How agile software development practitioners perceive the need for documenting quality requirements: a multiple case study

Woubshet Behutiye
M3S research unit
University of Oulu
Oulu, Finland
woubshet.behutiye@oulu.fi

Pilar Rodríguez
Universidad Politécnica de Madrid

Madrid, Spain
pilar.rodriguez@upm.es

Markku Oivo
M3S research unit
Universtiy of Oulu
Oulu, Finland
markku.oivo@oulu.fi

Sanja Aaramaa
Nokia
Oulu, Finland
sanja.aaramaa@nokia.com

Jari Partanen
Bittium Wireless Ltd.
Oulu, Finland
jari.partanen@bittium.com

Antonin Abhervé
Softeam
Paris, France
antonin.abhreve@softeam.fr

Abstract— **Agile software development (ASD) promotes minimal documentation and often prioritizes functional requirements over quality requirements (QRs). This may be beneficial in reducing the time to market of software. When considering QRs in ASD, the minimal documentation practice may be seen as a concern since QRs determine the success of software projects and are as well not easy to specify and document. Nevertheless, what do practitioners think of the necessity of documenting QRs in ASD? How do they perceive factors that may affect documentation of QRs in ASD? We conducted a multiple case study of three cases applying ASD, involving 12 participants. ASD practitioners identify that it is important to document QRs, and perceive that it contributes to ensuring quality, clarifying QRs, and helping in decision making. Time constraint, QR awareness and communication gaps on QRs influence the documentation of QRs in ASD. ASD teams may align their documentation practices to fit the sprint duration. The influence of QR awareness on documentation was dependent on project context and roles. Communication gaps can create confusion on QRs. Missing and outdated QR documentation may result in accruing technical debt, and lack of common understanding on QRs. The study synthesizes empirical evidence on the significance of documenting QRs in ASD and provides an insight into factors affecting documentation of QRs in ASD.**

Keywords-agile software development;documentation;quality requirements; non-functional requriements

I. Introduction

Agile software development (ASD) has been popular and widely adopted to meet the demands of dynamic business environments, where requirements are changing frequently and businesses have to stay competitive [1]. As a result, there have been many studies investigating different aspects (e.g. requirements engineering, adoption challenges and benefits, team dynamics, and Test driven development) in ASD. Recently, research in the engineering, documentation and management of Quality requirements (QRs)[1] in ASD are gaining traction. There are studies that investigate challenges of managing QRs in ASD [2]–[4], or those examining how QRs are engineered and managed in ASD [5], [6]. However, a topic that still requires more attention and clarification is the documentation of QRs in ASD.

QRs have important role in determining the success of projects [7]. In ASD, where minimal documentation is emphasized [8], QRs are often underspecified or not documented, and are neglected, and may lead to project failures [9]. On the other hand, the minimal documentation focus of ASD is seen as beneficial in delivering early return on investment and shortening the time to market [10].

Although ASD does not necessarily advocate a no need for documentation, studies reveal that practitioners may misinterpret the less emphasis on documentation as no need for documentation [11]. Such misinterpretations may be counterproductive when considering QRs. QRs may not be documented and thus, can be inappropriately treated in the development. Hence, understanding how ASD practitioners perceive the need for documentation is beneficial.

ASD practitioners are key stakeholders in the development and management of software. Understanding how practitioners perceive documentation is beneficial especially when considering QRs. This is important since the documentation of QRs have further economic implications, e.g. missing QR specification may incur documentation debt and increase maintenance costs [5], [12]. Additionally, understanding how practitioners perceive documentation of QRs would provide an insight into their approach to documentation of QRs in ASD. For instance, do practitioners identify documenting QRs in ASD important? If yes or no, what are their rationales? How do practitioners perceive factors that are reported in the scientific literature as influencing the documentation of QRs in ASD? Synthesizing empirical evidence on such aspects would give us an insight into whether documenting QRs in ASD is important or not, and as well improve our understanding of those factors claimed to affect documenting QRs in ASD.

[1] QRs, also referred as non-functional requirements (NFRs), describe the quality characteristics desired by a system to be developed such as reliability and maintainability [29].

Although there are studies investigating documentation or QRs in ASD [4], [13]–[16], there is not a study addressing practitioners' perception on the necessity of documenting QRs in ASD. In this paper, we provide empirical evidence on practitioners' perception on the needs of documenting QRs, and the factors that may affect documentation of QRs in ASD. We also examine practitioners' perception on the consequences of missing and outdated QR documentation in ASD. Therefore, our study addresses the following research questions:

RQ 1. How do practitioners perceive the importance of documenting QRs in ASD?

We investigate whether ASD practitioners consider documenting QRs important or not and synthesize their justifications on documenting or not documenting QRs.

RQ2. How do practitioners perceive factors that may influence documentation of QRs in ASD?

We collect practitioners' feedback on how they perceive factors that may influence documenting QRs in ASD. We collect their perception on how time constraints, QR awareness and communication gaps among team members affect documentation of QRs in ASD. These factors are reported to affect documentation and management of QRs in ASD [5].

