A Framework for Capturing Quality Requirements by Integrating the Requirement Engineering Elements in Agile Software Development Methods

Yuli Fitrisia[1], Rosziati Ibrahim[2]
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia[1, 2]
Faculty of Computer Engineering Technology, Politeknik Caltex Riau, Pekanbaru, Indonesia[1]

***Abstract*—The early phase of Agile Software Development (ASD) methods is Requirement Engineering (RE). Quality Requirement (QR) is a type of RE that needs to be captured at the initial development phase to reduce rework, time, and maintenance costs. However, QR is one of the issues mentioned in ASD, namely the need for more capability to elicit, analyze, document, and manage QR. Therefore, this research aims to propose a framework for capturing QR to address QR issues in ASD by integrating RE elements, namely the RE phases, Documentation, Roles, and RE techniques. This research was conducted in four phases: 1) undertaking a theoretical study, 2) conducting an exploratory study to identify the current practices and issues to capture QR in ASD, 3) constructing the framework by using the RE elements, and 4) evaluating the framework by conducting ASD practitioners' view using questionnaires. The questionnaires were then analyzed using descriptive statistics based on the average mean of each element. The result shows the average mean for all elements (4.25), the average mean of each element for the RE phases (4.36), the documentations (4.11), the roles (4.25), and the RE techniques (4.18). The mean distribution of each element is more than 4 out of 5 indicating that the framework to capture QR is verified. Thus, this framework can be used by ASD practitioners as a guideline to capture QR in ASD methods.**

***Keywords*—Quality requirement; requirement engineering; ASD; framework; ASD practitioners**

I. Introduction

In recent years, software practitioners have used many software development methods. Each software development method has its advantages and disadvantages. For example, traditional software development methods are suitable for requirements that are clearly defined from the initial phase of the Software Development Life Cycle (SDLC) [1]. However, these methods have several issues to be aware of, namely limitations in accommodating requirement change during development, lack of interaction with customer, and the client only seeing the product at the end of the project [2]. These issues are covered by Agile Software Development (ASD) methods [2-4].

ASD is a popular software development method widely used in today's business industry [5-8]. ASD methods have advantages in producing software products faster to markets, being flexible to changing requirements, and increasing customer collaboration [9-10]. In addition, based on the Agile annual report [11], ASD increases collaboration, facilitated teamwork, provided better alignment for business needs, better work environment, and better visibility capabilities in application development.

Requirement Engineering (RE) is the early stage of the Software Development Process, including ASD. It has two types of requirements, namely Functional Requirements (FR) and Quality Requirements (QR), also known as Non-Functional Requirements (NFR). QR presents the quality of the software product. However, the tendency to ignore QR, the product quality can be negatively impacted and even cause software product failure [9-11]. Therefore, it is important to capture QR early in the RE phase of the ASD to reduce time, rework, and maintenance cost.

ASD practitioners also realize the importance of documenting QR [15]. It can help with easy analyses and traceability, gives a ready-to-use template, and eases communication among ASD stakeholders. Roles are people who possess responsibilities for every process in software development. Assigning roles with their expertise helps managing QR and making the QR list consistent and unambiguous [16]. Selecting proper techniques for each RE phase is also important to ensure the phase is well-conducted and reduce the complexity of capturing QR.

ASD has issues in terms of lack of capability to handle QR [9,14-18]. For instance, eliciting and analyzing QR challenges, limited techniques for eliciting, modeling, and linking QRs with functional requirements, and inadequate user stories for specifying QR [12]. The other issues are the lack of QR documentation [9,17], and no explicit practice for QR [21]. Additionally, it also lacks ASD capabilities in managing QR through ASD artifacts, namely user stories, prioritizing functional requirements (FR), and the tendency to ignore QR [19-22]. Based on these previous issues, practitioners realize the importance of capturing QR early in the context of ASD [25]. It is also important for capturing QR systematically, which offers how to capture and validate QR [15,23,24,27]. It is also supported by the result of a survey conducted by López et al. [28] reporting that 50% of practitioners follow QR processes in a systematic, well-defined, or ad-hoc process. It can reduce the complexity of managing QR in ASD and improve product quality and customer satisfaction.

Based on literature studies conducted in this research, identified issues related to capture QR in ASD methods can be solved by implementing approaches, frameworks, and guidelines for QR in ASD. Furthermore, several studies have been conducted to capture QR. The QR elicitation guideline proposed by Younas et al. [29] was implemented to tackle QR in ASD methods at the elicitation phase by using several pieces of documentations, namely the QR glossary, historical data, and checklist table between FR and QR. It also involved the developer team, experts, and customers. A study by Younas et al. [30] proposed the Elicitation of Non-Functional Requirements in ASD using a Cloud Computing Environment. This methodology also involves the same RE phase. The documentations used were project history, template for NFR by Kopczy and system type. Then, the roles involved developer team, customers, and experts. According to Jarzębowicz et al. [31] proposed elicitation practices to capture QR in ASD. The roles involved developer team, customers, products owners, and experts. Another study by Behutiye et al. [32] proposed the QR documentation guideline within ASD. In this guideline, they only proposed documentation for QR. Then, Alhaizaey et al. [33] proposed a framework for Reviewing and Improving Non-Functional Requirements in ASD-based Requirements. This framework was implemented at the elicitation, analysis, and validation phase. The documentations used were FR user story, identified NFR, NFR bibliographic source, glossary and standards, and NFR description. It also involved the developer team and customers. Finally, Sherif et al. [24] proposed a framework to manage NFR in ASD. It was implemented at the elicitation, analysis, documentation, and validation phase. The documentations used were FR user story, checklist-based reading template, system type, domain type, project history, mapping sheet between QR, and QR user story. It also involved the developer team, product owners, and customers.

There are still several gaps in proposed previous solutions to capture QR. It shows that three [29] of six of the previous works implemented to capture QR in the elicitation phase only. However, based on the earlier discussion, it is important to capture the QR to construct the RE phase element by using complete phases, namely elicitation, analysis, documentation, and validation, to produce QR properly. Then, the documentation used in the previous study only focuses on partial documentation for each RE phase [33]. Additionally, several studies did not use a software quality model or standard as a reference to identify the QR [24]. However, it is needed because the quality model or standard has the metrics to measure the QR. It can be used to make sure the QR is measurable. One study focuses on QR documentation only [32]. It should need comprehensive documentation by constructing the documentation for each RE phase to give an easy analysis and get a clear QR list.

The other element is the roles involved. Several previous studies report that the roles did not focus on their expertise and involved the roles needed related to each RE phase [24,33]. The roles should be constructed to be suitable for their expertise and responsibility. As a result, it gives an impact on the process of capturing QR more effectively. In addition, several studies do not mention the RE technique used clearly for each RE phase [29, 31]. Furthermore, it also needs to define the RE technique that will be used in implementing the RE phase systematically.

