Agile Quality Requirements Elaboration: A Proposal and Evaluation

Wasim Alsaqaf[a], Maya Daneva[b] and Roel Wieringa
School of Computer Science, University of Twente, Enschede, The Netherlands
{w.h.a.alsaqaf, m.daneva, r.j.wieringa}@ utwente.nl

Keywords: Agile Projects, Quality Requirements, non-Functional Requirements, Large-Scale Distributed Projects, Requirements Engineering, Goal-Oriented Modelling, Design Science, Focus Group Study.

Abstract: The increasing success and user satisfaction of agile methods' application in their original context (e.g. small co-located teams), motivated large organizations to utilize agile methods to deal with the rapidly changing markets and the distributed global workforce. Several studies have reported a variety of quality requirements (QRs) challenges in large-scale distributed agile (LSDA) context, so a recent empirical study has identified 15 QRs challenges in LSDA projects. This paper proposes an approach based on the concept of goal documentation to deal with the 15 QRs challenges reported previously. Our proposal, the Agile Quality Requirements Elaboration (AQRE) approach, introduces a new organizational role and a two-step process to elaborate high-level goals(s) into epics and user stories alongside QRs. The fitness and the usefulness of AQRE are evaluated by using a focus group with eight practitioners in the IT department of a large Dutch government organization. The evaluation indicated that 12 of the 15 QRs challenges could be mitigated by the AQRE. Our main contribution is two-fold, (i) we proposed a solution approach to deal with QRs challenges in LSDA context, and (ii) our evaluation provided empirical evidence about its usefulness in real-world context.

1 INTRODUCTION

Agile software development methods have become increasingly popular in the last decade. In 2021, the 15th annual State of Agile report (Digital.ai, 2021) indicated the explosive increase of agile adoption. This is in contrast to the traditional non-agile approaches of software development which are characterized as heavyweight rigid processes (De Lucia and Qusef, 2010). Due to their rigidity, traditional non-agile approaches fail to keep up with the rapidly changing business environments (Helmy et al, 2012). Agile on the other hand presents itself as the way to deal with rapidly changing markets. In fact, responding to change is one of the values stated in the Agile Manifesto (Agile Alliance, 2001). Based on these values, agile practices have been developed to increase the involvement of the customers and to deliver software products faster. The agile practices include short iterations that end up with delivering a new incremental release, addressing as many requirements as needed to start an iteration, simple incremental designs that will be evolved during the iterations and peer reviews (Shore and Shane, 2007). The necessity of reacting quickly to the rapidly changing market, pushes large organizations to believe that the success stories of agile methods' application in the context they originally were designed for (e.g. small co-located teams) are repeatable in large-scale distributed agile (LSDA) context. The 15th annual State of Agile report (Digital.ai, 2021) states that the rapid changing economical market and the distributed global workforce do force organizations to take on their distributed agile chances. In line with this, software engineering scholars (Calefato and Ebert, 2019; Smart, 2018) describe two trends which are demanded in large-scale projects, namely, agile transformation and embracing team distribution. The application of agile methods – which were originally designed for the context of small and co-located teams – in LSDA however does not go smoothly (Smart, 2018; Conboy and Carroll, 2019). In particular, many studies have reported the neglect of quality requirements (QRs) in agile context (Maiti

[a] https://orcid.org/0000-0002-0253-428X
[b] https://orcid.org/0000-0001-7359-8013

Alsaqaf, W., Daneva, M. and Wieringa, R.
Agile Quality Requirements Elaboration: A Proposal and Evaluation.
DOI: 10.5220/0012143100003538
In *Proceedings of the 18th International Conference on Software Technologies (ICSOFT 2023)*, pages 668-679
ISBN: 978-989-758-665-1; ISSN: 2184-2833

and Mitropoulos, 2015; Alsaqaf, 2019; Inayat et al, 2014). In a recent empirical study (Alsaqaf et al, 2019) the authors have identified 15 challenges that LSDA projects cope with when it comes to the engineering of QRs. Given this background, in the present paper we propose and evaluate an approach to engineer QRs in LSDA where the identified challenges (Alsaqaf et al, 2019) could be mitigated. Our proposal was created by using a process grounded on Wieringa's Design Science (2014) and drawing on concepts from goal-oriented requirements engineering (GORE) (Pohl, 2010). Our very first evaluation with practitioners indicates that the approach may achieve its goal and could be useful. In what follows, we first present background and related work (Section 2). In Section 3 and 4, we introduce our proposed approach and its very first evaluation respectively. A discussion of the findings is in Section 5. We reflect on the limitations of our approach in Section 6 and conclude in Section 7.

2 BACKGROUND AND RELATED WORK

Three streams of research form the related work for this paper: (i) publications on challenges and possible solutions in the engineering of QRs in agile context; (ii) publications on the use of GORE, and (iii) publications on requirements modelling.

2.1 QRs Engineering in Agile

As already indicated, numerous empirical studies revealed the neglect of QRs in agile context (Maiti and Mitropoulos, 2015; Alsaqaf et al, 2019; Inayat et al, 2014). A recent one (Alsaqaf et al, 2019) has identified 15 challenges of five categories that LSDA cope with. For clarity, we list these QRs challenges in Table 1, where the first column shows the categories of the challenges as reported in (Alsaqaf et al, 2019) and the second column reports the specific challenges of each respective category in the first column.

These authors (Alsaqaf et al, 2019) have reported nine practices that agile practitioners use to cope with those challenges. However, the study indicated as well that the used practices could incur other challenges such as adding hierarchies to the agile teams which would take the flexibility of agile a step back towards waterfall.

