish¹.

not document QRs, leading to increased maintenance efforts when handling QR-related issues [6]. Despite all of these, practitioners identify QRs as important in ASD [7], [8]. Indeed, our recent study [9] revealed that ASD practitioners acknowledge the importance of documenting QRs. Practitioners perceive that documenting QRs helps ensure software quality, clarify QR tasks, and helps in decision-making during software development [9].

In ASD, there are limitations regarding the documentation of QRs [10]–[14]. For instance, informal QR documentation practices, such as relying on tacit knowledge, the infrastructure, other teams and organization for documentation of QRs,

and under-specification of QRs, are problematic [2], [15]. These practices are problematic since they can lead to outdated QR documents, system quality deterioration, scalability challenges, increased maintenance costs and unnecessary reworks [5], [10], [16]. ASD artifacts, such as user stories, are also seen as insufficient for specifying and documenting QRs [11], [17]. This is due to their focus on describing functionality or what a system should do, rather than describing QRs [18]. There is also a lack of clarity regarding the right level of documentation in ASD. For instance, developers misinterpret minimal documentation as no need for documentation [19]. The difficulty in defining and measuring QRs also worsens the challenge of documenting them in ASD [5], [20]. To mitigate the aforementioned challenges and limitations, we believe that QR documentation requires a systematic approach. In this regard, guidelines are beneficial since they can help ASD teams to systematically approach QR documentation. However, such guidelines are scarce [10].

Recently, we conducted a systematic mapping study (SMS) to synthesize the state of the art on management of QRs in ASD (i.e., including elicitation, prioritization, documentation, testing and handling of QRs), by focusing on the strategies and challenges of management of QRs [10]. We found that there are only a few guidelines supporting QR documentation in ASD [13], [15], [22], [24], [26]. Additionally, these guidelines tend to address only specific types of QRs, such as usability or security [15], [22], [24], [26]. Based on our findings from the SMS and to the best of our knowledge, only one guidelines proposal focused on the documentation of QRs in general, in ASD [13]. We also noticed that some of the existing guidelines overlook a wider perspective of the causes of underspecifying and not documenting QRs in ASD and lack validations [13], [15], [26].

We believe that guidelines that take into account a relatively wider perspective of the causes of the lack of documentation, motivations for documenting QRs, and consider the perspectives of different stakeholders in ASD, can better support the documentation and management of QRs. Such guidelines need to be light-weight, empirically built and evaluated to have the potential of supporting practitioners in their QR documentation tasks in ASD. It is necessary that such guidelines do not negatively affect the agility, e.g., the speed and simplicity, of the software development process.

Considering the significance of QRs in software quality management, their economic implications (e.g., their impact on software quality and maintenance costs) [27], informal QR documentation practices and inadequate artifacts which raise the need for systematic approach of QR documentation, and the need for guidelines for documenting QRs in ASD [10], [11], [28], we developed the Agile QR-Doc guidelines. The Agile QR-Doc is developed under the Q-Rapids project,¹ which was a European Union Horizon 2020 project made up of industrial and academia consortium. The project was

aimed at deriving a quality aware rapid software development framework, with a focus on improving software quality levels, shortening the time-to-market of software and increasing productivity by providing a tool support for quality aware rapid software development. The companies in the Q-Rapids consortium, provided a relevant and useful industrial context to explore the documentation and management of QRs in ASD and build our guidelines.

We applied a design science research methodology (DSRM) to develop the Agile QR-Doc guidelines [29]. In order to develop the guidelines, we explored the state of the practice of QR documentation in ASD, by conducting multiple case studies in companies applying ASD, within the Q-Rapids project consortium [8], [13], [30]. We also synthesized the state of the art of management of QRs in ASD through a SMS [10]. By using the findings of these studies ([8], [10], [13], [30]), we analyzed the knowledge on existing QR documentation and management challenges and practices in ASD to build the recommendations of our guidelines. We validated the guidelines by evaluating them with ten software practitioners from two ASD companies of the Q-Rapids consortium, through survey questionnaires and open discussion. These practitioners have relevant expertise on the topic and included roles such as specification engineers, software project managers, and software architects. Agility was given attention during the development and validation.

The Agile QR-Doc guidelines have 12 recommendations which are organized into two categories. The first category focus on raising awareness on the significance of QRs, their documentation and related QR documentation challenges. The second category focus on introducing artifacts and practices, which are aimed at supporting and enhancing QR documentation tasks in ASD. Examples of recommendations in the guidelines include recognizing the significance of QRs, applying lightweight artifacts to document QRs, ensuring the clarity of QRs, and utilizing documentation artifacts built for specific QRs. The guidelines provide the rationale behind each of the recommendations and ways to realize the recommendations along with illustrative examples.

Software practitioners can utilize our work to systematically approach and support QR documentation in ASD. For instance, they can adapt light-weight artifact such as 'Given/When/Then' template, and system stories to document QRs, utilize UserXstories to document usability QR, and document security QRs using abuser stories and misuse cases. Product owners can get insight into actions that they can consider for supporting QR documentation. By using the guidelines, practitioners can spot the indicators of QR documentation challenges and take corrective actions on time. Our work contributes to software engineering research by filling the gap in guidelines that support QR documentation in ASD. Researchers can utilize the knowledge created in building Agile QR-Doc, get a better understanding about the documentation and management of QRs in ASD. They can

¹<https://www.q-rapids.eu/about>

utilize the knowledge to develop solutions supporting QR documentation and management in ASD too.

Section II presents the related work. Section III presents the research methodology followed in the study. Section IV presents the Agile QR-Doc. Section V reports the validation of Agile QR-Doc. Section VI discusses the implications of the study and compares the Agile QR-Doc with related work. Section VII discusses the threats to the validity of the study. Finally, Section VIII concludes the paper.

II. RELATED WORK

This section presents the related work on QR documentation in ASD and the guidelines for documenting QRs in ASD.

A. DOCUMENTATION OF QRs IN ASD

Requirements documentation aims at conveying and sharing requirements needs among development teams and other stakeholders. ASD promotes minimal documentation practices, the rapid delivery of working software and early return on investment to customers [32]. These, combined with its capability to respond to frequently changing requirements, have increased the popularity of ASD, and it is currently well established in the software industry [33]. In ASD, the continuous interaction with the customer is thought to lessen the need for the specification and documentation of QRs [34]. For instance, practitioners may communicate QRs in white board discussions, understand them implicitly, and not document them properly [6], [13], [35].

ASD favor ‘*Working software over comprehensive documentation*’ [36]. However, this value is often misunderstood as no requirement for documentation, leading to informal requirement specifications, which result in problems in the long run [12]. Ambiguity in what minimal documentation means, coupled with the elusive nature of QRs [6], [20] can render the documentation of QRs in ASD more challenging. Consequently, QRs are usually underspecified and undocumented and are handled improperly, resulting in accruing technical debt (TD), increased maintenance costs, and at times, project failures [5].

Requirement engineering artifacts and practices in ASD are often insufficient for specifying and documenting QRs too [10], [11], [17]. For instance, specifying QRs, such as security and safety, in user stories is more difficult than specifying FRs. Documenting such QRs require end-to-end documentation covering many aspects [37]. Although we can clearly specify and mark FRs specified in user stories as done, it is not always easy to specify and mark QRs, such as maintainability and security, in user stories as done [23]. Since these QRs may change repeatedly during development, ensuring that they are specified and managed properly requires continuously revising their documentation [30]. The literature also reveals challenges due to practitioners’ lack of expertise in QRs affect QR documentation in ASD [21].

B. GUIDELINES FOR DOCUMENTING QRs IN ASD

The literature reveals guidelines proposals that support the documentation of QRs in ASD [13], [15], [22], [24], [26].

However, these guidelines, except [13], are focused on supporting the documentation of specific QRs, such as security and usability [22], [24], [26], and do not address the documentation of QRs in general.

Barbosa *et al.* [22] provide a guide for improving security measures in ASD based on an explorative literature review and interview study with practitioners. The guide recommends security backlogs to document security QRs and a corresponding security master who is responsible for defining and documenting security QRs in the security backlog. It also proposes evil user stories to document security requirements and recommends research vulnerabilities activity for the continuous identification and update of vulnerabilities and related security documents. The authors evaluated the guidelines by interviewing four practitioners.

Lee *et al.* [24] provide a usability pattern-based requirements analysis method and guidelines to improve usability specification in ASD. Their guidelines transform user tasks into features of a user interface specifying usability factors. They empirically evaluated the guidelines and methods and identified that these guidelines and methods supported usability requirements specification in ASD. Similarly, Cajander *et al.* [15] provide recommendations for addressing the documentation of usability and user experience in Scrum by interviewing practitioners. The recommendations included assigning clear responsibilities for handling usability and including clear and measurable usability goals in the specification.

There are also guidelines which lack empirical validation [13], [26]. For instance, our previous work [13] provides guidelines for documenting QRs in ASD. The guidelines consider classifying QRs by scope (i.e., system, group and local) and level of detail of the QRs (i.e., generic and detailed) and recommends using artifacts such as epics, user stories, and acceptance criteria. The system-wide scope applies to QRs covering the entire system and the group-wide scope applies to those QRs relating to more than one functionality. The local-scope applies to QRs relating to only one functionality or user story. For instance, a QR with a system-wide scope can be either generic or detailed. If it is generic, the recommendation is to use an epic to document the QR by clearly referring to the system. On the other hand, if it is detailed, the recommendation is to use a user story, clearly describing the functionalities to be covered by the QR. These guidelines lack empirical validation, thus, limiting their adoption to the industry.

Similarly, Wäyrynen *et al.* [26] propose recommendations to improve the documentation and management of security requirements in extreme programming (XP). The recommendations introduce a security engineer who will be responsible for documenting security-related user stories. This security engineer would be involved in pair programming to ensure the implementation of security requirements, documenting security architecture, and adding security review activity. These recommendations lack empirical validation, and this may limit their applicability.

Agile QR-Doc guidelines, which will be introduced shortly, complement existing guidelines merely focusing on specific types of QRs and those that lack empirical validation by industrial practitioners. The gap in QR documentation guidelines, combined with challenges due to informal QR documentation practices (e.g., not strictly defining and documenting QRs, under-specification, outdated and missing documentation of QRs in ASD) [2], [15], which in turn have a negative impact on software projects [10], [16], motivated the need for developing Agile QR-Doc guidelines.

III. RESEARCH METHODOLOGY

We adapted the DSRM by Peffers *et al.* [29] to build the Agile QR-Doc QR documentation guidelines. DSRM is a methodology used for conducting design science research, where artifacts that are aimed at solving a specific problem are developed and evaluated rigorously [29]. According to Peffers *et al.* [29], developing an artifact using DSRM involves the following activities: problem identification and motivation, defining the objectives of a solution, design and development, demonstration and evaluation, and communication. In what follows, we describe the DSRM activities applied to build the Agile QR-Doc set of guidelines.

Activity 1: Problem identification and motivation: the activity outlines a specific research problem and motivates the need for the design solution. QRs have great importance in determining the success of software projects [2]. However, in ASD, they are often neglected, under-specified and not documented resulting in poor quality software and increased maintenance costs [5], [14].

We conducted a preliminary work investigating the management of QRs in the Q-Rapids companies, and identified that ASD companies face QR documentation challenges such as the lack of traceability for QRs, difficulty in identifying dependencies between different QRs, and internally generated QRs being not documented [13]. We also conducted a SMS synthesizing the state of the art on challenges and strategies of managing QRs (i.e., including elicitation, prioritization, documentation, testing and handling of QRs), in ASD [10]. Our findings revealed that there are limitations with existing ASD artifacts in documenting QRs, challenges due to informal QR documentation practices and limited QR knowledge. There was a need for lightweight QR documentation and management strategies. We found that existing guidelines that support QR documentation and management are scarce. We observed that existing guidelines focused only on addressing specific QRs such as security and usability. Some of the guidelines lacked empirical validations ([13], [15], [26]), limiting their adoption in ASD. These findings reveal the need for solutions, e.g., guidelines that support systematic approach of QR documentation in ASD. Therefore, we defined our research problem, a gap in guidelines supporting QR documentation in ASD.