This paper presents research focuses on addressing three issues according to the gap findings, namely (i) lack of clarity on which the Requirement Engineering phase and the techniques used of the capture QR should be implemented, (ii) lack of documentation of QR, and (iii) the inadequate ability of user stories to capture QR by involving roles to address QR effectively. Based on these three issues, the following Research Questions (RQs) are formed. They are RQ1: how to determine the RE elements and each component in ASD methods, RQ2: how to design the framework for capturing QR using the elements and components of RQ1 and RQ3: how to evaluate the QR framework by using practitioner reviews. Thus, the objectives of this research are as follows: (i) to determine the RE elements and each component in the ASD method, (ii) to design a framework for capturing QR, and (iii) to evaluate the QR framework by using practitioner reviews.

The research significantly contributes to the body of knowledge in the field of Software Engineering, particularly in Requirement Engineering, on how QR is being addressed, captured, and documented in an ASD environment. Moreover, ASD practitioners can use this proposed framework as one of the best practices and guidelines for handling QR in an ASD environment, and the Stakeholders in an ASD environment can gain QR status transparency and track the QR. Then, it systematically captures the QR to reduce the complexity of managing QR in ASD, improve product quality, and finally improve customers' satisfaction.

This paper is outlined as follows: Section II presents an overview of related work and the gaps in previous work, Section III presents the research methodology, including theoretical study, exploratory study, framework design, and framework evaluation, Section IV explains the experimental results and analysis, and Section V presents the conclusion and future work.

II. Related Work

Quality Requirement (QR) is one of the RE types and is an important artifact that plays a crucial role in software project success [34, 35]. It can lead to increased costs or longer time-to-market due to the failure to meet QR needs properly [25,26]. Many software quality standards can be adopted in QR for software development. In the 1970s and 1980s, the authors in [36,37] proposed their own QR taxonomies. The ISO/IEC 25010 standard is the most widespread method of defining, categorizing, and managing QR. This standard is widely adopted in today's industry. It has nine quality categories, namely functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility, and safety [38]. Numerous recent studies have conducted extensive reviews on QR in the Requirement Engineering in ASD methods [13]. A study in [29], Younas et al. proposed activities of NFR Elicit guideline as depicted in Fig. 1.

According to Fig. 1, the first phase is the preliminary requirement to collect FRs that have been identified. The next step is to identify the software type and then to identify the QR from Glossary. It involves the historical data to make predictions about new QR. This elicitation guideline encompasses experts' involvement. This process ends when the experts and users finalize it. This guideline also uses a checklist table between FR and QR to manage changing requirements. In contrast, the limitation of this guideline is the mapping between FR and QR, which is done at the end of the process, for it can take time if there is a change, and the process should be repeated from the beginning. The other limitation is that there is no mapping between QRs to check conflict between them. Then, this guideline is only implemented in the elicitation phase, and there is no clear explanation of how to finalize the QR list between the experts and the users.

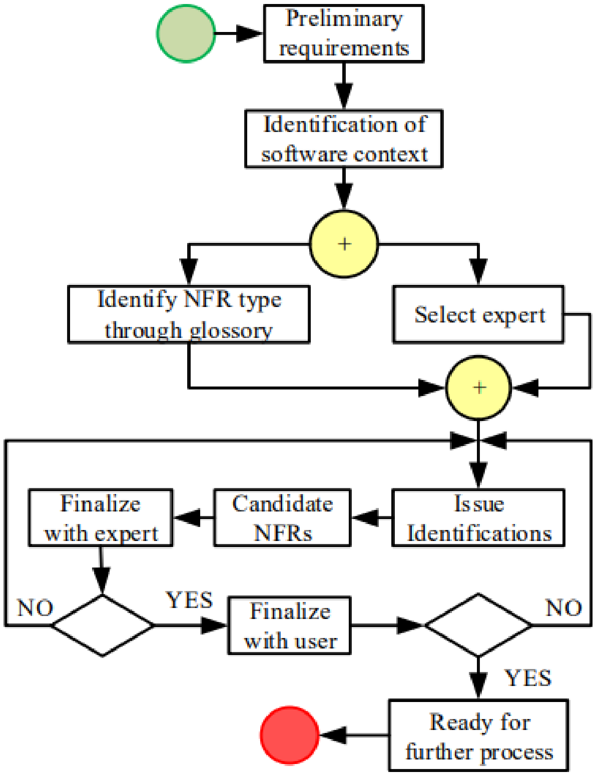

Fig. 1. QR elicitation guideline [29].

Similarly, in a study [30], Younas et al. proposed the Elicitation of Non-Functional Requirements in ASD using a Cloud Computing Environment. This proposed methodology is a continuation of their previous research on the QR elicitation guideline [29] as described in Fig. 2. From Fig. 2, it can be seen the differences of this present study from previous studies are the use of Natural Language Processing (NLP) for QR extraction in the elicitation phase and cloud computing tools for sharing data and communication. However, this methodology has no link between QR to check the conflict, and it is only implemented in the elicitation phase.

Jarzębowicz et al. [31], discussed elicitation practices to capture QR in ASD. The methodologies used in their study were a systematic literature review (SLR) and interviews with ten ASD practitioners. Based on their findings in the SLR study, techniques for elicitation include Customer-Developer meetings, Brainstorming, QR document circulation between Product Owner and QR Stakeholders (e.g., experts), QR catalog, and on the basis of Business Process Models are popular in several studies. The other findings based on interviews with ASD practitioners summarize that ASD practitioners mention that the presence of an analyst role contributes to more thorough QR elicitation. Then, for sources of requirements, several ASD practitioners argue that the Product Owners are capable of giving opinions on QR. In other cases, multiple stakeholders can consider eliciting requirements from their point of view, including other IT systems and document standards. The other sources are software developers and technical experts who will provide inputs based on their expertise. Then, the techniques used for QR elicitation are interviews, and workshops (including brainstorming and other kinds of group work) that are commonly used.

There are efforts to capture QR using the current practice. For instance, the elicit QR is non-verifiable and non-measurable, and not all QR needed in the software development is detected. For example, usability is recognized, but others still need to be defined. Therefore, based on the ASD practitioner's view, they state that it needs an active approach and guidance for all ASD stakeholders.

Another study done by Behutiye et al. [32] proposed the QR documentation guideline within ASD. This guideline, known as ASD QR-Doc, offers 12 recommendations to facilitate the documentation of QR in ASD. The guideline was validated by ASD professionals. They consider the ASD QR-Doc guideline to be straightforward for use in ASD, helpful in the early process and documentation of QR, and not obstructive to the ASD process. However, the guideline is only validated by practitioners, and this guideline needs to be clearly defined when QR is documented in the RE phase.