Table 1: The reported QR challenges.

Category	Challenges
1.Teams coordination and communication challenges	-Late detection of QRs infeasibility -Hidden assumptions in inter-team collaboration -Uneven teams maturity -Suboptimal inter-team organization
2. Quality assurance challenges	-Inadequate QRs test specification -Lack of cost-effective real integration test -Lengthy QRs acceptance checklist -Sporadic adherence to quality guidelines
3.QRs elicitation challenges	-Overlooking sources of QRs -Lack of QRs visibility -Ambiguous QRs communication process
4.Conceptual challenges of QRs	-Unclear conceptual definition of QRs -Confusion about QRs specification approaches
5. Architecture challenges	-Unmanaged architecture changes -Misunderstanding the architecture drivers

A 2017 systematic literature review (Alsaqaf et al, 2017) surveyed an array of proposed solutions to counter the neglect of QRs in agile context (e.g. (Kumar et al, 2013; Domah and Mitropoulos, 2015; Farid and Mitropoulos, 2012; Farid and Mitropoulos, 2013)). Those proposals introduced several new artefacts and roles to include in the engineering of QRs in agile processes. A 2022 study (Rahy and Baas, 2022) introduced as well two new artefacts e.g. Documentation Work Item and Safety Critical Work Item to treat QRs in an agile way. However, creating new artefacts or new roles and integrating them into the existing agile processes could be experienced by practitioners as 'heavyweight' additions that unexpectedly cause new problems (Alsaqaf et al, 2019). In the same vein, the review of Abheeshta et al. (Abheeshta et al, 2018) on the adoption of heavy-documented agile frameworks reported the "moving away from agile" as an important challenge among others. Another recent study (Sherif et al, 2022) explored published solutions to cope with QRs in agile context (e.g. (Maiti and Mitropoulos, 2015; Jeon et al, 2011)). However, the study (Sherif et al, 2022) also revealed the lack of empirical evidence that these proposed solutions work, which agrees with the

observation made by Inayat et al. (Inayat et al, 2015) indicating a lack of evidence about the effectiveness of new proposed solutions. Finally, Kopczyńska et al. (Kopczyńska and Nawrocki, 2014) described the SENoR method as response to the neglect of QRs in agile. SENoR includes several brainstorm sessions where QRs are elicited based on ISO25010 (ISO/IEC, 2011) and documented using predefined QRs templates. However, Sherif et al. (Sherif et al, 2022) pointed out that the narrow focus of SENoR uniquely on QRs in isolation of other requirements, is actually a weakness of this method.

2.2 Goal-Oriented Requirements Engineering

Requirements engineering (RE) is the process of elaborating stakeholders' intentions into specifications of the desired system or services (Lamsweerde et al, 1998). Therefore RE needs to understand those intentions that have to be fulfilled by the desired system of services. In RE, those stakeholders' intentions are referred to as *goals* (Pohl, 2010). Using goals to drive requirements has been popular (Horkoff, 2019) due to, among others, the following benefits (Pohl, 2010): (i) clarifying the context and the value of the system, (ii) driving and guiding the identifying of the system requirements, since for each goal a set of requirements can be defined, (iii) giving the possibility for identifying and evaluating solution alternatives, (iv) giving guidelines to identify irrelevant requirements, (v) giving guidelines for requirements completeness, (vi) giving rational for the relevance of requirements, (vii) giving guidelines for identifying and resolving requirements' conflicts and (viii) giving stability since goals are not subject for frequent changes while requirements are. In alignment with Pohl (2010), Daneva et al. (2007) highlighted the importance of assessing stakeholders' goals early in de system development phase to achieve a clear scope definition which would guide the identification of the most significant requirements. Whether these benefits have been observed in LSDA projects and to what extent Goal-Oriented Requirements Engineering (GORE) adds value in that context has not been empirically researched in much depth. However, one might think of the potential usefulness of GORE in LSDA, from the following perspective: in their work on QRs challenges in LSDA projects, Alsaqaf et al. (2019) reported several common root errors behind these challenges: (i) suboptimal priorities assigned to conflicted QRs, (ii) focusing too much on system's parts and losing the big picture, (iii) the emerging of relevant QRs late in the development phase. The authors (Alsaqaf et al, 2019) acknowledge that while these mechanisms were observed in LSDA projects, they are not unique to agile. As GORE was introduced to cope with such situations in 'traditional contexts', we thought that there would be no reason to assume that GORE would not work for agile projects. In fact, we believe that implementing GORE concepts could eliminate several of the mechanisms reported in (Alsaqaf et al, 2019), which in turn means that it can serve to mitigate the challenges in Table 1.

In GORE, goals can be of different levels of abstraction; for example, Cockburn (2000) reported the following levels: (i) Cloud level – a high-level business goal that need to be decomposed into sub-goals, (ii) Kite level – a decomposed goal from the Cloud level that represents an end-to-end system process, (ii) Sea level – a user goal decomposed from the Kite level that can be achieved by one person within the end-to-end system process, and (iv) Fish level – a task (not a goal by itself) carried out along with other tasks to achieve a user goal. Agile software development (ASD) however does not use these abstraction levels of goals. Instead, ASD uses the terms *themes, initiatives, epics and user stories*. Noreika et al. (2021) defined those agile terms as follows: (i) Theme is a logical organization and aggregation of related user stories to show they have something in common and is managed by business representatives in a project. No software development activities are required to achieve themes. (ii) Initiative is a composition of epics that drive toward a common business goal which should not span more than one year. Initiatives provide also the needed context to help companies make decisions regarding the course of direction. Software development activities are required to achieve an initiative. (iii) Epic is a set of related user stories that need no longer than a quarter to be completed. To achieve epics, software development activities are needed as well. (iv) User Story is the lowest level of granularity that means work to be completed within one to four weeks. Similarly to epics and initiatives, user stories need software development activities to be implemented. Based on the aforementioned description, throughout this paper, we treat agile initiatives as high level business goals (e.g. Cloud), epics − as sub-goals (e.g. Kite) and user stories − as user goals (e.g. Sea).