RQ3. What are the consequences of missing or outdated QR documentation in ASD?

We collect evidence on the possible consequences of missing and outdated QR documentation in ASD.

We identified that ASD practitioners perceive documentation of QRs important to ensure quality, for clarifying QRs, to assist decision making and owing their influence on the implementation of other features. Time constraint affects documentation of QRs in ASD. ASD teams' may tailor their documentation practices to fit the sprint durations. QR awareness is also perceived to affect documentation of QRs in ASD. However, its effect may be dependent on the project context and role. Communication gaps among team members may result from absence of team members, misinterpretations on QRs and may create confusion on QRs in ASD. Accumulation of technical debt, the lack of common understanding on QRs, and wrong implementations are some of the consequences of missing and outdated QR documentation in ASD.

The rest of the paper is structured as follows. Section II presents related work on the documentation of QRs in ASD. Section III describes the research method applied in the study. Section IV presents the results of the study. In Section V, we discuss our findings and threats to validity. Finally, in Section VI, we conclude the paper.

II. Related work

ASD encourages minimal documentation as highlighted in one of its four core values, "*Working software over comprehensive documentation*" [8]. Although the minimal documentation enables a quick delivery of working software, and early return on investments, studies reveal some concerns too. For instance, the focus on minimal documentation is misinterpreted as there is no need for documentation [11], or as 'just enough' documentation although it is unclear what 'just enough' documentation entail [13]. Such interpretations may be detrimental to software, especially when considering QRs in ASD. This is due to the nature of QRs (e.g. QRs are hard to define and measure) [17], and their impact on both software quality and cost, which in turn affect the success of software projects [5], [12]. Additionally, scientific studies reveal limitations of ASD' artefacts and practices used for specification and documentation of QRs [5], [6], [18], [19]. Although there have been ASD studies investigating either documentation or QRs, there are no studies that explicitly study ASD practitioners' perception of the need for documenting QRs in ASD. Existing studies focused on understanding documentation practices in ASD [13][20], [16], understanding ASD practitioners' perception of QRs [14], or examining documentation debt [12]. Few works explored QR documentation practices in ASD [4][21].

Hoda et al. [13] examined the documentation strategies applied in ASD by interviewing practitioners of 23 organizations in New Zealand and India. They identified that ASD practitioners use electronic back-ups of paper artefacts, document change decisions made by customers, document business terminologies into project dictionaries to enhance requirements elicitation, and approach collaboration with non-agile teams with traditional documentation. On the other hand, Voigt et al. [16] investigated documentation practices in ASD, by employing theoretical model of information and documentation. The authors found that satisfaction with information searches is correlated with the level of documentation for most type of information. They also identified that documentation on architecture and design models was insufficient and recommended that there should be more methods, and tools to support agile documentation.

Stettina and Heijstek [14] studied practitioners' perceptions of documentation in ASD, through survey. Their finding revealed that more than half of the 79 respondents in their study identify documentation important. They also found that ASD teams adopted collaboration tools in their work (e.g. issue trackers, wikis), to support documentation. In another study [20] the authors explored the documentation practices and the effect of formalism in ASD applying experiment with students. They identified that iterative documentation practices resulted in a more detailed textual information and that writing documentation task was perceived as intrusive task, which was assigned to less skilled team members and later on affected the cooperation within the teams. On the other hand, Kopczyńska et al. [15], surveyed 118 ASD practitioners to find their perception regarding the importance of QRs. They found that about 77% of their respondents perceive defining QRs at least important and 30% found it critical. Although the study explored the practitioners' perceptions of the importance of QRs, it does not explore documentation of QRs in ASD in detail.

In a previous work [4], we examined the documentation practices and challenges in ASD and proposed guidelines for documenting QRs. We identified that ASD teams applied artefacts (e.g. user stories, epics and acceptance criteria), wikis and different backlogs to document QRs and focused on face-to-face communication in smaller teams. Lack of traceability, missing lower-level detail information of QRs and difficulty for documenting internally generated QRs were challenges of documenting QRs in ASD. Our recent work, [21] explored in depth the QR documentation practices in ASD. We found that QR documentation practices were dependent on the needs of project contexts and affected by the experience of practitioners. Practitioners identified the level of abstraction, the traceability of QRs, optimal detail of information of QRs, and verification and validation are important aspects to consider when documenting QRs in requirement management repositories.

Amorndettawin and Senivongse [22], proposed a non-functional requirements pattern template to enhance identification of QRs in ASD. They evaluated the template with ten practitioners of a scrum team, by assigning them to write security and fault tolerance requirements. They found that the templates helped practitioners to write requirements faster and specifying them more comprehensively when compared with not using the templates.

Understanding how practitioners perceive the need for documenting QRs in ASD will pave the way to understand the practitioners' motivation behind their decision on documenting or not documenting QRs in ASD. Additionally, it provides an insight into factors that may affect documentation of QRs in ASD.