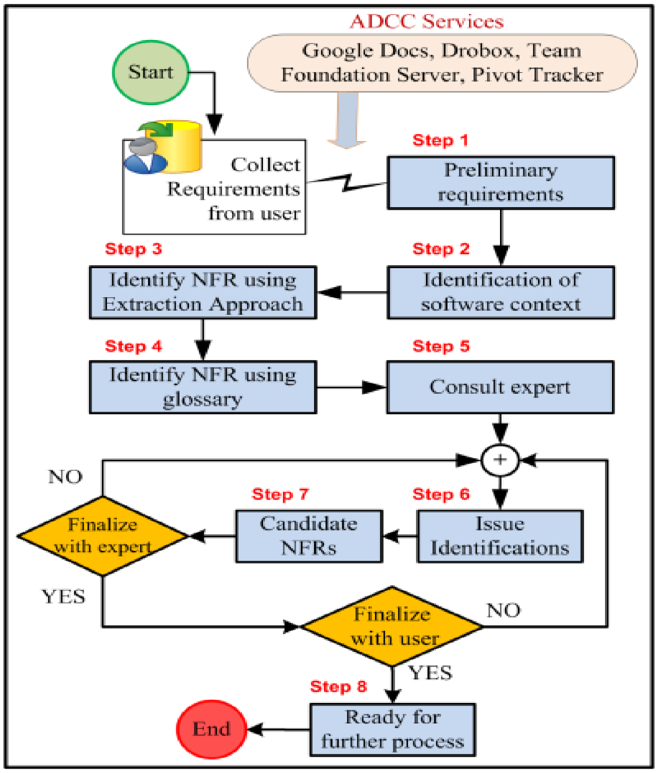
ADCC Services
Google Docs, Drobox, Team Foundation Server, Pivot Tracker
Start
Collect Requirements from user
Step 1
Preliminary requirements
Step 3
Step 2
Identify NFR using Extraction Approach
Identification of software context
Step 4
Step 5
Identify NFR using glossary
Consult expert
+
NO
Step 7
Step 6
Finalize with expert
Candidate NFRs
Issue Identifications
YES
Finalize with user
NO
YES
Step 8
End
Ready for further process

Fig. 2. QR elicitation methodology using cloud computing [30].

Alhaizaey et al. [33] proposed a framework for Reviewing and Improving Non-Functional Requirements in ASD-based Requirements. It uses NLP and Artificial Intelligence (AI) techniques to automate analyzing QR from user stories. Then, the artifact from this phase is reviewed and inspected for improving the QR. Three artifacts are produced from the previous phase, namely the processed user story, the identified QR, if any, and the description of the QR. The last phase is improving the requirements using four activities, namely rewriting a user story, rewriting [means] part of the user story, including QR as an Acceptance Criteria (AC), and including QR as a Definition of Done (DoD), as depicted in Fig. 3. However, this framework needs to explain how the documentation phase is conducted. It also only involves the developer team and clients to finalize the QR and needs to involve experts in validating the QR list.

Finally, Sherif et al. [24] proposed a framework to manage NFR in ASD called MANoR, which stands for Managing Agile Non-Functional Requirements. It provides two main stages and five main components. The stages are pre-analysis and post-analysis, and the components support various critical functions within requirements engineering, encompassing requirements elicitation, analysis, documentation, and validation. The main steps of the MANoR framework are depicted in Fig. 4. There are four areas for improvement and limitations associated with this approach. Firstly, there is no Quality Model from the glossary that can be used as a source to recommend QR. The quality model, particularly ISO 25010, has an advantage. For instance, it has the metrics to measure the QR lists. Secondly, the expert is not involved in the QR elicitation process, which can help identify QR more effectively. Thirdly, there is no mapping between FR and NFR during the QR analysis process to check for inconsistency. There is also no documentation of QR decisions that will help track them. Fourthly, the validation technique (reading technique) should be mentioned, and QR is only validated by the clients or customers.

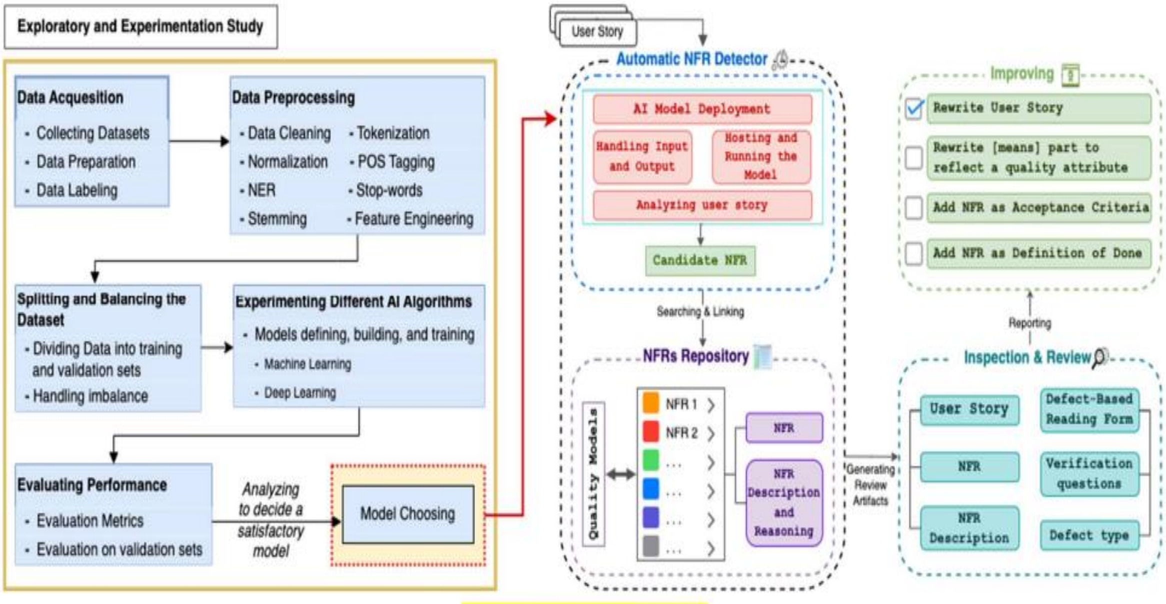

Fig. 3. Framework for elicitation, analysis, and reviewing QR [33].

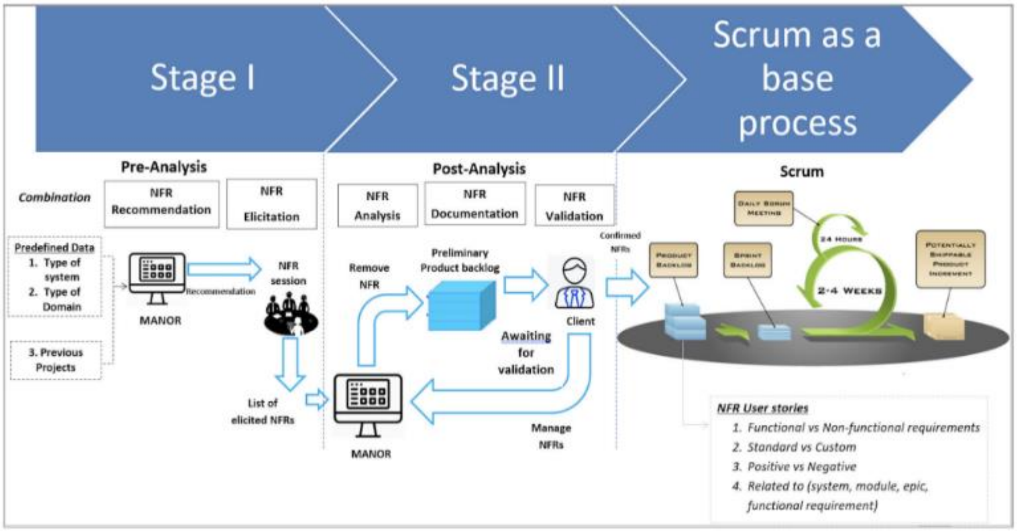

Fig. 4. MANoR framework [24].