2.3 Modelling Requirements

Models have been used widely in software development. They provide an abstract representation of a particular complex problem to simplify the

process of understanding the problem (Muller et al, 2012; Pastor et al, 2022). In more detail, Muller et al. (2012) and Pastor et al. (2022) explained modelling as a simplification of a system to be built with an intended goal in mind and an abstraction of a relevant part while ignoring irrelevant aspects. Moreover, Girvan and Paul (2017) reported the following benefits of using models, namely: models provide (i) an effective manner for discussion and collaboration, and (ii) an effective medium for communications. These benefits are in line with the agile way of working where individuals' interaction and customer collaboration are highly valued (Agile Alliance, 2001). The RE literature, e.g. (Horkoff, 2019), introduced many GORE frameworks where goals are used to identify significant requirements and are, in turn, modelled. Specifically, the i* framework (Yu, 1997) has been recognized as one of the most popular to model goals (Horkoff, 2019). Despite its broad use, the i* framework was experienced as not so easy to learn which hampered its spreading outside of the originating community (Dalpiaz et al, 2016). This experience forced the GORE community to come up with a response, which was in the form of the iStar 2.0 goal-based requirements modelling language (Dalpiaz et al, 2016). Throughout this paper, we will use iStar to refer the iStar 2.0 as described in (Dalpiaz et al, 2016). iStar is designed to provide means to model (i) different types of actors and their boundaries, (ii) independent elements e.g. goals, qualities, tasks and resources, and (iii) different types of relationships (Dalpiaz et al, 2016). iStar has an open-source online goal modelling tool[c], which we use throughout this paper as a documentation tool to explain our proposed approach. Our choice for the online tool of iStar was driven by the free of charge availability of the tool. Of course other documentation tools such as Microsoft Visio could be used for the purpose as well.

3 THE AQRE APPROACH

3.1 Our Process to Construct the AQRE Approach

For the purpose of this work, we followed a research process grounded on the design science guidelines of Wieringa (2014). We chose this approach as it is suitable for developing artefacts (i.e. methods) to solve problems in practical contexts. The design science inspired process (Wieringa, 2014) starts with formulating an objective that is expected to be achieved with the construction of the artefact (in our case, the proposal for an approach to engineering QRs in LSDA projects).

The overall objective of our proposed AQRE approach, is to help agile teams to deal with the QRs challenges reported in (Alsaqaf et al, 2019). For our approach to work, it needs to be embedded into the larger software development process of an organization (Pfleeger and Atlee, 2009). In our research context, this is the process of delivering a software system in a LSDA project. In line with this, we start on the premise that our proposed AQRE approach should be executed *before* the start of the first sprint to help create the initial version of the product backlog of the project and then enable the start of this first sprint. In a nutshell, our proposal, AQRE, represents a workshop session where participants discuss and elaborate software development initiatives into epics and user stories by using a goal-based requirements modelling language such as iStar. The AQRE approach consists of one mandatory role and two steps which are described in the next subsections.

3.2 The AQRE Role

Drawing on the industrial practice of McKinsey (Bucy et al, 2017) and other large companies (Sutherland et al, 2022), AQRE introduces the organizational concept of Initiative Owner (IO). While this term has been used in industrial experience reports (Bucy et al, 2017; Sutherland et al, 2022), to the best of our knowledge the term IO has not been elaborated in sufficient depth in scientific studies. For example, although Bucy et al. (2017) refer to this term, these authors did not provide a clear description of what they mean with it. Besides, Bucy et al. (2017) use the term IO in the context of organization's transition in general, and not related to agile in specific. Furthermore, Sutherland et al. (2022) describe the role of IO as synonym to the role of Product Owner (PO). To avoid any confusion, in AQRE we use the term IO to refer to the one who is responsible for achieving the initiative (e.g. the high-level goal(s)) of the organization and coordinating the activities needed to implement the initiative in question. This initiative will be decomposed into epic(s) and user stories and assigned to one or more agile teams to be implemented. In case of multiple agile teams, each would have its own PO. Each PO is then responsible for coordinating the work of his/her

[c] https://www.cin.ufpe.br/~jhcp/pistar/

own team based on the customer's values his/her team wants or deliver.

3.3 The Two AQRE Steps

Our proposed approach includes preparation of the AQRE workshop and its execution. Below, we describe the steps of AQRE in terms of activities that the AQRE participants would go through.

3.3.1 Preparation

The IO begins the initiative by providing a short description. The IO determines further who will be invited to the AQRE workshop (e.g. step two below) to discuss and elaborate the initiative. We note that the workshop participants should be chosen based on the needed knowledge and not based on their role in the organization. The types of needed knowledge are: (i) domain knowledge, (ii) QRs knowledge, (iii) enterprise architecture knowledge, (iv) infrastructure and maintenance knowledge. Based on the nature of the initiative, other particular types of knowledge could be needed (e.g. security, regulations and compliance, usability). In that case, the IO invites the people with that particular knowledge as well. Besides, the IO also ensures that the workshop's participants have sufficient knowledge of goals modelling techniques. If a participant has no prior exposure to these techniques, then the IO organizes a training session as part of the preparation step to educate the participants on goal documentation by means of models.