Activity 2: Define the objectives for a solution: this activity defines the objective of the solution or artifact developed to address the research problem. This research builds the Agile

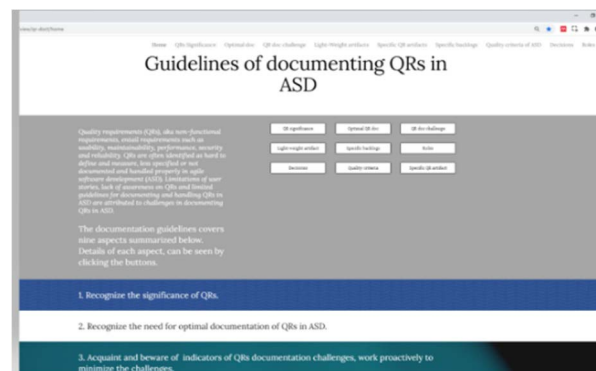


FIGURE 1. Screen capture of agile QR-Doc.

QR-Doc guidelines to solve the gap in guidelines that support QR documentation in ASD. The guidelines aim to support QR documentation in ASD, without compromising the software quality and agility of the process.

Activity 3: Design and development: we used knowledge and experience that we obtained from researching documentation and management of QRs in ASD [8], [10], [13], [30], as inputs to design and develop the Agile QR-Doc guidelines. These knowledge were gained by studying the state of the practice of documentation and management of QRs in ASD, with companies involved in the Q-Rapids consortium through case studies [8], [13], [30] and by synthesizing the state of the art of the management of QRs in ASD through a SMS [10].

We first developed the Agile QR-Doc guidelines by synthesizing nine recommendations. To synthesize these recommendations, we analyzed knowledge on the existing challenges of QR documentation and management in ASD, and the drawbacks of existing QR documentation practices and artifacts in ASD, we found from our studies [8], [10], [13], [30]. Our analysis revealed that QR documentation challenges are associated with the limited QR documentation awareness and associated factors, and the limitations of QR documentation artifacts and practices. Therefore, we built recommendations by focusing on: I) aspects related to raising awareness on QRs, documentation needs and challenges, and II) introducing artifacts and practices to support QR documentation. When building our recommendations, we used knowledge about QR documentation in ASD (e.g., practices, factors affecting QR documentation) that we obtained from our studies [8], [10], [13], [30]. Agility was a focus when building the recommendations. Following the synthesis of the recommendations of the Agile QR-Doc guidelines, we developed a prototype of the guidelines in the form of a website.² Figure 1 shows screen capture of the prototype.

Activity 4: Demonstration and evaluation: in our study, we applied one iteration of demonstration and evaluation. We demonstrated the use of Agile QR-Doc guidelines and evaluated them with ten practitioners of two ASD companies. The evaluation was used to determine how well the guidelines

² <https://sites.google.com/view/qr-doct/home>

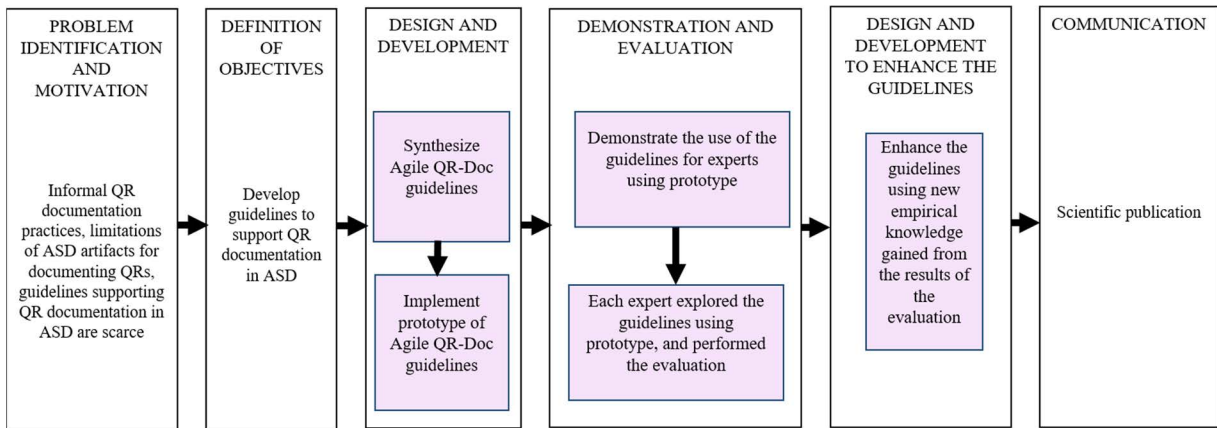


FIGURE 2. Design science research methodology adapted to build the guidelines.

support QR documentation in ASD, and collect feedback to enhance them. It also served to validate the applicability of the guidelines for industry. Informed consents were used in this step. Details regarding the demonstration and evaluation of the guidelines and their findings are discussed in Section V.

Activity 5: Design and development to enhance the guidelines: following the demonstration and evaluation activity, a DSRM research can continue to the communication step. It is also possible that researchers go back and redefine the objectives of the solution. Another possible path is to proceed with the design and development and enhance the artifact based on the findings of the evaluation. In our case, we iterated back to the design and development. We used the knowledge obtained from evaluating the guidelines to improve them and include 3 new recommendations.

Activity 6: Communication: involves reporting the research problem, its significance, the design artifact and its utility, and the design process to the relevant audience. In our case, we report all the aforementioned aspects in this publication. Figure 2 shows the DSRM applied in building Agile QR-Doc.

IV. AGILE QR-DOC: GUIDELINES TO SUPPORT QUALITY REQUIREMENTS DOCUMENTATION IN AGILE SOFTWARE DEVELOPMENT

The Agile QR-Doc QR documentation guidelines provide 12 recommendations aimed at supporting documentation of QRs in ASD. They aim to do so without negatively affecting the software quality and agility of the software development process. The recommendations follow the format: recommendation title, description, justification, ways to realize the recommendation, and related work. The recommendation title identifies the recommendation, and the description explains details of the recommendation. Then, we justify the reason for the recommendation, list the means to implement the recommendation, and finally, present related work regarding the justifications and means of realizing the recommendation. The recommendations are organized into two categories.

These are: A) Recommendations emphasizing awareness on QRs, their documentation and related issues and B) Recommendations that introduce artifacts and practices to support and enhance QR documentation.

A. RECOMMENDATIONS EMPHASIZING AWARENESS ON QRs, THEIR DOCUMENTATION AND RELATED ISSUES

In this category, the need for recognizing the importance of QRs, optimally documenting them and recognizing indicators of QR documentation challenges in ASD, are justified with the goal of supporting QR documentation. Strategies for realizing the recommendations are presented too.

1) RECOGNIZE THE SIGNIFICANCE OF QRs

Description: ASD teams and management need to acknowledge the importance of QRs in ASD and take actions that support raising awareness on QRs and their documentation.

Justification: In cases where ASD teams, customers, and management have limited knowledge regarding QRs, they may not recognize, specify, and handle QRs properly. ASD teams may ignore and not document QRs until there is a specific request for handling them or until the impact of neglecting QRs becomes visible [5], [6], [10], [11], [38], [39].

Ways to realize recommendation:

- a) ASD management and teams (e.g., developers, product owners, and testers) should develop an awareness on QRs and the skills for documenting and managing QRs.
 - Management and teams should invest in training on QRs in general, and on specific QRs such as security, and usability.
- b) ASD teams can utilize a company-specific agile playbook [8], for QRs, which is a lightweight guide describing QR classifications, their significance, and ways to document and manage them.
- c) ASD teams should adopt a ‘quality mindset’ by emphasizing QRs in the different steps of the software lifecycle and ensuring that corresponding QR needs are met.

d) ASD teams can assist customers in recognizing the importance and value of QRs, since customers in some domains may overlook important QRs or may not have adequate knowledge about QRs [40]. For instance, customers in the energy domain may overlook important QRs such as security [41].

Related Work: [5], [6], [8], [10], [30], [38]–[41]

2) RECOGNIZE THE NEED FOR OPTIMAL DOCUMENTATION OF QRs

Description: Agile teams should acknowledge the need for optimal documentation of QRs (i.e., a satisfactory level of QR documentation that will not compromise agility and product quality). They should take actions to meet the need for optimal documentation of QRs and start documenting QRs at early stages.

Justification: ASD’s focus on minimal documentation does not translate to the avoidance or neglect of documentation. However, evidence shows that practitioners misinterpret the ‘minimal documentation focus of ASD’ as a call for no documentation [19]. Such misinterpretations lead to the lack of proper documentation and management of QRs [8], not strictly specifying QRs [15] and confusions regarding QR deliverables [42]. Therefore, practitioners should adopt practices that enable optimal QR documentation.

Ways to realize the recommendation:

- a) Educate ASD teams about the benefits of optimal QR documentation.
- b) Establish processes that help ASD teams adopt optimal QR documentation (e.g. applying agile playbooks, QR documentation templates).

Related work: [5], [8], [10], [15], [19], [30], [42].

3) RECOGNIZE THE INDICATORS OF QR DOCUMENTATION CHALLENGES

Description: ASD teams and managers should be able to notice and identify different indicators of QR documentation challenges, the consequences of poor QR documentation practices and take actions to minimize or prevent them.

Justification: ASD teams can proactively work on minimizing the challenges of QR documentation if they distinguish the indicators and consequences of challenges of documenting and managing QRs [9].

Way to realize the recommendation:

ASD teams and managers should be aware of the indicators of QR documentation challenges, recognize the consequences of poor QR documentation practices, and utilize actions that help minimize or prevent the challenges (see Table 1).

Related work: [8]–[10], [13], [30], [43].

B. RECOMMENDATIONS THAT INTRODUCE ARTIFACTS AND PRACTICES TO SUPPORT AND ENHANCE QR DOCUMENTATION

In this category, different artifacts for documenting QRs in ASD are provided with examples. Practices that support QR

TABLE 1. QR documentation challenge indicators, mitigations for corresponding challenges and consequences of poor QR.

Indicator of QR documentation challenge	Mitigation action to minimize the challenge	Consequences of poor QR documentation (e.g. outdated QR documents, unclear QR specification, missing QR specification)
Outdated QR documents	Check and ensure that the QR documentation is up-to-date.	- System quality degradation [10] - Project failure [8]-[10] - Increased maintenance costs
Unclear QR documents and specifications	Review and clarify QR specifications.	- Rework [9],[10],[31] - Inability to scale software [31]
Challenges with traceability of QRs	Ensure traceability of QRs during documentation.	-Wrong implementations [13] - Damaging business relationships [31]
Missing QR goal specifications	Ensure the specification of clear and measurable QR goals and targets [9], [43].	- New members joining the team identify implementing and maintaining QRs difficult [10]

documentation and enhance aspects of documenting QRs in ASD are also suggested.

1) APPLY LIGHTWEIGHT ARTIFACTS TO DOCUMENT QRs

Description: ASD teams should be aware of lightweight artifacts that are available for specifying and documenting QRs and utilize them as needed.

Justification: ASD teams face challenges from confusions on QR specification approaches [44]. In this regard, they should acquaint themselves with lightweight artifacts that are used for documenting QRs in ASD, and utilize them as needed.

Ways to realize the recommendation:

- a) User stories with acceptance criteria: ASD suggests user stories to specify and document requirements by focusing on users’ goals. User stories can be used to document QRs by specifying them from users’ perspective and explicitly attaching an acceptance criteria for the QRs, and a user story Id, which identifies the user story and its version. An example of a user story for documenting QRs is as follows:

- User story Id: QRV1.0
User story: As an <end user>,
I want to <the app to support multiple language >
So that <I can access the app using Amharic, English, and Finnish language interfaces>
Acceptance criteria: perform internationalization tests to ensure support for Amharic, English, and Finnish languages and to ensure that the corresponding layout formats function correctly.