III. Research method

A. Study design and data collection

We conducted a multiple case study of three cases. The companies providing the cases vary in terms of their sizes, geographical location, and product domains. We designed a protocol for the multiple case study based on Runeson and Höst al. [23] case study guideline and conducted semi-structured interviews with 12 ASD practitioners. We communicated our research objective and applied key informant technique [24] to suggest participants' role such as project manager and specification engineers, to the champions of the cases. The champions recruited the relevant participants for our study. The purpose of the interviews was to synthesize knowledge on practitioners' perceptions on documenting QRs in ASD and as well as their perceptions on factors which may influence documentation of QRs in ASD. The interviews were conducted between September and October 2019. Each interview was audio recorded and lasted between 25 and 35 minutes. The interview script is accessible through https://rb.gy/1bjfne. Table I shows summary of the cases and Table II shows the participants' background.

B. Data analysis

We applied thematic analysis [25] in order to analyze the collected data. First, the audio recordings of the interviews were transcribed through professional service. Then, the first author read the interview transcriptions and labeled excerpts describing concepts that answer our research questions into codes, in NVivo which is a qualitative analysis tool. For instance, excerpts describing the importance of documentation of QRs in ASD, in each of the cases were coded as 'Significance of documentation of QRs in ASD'. We followed similar approaches to label concepts on the consequences of missing or outdated QR documentation and factors affecting documentation of QRs in ASD. Secondly, within each of the cases, we compared the concepts gathered under each of the codes with each other, and those that are related or recurring were refined and grouped into a theme. There were also non-recurring concepts. This step resulted in themes and non-recurring labels answering our research questions within each of the cases. Then, we compared themes and labels identified from each of the cases with each other and grouped closely related and similar themes, refined them into the final lists of bigger themes.

TABLE I. Summary of cases

Case	Product domain	Company size in number of employees	Software development method	ASD adoption (years)
A	Modelling tool	Over 900	ASD	15
B	Telecommunication Embedded systems	Over 600	Scrum	14
C	Telecommunication	Over 100,000	Large-scale distributed ASD	12

TABLE II. Summary of Interview Participants' background

ID	Case	Role	Software engineering experience (Years)	ASD experience (years)
P1	A	Project manager	21	11
P2	A	Project manager	12	3
P3	A	Executive manager	31	14
P4	A	Software architect	12	12
P5	B	DevOps tech lead	17	15
P6	B	Process coach	16	7
P7	B	Build manager	14	4
P8	B	Project manager	10	10
P9	C	Requirements specification engineer	24	10
P10	C	Software architect team lead	20	10
P11	C	Requirements specification engineer	24	7
P12	C	Product architect lead	15	2

IV. Results

A. ASD practitioners' perception on documenting QRs (RQ1)

All of the interviewees agreed up on the importance of documenting QRs in ASD. However, it was pointed out that the level of documentation needed in ASD may vary depending on the QR type and project context. For instance, the executive manager from case A explained, "*Yes quality requirements are important to document. The quality requirements are inherently part of the requirements for projects. In some projects, if you have safety aspects, for example, if you make a project on autonomous cars, quality is utterly important. Or if you have just the utility that you use just twice a year, quality is not that much important*". We found that the practitioners perceive documenting QRs important for ensuring quality, clarifying QRs, process conformance, assisting decision making on QRs, and due to QRs' influence on implementation of other features. Ensuring quality and clarity of QRs were the top recurring themes identified in all three cases. Table III summarizes the practitioners' perception on the importance of documenting QRs in ASD.

TABLE III. SUMMARY OF ASD PRACTITIONERS' PERCEPTIONS ON THE IMPORTANCE OF DOCUMENTING QRS IN ASD

Justification for documenting QRs in ASD	**Case A**	**Case B**	**Case C**
Ensuring quality	X	X	X
Clarity of QRs	X	X	X
Process conformance	X	X	-
Help in decision making	-	X	X
Influence implementation of other features	X	-	X

1) Ensuring quality: practitioners in all cases perceived documenting QRs is important to determine the acceptable level of quality for the QRs and as a means of establishing the required quality targets. For instance, a response from the software architect in case A shows how QR documentation contributes to ensuring quality by supporting verification, *"It is important to verify the way we can answer quality issue, what are the tasks, the concrete tasks to do to answer the quality issue, I think. And to do that, documentation of QRs is required"*. Similarly the DevOps tech lead in case B reflects on how documenting QRs in ASD support ensuring quality as follows, *"I think this kind of correct syntax for providing some acceptance testing and stuff like this, to get some gatekeeping levels for that, what is acceptable quality, and then those kinds of requirements link together as a cluster, having a whole understanding that what quality requirements go together and why. So this kind of tools like Jira support that kind of thing that you can use the format Given-When-Then to create those acceptance testing and acceptance requirements that really show the acceptable quality usually, and the acceptable test cases"*. The requirements specification engineer from case C describes how documenting QRs help define the desired quality targets and guide related work *"But for us, who are working with these systems, for example, the capacity issues are pretty well documented. The key quality targets are defined and this guide our work. So I think that those are good to document"*.