3.3.2 Workshop

The IO starts the workshop by giving background information such as organization's vision and goals, the initiative's description and how it fits within the organization's vision and goals. Hereafter, the participants start decomposing the initiative using the AND/OR goals modelling and decomposition technique described in (Pohl, 2010) and a modelling tool to visualize the models. Prior to the workshop, the IO assured that the participants understand the goals decomposition technique (see step one on the previous page). The objective of this process is to break down the initiative into smaller goals (e.g. matching the possible epics and user stories) based on the expected value of the customer (e.g. the who), defining possible alternative (sub)goals or solution's directions, defining QRs associated with the identified (sub)goals, identifying and resolving conflicts between (sub)goals in general, and QRs in specific, and defining and agreeing upon the scope of the initiative. The workshop could take hour(s) or day(s), depending on how complex the initiative is. Further, the workshop resulted in a model of decomposed (sub)goals similar to the one depicted in Figure 1. The IO is responsible for creating the resulting model. However, she could ask a participant to draw the model or she can draw it all by herself. Since our study focuses on the mitigation of the reported QRs challenges in (Alsaqaf et al, 2019), in the next subsection we elaborate further on that subject. We note that decomposing an initiative may result in subgoals (e.g. epics) that are big and vague to be technically implemented within the next sprint. In that case the AQRE workshop could be conducted again to elaborate each subgoal whenever it is needed.

3.3.3 QRs Elaboration

During the process of breaking down (sub)goals, the participants have to identify those QRs associated with the identified (sub)goals. The identified QRs can be further broken down into other related QRs. For example, a security quality attribute could be decomposed into e.g. Confidentiality, Integrity and Authenticity. Performance, for example, could be broken down into e.g. Capacity and Resource Utilization. To guide this process optimally we advise the participants to use a QRs framework of their choice, e.g. the ISO 25010 quality standards (ISO/IEC, 2011), the NFR framework (Chung et al, 2000), or the Sustainable Catalogue (Albuquerque et al, 2021). After identifying and elaborating the QRs, the participants have to provide those QRs with enough details in order to enable the development teams to make the right design decisions (concerning these QRs) at the right time. The details should at least include: (i) the estimated impact of (fully) having or (partially) missing the QR on the related (sub)goal, and (ii) broad specification of the QRs. This means that the participants should provide the development team with just enough QR's boundaries to be able to implement and test the right QRs correctly (Lauesen, 2002). For example, if the performance of collecting user data after logging in the system is an important QR, we can then specify what is an unacceptable performance like "user data should be collected and shown on screen in no longer than 7 seconds". Instead of "user data should be collected and shown on screen in 5 seconds". The last option will limit the decisions which the development teams can make, while the first option will give the development teams space to make their consideration, especially if there are conflicting performance and

usability requirements, e.g. the amount of data that should be shown after logging in the system.

4 OUR FIRST EVALUATION OF THE AQRE APPROACH

As stated earlier, the AQRE approach was developed to help mitigate QRs challenges experienced by agile practitioners (Alsaqaf et al, 2019). Therefore, to evaluate our proposed approach, we chose to design and conduct a focus group (FG) study (Krueger and Casey, 2014) to collect and analyze the opinions and perceptions of experts regarding the AQRE approach. We chose the FG-based evaluation strategy as recommended in (Wieringa, 2014; Lauesen, 2002). We chose this strategy, because (i) design science relies on feedback sessions from practitioners in the field where the proposed artefact is supposed to be used, and (ii) the FG approach lends itself naturally for the context of our evaluation efforts.

Our evaluation was set out to address two evaluation research questions (RQs): *RQ1. Does the goal model created by means of the AQRE workshop fit the project initiation context, according to the practitioners working in this context?* and *RQ2. In what way is this model useful, according to the practitioners in the FG study?*

As preparation for the FG, the first author developed a research protocol which was discussed and improved in collaboration with the second author. The final FG-based study consisted of the following two steps:

(i) Conducting a AQRE workshop within a project organization to provide a goal decomposition model of a real-world initiative,

(ii) Conducting the FG session itself with participants from the same organization (8 participants) to collect their opinions of the resulted goal decomposition model and their perceptions of the QRs model obtained by using AQRE.

Following FG research methodologists (Krueger and Casey, 2014; Morgan, 1996) we think that collecting experts' opinions based on an initiative from their own domain would provide us with a rich and meaningful feedback which would help us to improve the AQRE approach and use its improved version in further evaluation studies. We note that as this is our first evaluation, eight practitioners form a big enough focus group for the purpose of very early feedback-giving. Also, as per FG methodologists (Krueger and Casey, 2014; Morgan, 1996), the overall idea is to generate evaluative feedback from a variety of practitioners' perspectives and not strive for statistical analysis of the responses or for achieving consensus among the participants. To recruit FG participants, the first author used his professional network to first find a suitable organization which was willing to provide us with a real-world initiative and then to host the AQRE workshop and also to spend time in our FG-study afterwards. The AQRE workshop as well as the FG session were conducted in Dutch and they were performed at the IT department of a big government organization in the Netherlands.

The IT department has more than 300 employees and provides European business entities with the needed tools and software to communicate business-related issues with the Dutch government agencies as well as with other European business entities. The IT department of the host organization provided us with an initiative that had to be elaborated. The responsible employee for the initiative is a senior business analyst with more than eight years of professional experience in this job. In AQRE terminology, this practitioner will be the IO.