Note: in ASD, QRs may evolve and change. In such cases, the user story, and its acceptance criteria should be updated to reflect the corresponding change.

- b) Given/When/then templates: ASD teams utilize ‘Given <>/When <>/Then <>’ templates for specifying QRs [30]. For instance, a QR for battery consumption can be specified as:
 - Given <the battery level is less than 20%> When <the phone is up and running> Then <raise> an alert for battery saver mode>.
- c) Apply a system story to specify QR: utilize system stories to specify features related to QRs that could not be allocated in user stories. System stories describe QRs which spread over multiple user stories [25], [47]. For instance, we can write a system story for a reliability QR verifying the presence of a SIM card interface in a mobile phone device by including a brief summary of the rationale behind the system story, as shown in Table 2.

TABLE 2. System story illustration.

Field	Description
System Story Id	SS2
Description of the QR	All SIM card interfaces must be tested.
System story	System shall verify SIM card interface.
Rationale	We need to verify the SIM card interfaces in the mobile phone are present and functioning properly.

- d) Specify QRs as part of the Definition of Done (DoD) of FRs: the DoD is a criterion for accepting tasks as done. QRs can be documented in the DoD of epic-, story-, and task-level requirements. The recommendation when using a DoD is to keep the DoD list as small as possible [30].
 - An example of a DoD that can be attached to a user story describing a login form can be written as ‘Perform SSL (Secure Sockets Layer) certificate test for login form’.
- e) Apply iterative and incremental prototypes: companies that adopt model driven development in the context of ASD can apply iterative and incremental prototypes to document QRs. Here, the approach enables the discovery and documentation of QRs and the update of QRs as the system evolves [30]. It is also recommended that such prototypes are reviewed with a focus on QRs and the big picture of the system as the system evolves [45].
- f) Specify QRs in the definition of ready: The definition of ready concept provides a means of checking for what work or task can enter a sprint [10], [46]. We specify QRs in the definition of ready for user stories that contain a definition of the user story, its acceptance criteria, dependencies with other user stories, a list of QRs, architecture criteria, a person responsible for accepting the user story, and reviewers of the user story.

Related work:[10], [25], [30], [44]–[47].

2) UTILIZE DOCUMENTATION ARTIFACTS BUILT TO DOCUMENT SPECIFIC QRs, SUCH AS SECURITY AND USABILITY

Description: ASD teams may use documentation artifacts tailored to specify and document specific QRs, such as security and usability requirements.

Justification: Security and usability are commonly discussed QRs in the literature on the management of QRs in ASD [10]. However, ASD user stories are seen as inadequate for specifying and documenting these QRs. In this regard, the recommendation is to use artifacts tailored to specify and document security and usability [10].

Ways to realize the recommendation:

- a) Apply abuser stories to document security requirements: an abuser story describes the interaction of a malicious user posing a threat to a system and the potential threats and risks to the system that may occur from the malicious interaction [26]. Misuse stories and evil stories are other terminologies describing a similar concept. ASD teams should discuss abuser stories together with the customer team to ensure their relevance and importance. Abuser story examples for addressing security concerns that may arise from a potential malicious user with access to regular accounts in a web portal system of an institution can be written as follows:
 - As a malicious employee, I can change my account privileges, so that I can get access to admin rights and manipulate the system.
 - As a malicious employee, I want to brute force the login to steal business data.
- b) Apply misuse cases to document security requirements: misuse cases describe the sequence of events that may happen during security incidents, and they allow developers to visualize vulnerabilities and create countermeasures to resolve such vulnerabilities that may arise from the security incident [48]. Figure 3 shows a misuse case example of a malicious user of an online web store who may use the ‘add review comment’ feature on the website by adding malicious scripts in the review comments of the online web store. Here, we have three different actors: a regular user who adds review comments about a product with no malicious intentions, a malicious user whose purpose is to intrude the system by adding scripts in the review comments, and the admin, who moderates the review comments and detects malicious events. When a malicious user attempts to intrude the system by adding malicious comments, the admin can blacklist the ID and take further action.
- c) Apply UserXstories to document usability requirements: UserXstories are stories that combine the perspectives of personas (i.e., a representation of users of a system) and Nielsen’s heuristics (i.e., guidelines used for evaluating usability and user interface aspects) [49], to specify usability requirements in ASD [50]. The recommendation is to write UserXstories together with the acceptance criteria stipulated by user interface designers and the scrum team. A UserXstories template is shown below:
 - As a <persona> I want/need <goal> for this <interaction>,
 - Through/when <task>/<context>
 - I evaluate that my goal is achieved when <feedback>

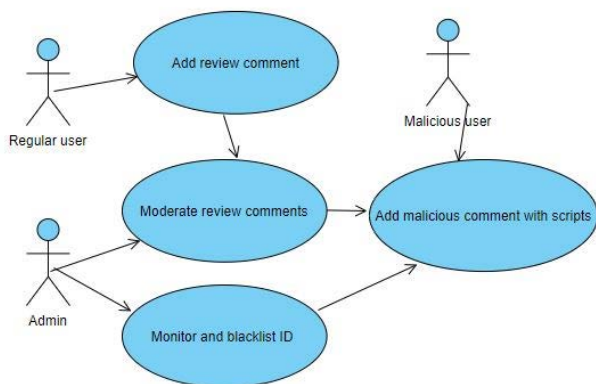


FIGURE 3. Misuse case illustration.

Acceptance criteria:

Checks <action> through <set of conditions> to satisfy <Nielsen’s heuristic(s)> of action and <Nielsen’s heuristic(s)> of feedback

We illustrate UserXstories by specifying usability and user interaction requirements for updating a booking by a customer in a flight-booking app.

- As a <customer>, I need to <update my flight booking details>; for this, <the app should allow me to select my booking using my booking reference code>, Through/when <updating my flight booking details>/<while I am logged in on the app> I evaluate that my goal is achieved when <my flight booking is updated>

Acceptance criteria:

Check <the app validates that the selected flight booking is permitted for rebooking> through <using the booking reference number>, to satisfy <H5 Nielsen’s heuristic(s)—error prevention> of action and <H9 Nielsen’s heuristic(s)—help users recognize, diagnose, and recover from errors> of feedback.

Check <the app displays the customer’s name and flight information> through <selecting from existing bookings or searching by booking reference number> to satisfy <H1 Nielsen’s heuristic(s)—visibility of system status> of action and <H6 Nielsen’s heuristic(s)—recognition rather than recall> of feedback.

Check <the app displays an updated booking when the customer updates the flight booking information> through <displaying a message saying that the flight booking is updated> to satisfy <H1 Nielsen’s heuristic(s)> of action, and <H 6 Nielsen’s heuristic(s)> of feedback.

Related work: [10], [21], [26], [48]–[51].

3) ALLOCATE ROLES AND RESPONSIBILITIES FOR DOCUMENTING AND MANAGING QRS IN ASD

Description: Assign roles to individuals with expertise on QRs to assist in the documentation of QRs in ASD. Empirical

evidence from the literature reveals that assigning roles to individuals with expertise in security, usability, and quality assurance have been helpful for documenting and managing QRs [17], [52], [53].

Justification: The lack of expertise in QRs is a challenge in documenting and managing QRs in ASD. Product owners and project managers may not always have the required skills for defining QRs, such as security and usability [10], [54]. They are also prone to forgetting QRs [55]. Some product domains may also have strict requirements and need separate organizations to handle QRs.

Ways to realize the recommendation:

- Assign security masters and security experts for security QRs: the literature reports that such roles are beneficial in specifying and analyzing the security requirements and the involvement in related discussions with agile teams [10], [30], [52].
- Assign usability experts and usability product owners for usability QRs: The literature reports that such roles have been beneficial in eliciting usability requirements, documenting, and facilitating usability discussions with agile teams. We can also have interaction designers that support the elicitation and documentation of QRs [10], [17], [53], [55].
- Assign separate organizations or teams to document and handle QRs in the process: companies with product domains that require strict regulations (e.g., embedded systems, telecommunications) and ASD, may employ separate organizations to handle documentation and manage QRs (e.g., security organization or security team, performance teams) [10], [30].

Related work: [10], [17], [30], [52]–[55].

4) ENSURE CLARITY OF QRS WHEN DOCUMENTING THEM

Description: To ensure a clear specification and documentation of QRs. The specification and documentation of QRs need to be clear and unambiguous, precise, and measureable.

Justification: The lack of clarity of QRs is a challenge reported in the ASD literature [54]. When product owners and project managers fail to specify QRs clearly, ASD teams may lack a common understanding of the QRs and consequently, adopt their own interpretations. These may lead to wrong implementations and at times, create friction among team members [8], [10], [11].

Ways to realize the recommendation:

- Use QR metrics, i.e., metrics for measuring QRs [56], to specify and document QRs in clear, precise, and measureable terms. For example, we can specify maintainability as ‘complexity of files should be below 20%’ by using the metric, complexity of files.
- Encourage open discussion to avoid ambiguous descriptions and definitions of QRs

Related work: [8], [10], [11], [54], [56].

TABLE 3. Levels of abstraction illustration.

Field	Description
Issue type, name	Story, test coverage story
Summary of story (should be a precise, clear, and measurable description of the goal).	Add test cases to get the test coverage up to 100%.
DoD	With the additional test cases, the test coverage should reach 100%.
Epic link	The linked epic is the testability QR (The system shall be testable)

5) APPLY LEVEL OF ABSTRACTION WHEN SPECIFYING QRS
Description: The level of abstraction of a QR describes the granularity level of the QR (i.e., the scale or level of detail of the QR). The recommendation is to apply different levels of abstraction (i.e., splitting and refining QRs from higher to lower levels with corresponding optimal information) when specifying and documenting the QRs.

Justification: Granular QR specifications are problematic in ASD [54]. An abstraction of requirements helps in communicating requirements better [57]. An appropriate level of abstraction can help in clarifying the QR task and achieving good effort estimation of the corresponding task [58].

Way to realize the recommendation:

Apply different levels of abstraction that entail optimal levels of detailed information appropriate for the corresponding granularity level of the QRs.

One way to do this is to split and specify QRs as epics, stories, and tasks at decreasing levels of granularity. An epic describes a high-level requirement [30]. A story represents a refined, lower level requirement of an epic [13]. An epic is split and refined into many stories, and a task represents a QR item to implement. For instance, one can specify a testability QR at the epic level as ‘The system shall be testable,’ and this epic can be refined into multiple stories and tasks. Table 3 shows a test coverage story, which is under the testability QR epic. We specify the story by providing descriptions and DoDs and linking it with the epic.

Related work: [13], [30], [54], [57], [58]

6) ENSURE THE TRACEABILITY OF QRS

Description: create and keep a linkage among different levels of abstraction of the QRs when documenting them. Document the dependencies among QRs and FRs when needed.

Justification: ASD faces limitations in traceability mechanisms for QRs [11], [13]. Limited traceability to higher-level requirements (both QRs and FRs) may slow down corrective actions taken to resolve errors during development [54], [59]. Practitioners can easily identify and correct QR issues when taking corrective actions based on documented QRs, which enable backward traceability to higher level requirements [28].

Way to realize the recommendation:

Record and keep a link between different levels of QRs (e.g., link QRs among epic-, story-, and task-level specifications of a specific QR). Table 4 illustrates an example.

TABLE 4. Recording links to ensure the traceability of QRs.

Field	Description
Project	X
Issue type	Story
Component	SW component z
Reporter	Mrs. P
Priority	Minor, Medium, Major
Summary of story	The loading time for the webpage should be under three seconds.
DoD	Given that <the webpage is loading>, When <accessed through browsers: Chrome, Firefox, Opera, Edge, and Safari>, Then <the loading time should be under three seconds>
Epic link	The linked epic is epic_Performance

Related work: [11], [13], [28], [54], [59].