2) Clarity of QRs: practitioners in all cases identify documenting QRs important to clarify the meaning and scope of the QRs. It was also explained that documentation is important since requirements change frequently and the QRs' definition may evolve through time. A project manager from case A explains importance of documentation for clarity as follows *"Well I think that it is important to document them. That always makes things clearer and, of course it will also help the same person who wrote that description"*. We also noticed that QRs may not always be clear to all stakeholders and that documenting them may help in clarifying them. The software architect team lead in case C stated, *"But unless those basic things like troubleshooting, software updates, resiliency, robustness, if they are not in place, no customer or product manager is going to upfront say that okay those are needed. But they are assumed to be there. So they need to be there properly framed. And when developing next releases of the products, unless those are documented well in the original development of the product, they are easily forgotten"*.

3) Process conformance: documentation of QRs was perceived important to ensure process conformance in cases A and B. The responses show that documenting QRs can help in determining whether practices and activities within a process are followed as recommended in the cases. For instance, a project manager from case A explained *"the idea is to always follow the same process and be sure that part of the development was not, or was taken into account by project manager for example. Quality is something which take a lot of time, and the fact that we formalize how you have to answer a quality issue forces you to take the time to resolve the issue"*. The DevOps tech lead in case B also pointed out how documentation can assist in conforming to specification process, *"I think that is the important thing, that people follow the same format and understand which of those requirements have been clustered together and why"*.

4) Help in decision making: two interviewees from cases B and C identified that documenting QRs may enhance decision making during elicitation, implementation and validation process. Properly defining and documenting QRs ensures what has been agreed upon the QRs among team members and thus facilitate decision making. According to them, unless QRs are documented, it is difficult to know what has been agreed up on previously regarding the QRs, as they can easily be forgotten.

5) QRs influence on the implementation of other features: two interviewees from cases A and C explained that QRs should be documented since they may affect the implementation of other features. For instance, the project manager from case A explained, *"When you get feedback from the testing team, the client, users or the tool itself, feedback needs to be related to some quality requirement. In our context, the feedback are so large, and can impact many features or functionalities, that is why we need to keep track by documenting the quality requirements"*.

B. ASD Practitioners' perception on factors that may influence documenting QRs in ASD (RQ2)

We asked the practitioners how they see factors that may influence the documentation of QRs in ASD. In particular, we collected their opinions on, whether time constraint and iteration cycles in ASD, QR awareness and communication gaps on QRs among development team members, may or may not influence documenting QRs in ASD. Table IV summarizes themes identified under the factors that may influence documentation of QRs in ASD.

1) The effect of time constraints and short iteration cycles on documenting QRs in ASD: All interviewees except one agreed that time constraints and short iteration cycles affect documentation of QRs in ASD.

a) Documentation of QRs left as the last thing to do: six *interviewees* from cases A, B and C reported that documentation of QRs is affected due to time constraints and short iteration cycles in ASD. When facing time pressure documentation becomes the first thing to compromise, QRs can be left as the last thing to do, not updated or documented and even forgotten. In such cases, ASD developers are focused on the implementation of features. For example, the executive manager from case A explained, *"Time constraint has always been a factor to have less documentation because* *it is the first thing you cut, if you do not have time. You do the code, you test and that is it'.*

b) *QRs were not defined and specified well and led to rework and additional iterations:* three interviewees from cases A and B identified that time constraints may result in poorly defined QR documents. They also noted that when QRs are not specified and documented properly, feature implementations may go in the wrong way and require additional iterations to improve the QRs and documentation.

c) Depends on sprint duration and the project: the DevOps tech lead from case B identified the sprint duration (e.g. 2 weeks, or a month) and the project context as an additional factor influencing the documentation of QRs in ASD. He described that achieving good documentation may be difficult in short sprint duration (e.g. two weeks) in some projects and that teams may apply additional practices to ensure documentation of QRs in ASD. The practices may be applying themes for short sprint cycles or including documentation and implementation in one long sprint duration as shown in his reply, *" there are themes that, in one sprint not everything is done usually. It is more about features, bug fixes, documentation, another feature sprint, another bug fixing sprint, another documentation sprint. Or if you have a longer sprint, then you can try to do all of those inside one sprint".*

d) Difficult to document when there are feature dependencies: the requirements specification engineer in case C described that in short iteration cycles documenting QRs that have dependencies on other features is difficult. He stated *"Of course there are those independencies between features. That is problematic. If we have new features, it's quite obvious that those will be documented so well and we have good background information already, so those are in better shape. But these features which are coming to the system, and those independencies could not be seen. From that, we have learned that it is problematic".*

e) Time constraints and short iterations do not affect documentation of QRs: in case C, the requirements specification engineer argued that QR documentation is not affected by time constraints. According to him, since there are multiple teams sharing the same goal and working together, time constraint does not influence documentation of QRs, *"I am still a little bit sceptic in that, even if we have short iterations and you get the feedback from other teams or customer or wherever, and you can fix it. But then when we have a lot of teams doing things to meet the same goal, and those teams' work cannot be isolated so well so, I think it does not minimize or decrease the need for documentation".*