4.1 Using the AQRE Workshop to Create a Goal Model

For the first step of our evaluation, the first author and the IO hold an AQRE workshop to elaborate the initiative. The PO of the potential agile team(s) that would implement the initiative has also attended the AQRE workshop. The participating PO has about fourteen years of experience as information manager and PO.

The AQRE workshop lasted about two and half hours and resulted in a goal decomposition model created by iStar (Dalpiaz et al, 2016) (Figure 1). In the first 30 minutes of the workshop, the iStar tool was introduced to the IO and the PO. Thereafter the actual goal modeling started. The IO as well as the PO were already familiar with goal decomposition technique.

4.2 Our FG: Data Collection and Analysis Process

As already mentioned, we chose for a FG-based evaluation strategy (Wieringa, 2014) to assess the fitness and the usefulness of the model resulting out of the application of the AQRE approach. Therefore, we set up a FG with experts to collect feedback on the deployment of the AQRE approach. As the AQRE is to be deployed on high-level goals, we considered for

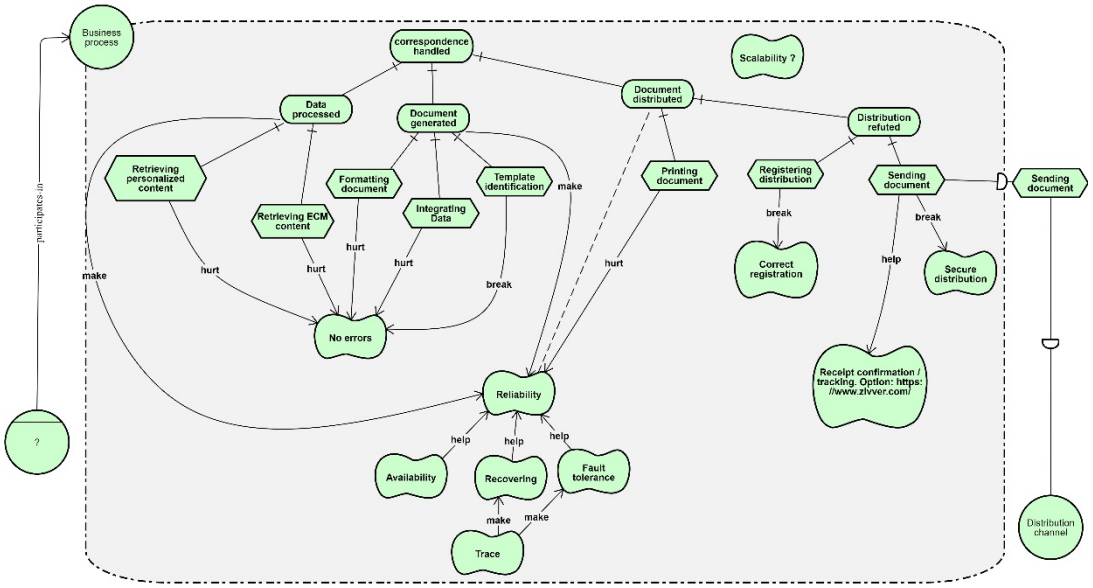

Figure 1: Goal decomposition by iStar.

inclusion practitioners possessing knowledge in one or more of the following: domain expertise, architecture knowledge and requirements knowledge. We approached relevant contacts of the host organization with those types of knowledges to our FG-study. Eventually eight experts accepted the invitation and attended the FG. We note that the participant of the workshop in Sec. 4.1 were not part of the FG, i.e. the FG-based evaluation includes completely different people. Table 2 presents the profiles of or FG participants. The first column describes their roles in the IT organization and the second describes the number of participants for each role. The average years of IT experience and agile experience of the participants for each role are described in the third and the fourth columns, respectively. We note (as per Table 2) that the participants have gathered more IT experience in 'heavyweight' (or traditional) way of working than in Agile.

Table 2: Roles of participants.

Role	Number of participants	Average years of experience in IT	Average years of experience in Agile
PO	2	25	6
Architect	1	26	3
Domain expert	2	22	1
Business analyst	3	23	4

The FG was guided by the first author, was 45 minutes long and was video-recorded. At the beginning of the FG, the QRs challenges reported in (Alsaqaf et al, 2019) were explained in a nutshell. The first author then explained the initiative used in the AQRE workshop and presented the resulting model as depicted in Figure 1. Thereafter, the participants were given the time to express their opinions and thoughts. To trigger the conversation, the first author asked the participants what they think of the model on Figure 1 in regard to whether it fits the context, whether the participants think it is useful and in what way they perceive it as useful. The video-recording was analysed immediately after the FG meeting. The process of data analysis consisted of two phases and started by having the first two authors watching the recorded video of the FG and analysing the content (Phase 1). The grounded theory analysis techniques as described in (Charmaz, 2006) were followed to extract knowledge from the contents. Contrasted interpretations made by authors during the data analysing process were resolved by conducting an argumentative discussion (Hitchcock, 2002) to reach a shared interpretation. In the next phase (Phase 2), the concept-mapping practices (Conklin, 2003) and argumentative discussion (Hitchcock, 2002) were applied to identify which QRs challenges reported in (Alsaqaf et al, 2019) could be mitigated by using the AQRE approach. In Section 4.3, Table 3 summarizes the results of Phase 2. The first column of the table shows the reported categories and their related challenges, while the second indicates how the AQRE approach could mitigate the related challenge in the

first column. A dash "-" in the second column means that no remedy has been identified by the practitioners, which also means that the AQRE approach has no answer yet for the related challenge.