7) DOCUMENT DECISIONS RELATED TO QRS

Description: document important QR-related decisions, the rationale behind the decisions, and the affected components. QR-related decisions should be documented and saved in requirements management repositories that can later serve as knowledge repositories.

Justification: Informal QR documentation practices lead to missing knowledge and rationale behind already made decisions regarding QR [2]. Documenting QR decisions will help minimize knowledge that may otherwise remain tacit to practitioners or be lost when these practitioners leave the company [21], [30], [55].

Ways to realize the recommendation:

Small, collocated, and collaborative ASD teams may prefer to communicate QRs in whiteboard meetings and decide to document only important QR decisions [13]. In such scenarios, they should precisely document the decision regarding the respective QR along with a rationale for the decision as shown in Table 5.

TABLE 5. Documenting QR decisions.

Field	Description
Decision Id	24
Decision summary	Run penetration testing.
Affected component	SW X
Rationale	Detect and identify potential vulnerabilities of the system.

Related work: [2],[13], [21], [30], [55].

8) APPLY SEPARATE OR SPECIFIC BACKLOGS TO DOCUMENT QRS

Description: utilizing separate or specific backlogs, which run in parallel to the product backlog, to document QRs identified during the development process and distinct QRs. When applying a specific or separate backlog, we should ensure

that it does not affect the agility of the software development process.

Justification: QRs which are discovered during the development process of ongoing sprint can be left undocumented [13]. At times QRs such as security may not receive adequate attention compared with other functional requirements in sprint backlogs [60]. Utilizing a distinct backlog to document specific QRs can help to give more focus to QRs [41].

Ways to realize the recommendation:

- a) Improvement backlog: a separate backlog that runs in parallel to sprint backlogs and is used to document QRs and improvement items which are identified by development team during the development process of an ongoing sprint [30].
- b) Security backlog: separate backlogs allocated for specifying security requirements in ASD [41]. Items within the security backlog can be written in the form of abuser or misuse stories to describe how a person with malicious intent could do harm. Each item is prioritized according to risk and needs to be linked to product backlog items. Security backlogs can be better maintained by including a person who has expertise in security [10].
- c) Shadow backlog: a separate backlog that contains issues such as inconsistencies and omitted usability tasks [61].

Related work: [10], [13], [30], [41], [60], [61].

9) WHEN SPECIFYING QRS ENSURE DOCUMENTING THE MEANS TO VERIFY AND VALIDATE THEM

Description: While specifying QRs at different levels of hierarchy, it is important to clearly document how to verify and validate the corresponding QRs.

Justification: in ASD verification of QRs can be problematic in cases where QRs are defined without reaching agreement on the verification method [16]. Failing to specify the verification and validation aspects of QRs can lead to an underperforming product that does not meet the QR expectations of end users [16]. For instance, when the verification aspect of capacity of a login page is not considered and specified clearly, testing teams may not verify and validate the requirement, and the companies may deliver software that underperforms and does not meet end users' expectations [11].

Ways to realize the recommendation:

Specify how the QR is to be verified and validated:

- Peer reviews, design reviews, code reviews, application program interface (API) reviews [30].
- Customer reviews (review sessions with customers and/or users to verify and validate QRs) [30].
- Continuous integration [10].
- DoDs and checking them against acceptance criteria [30].

Related work: [10], [11], [16], [30].

V. VALIDATING AGILE QR-DOC GUIDELINES

This section presents validation of the Agile QR-Doc. Section A presents the validation design and Section B presents the findings from validation of the guidelines.

A. AGILE QR-DOC GUIDELINES VALIDATION DESIGN

In order to validate the guidelines and ensure their suitability for ASD, we considered:

- 1) Evaluating the usefulness, relevance, and the understandability of the guidelines, with respect to supporting QR documentation in ASD. The measurement of information quality aspects, i.e., of perceived usefulness, relevance, and understandability, provide an insight into users' satisfaction levels regarding the quality of information systems [62].
- 2) Evaluating the impact on the agility of the software process, i.e., in terms of speed, and leanness or being simple enough to use in ASD. Speed and leanness or simplicity are relevant aspects which are used to measure agility of software process [71]. We believe that these provide an insight on the compatibility of the guidelines with ASD. It helps understand whether the guidelines are simple enough to use in supporting QR documentation tasks in ASD, and determine that their applications do not compromise the speed of development.
- 3) Evaluating the level that important aspects for documenting QRs in ASD are covered in the guidelines. This helps determine whether the guidelines cover important aspects for documenting QRs in ASD.
- 4) Gathering suggestions for enhancing the guidelines.

Accordingly, we formulated the following research questions:

RQ1. How do practitioners perceive the usefulness, relevance, and understandability of the guidelines in supporting QR documentation?

RQ2. How do practitioners perceive the impact of the guidelines with respect to agility, i.e., the speed and leanness or simplicity, of the software development process?

RQ3. Do practitioners perceive that the guidelines cover the important aspects to consider when documenting QRs in ASD? What aspects do they consider important when documenting QRs?

RQ4. What do practitioners suggest for improving the guidelines?

1) PARTICIPANTS USED FOR VALIDATING THE GUIDELINES

Ten software practitioners, from two companies within the Q-Rapids consortium, participated in validating Agile QR-Doc (see Table 6). These two companies have rich experiences of ASD. QRs have great importance in their contexts. The first company (C1) operates in the embedded systems and telecommunications domain. At the time we run the validation, C1 has been applying ASD for 14 years and was using Scrum. It is medium-sized company with over 600 employees. C1 applies QR documentation practices and tools (e.g., applying agile playbook guidelines, and using JIRA

TABLE 6. Participants of the validation.

Participant ID	Role	Software engineering experience (Years)	ASD experience (Years)	Company
P1	Software project manager	10	10	C1
P2	Software quality manager	10	10	
P3	DevOps Tech Lead	15	15	
P4	Software Engineer/Process coach	14	6	
P5	Specification engineer	30	10	
P6	Specification engineer	24	10	C2
P7	Software architecture team lead	21	10	
P8	System specification group leader	15	2	
P9	Specification engineer	24	7	
P10	Architect	< 1 year	< 1 Year	

to document QRs). The second company (C2) is a global, large-scale telecommunications company with over 100,000 employees. It has been applying ASD for 12 years and uses large-scale distributed ASD.

C2 applies varying QR documentation practices and tools depending on the QR types and organizational level. DOORS, JIRA, and Accept 360 are some of the tools used to document QRs.

We used the key informant technique [63] and consulted the representatives of the companies to recruit participants who have experience in requirements specification and management, to validate the guidelines. The key informant technique offers collecting feedback from relevant experts on a specific topic. In our study, we had participants, who have broad experience in software engineering and ASD.

2) DEMONSTRATION AND EVALUATION TO VALIDATE THE GUIDELINES

Below, we describe the demonstration and evaluation activity, in the DSRM process, that we applied in each company to validate the guidelines.

Company 1: We organized the demonstration and evaluation session with five practitioners of the case, in October 2019. The duration of the demonstration and evaluation was 40 minutes.

- 1) First, a researcher presented the objective, scope, and the intended target of the guidelines.

- 2) Then the researcher demonstrated the use of Agile QR-Doc guidelines by going through the recommendations of the guidelines and briefly explaining how the recommendations can be used to support QR documentation, using the prototype.
- 3) We asked each participant to explore the guidelines in detail for ten minutes. Each participant explored the guidelines independently using the prototype for ten minutes.
- 4) Following the exploration, we asked the participants to provide their suggestions for improving the guidelines in open discussion. Their feedback was audio recorded and later transcribed for data analysis.
- 5) We asked each participant to answer a survey questionnaire. The survey questionnaire was structured in a way that it addressed **RQ1-RQ4** (see Appendix). We used the Webropol online survey tool to collect the participants' feedback through an online questionnaire.

Company 2: the demonstration and evaluation were done with five practitioners of the case separately, in October 2019. We performed the demonstration and evaluation with each of the practitioners separately. We did this because we were not able to organize the session by involving all participants at the same time. The demonstration and evaluation session with each of the practitioners lasted between 35 and 40 minutes. We followed similar steps as in company 1, i.e., the steps 1-5 described above.

In the survey questionnaire, we used a five-point Likert scale (1 = strong disagreement, 2 = disagreement, 3 = neutral, 4 = agreement, and 5 = strong agreement) to collect feedback. We collected feedback regarding the usefulness, relevance, and the understandability of the guidelines and the impact of applying them in terms of agility. We also collected feedback using an open-ended question in the survey regarding the important aspects that the ASD practitioners consider when documenting QRs. We audio recorded the sessions of the demonstration and evaluation in both companies. The audio was professionally transcribed for analysis purpose.

3) DATA ANALYSIS OF COLLECTED FEEDBACK

In order to analyze data collected during the demonstration and evaluation phase, we applied quantitative and qualitative analyses. We used quantitative analysis for the results of the survey questionnaire. We report descriptive statistics, such as the sample size (N), minimum (Min), median (Mdn), and maximum (Max) values of the ratings. We also used a one-sample Wilcoxon signed rank test [64], which is used to examine variations in the responses of the participants in cases of small sample sizes [64], [65]. We applied the one-sample Wilcoxon signed rank test to examine how the evaluation results differed from the median value, which was 3 or neutral in our scale. We report the results of the one-sample Wilcoxon signed rank observed value (V) and the

significance level (p) by setting a 95% confidence interval for tests.

In order to analyze the qualitative data, we applied thematic analysis [66], to synthesize aspects that participants identify as important in documenting QRs and their suggestions for improving the guidelines. We first coded each of the responses as ‘important aspects’ and ‘suggestions for improvement’. Then we compared the responses under each of these labels. We then grouped and labelled closely related concepts under themes. However, during this process there were also non-recurring concepts that did not form a theme. These are reported as non-recurring themes.

B. RESULTS FROM AGILE QR-DOC VALIDATION

In this section, we report the findings from the validation of the Agile QR-Doc guidelines. Figure 4 plots summary of the findings of RQ1 and RQ2, regarding the usefulness, relevance, and the understandability of the guidelines and their impact on agility.

1) USEFULNESS, RELEVANCE, AND THE UNDERSTANDABILITY OF THE GUIDELINES IN SUPPORTING QR DOCUMENTATION (RQ1)

We found that the practitioners perceived the guidelines to be useful to support the optimal documentation of QRs, as evidenced by the descriptive statistics and the one-sample Wilcoxon sign ranked test result ($N = 10$, $Min = 3$, $Max = 5$, $Mdn = 4$, $V = 45$, $P = 0.006008$). Similarly, nine respondents perceived the guidelines as useful for improving software product quality ($N = 9$, $Min = 4$, $Max = 5$, $Mdn = 4$, $V = 45$, $P = 0.007412$). However, one of the respondents did not respond to the question.

Regarding relevance, we asked whether they consider the guidelines helpful in identifying relevant actions for documenting QRs in ASD and whether the guidelines are appropriate for documenting QRs. The participants reported that the guidelines are helpful in identifying relevant actions for documenting QRs in ASD ($N = 10$, $Min = 2$, $Max = 5$, $Mdn = 4$, $V = 41$, $P = 0.02459$) and are appropriate for documenting QRs in ASD ($N = 10$, $Min = 3$, $Max = 5$, $Mdn = 4$, $V = 28$, $P = 0.01766$).

Regarding understandability, the participants identified the guidelines as understandable enough to use for documenting QRs in ASD ($N = 10$, $Min = 3$, $Max = 5$, $Mdn = 4$, $V = 45$, $P = 0.007412$). Overall, the evaluation of the information quality aspects of the guidelines indicated that the guidelines were perceived as useful, relevant, and understandable to support the documentation of QRs in ASD.