2) The effect of QR awareness on documenting QRs in ASD: We asked the interviewees whether a limited or inadequate knowledge on QRs affects documentation of QRs in ASD or not. All the interviewees pointed out that QR awareness may affect documentation of QRs. We identified four themes on the effect of QR awareness on documentation of QRs in ASD.

a) QR awareness of practitioners and customers affect documentation of QRs: this was shared by nine interviewees of the three cases. The interviewees identified that knowledge on QRs by different stakeholders may affect the likelihood of documenting QRs in ASD. For instance, a project manager from case A described how practitioners' knowledge on QRs affects documentation in ASD, *"I would say because for me, when you design or when you think about your quality requirements, it is related to some particular issue, what is the main, the core, the important feature of your tool, what it is part of? If it is security, if it is code quality, and by thinking or be defining the quality requirement, you also think about what it is needed the documentation… if you do not think in terms of quality requirements, it will have an impact on the documentation".* The software architecture team lead from case C noted that QRs may not be obvious to junior developers, and indicated that a formal approach to QRs may help in documenting them,*"Unless you have that kind of, a checklist of the quality areas, the non-functional requirement domains to be documented and different views on the product behavior, they are not that easy to invent by yourself. Even though you could be inventing them yourself and figuring them out, that easily leads to very fragmented categorization of those non-functional requirements. So having a canonical way of looking at the non-functional requirement area helps with the documentation as well*". ASD practitioners' knowledge on QRs may not be enough alone. For example, the build manager from case B shared his experience on how inadequate QR knowledge by customers and the lack of documentation of QRs affected their work as follows, *"from the previous company I was doing sub-contracting and the customer had neglected the quality requirements and we struggled to get the needed information from them because they did not have even their own. It affected our work too, and caused delaying the deliveries because the quality requirements on their part were neglected. They did not have much knowledge on quality requirements and then the lack of the documentation to that, it affected our work also".*

b) Depends on the role: a project manager from case A and the process coach from case B noted that whether QR awareness affects documentation of QRs in ASD depends on the role. For instance, the project manager argued that roles such as product owners and project managers should care about and have QR knowledge since they need to decide whether quality level of a product is met, thus ensuring product readiness. On the contrary, the interviewee argued that roles such as developers may emphasize more on the development and may not care much about QRs and their documentation. The process coach from case B pointed out that poor specification and documentation of QRs carried out by managerial roles at higher levels may affect how developers at lower level document and implement QRs.

*c) Depends on the project context: a*ccording to the DevOps tech lead in case B, whether QR awareness affect documentation of QRs or not, depends on the project context. He described that certain projects may value functional requirements over QRs and treat QRs as an afterthought to be taken care of by a quality organization. On the contrary, other projects may value and be thoughtful about QRs and emphasize documenting QRs also.

d) Depends on the QR type: according to the requirements specification engineer from case C it is difficult to specify and document QRs such as capacity is in the early phases of development, because they may require rework and even hardware changes, *"We can set some target that we are reaching, we want to have 500 users at the same time. But then, at some point, when all functional requirements are* *implemented and they started to be tested there might be some surprises that we are not reaching what we set at early point as the capacity target. So it might be that we are getting only 450 for some reason. At that point it is very expensive to start to do it again, to all those functional requirements and maybe even hardware need to be updated that we reach that original 500. So it means that it is not so easy to define these non-functional requirements in early phases".*

3) The effect of communication gaps on QRs among team members on documenting QRs in ASD: We asked interviewees, whether communication gaps among ASD team members affect or do not affect documenting QRs. Except for two of the interviewees, all agreed that communication gap may affect documenting QRs in ASD.

a) Communication gaps affect the documentation of QRs: nine interviewees from all cases explained that communication gaps among team members regarding QRs can affect the documentation of QRs in ASD. The interviewees described members absence, misunderstandings and misinterpretation of QRs, not discussing QRs and missing QR documentation can create communication gaps and affect documentation. However, they identifed practices that help minimize the communication gaps. For instance, they recommended open discussion of QRs among team members in order to minimize misunderstandings of QR and improve documentation of QRs in ASD.

b) Communication gaps will not arise since QRs are specified and documented in early stages: Two interviewees from cases A and B presume that communication gaps from misunderstanding may not occur as QRs are agreed up on and specified at the beginning. For instance the project manager from case A explained, *"I cannot think that a misunderstanding on quality requirements can really happen. Because, you just set them from the beginning, so if you decide for example to have certain accessibility, because your clients have some particular needs, you just state that from the beginning, you define it".* Documenting QRs in early stages is assumed to facilitate communication of QRs, hence preventing QR communication gap among team members.

c) Communication gaps can create confusion: the DevOps tech lead in case B argued that if there are communication gaps among team members regarding QRs, it is likely that team members document QRs in their own way, which can create confusions on the tasks and unnecessary conflicts. He suggested that specifying and documenting QRs clearly in early stages may help to prevent such confusions.