4.3 Findings

In this section, we summarize our findings based on the analysis process we described in the previous section. We first report the extracted knowledge resulting from Phase 1. Then, Table 3 summarizes the result of Phase 2.

By analysing the shared feedback by the workshop participants and the FG participants, we derived three themes: requirements visualization, requirements documentation and requirements analysis. These are described below.

4.3.1 Theme: Requirements Visualization

The participants indicated that they experienced the AQRE approach as a way to visualize requirements in general, and QRs in particular. They felt that the AQRE approach could establish a visualized discussion to elaborate goals into requirements be it functional requirements, or QRs. A participant of the FG, expressed her opinion as "*This visualization tool will help conduct the right discussion with the right persons to elaborate the requirements, however, we need somehow to document the outcomes of the discussion*". Another FG-participant agreed with the need of visualizing requirements and suggested the use of Rich Picture (Berg and Pooley, 2013) instead of a goal-based requirements modelling language tool. While AQRE does not recommend one visualization tool above another, the choice of a visualization tool should not result in multiple of vague interpretations of the requirements, which is the case in Rich Picture (Berg and Pooley, 2013).

4.3.2 Theme: Requirements Documentation

A heavy documented requirements process was recognized by the participants as a source of annoyance. A participant reacted with: "*I will be happy if this* approach *(e.g. AQRE) results in less documentation. Our process of elaborating requirements from initiative to epics to user stories results usually in long documents*". The same participant expressed her concerns and emphasized the feeling of losing control, which people who are used to heavy documented requirements process might develop by implementing the AQRE approach. Another participant elaborated further: "*We have ever started with an initiative of two pages, nowadays we write initiatives of more than 10 pages. Specially the requirements for publishing tenders need to be very concrete, detailed and SMART, something I do not see happening when using the AQRE* approach". However, the AQRE approach does not advocate "not documenting the requirements at all". It provides only instruments to elaborate the requirements in a face-to-face conversational way and documenting only what is needed while leaving enough space for decisions to be made at the right time by the right people.

4.3.3 Theme: Requirements Analysis

A FG-participant has recognized the need for delaying the decisions to the right moment and giving software developers the space to make the right decision when enough detailed information is emerged "*I see the AQRE approach as a discussion instrument to elaborate and capture the requirements (e.g. functional and quality requirements) while giving the chance for trade-offs to be made at the right time*". Another participant proposed to add priorities to the defined (sub)goals to guide later the development process. The same participant proposed as well to assign the (sub)goals to individual PO's to be able to elaborate and coordinate the requirements further within and between individual teams.

4.3.4 Mapping the Practitioners Perceptions of AQRE Against the Challenges

Once the themes crystalized, our next step was to map the practitioners' feedback against the QRs challenges from Table 1. (For interested readers, we suggest them refer to ref. (Alsaqaf et al, 2019) which explains in detail how the QRs challenges have been identified based on case study research.) The mapping is presented in Table 3. It indicates that AQRE addresses 12 out of the 15 QRs challenges. The three QRs challenges that are not addressed are *Uneven teams maturity*, *Suboptimal inter-team organization* and *Lack of cost-effective real integration test*. We should note, however, that these three challenges are organizational in nature and therefore might demand an approach that is grounded more on social and organizational behavior and respective theoretical models than on RE concepts (such as GORE).

Table 3: QRs challenges could be mitigated with AQRE.

QRs challenges (Alsaqaf et al, 2019)	How AQRE helps
Late detection of QRs infeasibility	AQRE workshop elaborates QRs related to identified (sub) goals
Hidden assumptions in inter- team collaboration	Early Face-Face communications and discussion will point participants to the same direction
Uneven teams maturity	-
Suboptimal inter-team organization	-
Inadequate QRs test specification	Elaborating QRs and making decision just in time
Lack of cost-effective real integration test	-
Lengthy QRs acceptance checklist	Elaborating QRs and making decision just in time
Sporadic adherence to quality guide-lines	Elaborating QRs and making decision just in time
Overlooking sources of QRs	AQRE workshop involves essential knowledge
Lack of QRs visibility	AQRE approach visualize requirements
Ambiguous QRs communication process	AQRE workshop enforce face-to-face communication
Unclear conceptual definition of QRs	AQRE approach explicitly mentions QRs
Confusion about QRs specification approaches	Elaborating QRs and making decision just in time
Unmanaged architecture changes	AQRE approach involve architectural knowledge early in the process
Misunderstanding the architecture drivers	AQRE approach involve architectural knowledge early in the process

5 DISCUSSION

The results of this research indicates the complexities of confronting the QRs challenges in LSDA projects by using one only method. Although our proposal seems to address 12 out of the 15 QRs challenges identified in (Alsaqaf et al, 2019), there are three more that remain to be dealt with. We consider this as a clear signal that it might not be realistic to expect that a single method might exist to deal with all QRs challenges at once. As the three challenges are organizational in nature, we think that these could be most effectively delt with presumably by introducing some project management actions that target the social behavior of project team members and their level of professional maturity. In line with this, it seems worthwhile rethinking our AQRE approach in terms of possible extensions regarding what it could be supplemented with from organizational perspective, in order to address the full spectrum of QRs challenges.

Furthermore, it is important to note that in our case, educating the two practitioners, the IO and the PO, took about 30 minutes. We think that this is not too much, knowing their solid professional experience (i.e. the IO had 8 years while the PO had 14 years of experience in their fields, though not all years were in agile projects). This indicates the possible cost-effectiveness of AQRE in terms of learnability and ease of adoption. Of course, as this is our very first evaluation, more evaluations through case studies are needed in order to have an accurate understanding of the cost-effectiveness of our approach. This is our next step. At the time of writing this paper, the authors are already in a conversation with another project organization willing to host a case study.