In summary, based on related studies discussed, the solutions focus on capturing QR in ASD from different points of view, namely frameworks, approaches, models, and guidelines as previously mentioned. According to the gaps identified in related work, therefore, this research proposes a framework for capturing the QR by integrating the RE elements in ASD methods comprising 1) the RE phase, 2) documentation, 3) the roles involved, and 4) the RE technique.

III. Methodology

Fig. 5 shows that the research methodology consists of four phases. The first phase is theoretical study, the second phase is exploratory study, the third phase is framework design, and the fourth phase is framework evaluation. Each phase has key activities and outputs of the activities.

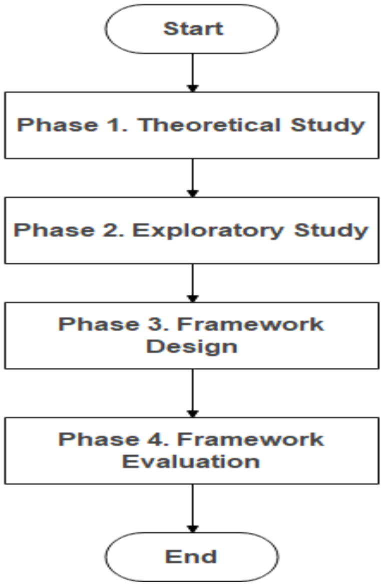
Start
Phase 1. Theoretical Study
Phase 2. Exploratory Study
Phase 3. Framework Design
Phase 4. Framework Evaluation
End

Fig. 5. Research methodology.

A. Theoretical Study

The first phase was conducted by reviewing the literature to explore the concept related to the study, current practices, and issues to capture QR in ASD, identifying the element constructed in the framework, and finding the gap by analyzing the related work in ASD from various sources like journals, conferences, books, and other sources. Then, based on the gap finding from existing related work, this study proposes a new solution for the elements and the components to construct for capturing QR in ASD. The outputs of this phase are a clear explanation of current practices and the issues of capturing QR in ASD. It also gets the elements and the components constructed in the framework.

B. Exploratory Study

The second phase was conducted by doing a qualitative study by interviewing ASD practitioners with a small number of respondents to get detailed information about their opinions [39]. The interview aims to explore current practices to capture QR in the ASD industry and the issues based on practitioners' views. The exploratory study was conducted by an online interview involving 30 ASD practitioners. The summary of current practices when capturing QR in ASD is categorized into three phases, namely RE phase, during development, and during product release to market or customers.

The result percentage indicates that the QR is identified majority during development and release to market is 33.33%. Then, during the RE phase and development is 26.67%, during development is 20%, in RE phase is 10%, and product release to market is 10%. The other findings are that QR is identified the majority during development is 80%. Then, during product release to market is 43.33%, and during the RE phase is 36.67%. On the other hand, it also shows the weaknesses of current practice in capturing QR based on interview findings, namely:

- Identified QR in the RE phase: More time is needed in the planning phase. Therefore, it takes time to start the development process. Only the developer team was involved in identifying the QR in the RE phase. Sometimes, people need to be made aware of QR. Consequently, they should have added one more task for improvement, but it took time.
- Identified QR during development: It needs more rework, cost, and time to adjust the QR because it changes most in software architecture, namely in design and code, adding a new story to fix the bug in Product Backlog. It is challenging to identify QR during development because the document is not updated based on requirement changes. Sometimes, the QR is ignored because there is no QR documentation. It can change the project timeline, and the workload also increases a lot.
- Identified QR after release to market or customers: It needs more rework because it changes most in software architecture, namely in design and code. The application appearing on the device does not meet the user's requirements. It needs more time, it can change the project timeline, product backlog items increase, and the workload also increases a lot.

C. Framework Design

The third phase is to construct the framework based on a theoretical study and exploratory study findings. The theoretical study is based on the literature review, and the exploratory study is based on interviews with ASD practitioners. The output of this phase is the proposed framework for capturing QR by integrating the RE element in ASD based on studying the existing model gaps outlined in related work. It preserves the strengths of these existing models and tries to overcome their limitations. It also constructs various aspects for comprehensive QR.

This present study argues that it is important to capture QR by implementing in the RE phases as the foundation of the software lifecycle, which are elicitation, analysis, documentation, and validation [24]. The QR was identified during an initial iteration of Agile Software Development and then refined in further iteration [36,37]. The authors in [31] also found that early identification of QR at the beginning of a software project is better. It also supports an exploratory study; if QR were defined during the RE phase, it would not require much effort for future tasks. When QR is identified early, it can help to produce QR properly, help to identify project effort, cost, and size, and reduce rework.

The documentation/artifact should need comprehensive documentation, usefulness, relevance, and understandability for supporting QR documentation and its impact on ASD practice [32]. The ASD manifesto focuses on the development of working software over comprehensive documentation [42]. However, a lack of QR documentation can cause misinterpretation and rework [31]. It also supports an exploratory study, which makes it challenging to identify QR during development because the document was not updated based on requirement changes, and sometimes the QR is ignored because there is no QR documentation. Therefore, it is important to include documentation as the element to capture QR. Constructing the documentation for each RE phase can give easy analysis, clear tasks, traceability, clear documentation, and well-documented QR [16]. It can help communication among ASD Stakeholders.

The other element is the roles involved. In several previous studies reviewed, it was found that the roles need to include capturing QR to focus on their expertise [31]. The roles for each RE phase need to be constructed in a way that is suitable for their expertise and responsibility [32]. It also supports an exploratory study result, because only the developer team is involved in identifying the QR, sometimes people are not aware of the QR. Consequently, they should have added one more task for improvement, which took more time. Furthermore, it is important to involve roles with their expertise for each RE phase because it can impact the process of capturing QR more effectively. According to Aljallabi et al. [16] stated that it also provides proper QR results with more reliable results due to invented different points of view.

It also needs to define the RE technique that is used to capture QR, which is in line with ASD practice on direct communication [31]. RE techniques were needed to conduct the RE phase systematically by the team [1]. If we can choose the right techniques, it can be produced and conducted to capture QR more effectively and clearly. It also helps to capture the impact of change of requirement, which is understandable by all stakeholders, and check for errors and inconsistencies. Therefore, this research proposes a framework to integrate the RE elements, namely, 1) the RE phase, 2) documentation/artifact, 3) the roles involved, and 4) the RE technique. These four elements are needed to capture QR in ASD. The following sub-sections define the RE element and its relation to the component of each element.

1) Constructing RE phase element: The RE phase was constructed by using the four-phase component of RE as the first element, namely elicitation, analysis, documentation, and validation [1]. It is important to capture QR by using complete phases to produce QR properly [24]. It is also supported by a study [33], who argue that validating the QR is a crucial requirement process as the last phase of RE. Therefore, it should be conducted in all phases of RE to capture the QR.

The elicitation phase is the first phase of RE that aims to understand the tasks performed by stakeholders and how a new system could support their tasks [1]. This phase is the foundation of project success and aims to explain QR to the stakeholders. This session also determines the QR based on the element used. The output of this phase is the list of elicit QR. The second phase is the analysis phase aiming to find consistency between FR and QR [16]. It is also to make sure there is no conflict between QRs [24].