C. *ASD practitioners' perception on the consequences of missing and outdated QR documentation (RQ3)*

We found five perceived consequences of missing and outdated QR documentation in ASD, summarized in Table V.

1) Technical debt accumulation: four interviewees from all cases explicitly stated that missing and outdated documentation of QRs in ASD may lead to accumulation of technical debt. Additionally, others reported system quality and performance degradation (two interviewees from cases B and C), increased development time (three interviewees from all cases), increased maintenance costs (two interviewees from B and C) and rework (one interviewee from case C) which are indicators of technical debt, as the consequences of missing and outdated documentation.

2) Practitioners may not know what the QRs cover, and will not have the understanding of the current behavior: three interviewees from cases A and B reported that when QRs are not documented it will not be clear to know what the QRs cover. The DevOps tech lead from case B, for instance, explained*, "For me, the first thing that comes to mind is that, I do not know when I am ready, I am not ready when I have done the features, I want to know how it has been used, is it working well? If I do not have any quality metrics and quality requirements to tow those metrics, I really don't know that. I have the functionality. I just have no understanding of the current behavior of the usage model, or is it actually useful or anything like that".* Additionally, it was noted that in such cases, practitioners may face frustrations and become demotivated, as they spend added time revisiting old features.

TABLE IV. SUMMARY OF FACTORS AFFE CTING DOCUMENTATION OF QRS IN ASD

Factors affecting QR documentation in ASD	**Themes**	**Case A**	**Case B**	**Case C**
The effect of time constraint and short iteration cycles	Documentation of QRs left as the last thing to do	X	X	X
	QRs not defined and specified well, and led to rework and additional iterations	X	X	-
	Depends on sprint duration and project	-	X	-
	Difficult to document when there are feature dependencies	-	-	X
	Time constraints and short iterations do not affect documentation of QRs	-	-	X
The effect of QR awareness on documenting QRs in ASD	QR awareness of practitioners and customers affect documentation of QRs	X	X	X
	Depends on the role	X	X	-
	Depends on the project context	-	X	-
	Depends on the QR type	-	-	X
Opinion on the effect of communication gap on documenting QRs in ASD	Communication gap affects documentation	X	X	X
	Communication gap will not arise since QRs are specifed and documented in early stages	X	X	-
	Can create confusion	-	X	-

3) Lack of common understanding of QRs: three interviewees from cases A and C reported that missing QR documentation may result in the lack of common understanding of QRs, and create confusion among practitioners. Different interpretations of QRs can further lead to causing frictions among team members. A project manager from case A indicated that ensuring the right level of documentation of QRs may help in avoiding the confusions on QRs that may happen from missing and outdated QR documentations.

4) Informal Quality management process: a project manager from case A noted that missing and outdated documentation of QRs may lead to informal quality management processes. Additionally, it can mean that quality is not at the center of development.

5) Wrong implementations leading to unhappy customers: the requirements specification engineer from case C revealed that not documenting QRs may result in wrong implementation and unhappy customer, "*the worst-case scenario is if those basic quality requirements are not documented. Then, they are not implemented into the product and the lack of some basic capability, recovery capability and so on is only found out after the product has been shipped to the field and then we have an unhappy customer*".

TABLE V. CONSEQUENCES OF MISSING OR OUTDATED QR DOCUMENTATION IN ASD

Consequence of missing or outdated QR documentation in ASD	**Case A**	**Case B**	**Case C**
Technical debt accumulation	X	X	X
Practitioners may not know what the QRs cover, and will not have understanding of current behavior	X	X	-
Lack of common understanding of QRs	X	-	X
Informal quality management process	X	-	-
Wrong implementation leading to unhappy customer	-	-	X

V. DISCUSSION

A. *The significance of documentation of QRs in ASD*

All practitioners in our study recognize the importance of documenting QRs in ASD. Stettina and Heijstek [13], reported that more than half of ASD practitioners in their study perceive documentation important although they did not specifically asked about QR documentation. Additionally, in our study, we found that project context and QR type affect the level of documentation of QRs required in ASD. For instance, the level of documentation needed for projects employing safety is more than those projects implementing basic utility software or a web app.

Documenting QRs was seen as a means of ensuring quality in all the three cases. By specifying QRs targets and acceptable level of quality, documenting QRs supports to ensure software quality. We also noticed documenting QRs is perceived to improve practitioners and customers' understanding of QRs. We found the result beneficial since the lack of QR awareness by ASD customers and practitioners are challenges in managing QRs [5].

QRs define desired quality properties of software and affect the implementation of interdependent features. ASD practitioners perceived that documenting QRs may help in keeping track of changes and the traceability of QRs. This is important since lack of traceability of QRs is a challenge in ASD [4][26]. Keeping the traceability of QRs was identified as one important aspect to consider while documenting QRs in ASD requirements management repositories [21]. We also observed that documentation can be used to enforce conformance to specification and documentation in ASD, thus ensuring QRs are properly documented and not neglected.