One could argue that using AQRE to elaborate QRs from high-level goals before starting the first sprint looks more waterfall since it contrasts with the agile "just-in-time" identification of the requirements. We think that QRs as they impact the software architecture should be identified as soon as possible while leaving enough space to specify them just-in-time. That is precisely what the AQRE does, it identifies the QRs as early as customer's intentions have been decomposed. The exact specification of those QRs are left to be just-in-time identified.

Did our practitioners feel that AQRE made their agile project less agile and more heavyweight? The FG results did not suggest this. One possible explanation could be that the practitioners had long experience in "heavyweight" projects in a large government organization that values documentation. As their organization was experimenting with agile and figuring out how to achieve a good balance between agility, flexibility and documentation focus, the participants might not have perceived our goal model as a source of additional bureaucratic and documentation efforts. In their views, the benefits of AQRE clearly outweighed the 2 hours invested time

for creating the AQRE model (Figure 1). Of course, we might have obtained other results if the AQRE approach would have been tried out in an agile organization where a large-scale framework had been adopted for many years and a history of many agile projects existed.

Based on the feedback from the FG participants, we recognized an interesting contradiction: on one side, there was willingness for less documentation while on the other side, there was the fear of losing control by documenting less. It seems that the FG participants associate the sense of control over a project with the presence of documentation. More documentation means demonstrating control and vice versa. We think that the culture of the organization may play a significant role in promoting more or less documentation. We note that the evaluation of the AQRE approach occurred in a government organization and the practitioners involved in the FG had far more experiences in traditional software development than in agile (see Table 2).

6 LIMITATIONS

This study had some limitations. First, our newly proposed approach addresses 12 of the 15 QRs challenged. While this indicates important progress, in order to cope with the remaining three challenges, one needs to supplement it with some management actions. We consider this an important aspect and it is our intention to look into those three QRs challenges in our follow-up research.

Second, there are some validity threats (Wieringa, 2014) concerning our very first evaluation. Because all practitioners involved in our FG study operate in a large-scale government organization in the Netherlands, the themes that came out of the FG might differ from those that might possibly have come out if the demonstration of the AQRE approach and its evaluation had happened in a private agile company, for example one that is incorporated in the past decade and has always been agile (i.e. never worked with heavy-weight approaches).

Next, the FG moderator (i.e. the first author) had experience in the host organization, so some bias could be passed to the data analysis process. However, we employed disciplined steps and systematic techniques, e.g. coding, reflexivity, memoing, and discussion of the analytic process with the other researchers, all of which helped us reduce bias.

Finally, we also would like to mention that while we cannot claim universal generalizability of our very first evaluation results, it could possibly be realistic to expect that if we use the AQRE approach in similar type but physically different organizations, we might find some similar observations. This is due to the similarities in the contexts of our organization with others in the same sector (Ghaisas et al, 2013). For example, other Dutch large-scale government IT departments who embark on LSDA and experience similar challenges like those described in Table 1.

7 CONCLUSION AND IMPLICATIONS

In this paper we proposed the AQRE approach for elaborating requirements in general and QRs in specific from high-level goal(s) in LSDA projects. The proposed approach consists of one role (e.g. IO) and two steps (e.g. Workshops preparation, Workshop execution). Adhering to the agile principles is heavily taken into consideration since our proposed approach leans on collaborations and individuals interactions. The primary objective of AQRE is to identify a remedy to the QRs challenges reported in (Alsaqaf et al, 2019). The fitness and usefulness of the proposed approach is further evaluated using a FG as recommended in (Wieringa, 2014). This very first evaluation of the AQRE indicated that the proposed approach could mitigate 12 of the 15 QRs challenges reported in (Alsaqaf et al, 2019). The other three challenges are organizational in nature (e.g. Uneven teams maturity, Suboptimal inter-team organization, Lack of cost-effective real integration test) and may need additional organizational practices to deal with them. Our highest priority is to evaluate the AQRE approach further in other real-world projects in order to improve its effectiveness and usefulness beyond the context in which we did our very first FG evaluation. To this end, we plan two FG studies in two different contexts: one in a consulting company with professional consultants providing agile project management services to agile project organizations in a wide range of business sectors, and a second FG study in a national professional association whose members are requirements engineering practitioners, some of which work primarily in agile contexts. This evaluation effort would help us better assess the applicability of AQRE and its generalizability across contexts.

REFERENCES

Abheeshta, P., et al. (2018). “Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review”. PROFES(9459): 334–351.

Agile Alliance. (2001). “Manifesto for Agile Software Development”. http://www.agilemanifesto.org.

Albuquerque, D., et al. (2021). “A Sustainability Requirements Catalog for the Social and Technical Dimensions”. ER(1): 381–394.

Alsaqaf, W., et al. (2017). “Quality requirements in large-scale distributed agile projects - A systematic literature review”. LNCS 10153: 219–234.

Alsaqaf, W., et al. (2019). “Quality Requirements Challenges in the Context of large-scale distributed agile: An Empirical Study.” Information and Software Technology.

Berg, T., Pooley, R. (2013). “Rich Pictures: Collaborative Communication Through Icons”. Systemic Practice and Action Research 26(4):361–376.

Bucy, M., et al. (2017). “Keeping transformations on target”. McKinsey & Company(March): 1–17. [Online]. Available: https://www.mckinsey.com/capabilities/rts/our-insights/keeping-transformations-on-target.