The purpose of the documentation phase as the third phase is to write down software requirements into a Software Requirements Specification (SRS) document [1]. It can be used to document user requirements and system requirements, namely the FR and QR in the user story written in the product backlog. It can also be used to document the decision on the QR. The last phase is the validation phase, which aims to check the requirements meet the customers' expectations [1]. The checking process consists of a validity check, consistency check, completeness check, realism check, and verifiability check. It is also to ensure that the QR list is clearly defined, that there is no error interpretation, to check areas where clarification may be required, and that there is no missing information. According to Sherif et al. [24], this phase also aims to reach an agreement among stakeholders regarding QR on the same view for the software being developed.

2) Constructing QR documentation element: The documentation element used in this framework can provide a ready-to-use template for easy analysis and traceability. It was constructed at each RE phase by integrating documentation components as a reference for capturing the QR. QR should be documented along with FR [43].

System-type document in elicitation helps users identify relevant QR based on different types of systems [40,41]. Domain-type document in elicitation helps users identify relevant QR based on various application domains [44]. ISO 25010 Quality Model in elicitation has advantages, namely providing a more detailed QR and metrics on how to measure the QR [29,30]. Project history document in elicitation is useful to define the QR for the next project based on historical data. [24]. A mapping sheet between FR and QR document is used in the analysis to check the consistency between FR and QR [16]. A mapping sheet between QR document is used in the analysis to check the conflict between QRs [24].

In this research, a separate user story is used to document QR, which consists of the FR user story and the QR user story. This helps to manage QR during the development process, for example, during the sprint [45]. The functional user story in the documentation aims to document functional requirements in ASD. QR user story in documentation to document the Quality Requirements list in ASD. QR decision in documentation as a history to decide the QR [32]. According to Sabaliauskaite et al. [46] Checklist-based reading document in validation is used to check the properties of documents and what problems or defects should be identified based on the list of questions.

3) Constructing the involvement of roles element: The roles constructed into the element of assigning roles with their expertise are helpful for managing QR and making the QR list consistent and unambiguous. Then, different points of view for the validation phase are involved in working together to produce QR, which results in more reliability and understanding for all stakeholders. It is constructed at each RE phase by integrating role components to capture the QR.

The developer team in elicitation is a person who understands the technical side related to QR that is elicited and is responsible for finishing the QR item in the product backlog [24]. Customers in elicitation are the stakeholders who own the system and need to be explained about QR, which is elicited [29]. An expert in elicitation helps elicit the QR; that is, someone who has more knowledge of QR and is concerned with the fulfillment of QR [29]. The Product Owner in elicitation has the responsibility of managing and optimizing the product backlog to ensure the product value is maximized which aligns with the FR and QR that should be elicited [47]. According to Jarzębowicz et al. [31] QR.

The developer team, in analysis as a technical side, defines consistency and conflict in the mapping sheet according to a clear justification based on the developer's knowledge [16]. The developer team in documentation manages the document if there is a change in these documents. The developer team in validation is a technical team that finishes the QR in the Product Backlog and ensures the QR can be tested. The expert in validation can help refine the QR and interdependencies among QRs and validate the QR. The customer is also involved in validation to make sure the QR list is understandable to the customer. The Product Owner, in validation, ensures the QR list is valid and included in the Product Backlog.

4) Constructing the RE technique element: The RE technique presents how the RE phase was conducted. It was constructed at each RE phase by integrating the components of the RE techniques to capture the QR. The Interview and Brainstorming techniques in elicitation are commonly used and popular techniques in the elicitation phase [31]. The interview aims to discover information and to understand the system to be developed based on asking questions to the stakeholders [1]. Additionally, a study [48] stated that Brainstorming aims to gather information in many creative ways by conducting work group meetings involving roles.

The interaction matrix technique is used in the analysis of two-dimensional requirements to assess the inconsistency between FR vs. QR and the conflict between QR where each requirement is compared to the other [49]. The Structured Natural Language technique in documentation is the documentation technique for writing the FR and QR using natural language on a standard form or template where each field provides information on the requirements [1]. The checklist-based reading (CBR) technique in validation aims to detect defects in the requirements based on a list of questions [49].

There are four major elements constructed in the proposed framework to capture QR, namely: 1) RE Phase, 2) Documentation/Artifact, 3) Roles Involved, and 4) RE technique. Each element has its components for the proposed framework, as depicted in Fig. 6.

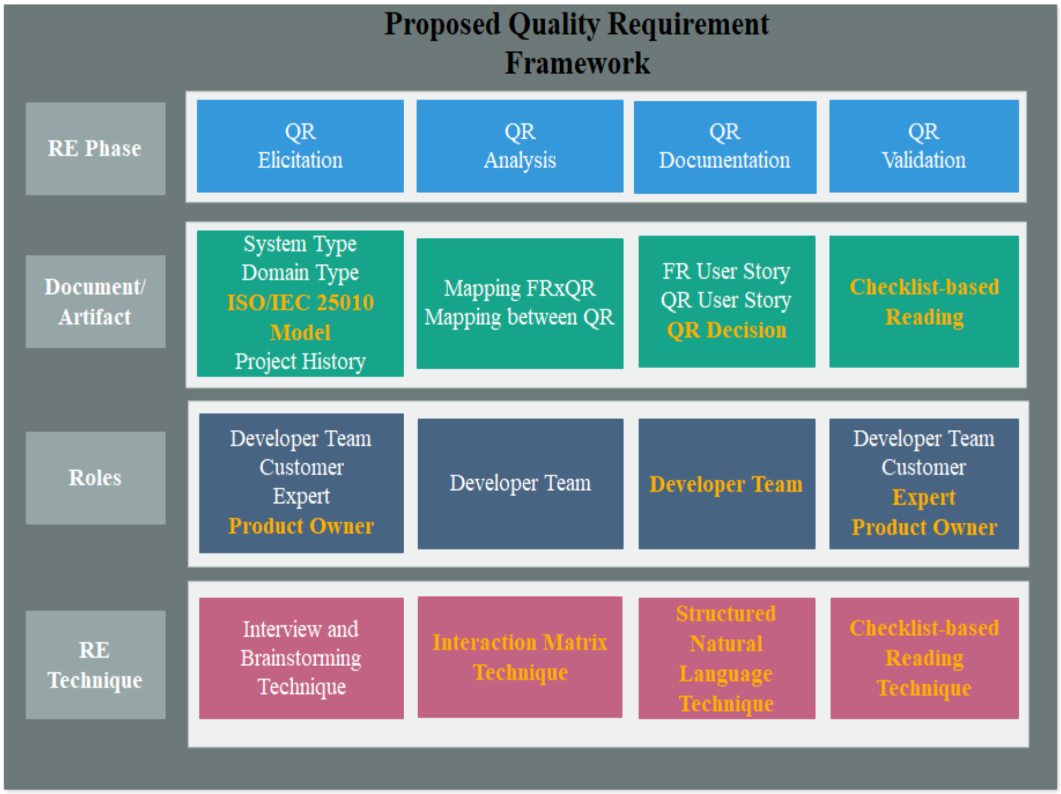

Fig. 6. The proposed QR framework.