B. *Factors influencing documentation of QRs in ASD*

We observed that most practitioners may compromise documentation of QRs due to time constraint, and that QRs may be underspecified, consequently leading to rework in later phases. Sprint duration influenced the documentation of QRs. ASD teams may tailor their software development process to meet QR documentation needs of projects depending on the sprint duration (e.g. assigning a sprint dedicated to documentation when the sprint duration is short).

In large-scale distributed ASD setting, where there are multiple teams, and each team is responsible for its own task time constraint may be perceived as not affecting documentation of QRs. Assigning clear responsibilities for QR tasks have been reported as a practice to address QR challenges [5]. Perhaps, the clarity on the tasks and responsibilities may have helped minimize the influence of time constraints on documenting QRs in the specified large-scale distributed ASD setting.

Whether QR awareness influences documenting QRs or not may be dependent on the project context. Some projects value QRs and instill activities in their development process that ensure QRs are documented and managed properly. We also noticed that in ASD, QR awareness of both practitioners and customers affect documentation. The lack of QR awareness by customers is a challenge in ASD [5], [27]. When considering practitioners, QR awareness may be seen a necessity for roles such as project managers and product owners and deemed less important for developers in some cases. Additionally, junior developers may not have sufficient level of QR awareness, and formal documentation practices were recommended to improve their understanding of QRs. We recognized that in embedded system and telecommunications domain, where software is linked with hardware implementations, upfront knowledge on QRs and specification may not be easy.

Most of the practitioners considered that communication gaps affect documenting QRs in ASD. In some cases, QR specifications done in early phases are presumed to mean that there will not be communication gaps on QRs. However, as QRs evolve through time in the development process, there is always a probability for communication gaps. In this regard, documentation of QRs should be continuously updated. We also noticed that open discussion on QRs is encouraged to minimize communication gaps on QRs among team members.

C. *Consequence of missing or outdated QR documentation*

Accumulation of technical debt, with increased development and maintenance time and system quality degradation was mainly reported as the consequence of missing and outdated QR documentation. Missing and outdated QR documentation make QRs and their tasks unclear. The extra time spent on clarifying the QRs may demotivate ASD practitioners. Moreover, it may also create friction among the practitioners, as there will not be common understanding on the QRs. Imprecise QR specifications have led to misinterpretations [28]. When QRs are not documented and managed properly, software implementation may not meet customers' expectation and result in bad relationships.

D. *Threats to validity*

Construct validity: we communicated the research objective of our study and suggested potential participant roles using key informant technique, to the champions of the cases. These enabled us to collect relevant information from ASD practitioners. We minimized threats from misinterpretation of concepts and interview questions, by describing the research objectives and clarifying concepts and interview questions to the participants. For instance, we clarified that we treat QRs and non-functional requirements as equivalents.

External validity: to improve the generalizability of our study, we synthesized evidence from three cases of varying ASD contexts and by involving participants with different roles and experiences. Although, our study is based on 12 interviews, we believe that the findings can partly be generalized to similar contexts.

Reliability: we applied a protocol guiding our interview and collected data systematically, by recording audio of the interviews to increase the reliability of the study.

VI. Conclusions

The paper explored practitioners' perception on documenting QRs in ASD. We found that practitioners identify documenting QRs in ASD important for ensuring quality, clarifying QRs and their tasks, enforcing process conformance on QRs, enhancing decision making and due to QRs' influence on the implementation of other features. Time constraints, QR awareness and communication gaps on QRs are perceived to affect documentation of QRs in ASD. ASD teams may tailor documentation practices according to the sprint duration. QR awareness is deemed important for project managers and product owners than developers. Open discussion on QRs was recommended to minimize communication gaps on QRs. Missing and outdated QR documentation may lead to incurring technical debt, lack of common understanding on QRs, informal QR management and wrong implementations.

Our study contributes both to the software industry and software engineering research. Software practitioners (e.g. project managers, product owners, and developers) can learn about the significance of documenting QRs in ASD, and get informed that documenting QRs can contribute to ensuring software quality, and facilitating software development by clarifying QR tasks and supporting decision making in ASD. They can also benefit from knowledge on factors influencing the documentation of QRs (QR awareness of practitioners and customers, QR types and project context), and the consequences of outdated QR documentation (e.g. lack of common understanding on QRs). For the Software engineering research, our study provides empirical evidence on documentation of QRs in ASD. Researchers may utilize the findings to get an insight into the research area. As future work, we aim to extend our work to provide recommendations for optimal documentation of QRs in ASD.

Acknowledgment

This study is partially funded by the Q-Rapids project, European Union's Horizon 2020 research and innovation funded program under grant agreement N° 732253.

References

[1] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula, "Survey on agile and lean usage in finnish software industry," in International Symposium on Empirical Software Engineering and Measurement, 2012, pp. 139–148.

[2] W. Alsaqaf, M. Daneva, and R. Wieringa, "Understanding challenging situations in agile quality requirements engineering and their solution strategies: Insights from a case study," in Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018, 2018, pp. 274–285.