Calefato, F., Ebert, C. (2019). “Agile Collaboration Distributed Teams”. IEEE Software 36(1): 72–78.

Charmaz, K. (2006). “Constructing grounded theory: a practical guide through qualitative analysis”. SGAE. 1st edition.

Chung, L., et al. (2000). “Non-Functional Requirements in Software Engineering”. Springer. NY, 1st edition.

Cockburn, A. (2000). “Writing Effective Use Cases”. Addison Wesley. 1st edition.

Conboy, K., Carroll, N. (2019). “Implementing Large-Scale Agile Frameworks: Challenges and Recommendations.” IEEE Software 36(March/April):1–9.

Conklin, J. (2003). “Dialog Mapping: Reflections on an Industrial Strength Case Study”. Visualizing argumentation:1–1.

Dalpiaz, F., et al. (2016). “iStar 2.0 Language Guide”. [Online]. Available: https://sites.google.com/site/istarlanguage.

Daneva, M., et al. (2007).“Exploiting a Goal-Decomposition Technique to Prioritize Non-functional Requirements”. 10th WER.

De Lucia, A., Qusef, A. (2010). “Requirements engineering in agile software development,” Journal of Emerging Technologies in Web Intelligence 2(3): 212–220.

Digital.ai Software, (2021). “ 15th State of Agile Report”. Digital.ai.

Domah, D., Mitropoulos, F, J. (2015) “The NERV Methodology : A Lightweight Process for Addressing Non-functional Requirements in Agile Software Development”. Southeastcon: 1-7.

Farid, W, M., Mitropoulos, F, J. (2012). “Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes,”. Southeastcon: 1-7.

Farid, W, M., Mitropoulos, F, J. (2013). “Visualization and scheduling of non-functional requirements for agile processes”. Southeastcon: 1-8.

Ghaisas, S., et al. (2013). “Generalizing by Similarity : Lessons Learnt from Industrial Case Studies”. CESI: 37–42.

Girvan, L., Paul, D. (2017). “Agile and Business Analysis Practical guidance for IT professionals”. Bcs Learning & Development Limited. 1st edition.

Helmy, W., et al. (2012). “Requirements engineering methodology in agile environment,” International Journal of Computer Science Issues 9(5): 293–300.

Hitchcock, D. (2002). “The Practice of Argumentative Discussion”. Argumentation(16): 287–298.

Horkoff, J., et al. (2019). “Goal-oriented requirements engineering: an extended systematic mapping study”. RE 24:133–160.

Inayat, I., et al. (2014). “A systematic literature review on agile requirements engineering practices and challenges”. Computers in Human Behavior 51(October): 915–929.

Inayat, I., et al. (2015). “Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed”. XP Workshops.

ISO/IEC. (2011). “ISO/IEC 25010:2011 - Systems and software Quality Requirements and Evaluation (SQuaRE)”. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733.

Jeon, S., et al. (2011). “Quality attribute driven agile development”. SERA: 203–210.

Kopczyńska, S., Nawrocki, J. (2014). “Using non-functional requirements templates for elicitation: A case study”. RePa: 47-54.

Krueger, R, A., Casey, M, A. (2014). “Focus Groups: A Practical Guide for Applied Research”. SAGE. 5th edition.

Kumar, M., et al. (2013). “A Hybrid Approach of Requirement Engineering in Agile Software Development”. ICMIRA: 515–519.

Lamsweerde, van, A., et al. (1998). “Managing Conflicts in Goal-Driven Requirements Engineering”. IEEE Transactions on Software Engineering 24(11): 908–926.

Lauesen, S. (2002). “Software Requirements: Style and Techniques”. Addison-Wesley. 1st edition.

Maiti, R, R., Mitropoulos, F, J. (2015). “Capturing , Eliciting , Predicting and Prioritizing (CEPP) Non-Functional Requirements Metadata During the Early Stages of Agile Software Development.” SoutheastCon:1-8.

Morgan, D, L. (1996). “Focus Groups as Qualitative Research” . SAGE. 2nd edition.

Muller, P., et al. (2012). “Modeling modeling modeling”. Softw Syst Model 11(3): 347–359,

Noreika, K., Gudas, S. (2021). “Modelling the alignment between agile application development and business strategies”. BIR-WS 2991: 59–73.

Pastor, O., et al. (2022). “Teaching Modeling in the time of Agile Development”. Computer (Long Beach Calif) 55(June): 73–76.

Pfleeger, S., Atlee, J. (2009). “Software Engineering: Theory and Practice”. (2009). Pearson. 4th edition.

Pohl, K. (2010). “Requirements Engineering Fundamentals, Principles, and Techniques”. Springer Berlin. Heidelberg, 1st edition.

Rahy, S., Baas, J, M. (2022). “Managing non-functional requirements in agile software development,” IET Software 16(1): 60–72.

Sherif, E., et al. (2022). “Managing non-functional requirements in Agile Software Development”. ICCSA: 205–216.

Shore, J., Shane, W. (2007). “The Art of Agile Development”. O’Reilly Media. 1st edition.

Smart, J. (2018). “To Transform to Have Agility, Dont Do a Capital A, Capital T Agile Transformation”. IEEE Software 35(6): 56–60.

Sutherland, J., et al. (2022). “Embracing Agile: How to master the process that’s transforming management”. Harvard Business Review 94(5): 40–50.

Wieringa, R, J. (2014). “Design Science Methodology for Information Systems and Software Engineering”. Springer. Dordrecht, 1st edition.

Yu, E, S, K. (1997). “Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering,”. ISREI: 226–235.