2) THE IMPACT OF THE GUIDELINES WITH RESPECT TO THE AGILITY, LEANNESS OR SIMPLICITY AND SPEED, OF THE SOFTWARE DEVELOPMENT PROCESS (RQ2)

The practitioners perceived the guidelines as simple enough to use in ASD ($N = 10$, $Min = 3$, $Max = 5$, $Mdn = 4$, $V = 45$, $P = 0.006008$). Regarding agility in terms of speed, we asked if they thought that the guidelines did not

hinder ASD process. They perceived that the guidelines did not hinder ASD process ($N = 10$, $Min = 2$, $Max = 5$, $Mdn = 4$, $V = 32$, $P = 0.04183$). However, one respondent disagreed, whereas two others were neutral. The practitioners also perceived that the guidelines are helpful in the earlier consideration and documentation of QRs ($N = 10$, $Min = 3$, $Max = 5$, $Mdn = 4$, $V = 45$, $P = 0.007412$).

3) COVERAGE OF IMPORTANT ASPECTS FOR DOCUMENTING QRs IN ASD WITHIN THE GUIDELINES AND ASPECTS THAT THE PRACTITIONERS CONSIDER IMPORTANT WHEN THEY DOCUMENT QRs (RQ3)

We asked the participants whether they thought the guidelines covered important aspects to be considered when documenting QRs in ASD. All the ten participants agreed that the guidelines covered important aspects that needed to be considered. Three participants further explained why they thought that the guidelines covered important aspects. The DevOps tech lead (P3) stated ‘*on a high level, this serves as a good reminder of things to consider when documenting QRs*’. The specification engineer (P6) noted, ‘*with a short introduction, the guidelines seem to cover most important issues*,’ and another specification engineer (P5) commented, ‘*the decomposition from the top-level guidelines makes sense nicely*.’

In Table 7, we summarized the themes and corresponding statements reported as important aspects to consider in documenting QRs by the participants. We found that the participants considered the clarity of QRs to be the most important aspect when documenting QRs. Clarity refers to ensuring a clear definition and quantification of the QR and minimizing the likelihood of misinterpretation when documenting QRs. Recognizing the importance of QRs and the need to document them was the second top recurring aspect. The traceability of QRs and the verification and validation of QRs were the third most recurring themes. Additionally, the participants reported five non-recurring aspects that are important to consider when documenting QRs. These include costs and related performance impacts, defining processes for handling QRs, the impact on product, dependency on other QRs, and expertise of practitioners defining QRs.

4) SUGGESTIONS FOR IMPROVING THE GUIDELINES (RQ4)

The practitioners reported suggestions to improve the guidelines. Table 8 summarizes the participants’ suggestions of improvement.

The top recurring suggestion was including aspects on how to verify and validate the QRs when documenting them. According to the participants, documenting QRs in ASD requires the specification of a means for verifying and validating them. The second top suggestion concerns backlogs suggested in the guidelines to document QRs. According to the feedback, including such backlogs may incur additional costs and be unfavorable in small-sized companies. There was a suggestion to clarify the details of documenting QRs in separate or specific backlogs. In addition, we found suggestions for including quality goals and process quality

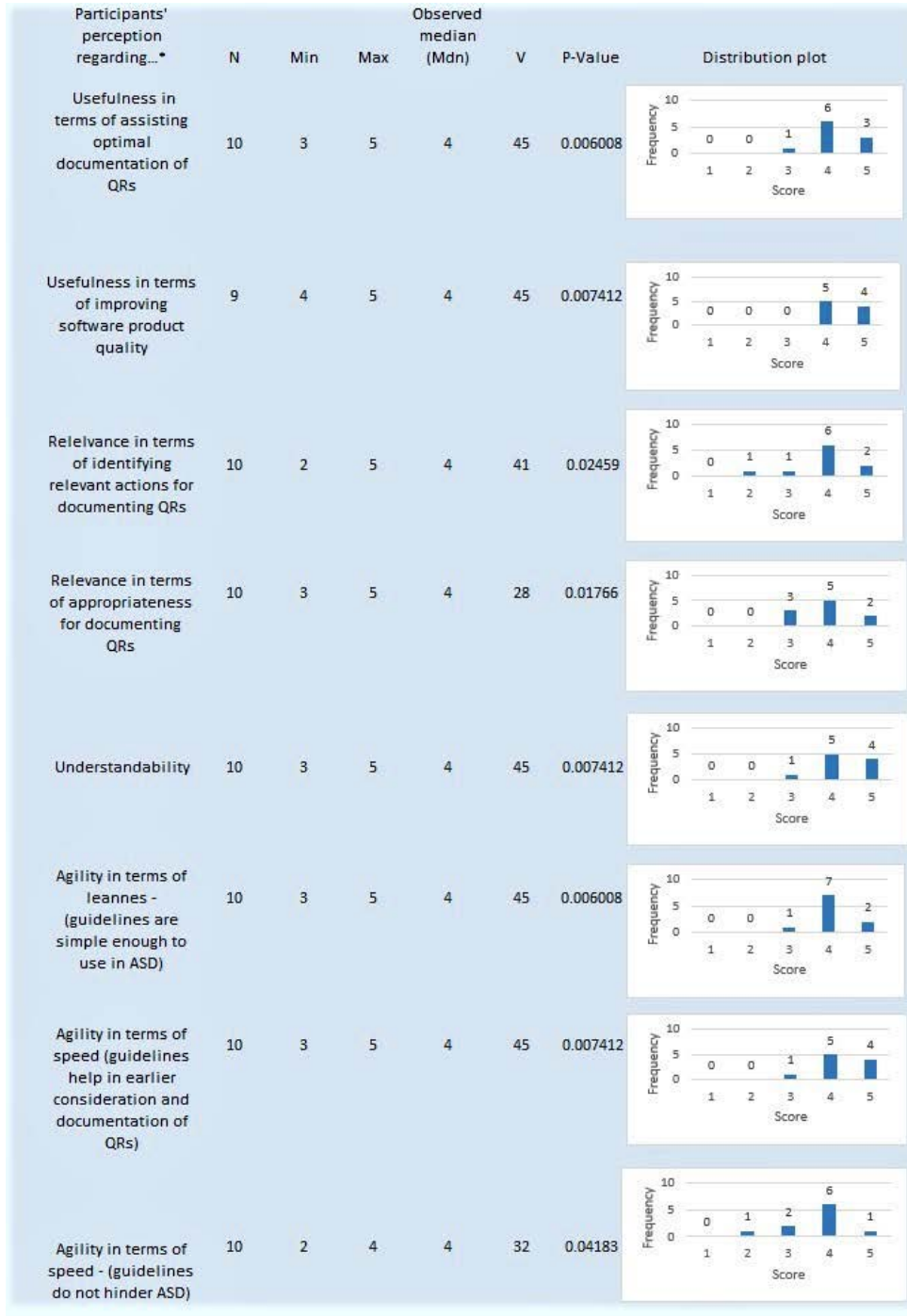


FIGURE 4. Plot and statistical summary of findings of RQ1 and RQ2.

aspects when documenting QRs. One suggestion also emphasized the need for the earlier consideration of QRs (i.e., the definition and documentation of QRs should begin together with FRs).

VI. DISCUSSION

This section discusses the implications of the study and compares Agile QR-Doc guidelines with related work.

A. IMPLICATIONS FROM THE EVALUATION FINDINGS

The results from the evaluation of the guidelines validated the suitability of Agile QR-Doc for the industry. The practitioners identified the guidelines to be understandable enough and useful in supporting QR documentation and in identifying relevant actions for documenting QRs in ASD. Although these findings are based on feedback from a small sample, they are remarkable and suggest the ease of understandability

TABLE 7. Aspects practitioners identify important when documenting QRs.

Theme, (# of labels under theme)	Important aspects to consider when documenting QRs in ASD reported by the participants
Clarity of QRs is an important aspect, (4)	Properly and clearly documenting the QR and its importance with respect to the project (quality). Clarity and quantification of QRs in order to make measurement, monitoring, and reporting easier. Clear terminology and discipline in requirement traceability, requirement categorization. No possibilities to interpret QRs wrongly.
QRs are as important as FRs and need to be documented, (3)	It is important to have QRs. It is important to have QRs documented. Optimal documentation and significance of QRs. They are as important as FRs. They need to be documented as well so that what has been agreed on is known.
Traceability of QRs, (2)	Clear terminology and discipline in requirement traceability and requirement categorization. Transparency through higher-level requirements down to actual implementation.
Verifying and validating QRs, (2)	Definition of ready and DOD and how to validate QRs. DoD and how to verify QRs.
Costs and related performance impacts, (1)	Performance effects, cost of getting required quality target.
Defining processes for handling QRs, (1)	Defining processes: how the common QRs are used in day-to-day ASD.
Impact on product	Impact on product.
Expertise of practitioners regarding QRs, (1)	Author expertise needs to be at a good level for related requirement.
Dependency on other QRs, (1)	Dependencies on other QRs.

of the guidelines and their usefulness and compatibility with ASD.

It is crucial that strategies aimed at improving or supporting the documentation and management of QRs in ASD do not negatively affect the agility of software development processes [10]. Regarding the impact of applying the guidelines in terms of agility (i.e., how simple enough they are to use in ASD, and how they affect the speed of the development process in ASD), the practitioners perceive the guidelines as simple enough to use in ASD, not hindering the speed of ASD and helpful in the early consideration and documentation of QRs. However, we noticed that one respondent disagreed when asked if the guidelines did not hinder the speed of ASD. This could be because some recommendations were not described and justified adequately in the initial version of the guidelines. Therefore, taking the feedback into consideration we improved the guidelines by refining the descriptions, justifications, and providing illustrative examples. For instance, the recommendations of separate backlogs have been refined to clarify that their implementation should not compromise the agility of the process.

TABLE 8. Summary of suggestions for improvement.

Theme (# of suggestions under the theme)	Suggestion
Verification and validation aspect (3)	<p>Process coach (P4), <i>'But that part is missing especially how you validate the QRs. For instance, [the] capacity you can measure, but some other QRs, how do you validate [them]. . . It is not only that. Nevertheless, QRs affect the architecture also. Let us say for instance, security.'</i></p> <p>Software architecture team lead (P7), <i>'Is there something on the, verification of the quality requirements? Did you have some, how to handle them in the continuous integration and verification? Perhaps that might be one aspect to look at if it is not because for me, the requirements, they are just paper unless you have a test case for them and you sometimes verify it. So, it's . . . just an expert opinion that okay, somebody says that okay, we implemented that in software, unless you verify it somehow.'</i></p> <p>System specification group leader (P8), <i>'What came to my mind concerning this quality of requirement, what many times may be a difficult thing, what, also perhaps needs to be figured out and documented is . . . how you verify those requirements and . . . how to make sure that you comply with these quality requirements.'</i></p> <p>System specification group leader (P8), <i>'How it is then verified and . . . [how do] you fulfil this quality requirement? Have you given any thought on that? So, how to verify it and if it is verified . . . how you prove that you fulfil that quality requirement.'</i></p>
Concerns on specific backlogs (2)	<p>System specification group leader (P8), <i>'So for these difficult topics, you need to have specific backlogs and, of course kind of, as I'm in a, or we are in a big company, it's easy for us, we have a lot of people, so we can manage specific areas specifically, and we can have specific people allocated to some specific things to do. But of course, if you have a smaller company, then it's not that easy.'</i></p> <p>Software architecture team lead (P7), <i>'Maybe this specific backlogs topic, that could be widened to show . . . what . . . the different approaches for handling quality requirements in backlogs [are], and then these specific backlogs might be one option, and then those are the options for complete coverage. Because now . . . from the title, one might get the impression that okay, specific backlogs are really . . . the recommended way to go.'</i></p>
Quality goals (1)	DevOps tech lead (P3), <i>'It is important that we include also the quality goals.'</i>
Quality of process (1)	DevOps tech lead (P3), <i>'Aspect on quality of the process should be included'</i>
Emphasize defining and documenting QRs at early phases and higher levels, like FRs.(1)	Process coach (P4), <i>'When do you need to define QRs? Because QRs, I could see, are as important as FRs, and they should be defined at [the] top level, at the same time as FRs because they affect the design as well, and you should know the QRs when you are defining the architecture.'</i>

The findings of the evaluation revealed that the guidelines covered important aspects for documenting QRs in ASD.