[3] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects - A systematic literature review," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10153 LNCS, 2017, pp. 219–234.

[4] W. Behutiye, P. Karhapää, D. Costal, M. Oivo, and X. Franch, "Non-functional requirements documentation in agile software development: Challenges and solution proposal," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10611 LNCS, Cham, 2017, pp. 515–522.

[5] W. Behutiye et al., "Management of quality requirements in agile and rapid software development: A systematic mapping study," Information and Software Technology, vol. 123, p. 106225, Nov. 2020.

[6] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," Information and Software Technology, vol. 110, pp. 39–55, Jun. 2019.

[7] L. Chung and B. A. Nixon, "Dealing with non-functional requirements," in Proceedings of the 17th international conference on Software engineering - ICSE '95, 1995, pp. 25–37.

[8] K. Beck et al., "Agile Manifesto," Software Development, 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed: 24-Mar-2020].

[9] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," IEEE Software, vol. 25, no. 1, pp. 60–67, 2008.

[10] A. Sidky and J. Arthur, "Determining the applicability of agile practices to mission and life-critical systems," in Proceedings - International Conference on Software Engineering, 2007, pp. 3–12.

[11] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," Journal of Systems and Software. 2012.

[12] T. S. Mendes, M. A. F. De Farias, M. Mendonça, H. F. Soares, M. Kalinowski, and R. O. Spínola, "Impacts of agile requirements documentation debt on software projects: A retrospective study," in Proceedings of the ACM Symposium on Applied Computing, 2016, vol. 04-08-Apri, pp. 1290–1295.

[13] R. Hoda, J. Noble, and S. Marshall, "Documentation strategies on agile software development projects," International Journal of Agile and Extreme Software Development, vol. 1, no. 1, p. 23, 2012.

[14] C. J. Stettina and W. Heijstek, "Necessary and Neglected ? An Empirical Study of Internal Documentation in Agile Software Development Teams," in 29th ACM International Conference on Design of Communication (SIGDOC 2011), 2011, pp. 159–166.

[15] S. Kopczyńska, M. Ochodek, and J. Nawrocki, "On Importance of Non-functional Requirements in Agile Software Projects—A Survey," in Studies in Computational Intelligence, 2020.

[16] S. Voigt, J. Von Garrel, J. Müller, and D. Wirth, "A Study of Documentation in Agile Software Projects," in International Symposium on Empirical Software Engineering and Measurement, 2016.

[17] B. Paech and D. Kerlow, "Non-Functional Requirements Engineering - Quality is essential," Proceedings of the 10th Anniversary International Workshop on Require- ments Engineering: Foundation of Software Quality (REFSQ'04), pp. 237–250, 2004.

[18] A. Martakis and M. Daneva, "Handling requirements dependencies in agile projects: A focus group with agile software development practitioners," in Proceedings - International Conference on Research Challenges in Information Science, 2013.

[19] G. Erdogan, P. H. Meland, and D. Mathieson, "Security testing in agile web application development - A case study using the EAST methodology," in Lecture Notes in Business Information Processing, vol. 48 LNBIP, 2010, pp. 14–27.

[20] C. J. Stettina, W. Heijstek, and T. E. Fægri, "Documentation work in agile teams: The role of documentation formalism in achieving a sustainable practice," in Proceedings - 2012 Agile Conference, Agile 2012, 2012.

[21] W. Behutiye, P. Seppänen, P. Rodríguez, and M. Oivo, "Documentation of Quality Requirements in Agile Software Development," in Proceedings of the Evaluation and Assessment in Software Engineering, 2020, pp. 250–259.

[22] M. Amorndettawin and T. Senivongse, "Non-functional Requirement Patterns for Agile Software Development," in Proceedings of the 2019 3rd International Conference on Software and e-Business, 2019, pp. 66–74.

[23] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," Empirical Software Engineering, vol. 14, no. 2, pp. 131–164, Apr. 2009.

[24] M. N. Marshall, "The key informant technique," Family Practice, vol. 13, no. 1, pp. 92–97, 1996.

[25] D. S. Cruzes and T. Dybå, “Recommended steps for thematic synthesis in software engineering,” in International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 275–284.

[26] D. Baca, M. Boldt, B. Carlsson, and A. Jacobsson, “A novel security-enhanced agile software development process applied in an industrial setting,” in Proceedings - 10th International Conference on Availability, Reliability and Security, ARES 2015, 2015.

[27] A. Tetmeyer, D. Hein, and H. Saiedian, “A tagging approach to extract security requirements in non-traditional software development processes,” International Journal of Secure Software Engineering (IJSSE), vol. 5, no. 4, pp. 31–47, 2014.

[28] C. W. Ho, M. J. Johnson, L. Williams, and E. M. Maximilien, “On Agile performance requirements specification and testing,” in Proceedings - AGILE Conference, 2006, 2006, vol. 2006, pp. 47–52.

[29] K. Wiegers and J. Beatty, “First things first: Setting requirement priorities,” in Software Requirements,Microsoft Press,Redmond, Washington, 2013.