D. Framework Evaluation

A practitioner evaluated this framework to verify it. This phase was conducted using quantitative research through questionnaires. The phases are instrument design, instrument validation, pilot study, sampling, data collection, and feedback analyses [50]. The questionnaires consist of two sections: demographic information and elements and the components of the QR framework. These instruments are used to verify the proposed framework to confirm whether the ASD practitioners' review agrees or disagrees with the proposed elements and the components of the framework to capture QR in ASD [50], [51]. The result of the framework evaluation is explained in the following Section.

IV. Results

This section explains the findings of the evaluation by using questionnaires for ASD practitioners classified into sub-sections: A) the Pilot Study, B) Data Collection and Sampling, C) Data Analysis, and D) the Result.

A. Pilot Study

A pilot study was conducted by involving 30 ASD practitioners to check the reliability of the instrument and get feedback for the questionnaires before conducting the sampling survey. The pilot study result shows that it achieves the reliability threshold (> 0.7) based on Cronbach's alpha coefficient [52]. It consists of all elements (0.81), RE phase (0.72), document (0.91), roles (0.90), and RE technique (0.82), which means that the questionnaires are acceptable. The other result is questionnaire feedback from the respondents. The questionnaires were refined to improve the quality and avoid misinterpretation of the questions.

B. Data Collection and Sampling

The respondents of this questionnaire survey are ASD practitioners gathered using Snowball Sampling as one of the non-probability sampling techniques. Then, the questionnaires were distributed using an online survey from various channels, namely alumnae and their networking, colleagues in the industry, and post questionnaires in ResearchGate, LinkedIn Group for ASD, and ASD Community Indonesia. The total number of respondents who filled out the questionnaire was 170 people for one month, as presented in Table I.

TABLE I. Questionnaire Response Rate

Description	Frequency	Percentage
Total questionnaires received	170	100%
Total rejected questionnaires	12	7.06%
Total usable questionnaires	158	92.94%

Table I shows that the total number of questionnaires received was 170, only 12 (7.06%) questionnaires were rejected due to outliers, and 158 (92.94%) questionnaires were used in this research.

C. Data Analysis

The data were analyzed using descriptive statistics. Data preparation aims to ensure that the data are free from errors by processing the cleansing data. It is important to make sure the result is accurate. According to Ibrahim et al. [50], the steps for data preparation start with screening the data and coding data, as well as checking missing data, suspicious response rates, outliers, and normality by using SPSS software.

1) *Screening and coding data*: This step was done by ensuring the data types were numeric and changing the measure data to scale. It was done at the variable view by changing the code manually.

2) *Checking for missing data*: This step aims to analyze the data that the respondents fill out. If there is missing data, it can cause an error. Based on the data, no missing values for all items were found. According to Hair et al. [53], if there are missing values, responses can be excluded by less than 10%, and the result is acceptable.

3) *Suspicious response rate*: This step is to identify the answer pattern if the respondents fill in the same values for all questions [53]. It can be excluded from the data that needs to be analyzed. The result is that no suspicious response rate was found.

4) *Outliers*: When using parametric or non-parametric tests, outliers can state the error rates and substantial distortions of parameter and statistic estimates [54]. Standardized Z-scores can be used to analyze the outliers based on the variables, and the values are then examined. The acceptable value of the Z-score is between -3.29 and +3.29, which indicates no outliers [55]. According to the Z-scores analysis, 12 outliers were found in this dataset, and it was removed from the dataset.

5) *Normality*: Before the data can be analyzed, it must meet the normal distribution, where each construct item must meet the normality [56]. The values of skewness and kurtosis can estimate the symmetry and data distribution. The authors in [53] stated that the value of the standard error of skewness close to zero is acceptable. While the authors in [56] stated that the acceptable value of the standard error of kurtosis should not exceed 10. The result shows that the standard error of skewness is 0.19, and the standard error of kurtosis is 0.38. This means that the data are close to the normal distribution and can be used to analyze the dataset.

D. Results

This section explains the results of the framework evaluation using the questionnaires by ASD practitioners. It consists of two sub-sections, namely demographic information and descriptive statistics.

1) *Demographic information*: This section describes the respondent's background and the organizational background.

a) *Respondents' background*: This section indicates their position in the organization and years of experience in ASD. According to respondents' positions, programmers are the majority of the respondents, about 46.20%. The second position is System Analyst, about 13.92%, and Quality Assurance (QA)/Tester, about 11.39%, followed by Team Leader, about 10.13% and Project Manager, about 6.33%, Product Owner, about 5.06%, Scrum Master, about 1.90%, and Others about 5.06%. The respondents' experiences with Agile Software Development methods are depicted in Table II.

TABLE II. EXPERIENCE IN AGILE SOFTWARE DEVELOPMENT METHODS

Positions	<1 Year	1-5 Years	6-10 Years	>10 Years	Total
Programmer	2	65	6	0	73
System Analyst	1	19	2	0	22
Quality Assurance/Tester	0	18	0	0	18
Team Leader	1	13	2	0	16
Project Manager	1	3	5	1	10
Product Owner	0	6	1	1	8
Scrum Master	0	1	2	0	3
Others	1	7	0	0	8
Total	6	132	18	2	158

Table II reports that the majority of the respondents' experiences were between 1 and 5 years, comprising 132 respondents, and among them, 65 respondents are programmers. The ASD experiences of more than 10 years are 2 respondents, namely Project Manager and Product Owner presented in Table II.

b) *Organizational background*: This section describes the organization sector in the industry. The majority of respondents in the organization sector are banking/financial/insurance, about 29.11%. Then, the percentage of Software Houses is 27.22%, the percentage of the Oil and Gas and other mining industries is 13.29%, and other sectors are depicted in Table III.

2) *Descriptive statistics*: Descriptive statistics is used to measure the tendency and frequency of each item. Table IV presents that the respondents mostly agree (4) and strongly agree (5) for all items based on the Likert scale, which consists of Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), and Strongly Agree (5).

From Table IV, it can be seen that the average mean of all elements is 4.25. The average mean of the RE phase element is 4.36 confirming that the first research problem in terms of lack of clarity on which the Requirement Engineering phase to capture QR should be implemented is answered. Then, the average mean of the RE technique is 4.18 confirming that the first research problem in terms of lack of clarity on which the techniques used to capture QR implemented is also answered.

The average mean of documents used is 4.11 confirming that the second research problem in terms of lack of documentation of QR is answered. Then, the average mean of roles involved is 4.25 confirming that the third research problem in terms of the inadequate ability of user stories to capture QR by involving roles to address QR effectively is answered.

Table IV presents that our finding extends the previous work of [29-31] on how to capture QR, which was only implemented in the elicitation phase. According to Table VI, each of the RE phases is needed to implement for capturing QR early. It also extends the previous work of [24], according to Table IV, in terms of documentation used, roles involved, and the RE techniques are also needed for each RE phase. Furthermore, it extends the previous work of Alhaizaey et al. [33], according to Table IV, in terms of documentation used and the roles involved are also needed for each RE phase. On the other hand, according to Table IV, our findings are in line with the documentation used [32] and in terms of implementation for capturing QR in all the RE phases [24], as described in the research gap. In summary, this study fills the gaps from previous works by implementing all the RE elements. Additionally, it also uses comprehensive documentation, the roles involved, and the RE techniques for each RE phase. According to the statistical results, the mean distribution is more than 4 out of 5 suggesting that the framework (Fig. 6) used to capture QR is verified and shows a positive impact on dealing with capturing QR in ASD. However, this framework still needs to prove its effectiveness by conducting validation using case studies.