This shows the suitability and adequacy of the practices, artifacts and knowledge conveyed in the recommendations for supporting QR documentation in ASD.

The clarity of QRs was the most recurring important aspect that practitioners considered in documenting QRs. In ASD, conceptual challenges regarding QRs can lead to unclear QR specifications and confusions regarding QR specification approaches [11]. Agile QR-Doc include suggestions such as the agile playbook that can help practitioners gain knowledge on QRs and ways of implementing them. Such practices help minimize confusions and unclear QR specifications in ASD. In distributed ASD settings, where achieving shared understanding about QRs is challenging [9], [44], agile playbook is helpful. Additionally, light-weight artifacts that illustrate different means of specifying QRs (e.g., specifying QRs in system stories, documenting them as DOD of functional requirements) give practitioners insight into these approaches.

Traceability was another important aspect that the practitioners considered for documenting QRs in ASD. This is an important finding considering that traceability has been reported as QR documentation challenge in ASD. For instance, tracing security QRs with informal agile documentation practices is difficult [59]. Identifying dependencies between QRs is challenging in the presence of complex backlog structure [13]. Practitioners also identified the importance of QRs, and the need to document them early, and ensuring verification and validation, as important when documenting QRs in ASD. Considering the feedback, we included recommendations about ensuring traceability in QR documentation.

The practitioners evaluated the initial version of the guidelines that contained only nine recommendations, i.e., recommendations: A) 1-3, B) 1-4 and B) 7-8. The feedback from the evaluation helped us enrich the recommendations in our guidelines. For instance, there were concerns regarding applying separate or specific backlogs to document QRs (e.g., using security backlogs to document security requirements and using shadow backlogs to document usability requirements). The practitioners noted that implementing such backlogs may require additional resources and can be expensive in small-sized companies. Additionally, they noted the need for enhancing the descriptions in our guidelines. There were also suggestions regarding the inclusion of verification and validation aspects when documenting QRs. Following the evaluation, we updated the guidelines by refining the recommendations and including new ones, i.e., recommendations B) 5-6 and B) 9. We also included detailed rationale for the recommendations, and provided additional illustrative examples.

B. IMPLICATIONS FOR SOFTWARE ENGINEERING INDUSTRY, RESEARCH AND EDUCATION

The Agile QR-Doc guidelines contribute to the software engineering industry, research and education.

To the software engineering industry, our work contributes by providing guidelines for supporting QR documentation

task in ASD. Practitioners can familiarize themselves with the QR documentation practices, artifacts and knowledge in the guidelines and apply them to proactively support QR documentation in ASD. For instance, requirements engineers and developers who may find documenting QRs, such as usability, difficult, can identify recommendations such as UserXstories artifacts, shadow backlogs, and using usability experts. These recommendations can help them better document and manage usability in ASD. Likewise, managers and product owners can learn how limited awareness on QRs and misconceptions on minimal documentation can negatively affect QR documentation. They can identify potential mitigation actions such as investing in QR trainings, adopting quality mindset development and agile playbook, which they can adapt to the betterment of the QR documentation in ASD.

To the software engineering research, our work contributes by filling the gap in guidelines that support the documentation of QRs in ASD. It also demonstrates the application of DSRM in building a design science artifact. We reviewed the state of the art of management of QRs in ASD, and explored the documentation of QRs in ASD cases, when building the guidelines. Researchers can utilize the knowledge created in the process of building our guidelines to get better understanding about the documentation and management of QRs in ASD. The recommendations, their justifications, and realization techniques give insight into the research of QR documentation in ASD too. Researchers can also utilize our work to build new proposals for facilitating documentation and management of QRs in ASD.

We also believe that software engineering education can benefit from the guidelines. For instance, when these guidelines are incorporated in requirements engineering exercises, students can learn about the importance of QRs, and the need for optimally documenting them in ASD. They can be familiarized with the artifacts, practices and knowledge proposed in the recommendations too.

C. COMPARISON TO RELATED WORK

In the agile requirement engineering literature, we identified guidelines that support the documentation of QRs in ASD.

Barbosa *et al.* [22] proposed guidelines for integrating security QRs in ASD that were evaluated by three experts. Wäyrynen *et al.* [26] proposed recommendations to improve the documentation and management of security requirements in XP. Lee *et al.* [24] introduced guidelines to improve usability specification in ASD. Cajander *et al.* [15] provided recommendations for the documentation of usability and user experience. These guidelines focused only on security and usability. Our previous work provided guidelines for documenting QRs in ASD [13]. Agile QR-Doc guidelines address the documentation of QRs in general, within the context of ASD.

Recruiting large number of participants in empirical studies is a challenge in software engineering studies [70]. This is also witnessed in existing QR documentation guidelines proposals in ASD as shown here. We validated Agile

QR-Doc by evaluating them with ten practitioners from two ASD companies. Barbosa and Sampaio [22] evaluated their guidelines with three experts, and Lee et al. [24] evaluated their guidelines with ten practitioners. However, the guidelines by Cajander et al. [15], Wäyrynen et al. [26], and Behutiye et al. [13] were not evaluated empirically. In this regard, we call for more industry-academia collaborations since they can offer opportunities for the development and validation of guidelines, tools and other proposals in software engineering.

Most studies on documenting and managing QRs in ASD do not examine the impact on agility [3]. Only the work by Lee et al. [24] and our work examined the perceived impact of applying the guidelines on agility. A common limitation among all guidelines including our own Agile QR-Doc is that they have not been applied in industrial projects. Extending the application of these guidelines beyond evaluations will be beneficial for the software industry. Additionally, it offers opportunity to enhance the guidelines and strengthen their adoption in the industry. In Table 9, we present the comparison of these guidelines.

VII. THREATS TO VALIDITY

We use the validity aspect classification by Wohlin et al. [67] to discuss the validity threats in our study. We also discuss the measures we took to minimize the threats and improve the validity of our work.

Construct validity: explains threats that may arise from differences in the interpretations of operational measures that are studied. We mitigated potential construct validity threats in the design of the guidelines by adapting a DSRM to systematically develop the guidelines. A grounded theory would have been alternative approach to adopt when developing the guidelines. However, we found DSRM models suitable and easier to adapt. We found DSRM model by Peffers et al. [29], relatively easier to adopt and guide the development of our guidelines, since they helped clearly distinguish the design artifact and knowledge created in the process of building the artifact. Additionally, when building the guidelines, we applied rigorous steps and relevant knowledge gained both from the state of the art, and the practice on documentation and management of QRs in ASD.

In the validation design, we employed the key informant technique and considered practitioners who have relevant background and experience. The authors of the paper reviewed the guidelines, and the survey questionnaire before the validation process, i.e., demonstration and evaluation activity. During the demonstration and evaluation, a researcher was present and provided clarifications regarding the guidelines and the survey questionnaire to the participants.

Internal validity: inspects the presence of factors that we do not have control over or factors we may not have measured, which may influence the outcome of our findings. We acknowledge that there might be factors that we did not consider, which could have influenced our findings. For

TABLE 9. Comparison with related work.

Study	Focus	Have the guidelines been evaluated empirically?	Was it a student-expert- or practitioner-based evaluation?	Have the guidelines been applied or used in projects (industry or student projects)?
Agile QR-Doc guidelines (This paper)	QR documentation in ASD	Yes	Practitioner-based evaluation with ten practitioners from two ASD companies	No
Barbosa and Sampaio [22]	Integrating security into ASD, of which documentation is one aspect	Yes	Expert-based evaluation with three experts	No
Behutiye et al. [13]	QR documentation in ASD	No	No	No
Cajander et al. [15]	Including user perspectives, usability, and user experiences	No	No	No
Lee et al. [24]	Usability specification	Yes	Practitioner-based evaluation with ten practitioners (agile user-centered design specialists)	No
Wäyrynen et al. [26]	Security	No	No	No

instance, considering QR documentation practices from other ASD contexts, than those involved in our study can return practices and artifacts that were not included in our guidelines. It can also identify aspects that we have overlooked in our guidelines too. To minimize such threats, we included knowledge synthesized from studying the state of the art of management of QRs in ASD.

While there might have been factors, which we are unaware of, that influenced the outcome of the validation process, we took measures to minimize threats to internal validity. For instance, we triangulated data by transcribing the audio recorded during the demonstration and evaluation, listening to the audio recordings, and using the survey feedback.

External validity: explains the degree that one can generalize the findings of a study to wider scope. In order to improve the generalizability of the guidelines, we used knowledge gained both from studying the state of the practice of QR documentation in ASD cases, and synthesis of the state of

the art on the topic to build the guidelines. Ten software practitioners validated the guidelines by assessing the usefulness, relevance, understandability and impact on agility of the guidelines for supporting QR documentation in ASD.

We acknowledge the threat to external validity due to the small number of participants used in validating the guidelines. However, to mitigate this threat, we took the following measures. Firstly, we recruited practitioners (e.g., including roles such as specification engineers, software project managers and software architects) who have rich experience in software engineering and ASD for validating the guidelines. Secondly, we applied established quantitative analysis technique to interpret our findings. We used, the one-sample Wilcoxon signed rank test, which is used to evaluate variations in the response of participants of small sample sized studies to make appropriate interpretations. The technique is advantageous for analyzing small sample sized data [68] and is used, for example, in medical studies with small number of participants. For example, Zhang *et al.* [69] utilize one-sample Wilcoxon signed rank for analyzing clinical data of 8 patients. Additionally, we incorporated the descriptive statistics when analyzing and reporting the quantitative data findings. Another measure we took to minimize the threat was collecting qualitative data. This helped us supplement the quantitative data. Thus the validation utilized both quantitative and qualitative data. We agree that it is important to have additional studies to extend the generalizability of the study. In this regard, we plan to conduct additional validation studies of the guidelines in varying ASD contexts (e.g., ASD projects with varying product domains, team sizes, ASD methods and).

Conclusion validity: explains the reliability regarding the data collection and analysis used in deriving the findings of a study. We used the Webropol tool to collect the survey feedback systematically. This helped us ensure proper data collection. We recorded audio of the demonstration and evaluation sessions and transcribed the data. We also applied data analysis techniques suitable for analyzing our data. For instance, we used a one-sample Wilcoxon ranked test for testing variations in the opinions of small-sized teams. Additionally, in order to enhance the reliability of our study, the second and third authors reviewed and discussed the findings from the data analysis with the first author.

VIII. CONCLUSION

This paper developed the Agile QR-Doc guidelines with the aim of addressing the gap in guidelines that support QR documentation in ASD. We adapted DSRM to build the guidelines. Therefore, a rigorous process was followed utilizing knowledge and experience gained from studying QR documentation and management in ASD cases, and reviewing the state of the art on the management of QRs in ASD.

The guidelines were validated by ASD practitioners. Practitioners identified them understandable and useful for identifying relevant actions for documenting QRs. They also

identified Agile QR-Doc guidelines as simple enough to use in ASD, helpful in the early consideration and documentation of QRs, and not hindering the ASD process. According to them, the guidelines cover important aspects to consider when documenting QRs.

The guidelines presented recommendations focusing on raising awareness on the significance of QRs and their documentation and those proposing lightweight artifacts and practices for documenting QRs in ASD. Ensuring clarity of QRs, recognizing the significance of QRs and their documentation, ensuring the verification and validation aspects when documenting QRs, and ensuring traceability of QRs were some of the recommendations favored by the practitioners.

Our work contributes to the software engineering industry by providing guidelines for supporting QR documentation in ASD. For instance, requirements specification engineers can learn about the different light-weight artifacts used for documenting QRs in ASD. Junior developers can also benefit from the examples of artifacts and practices for documenting QRs presented in the guidelines.

Our work contributes to software engineering research since it addresses the gap in guidelines supporting QR documentation in ASD. Researchers can get a better insight into the documentation and management of QRs in ASD, by using the knowledge created in the design process of Agile QR-Doc. These include knowledge from the state of the art, the state of the practice of QR documentation and knowledge gained during the validation of the guidelines.