TABLE III. ORGANIZATION SECTORS

Organization Sectors	Frequency	Percentage
Banking/Financial/Insurance	46	29.11%
Software House	43	27.22%
Oil and Gas, and other mining industries	21	13.29%
Education/Training	9	5.70%
Telecommunication	6	3.80%
E-Commerce	5	3.16%
Manufacturing	5	3.16%
Transportation & Storage	4	2.53%
Healthcare	4	2.53%
Agriculture, Hunting & Forestry	4	2.53%
Construction	3	1.90%
Others	8	5.06%
Total	158	100%

TABLE IV. DESCRIPTIVE STATISTICS AND VARIABLES

Item		Median	Mode	Mean	Average Mean
Element	RE Phase	4	4	4.32	4.25
	Document	4	5	4.41	
	Roles	4	4	4.16	
	RE Technique	4	4	4.09	
RE Phase	Elicitation	4	4	4.04	4.36
	Analysis	5	5	4.42	
	Documentation	5	5	4.46	
	Validation	5	5	4.50	
Document (elicitation phase)	System-type document	4	4	4.04	4.11
	Domain-type document	4	4	4.09	
	ISO/IEC 25010 Model	4	4	4.11	
	Project History	4	5	4.29	
Document (analysis phase)	Mapping sheet between FR vs QR document	4	4	4.04	
	Mapping sheet between QR documents	4	4	4.00	
Document (documentation phase)	FR user story	4	5	4.29	
	QR user story	4	4	4.08	
	QR Decision	4	4	4.08	
Document (validation phase)	Checklist-based Reading document	4	4	4.04	
Roles (elicitation phase)	Developer Team	5	5	4.53	4.25
	Customer	4	5	4.27	
	Expert	4	5	4.29	
	Product Owner	5	5	4.46	
Roles (analysis phase)	Developer Team	4	4	4.15	
Roles (documentation phase)	Developer Team	4	4	3.80	
Roles (validation phase)	Developer Team	4	5	4.29	
	Customer	4	4	4.10	
	Expert	4	4	4.22	
	Product Owner	4	5	4.37	
RE Technique (elicitation phase)	Interview and Brainstorming Technique	5	5	4.46	4.18
RE Technique (analysis phase)	Interaction Matrix technique	4	4	4.03	
RE Technique (documentation phase)	Structured Natural Language technique	4	4	4.03	
RE Technique (validation phase)	Checklist-based reading technique	4	4	4.20	

V. CONCLUSION AND FUTURE WORK

This research proposes a framework for capturing QR by integrating the RE elements in Agile Software Development methods. The research result based on the framework verification using the questionnaire confirms that the ASD practitioners' review agrees with the proposed element and the component to capture QR in ASD. Thus, this framework offers a comprehensive way to handle QR in ASD while aligning with ASD Practice to reduce rework, time, cost, and even project failure. Furthermore, this framework also emphasizes that the RE phase should be iterative along with the process of capturing functional requirements, even requirements (either FR or QR) that arise in the middle of development to accommodate requirement change. For future research, this framework will be validated by using case studies to evaluate its effectiveness, which is in line with the ASD practice, to see the importance and impact of QR on ASD. The case studies will be implemented in selected companies that have used ASD methods for software development.

ACKNOWLEDGMENT

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research. Communication of this research is made possible through monetary assistance from Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

REFERENCES

[1] I. Sommerville, Software Engineering, 10th ed. England: PEARSON, 2016.

[2] S. Najihi, S. Elhadi, R. A. Abdelouahid, and A. Marzak, "Software Testing from an Agile and Traditional view," in Procedia Computer Science, Niagara Falls: Elsevier, 2022, pp. 775–782.

[3] A. S. Alhazmi, "Integrating Design Thinking Model and Items Prioritization Decision Support Systems into Requirements Management in Scrum," Florida Atlantic University, Florida, 2021.

[4] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development Methodologies and Trends," International Journal of Interactive Mobile Technologies, vol. 14, no. 11, 2020.

[5] A. Muhammad, A. Siddique, Q. N. Naveed, U. Saleem, M. A. Hasan, and B. Shahzad, "Investigating Crucial Factors of Agile Software Development through Composite Approach," Intelligent Automation and Soft Computing, vol. 27, no. 1, 2021.

[6] N. Govil and A. Sharma, "Validation of agile methodology as ideal software development process using Fuzzy-TOPSIS method," Advances in Engineering Software, vol. 168, 2022, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0965997822000357

[7] D. Satria, D. I. Sensuse, and H. Noprisson, "A systematic literature review of the improved agile software development," in 2017 International Conference on Information Technology Systems and Innovation (ICITSI), 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8267925

[8] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," Comput Stand Interfaces, vol. 49, 2017, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0920548916300708?via%3Dihub

[9] P. Serrador and J. K. Pinto, "Does Agile work? — A quantitative analysis of agile project success," International Journal of Project Management, vol. 33, 2015, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0263786315000071?via%3Dihub

[10] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," Inf Softw Technol, vol. 50, 2008, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950584908000256

[11] V. One, "16th Annual State of Agile Report," 2022. [Online]. Available: https://stateofagile.com/

[12] W. Behutiye et al., "Management of quality requirements in agile and rapid software development: A systematic mapping study," Inf Softw Technol, vol. 123, 2020, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S095058491930240X

[13] D. Kumar, A. Kumar, and L. Singh, "Non-functional Requirements Elicitation in Agile Base Models," Webology, vol. 19, 2022, [Online]. Available: https://www.researchgate.net/publication/358057432_Non-functional_Requirements_Elicitation_in_Agile_Base_Models

[14] S. Rahy and J. M. Bass, "Managing non-functional requirements in agile software development," IET Software, vol. 16, 2021, [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/sfw2.12037

[15] W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen, and A. Abhervé, "Towards optimal quality requirement documentation in agile software development: A multiple case study," Journal of Systems and Software, vol. 183, 2021, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121221002090

[16] B. M. Aljallabi and A. Mansour, "Enhancement approach for non-functional requirements analysis in Agile environment," in 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), IEEE, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7381407

[17] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," Inf Softw Technol, vol. 110, pp. 39–55, 2019.

[18] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de O. Neto, "Requirements engineering challenges and practices in large-scale agile system development," Journal of Systems and Software, vol. 172, 2021, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121220302417

[19] L. López et al., "Quality measurement in agile and rapid software development: A systematic mapping," J Syst Softw, vol. 186, 2021, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121221002661

[20] E. Sherif, W. Helmy, and G. H. Galal-Edeen, "Managing Non-functional Requirements in Agile Software Development," in International Conference on Computational Science and Its Applications, Malaga, Spain: Springer, 2022, pp. 205–216.