In the future, we consider extending the validation of the guidelines, with more cases of varying contexts (e.g., domain, product type, and ASD methodology) and participants of varying backgrounds. We also plan to utilize the guidelines with students in requirements engineering exercises.

APPENDIX. SURVEY QUESTIONNAIRE

1. Role
2. Experience in software development in years
3. Experience in agile software development
4. In my opinion, the guidelines are useful in assisting the optimal documentation of QRs in ASD. (**Usefulness in terms of assisting the optimal documentation of QRs**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
5. In my opinion, the guidelines may help in improving software product quality. (**Usefulness in terms of improving software product quality**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree

6. The guideline helps in identifying relevant actions when documenting QRs in ASD. (**Relevance in terms of identifying relevant actions**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
7. I find the artifact recommendations in the guidelines appropriate for documenting QRs in ASD. (**Relevance in terms of appropriateness for documenting QRs**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
8. I find the guidelines understandable enough to use while documenting QRs in ASD. (**Understandability**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
9. I find the guidelines helpful for the early consideration and documentation of QRs. (**Agility, in terms of speed**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
10. In my opinion, the recommendations in the guidelines do not hinder the ASD process. (**Agility, in terms of speed**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
11. I find the guidelines simple enough to use in agile software development. (**Agility, in terms of leanness**)
 - Strongly agree
 - Agree
 - Neutral
 - Disagree
 - Strongly disagree
12. Do you think the guidelines cover important aspects to consider while documenting QRs in ASD?
 - Yes, please elaborate.
 - No, please elaborate.

Please list at least two important aspects you would like to consider when documenting QRs in ASD.

REFERENCES

- [1] K. Wiegers and J. Beatty, *Software Requirements*. Redmond, WA, USA: Microsoft Press, 2013.
- [2] B. M. Aljallabi and A. Mansour, "Enhancement approach for non-functional requirements analysis in agile environment," in *Proc. Int. Conf. Comput., Control, Netw., Electron. Embedded Syst. Eng. (ICCNEEE)*, Sep. 2015, pp. 428–433, doi: [10.1109/ICCNEEE.2015.7381407](https://doi.org/10.1109/ICCNEEE.2015.7381407).
- [3] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects—A systematic literature review," in *Requirements Engineering: Foundation for Software Quality (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10153. Springer-Verlag, 2017, pp. 219–234, doi: [10.1007/978-3-319-54045-0_17](https://doi.org/10.1007/978-3-319-54045-0_17).
- [4] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: A systematic literature review," *Comput. Stand. Interface*, vol. 49, pp. 79–91, Jan. 2017, doi: [10.1016/j.csi.2016.08.011](https://doi.org/10.1016/j.csi.2016.08.011).
- [5] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: An empirical study," *Inf. Syst. J.*, vol. 20, no. 5, pp. 449–480, Nov. 2007, doi: [10.1111/j.1365-2575.2007.00259.x](https://doi.org/10.1111/j.1365-2575.2007.00259.x).
- [6] P. Mohagheghi and M. E. Aparicio, "An industry experience report on managing product quality requirements in a large organization," *Inf. Softw. Technol.*, vol. 88, pp. 96–109, Aug. 2017, doi: [10.1016/j.infsof.2017.04.002](https://doi.org/10.1016/j.infsof.2017.04.002).
- [7] S. Kocpzyńska, M. Ochodek, and J. Nawrocki, "On importance of non-functional requirements in agile software projects—A survey," in *Studies in Computational Intelligence*, vol. 851. Springer-Verlag, 2020, pp. 145–158, doi: [10.1007/978-3-030-26574-8_11](https://doi.org/10.1007/978-3-030-26574-8_11).
- [8] W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen, and A. Abherve, "How agile software development practitioners perceive the need for documenting quality requirements: A multiple case study," in *Proc. 46th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2020, pp. 93–100, doi: [10.1109/SEAA51224.2020.00025](https://doi.org/10.1109/SEAA51224.2020.00025).
- [9] W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen, and A. Abherve, "Towards optimal quality requirement documentation in agile software development: A multiple case study," *J. Syst. Softw.*, vol. 183, Jan. 2022, Art. no. 111112, doi: [10.1016/j.jss.2021.111112](https://doi.org/10.1016/j.jss.2021.111112).
- [10] W. Behutiye, P. Karhapää, L. López, X. Burgués, S. Martínez-Fernández, A. M. Vollmer, P. Rodríguez, X. Franch, and M. Oivo, "Management of quality requirements in agile and rapid software development: A systematic mapping study," *Inf. Softw. Technol.*, vol. 123, Jul. 2020, Art. no. 106225, doi: [10.1016/j.infsof.2019.106225](https://doi.org/10.1016/j.infsof.2019.106225).
- [11] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," *Inf. Softw. Technol.*, vol. 110, pp. 39–55, Jun. 2019, doi: [10.1016/j.infsof.2019.01.009](https://doi.org/10.1016/j.infsof.2019.01.009).
- [12] A. D. Lucia and A. Qusef, "Requirements engineering in agile software development," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 212–220, Aug. 2010, doi: [10.4304/jetwi.2.3.212-220](https://doi.org/10.4304/jetwi.2.3.212-220).
- [13] W. Behutiye, P. Karhapää, D. Costal, M. Oivo, and X. Franch, "Non-functional requirements documentation in agile software development: Challenges and solution proposal," in *Proc. Int. Conf. Product-Focused Softw. Process Improvement* in Lecture Notes in Computer Science, vol. 10611. Cham, Switzerland: Springer, 2017, pp. 515–522, doi: [10.1007/978-3-319-69926-4_41](https://doi.org/10.1007/978-3-319-69926-4_41).
- [14] A. Jarzębowski and P. Weichbroth, "A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices," in *Lean and Agile Software Development (Lecture Notes in Business Information Processing)*, vol. 408, A. Przybyłek, J. Miler, A. Poth, and A. Riel, Eds. Cham, Switzerland: Springer, 2021, pp. 91–110, doi: [10.1007/978-3-030-67084-9_6](https://doi.org/10.1007/978-3-030-67084-9_6).
- [15] A. Cajander, M. Larusdottir, and J. Gulliksen, "Existing but not explicit—The user perspective in scrum projects in practice," in *Human-Computer Interaction—INTERACT 2013 (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8119. Berlin, Germany: Springer, 2013, pp. 762–779, doi: [10.1007/978-3-642-40477-1_52](https://doi.org/10.1007/978-3-642-40477-1_52).
- [16] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, "Challenges and practices in aligning requirements with verification and validation: A case study of six companies," *Empirical Softw. Eng.*, vol. 19, no. 6, pp. 1809–1855, Dec. 2014, doi: [10.1007/s10664-013-9263-y](https://doi.org/10.1007/s10664-013-9263-y).
- [17] S. Anwar, Y. H. Motla, Y. Siddiq, S. Asghar, M. S. Hassan, and Z. I. Khan, "User-centered design practices in scrum development process: A distinctive advantage?" in *Proc. 17th IEEE Int. Multi Topic Conf.*, Dec. 2014, pp. 161–166, doi: [10.1109/INMIC.2014.7097330](https://doi.org/10.1109/INMIC.2014.7097330).

- [18] F. Gilson, M. Galster, and F. Georis, "Extracting quality attributes from user stories for early architecture decision making," in *Proc. IEEE Int. Conf. Softw. Archit. Companion (ICSA-C)*, Mar. 2019, pp. 129–136, doi: 10.1109/ICSA-C.2019.00031.
- [19] T. Dingsøy, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *J. Syst. Softw.*, vol. 85, no. 6, pp. 1213–1221, Jun. 2012, doi: 10.1016/j.jss.2012.02.033.
- [20] B. Kitchenham and S. L. Pfleeger, "Software quality: The elusive target [special issues section]," *IEEE Softw.*, vol. 13, no. 1, pp. 12–21, Jan. 1996, doi: 10.1109/52.476281.
- [21] G. Erdogan, P. H. Meland, and D. Mathieson, "Security testing in agile web application development—A case study using the EAST methodology," in *Agile Processes in Software Engineering and Extreme Programming* (Lecture Notes in Business Information Processing), vol. 48. Springer-Verlag, 2010, pp. 14–27, doi: 10.1007/978-3-642-13054-0_2.
- [22] D. A. Barbosa and S. Sampaio, "Guide to the support for the enhancement of security measures in agile projects," in *Proc. 6th Brazilian Workshop Agile Methods (WBMA)*, Oct. 2015, pp. 25–31, doi: 10.1109/WBMA.2015.9.
- [23] M. Daneva, E. van der Veen, C. Amrit, S. Ghaisas, K. Sikkil, R. Kumar, N. Ajmeri, U. Ramteerthkar, and R. Wieringa, "Agile requirements prioritization in large-scale outsourced system projects: An empirical study," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1333–1353, May 2013, doi: 10.1016/j.jss.2012.12.046.
- [24] S.-H. Lee, I.-Y. Ko, S. Kang, and D.-H. Lee, "A usability-pattern-based requirements-analysis method to bridge the gap between user tasks and application features," in *Proc. IEEE 34th Annu. Comput. Softw. Appl. Conf.*, Jul. 2010, pp. 317–326, doi: 10.1109/COMPSAC.2010.39.
- [25] R. E. Maria, L. A. Rodrigues, and N. A. Pinto, "ScrumS: A model for safe agile development," in *Proc. 7th Int. Conf. Manage. Comput. Collective Intell. Digit. EcoSyst.*, Oct. 2015, pp. 43–47, doi: 10.1145/2857218.2857225.
- [26] J. Wäyrynen, M. Bodén, and G. Boström, "Security engineering and eXtreme programming: An impossible marriage?" in *Extreme Programming and Agile Methods—XP/Agile Universe 2004* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3134. Springer-Verlag, 2004, pp. 117–128, doi: 10.1007/978-3-540-27777-4_12.
- [27] L. M. Cysneiros and E. Yu, "Non-functional requirements elicitation," in *Perspectives on Software Requirements*, J. C. S. do Prado Leite and J. H. Doorn, Eds. Boston, MA, USA: Springer, 2004, pp. 115–138, doi: 10.1007/978-1-4615-0465-8_6.
- [28] M. Oriol, S. Martínez-Fernández, W. Behutiye, C. Farré, R. Kozik, P. Seppänen, A. M. Vollmer, P. Rodríguez, X. Franch, S. Aaramaa, A. Abherve, M. Choraś, and J. Partanen, "Data-driven and tool-supported elicitation of quality requirements in agile companies," *Softw. Quality J.*, vol. 28, no. 3, pp. 931–963, Sep. 2020, doi: 10.1007/s11219-020-09509-y.
- [29] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manage. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.
- [30] W. Behutiye, P. Seppänen, P. Rodríguez, and M. Oivo, "Documentation of quality requirements in agile software development," in *Proc. Eval. Assessment Softw. Eng.*, Apr. 2020, pp. 250–259, doi: 10.1145/3383219.3383245.
- [31] W. N. Behutiye, P. Rodríguez, M. Oivo, and A. Tosun, "Analyzing the concept of technical debt in the context of agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 82, pp. 139–158, Feb. 2017, doi: 10.1016/j.infsof.2016.10.004.
- [32] A. Sidky and J. Arthur, "Determining the applicability of agile practices to mission and life-critical systems," in *Proc. 31st IEEE Softw. Eng. Workshop (SEW)*, Mar. 2007, pp. 3–12, doi: 10.1109/SEW.2007.61.
- [33] S. Capgemini and M. Focus. (2019). *World Quality Report 2018–19, Tenth Edition*. [Online]. Available: <https://www.capgemini.com/service/world-quality-report-2018-19/>
- [34] A. Sillitti and G. Succi, "Requirements engineering for agile methods," in *Engineering and Managing Software Requirements*. Berlin, Germany: Springer, 2005, pp. 309–326, doi: 10.1007/3-540-28244-0_14.
- [35] V. T. Heikkilä, D. Damian, C. Lassenius, and M. Paasivaara, "A mapping study on requirements engineering in agile software development," in *Proc. 41st Euromicro Conf. Softw. Eng. Adv. Appl.*, Aug. 2015, pp. 199–207, doi: 10.1109/SEAA.2015.70.
- [36] K. Beck. (2001). *Agile Manifesto*. Software Development. Accessed: Mar. 24, 2020. [Online]. Available: <http://agilemanifesto.org/>
- [37] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, "Requirements engineering challenges in large-scale agile system development," in *Proc. IEEE 25th Int. Requirements Eng. Conf. (RE)*, Sep. 2017, pp. 352–361, doi: 10.1109/RE.2017.60.
- [38] A. Tettmeyer, D. Hein, and H. Saiedian, "A tagging approach to extract security requirements in non-traditional software development processes," *Int. J. Secure Softw. Eng.*, vol. 5, no. 4, pp. 31–47, Oct. 2014, doi: 10.4018/ijssse.2014100102.
- [39] C. R. Camacho, S. Marczak, and D. S. Cruzes, "Agile team members perceptions on non-functional testing: Influencing factors from an empirical study," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 582–589, doi: 10.1109/ARES.2016.98.
- [40] X. Wang, L. Zhao, Y. Wang, and J. Sun, "The role of requirements engineering practices in agile development: An empirical study," in *Requirements Engineering* (Communications in Computer and Information Science), vol. 432. Cham, Switzerland: Springer, 2014, pp. 195–205, doi: 10.1007/978-3-662-43610-3_15.
- [41] T. Nicolaysen, R. Sassoon, M. B. Line, and M. G. Jaatun, "Agile software development," *Int. J. Secure Softw. Eng.*, vol. 1, no. 3, pp. 71–85, Jul. 2010, doi: 10.4018/ijssse.2010070105.
- [42] M. Budwig, S. Jeong, and K. Kelkar, "When user experience met agile: A case study," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, Apr. 2009, pp. 3075–3083, doi: 10.1145/1520340.1520434.
- [43] G. K. Hanssen and T. E. Fægri, "Agile customer engagement: A longitudinal qualitative case study," in *Proc. ACM/IEEE Int. Symp. Int. Symp. Empirical Softw. Eng. (ISESE)*, Sep. 2006, pp. 164–173, doi: 10.1145/1159733.1159759.
- [44] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *J. Syst. Softw.*, vol. 172, Feb. 2021, Art. no. 110851, doi: 10.1016/j.jss.2020.110851.
- [45] M. Kapyaho and M. Kauppinen, "Agile requirements engineering with prototyping: A case study," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 334–343, doi: 10.1109/RE.2015.7320450.
- [46] K. Power, "Definition of ready: An experience report from teams at Cisco," in *Agile Processes in Software Engineering and Extreme Programming* (Lecture Notes in Business Information Processing), vol. 179. Springer-Verlag, 2014, pp. 312–319, doi: 10.1007/978-3-319-06862-6_25.
- [47] J. Cleland-Huang and M. Vierhauser, "Discovering, analyzing, and managing safety stories in agile projects," in *Proc. IEEE 26th Int. Requirements Eng. Conf. (RE)*, Aug. 2018, pp. 262–273, doi: 10.1109/RE.2018.00034.
- [48] N. Newton, C. Anslow, and A. Drechsler, "Information security in agile software development projects: A critical success factor perspective," in *Proc. 27th Eur. Conf. Inf. Syst. (ECIS)*, Stockholm, Sweden, Jun. 2019. [Online]. Available: https://aisel.aisnet.org/ecis2019_rp/92
- [49] J. Nielsen. (2005). *Heuristic Evaluation Ten Usability Heuristics*. Accessed: Mar. 8, 2021. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/> and <https://pdfs.semanticscholar.org/5f03/b251093aee730ab9772db2e1a8a7eb8522cb.pdf>
- [50] J. Choma, L. A. M. M. Zaina, and D. Beraldo, "UserX story: Incorporating UX aspects into user stories elaboration," in *Human-Computer Interaction. Theory, Design, Development and Practice* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9731. Springer-Verlag, 2016, pp. 131–140, doi: 10.1007/978-3-319-39510-4_13.
- [51] V. Kongsli, "Towards agile security in web applications," in *Proc. Companion 21st ACM SIGPLAN Conf. Object-Oriented Program. Syst., Lang., Appl. (OOPSLA)*, 2006, pp. 805–808, doi: 10.1145/1176617.1176727.
- [52] T. Ayalew, T. Kidane, and B. Carlsson, "Identification and evaluation of security activities in agile projects," in *Secure IT Systems* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8208. Springer-Verlag, 2013, pp. 139–153, doi: 10.1007/978-3-642-41488-6_10.
- [53] L. Schwartz, "Agile-user experience design: Does the involvement of usability experts improve the software quality? State of the art and a first experiment," *Int. J. Adv. Softw.*, vol. 7, no. 3, pp. 456–468, 2014.
- [54] P. Karhapää, W. Behutiye, P. Rodríguez, M. Oivo, D. Costal, X. Franch, S. Aaramaa, M. Choraś, J. Partanen, and A. Abherve, "Strategies to manage quality requirements in agile software development: A multiple case study," *Empirical Softw. Eng.*, vol. 26, no. 2, p. 28, Mar. 2021, doi: 10.1007/s10664-020-09903-x.
- [55] J. Nawrocki, M. Ochodek, J. Jurkiewicz, S. Kopczyńska, and B. Alchimowicz, "Agile requirements engineering: A research perspective," in *SOFSEM 2014: Theory and Practice of Computer Science* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8327. Springer-Verlag, 2014, pp. 40–51, doi: 10.1007/978-3-319-04298-5_5.

- [56] L. López, X. Burgués, S. Martínez-Fernández, A. M. Vollmer, W. Behutiye, P. Karhapää, X. Franch, P. Rodríguez, and M. Oivo, "Quality measurement in agile and rapid software development: A systematic mapping," *J. Syst. Softw.*, vol. 186, Apr. 2022, Art. no. 111187, doi: [10.1016/j.jss.2021.111187](https://doi.org/10.1016/j.jss.2021.111187).
- [57] S. Park and F. Maurer. (2008). *The Requirements Abstraction in User Stories and Executable Acceptance Tests*. Accessed: Dec. 2, 2019. [Online]. Available: http://ase.cpsc.ucalgary.ca/uploads/Publications/park_maurer_agile2008.pdf
- [58] O. Liskin, R. Pham, S. Kiesling, and K. Schneider, "Why we need a granularity concept for user stories," in *Agile Processes in Software Engineering and Extreme Programming (Lecture Notes in Business Information Processing)*, vol. 179. Springer-Verlag, 2014, pp. 110–125, doi: [10.1007/978-3-319-06862-6_8](https://doi.org/10.1007/978-3-319-06862-6_8).
- [59] J. Peeters, "Agile security requirements engineering," in *Proc. Symp. Requirements Eng. Inf. Secur.*, vol. 12, 2005, pp. 1–4.
- [60] S. Türpe and A. Poller, "Managing security work in scrum: Tensions and challenges," in *Proc. CEUR Workshop*, vol. 1977, 2017, pp. 34–49.
- [61] A. Y. Wale-Kolade, "Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers," *J. Syst. Softw.*, vol. 100, pp. 54–66, Feb. 2015, doi: [10.1016/j.jss.2014.10.036](https://doi.org/10.1016/j.jss.2014.10.036).
- [62] V. McKinney, K. Yoon, and F. Zahedi, "The measurement of web-customer satisfaction: An expectation and disconfirmation approach," *Inf. Syst. Res.*, vol. 13, no. 3, pp. 296–315, Sep. 2002, doi: [10.1287/isre.13.3.296.76](https://doi.org/10.1287/isre.13.3.296.76).
- [63] M. N. Marshall, "The key informant technique," *Family Pract.*, vol. 13, no. 1, pp. 92–97, 1996, doi: [10.1093/famp/13.1.92](https://doi.org/10.1093/famp/13.1.92).
- [64] S. S. Mangiafico. (2016). *Summary and Analysis of Extension Program Evaluation in R Factorial ANOVA?: Main Effects, Interaction Effects, and Interaction Plots*. [Online]. Available: rcompanion.org/documents/RHandbookProgramEvaluation.pdf
- [65] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, p. 80, Dec. 1945, doi: [10.2307/3001968](https://doi.org/10.2307/3001968).
- [66] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, Sep. 2011, pp. 275–284, doi: [10.1109/esem.2011.36](https://doi.org/10.1109/esem.2011.36).
- [67] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012, doi: [10.1007/978-3-642-29044-2](https://doi.org/10.1007/978-3-642-29044-2).
- [68] P. B. Dao, "On Wilcoxon rank sum test for condition monitoring and fault detection of wind turbines," *Appl. Energy*, vol. 318, Jul. 2022, Art. no. 119209, doi: [10.1016/j.apenergy.2022.119209](https://doi.org/10.1016/j.apenergy.2022.119209).
- [69] C. Zhang, J. Gao, J. He, C. Liu, X. Lv, X. Yin, Y. Deng, Z. Lu, and Z. Tian, "Regulatory T-cell expansion in oral and maxillofacial langerhans cell histiocytosis," *Oral Surgery, Oral Med., Oral Pathol. Oral Radiol.*, vol. 130, no. 5, pp. 547–556, Nov. 2020, doi: [10.1016/j.oooo.2020.08.005](https://doi.org/10.1016/j.oooo.2020.08.005).
- [70] Y.-G. Guéhéneuc and F. Khomh, "Empirical software engineering," in *Handbook of Software Engineering*. Cham, Switzerland: Springer, 2019, pp. 285–320, doi: [10.1007/978-3-030-00262-6_7](https://doi.org/10.1007/978-3-030-00262-6_7).
- [71] A. Qumer and B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering," *Inf. Softw. Technol.*, vol. 50, no. 4, pp. 280–295, Mar. 2008, doi: [10.1016/j.infsof.2007.02.002](https://doi.org/10.1016/j.infsof.2007.02.002).



WOUBSHET BEHUTIYE received the M.Sc. degree in information processing science from the University of Oulu, in 2015. He is currently a Postdoctoral Researcher with the M3S Software Engineering Research Unit, University of Oulu. His research interests include agile software development, continuous software engineering, requirements engineering, quality requirements, technical debt management, and evidence-based software engineering. He was actively involved in European Union's H2020 Q-Rapids Project, "Quality aware rapid software development" work package as a Researcher.



PILAR RODRÍGUEZ received the B.Sc., M.Sc., and Ph.D. degrees in computer science, in 2006, 2008, and 2013, respectively. She is currently an Assistant Professor with the Universidad Politécnica de Madrid, Spain, and a Docent at the University of Oulu, Finland. Her research interests include agile and lean software development, software quality, value-based decision-making, and human factors in software engineering. She is a member of the TSE Review Board and has served as a Reviewer for other leading SE journals, such as *ESEM*, *IST*, *JSS*, and *SQJ*. She has been a PC Member of conferences, such as *ESEM* and *XP*, where she received the XP2021 Best Reviewer Award.



MARKKU OIVO received the Ph.D. and E.M.B.A. degrees. He has been a Professor and the Head of the M3S Research Unit, University of Oulu, Finland, since 2002. He has worked both in academy and industry in Finland, USA, France, Germany, Italy, and Spain. From 2000 to 2002, he was the Vice President and the Director of Research and Development at Solid Company. He held several positions at VTT, in 1986 and 2000, University of Maryland, from 1990 to 1991, Schlumberger Ltd., Paris, from 1994 to 1995, Fraunhofer IESE, from 1999 to 2000, the University of Bolzano, from 2014 to 2015, the Universidad Politécnica de Madrid, in 2015, and Kone Company, from 1982 to 1986. He has initiated and managed more than 100 national and international research projects. He has served as the chair and a committee member in organizing numerous international conferences and has been a reviewer for top scientific journals. He is a Founding Member and the Chair of the Steering Committee of ISERN.

...