[21] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements engineering: A systematic mapping study in agile software development," J Syst Softw, vol. 139, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0164121218300141

[22] D. Ismail and Arviansyah, "A Systematic Literature Review and Delphi Study on Agile Software Development Challenges," in 6th International Conference on Management in Emerging Markets (ICMEM), Bandung: IEEE, 2021.

[23] A. Muhammad, A. Siddique, M. Mubasher, A. Aldweesh, and Q. N. Naveed, "Prioritizing Non-Functional Requirements in Agile Process Using Multi Criteria Decision Making Analysis," IEEE Access, vol. 11, 2023, [Online]. Available: https://ieeexplore.ieee.org/document/10061380

[24] E. Sherif, W. Helmy, and G. H. Galal-Edeen, "Proposed Framework to Manage Non-Functional Requirements in Agile," IEEE Access, vol. 11, pp. 53995–54005, 2023.

[25] W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen, and A. Abhervé, "How agile software development practitioners perceive the need for documenting quality requirements: a multiple case study," in 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia: IEEE, 2020.

[26] A. M. Almanaseer, W. Alzyadat, M. Muhairat, S. Al-Showarah, and A. Alhroob, "A proposed model for eliminating nonfunctional requirements in Agile Methods using natural language processes," in 2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA), IEEE, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/10009796

[27] S. Kopczyńska, M. Ochodek, and J. Nawrocki, "On Importance of Non-functional Requirements in Agile Software Projects—A Survey," in Integrating Research and Practice in Software Engineering, Switzerland: Springer, 2020, pp. 145–158. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-26574-8_11

[28] L. López, J. Partanen, P. Rodríguez, and S. Martínez-Fernández, "How Practitioners Manage Quality Requirements in Rapid Software Development: A Survey," in 2018 IEEE 1st International Workshop on Quality Requirements in Agile Projects (QuaRAP), IEEE, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8501270

[29] M. Younas, I. Ghani, D. N. A. Jawawi, and R. Kazmi, "Non-Functional Requirements Elicitation Guideline for Agile Methods," Journal of Telecommunication, Electronic and Computer Engineering, vol. 9, 2017.

[30] M. Younas et al., "Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment," IEEE Access, vol. 8, 2020, [Online]. Available: https://ieeexplore.ieee.org/document/9178791

[31] A. Jarzębowicz and P. Weichbroth, "A Qualitative Study on Non-Functional Requirements in Agile Software Development," IEEE Access,

vol. 9, 2021, [Online]. Available: https://ieeexplore.ieee.org/document/9371679

[32] W. Behutiye, P. Rodriguez, and M. Oivo, “Quality Requirement Documentation Guidelines for Agile Software Development,” IEEE Access, vol. 10, 2022, [Online]. Available: https://ieeexplore.ieee.org/document/9810243

[33] A. Alhaizaey and M. Al-Mashari, “A Framework for Reviewing and Improving Non-Functional Requirements in Agile-based Requirements,” in 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal: IEEE, 2023.

[34] L. Chung and J. C. S. do P. Leite, “On Non-Functional Requirements in Software Engineering,” in Conceptual Modeling: Foundations and Applications, vol. 5600, Springer, 2009, pp. 363–379.

[35] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, and T. Suzuki, “Non-functional requirements in industry - three case studies adopting an experience-based NFR method,” in 13th IEEE International Conference on Requirements Engineering (RE’05), Paris, France: IEEE, 2005.

[36] B. W. Boehm, J. R. Brown, and H. Kaspar, Characteristics of software quality, 2nd ed. North-Holland Publishing Company, 1978.

[37] J. McCall, “Factors in software quality,” 1977.

[38] ISO/IEC JTC 1/SC 7, “ISO/IEC 25010,” https://iso25000.com/index.php/en/iso-25000-standards/iso-25010.

[39] C. Boyce and P. Neale, CONDUCTING IN-DEPTH INTERVIEWS: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input, vol. 2. Pathfinder International, 2006.

[40] S. W. Ambler, “Beyond functional requirements on agile projects - Strategies for addressing nonfunctional requirements,” Doctor Dobbs Journal, vol. 33, no. 10, pp. 64–66, 2008.

[41] V. Sachdeva and L. Chung, “Handling non-functional requirements for big data and IOT projects in Scrum,” in 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, IEEE, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7943152

[42] K. Beck et al., “Manifesto for Agile Software Development.”

[43] D. Mairiza and D. Zowghi, “Constructing a Catalogue of Conflicts among Non-functional Requirements,” in International Conference on Evaluation of Novel Approaches to Software Engineering, Athens, Greece: Springer, 2010, pp. 31–44.

[44] D. Mairiza, D. Zowghi, and N. Nurmuliani, “An investigation into the notion of non-functional requirements,” in Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland: Association for Computing Machinery, 2010, pp. 311–317.

[45] A. E. Sabry and S. S. El-Rabbat, “Proposed framework for handling architectural NFR’s within scrum methodology,” in International Conference Software Engineering Research and Practice, SERP, 2015, p. 238.

[46] G. Sabaliauskaite, F. Matsukawa, S. Kusumoto, and K. Inoue, “An Experimental Comparison of Checklist-Based Reading and Perspective-Based Reading for UML Design Document Inspection,” in Proceedings International Symposium on Empirical Software Engineering, Nara, Japan: IEEE, 2002.

[47] N. K. Rad and F. Turley, Agile Scrum Handbook, 2nd ed. Van Haren Publishing, Zaltbommel, 2018.

[48] Miro, “What is brainstorming?,” https://miro.com/brainstorming/what-is-brainstorming/.

[49] K. Wiegers and J. Beatty, Software Requirements, 3rd Edition. Redmond, Washington: Microsoft Press, 2013.

[50] R. Ibrahim, N. A. M. Asri, S. Jamel, and J. A. Wahab, “Validation of Requirements for Transformation of an Urban District to a Smart City,” Int J Adv Comput Sci Appl, vol. 12, no. 7, pp. 322–328, 2021.

[51] N. A. M. Asri, R. Ibrahim, and S. Jamel, “Designing a Model for Smart City through Digital Transformation,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 1.3, pp. 345–351, 2019.

[52] J. Frost, “Cronbach’s Alpha: Definition, Calculations & Example,” https://statisticsbyjim.com/basics/cronbachs-alpha/#:~:text=Cronbach%E2%80%99s%20alpha%20coefficient%20measures%20the%20internal%20consistency%2C%20or,agreement%20on%20a%20standardized%200%20to%201%20scale.

[53] J. F. Hair, G. T. M. Hult, C. M. Ringle, and M. Sarstedt, “A Primer on Partoa; east Squares Structural Equation Modelling (PLS-SEM),” International Journal of Research & Method in Education, vol. 38, 2016.

[54] D. W. Zimmerman, “A note on the influence of outliers on parametric and nonparametric tests,” Journal of General Psychology, vol. 121, no. 4, pp. 391–401, 1994.

[55] B. G. Tabachnick and L. S. Fidell, Using Multivariate Statistics, 7th ed. Boston: Pearson Education, 2021.

[56] Holmes-Smith P, Coote L, and Cunningham E, Structural Equation Modeling. School Research, Evaluation and Measurement Services, 2006.