ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Current Practices and Required Knowledge of Non-Functional Requirements Elicitation in Agile Context: Key Results of an Empirical Study

Mohammed Saleh[1], Fauziah Baharom[2] and Shafinah Farvin Packeer Mohamed[3]

ABSTRACT

The Non-Functional Requirements (NFRs) play a crucial role in guaranteeing the security, usability, reliability, performance, and effectiveness of the system, underscoring their significant importance. Nevertheless, within Agile methodologies, NFRs are frequently overlooked until the later stages of the Agile process. The primary hurdle in the domain of Agile requirements elicitation lies in the absence of early-phase guidelines for eliciting NFRs. Even though the guidelines for NFRs elicitation are considered important, the empirical studies on the current elicitation practices of NFRs in the Agile context are notably scarce, especially those related to security ones. Also, there is a lack of empirical studies on the elicitation knowledge of NFRs. Therefore, this study was surveyed to investigate the current elicitation practices and the required knowledge of NFRs focusing on security requirements (SRs) in Agile environments. The survey utilized the questionnaire to collect the data, which included 46 questions with sub-questions. These questions were answered by 121 software practitioners. The questionnaire outcomes defined sixteen (16) practices of NFRs elicitation and ten (10) practices of SRs elicitation. Also, the study defined the required knowledge for elicitation in an Agile context. On top of that, the vast majority of respondents agree that the guidelines for NFR elicitation are significant and will help enhance the understanding of the stakeholders about the elicitation process. Thus, the outcomes of this study will be used to propose a guideline for NFR elicitation in an Agile context.

Article information:

Keywords: Agile Requirements Engineering, Non-functional Requirements Elicitation, Security Requirements, Required Elicitation Knowledge

Article history:
Received: February 12, 2024
Revised: March 25, 2024
Accepted: June 27, 2024
Published: July 13, 2024 (Online)

DOI: 10.37936/ecti-cit.2024183.255650

1. INTRODUCTION

Agile software methods have undergone continuous development and evolution since the early 1990s. These methods employ a short development life cycle characterized by an iterative and incremental process, aiming to deliver a high-quality product within the estimated budget and time constraints. As a result, Agile methods have gained widespread use, particularly in business sectors where requirements exhibit a degree of instability [1], [2]. Requirements form the foundation for all software products, aiming to ensure accurate comprehension of customer needs [1], [3].

Requirements engineering (RE) stands as a critical software process, delineating the methods for collecting, documenting, reviewing, and achieving requirements [4]. In traditional approaches, such as the waterfall method, requirements are thoroughly and comprehensively outlined before commencing design work. Conversely, in Agile methods such as eXtreme Programming (XP), SCRUM, and Dynamic Systems Development Method (DSDM), the approach diverges. Typically, requirements are conveyed straightforwardly by generating user stories at the onset of the Sprints [5]. When NFRs are neglected, notable effort and cost problems arise. In the initial phases of a project, customers may not have a complete awareness of all NFRs. Instead, their focus tends to be on fully developing functional requirements (FRs), which can result in challenges associated with a reduction in software quality and an escalation in maintenance expenses [6]. Despite the significance of NFRs, they are typically informally addressed in Agile methods [8]. For instance, SCRUM

[1,2,3] The authors are the Universiti Utara Malaysia, Malaysia, E-mail: m.rawajfeh@yahoo.com, fauziah@uum.edu.my and shafinah@uum.edu.my

captures requirements on 3x5 index cards, with each requirement described in 3–5 sentences, a format that falls short of capturing detailed NFRs [7].

Moreover, gathering all requirements from a single person, such as the Product Owner (PO) in SCRUM, proves challenging, particularly when the PO lacks knowledge during the elicitation stage [7], [9]. Therefore, fostering good knowledge and understanding among software practitioners becomes crucial to facilitating NFR elicitation [10]. Clearly defining the required knowledge for NFR elicitation enhances the comprehension of software practitioners in requirements elicitation, guiding them on what considerations to bear in mind during this stage.

On top of that, recent studies have highlighted that the primary problem in requirements elicitation in the Agile context is the absence of elicitation guidelines for NFRs [1], [11], [12]. A guideline for requirements elicitation is a collection of activities offering guidance on how to carry out the process of gathering requirements from stakeholders [12]. While the elicitation of NFRs is acknowledged as crucial, there is a notable dearth of empirical studies examining current practices in NFRs elicitation within the context of Agile companies [13], [14].

This gap in research is particularly pronounced concerning the elicitation of SRs [15]. Undoubtedly, identifying the current practices of NFR elicitation can help get more information about the NFRs and obtain suggestions for the composition of the NFR elicitation procedure. Besides, it helps formulate the guideline for NFR elicitation in an Agile context. Moreover, it is impossible to create secure software without keeping security in mind from the beginning of elicitation [15]. Thus, this paper conducted an exploratory study to define the current practices for NFR elicitation, focusing on SRs.

The structure of the paper is as follows: Section 1 presents an introduction to Agile RE and NFRs elicitation. Section 2 discusses the pertinent literature concerning Agile NFR elicitation. Section 3 outlines the research methodology. Section 4 explores the study's findings. Section 5 provides a detailed elaboration on these findings. Finally, Section 6 presents conclusive comments, summarizing key points and suggesting directions for future research.

2. EXISTING STUDIES RELATED TO NFRS ELICITATION

Several studies, including Alam *et al.* [2], Inayat *et al.* [16], and Elghariani *et al.* [17], have explored Agile RE practices through comprehensive literature reviews. Alam *et al.* [2] identified fourteen (14) Agile RE practices along with the challenges practitioners encounter in their implementation. Inayat *et al.* [16] outlined seventeen practices illustrating how Agile methods address requirements, and Elghariani *et al.* [17] presented findings similar to those of Inayat *et al.* [16]. However, these investigations generally focused on Agile RE as a whole, with limited emphasis on NFR elicitation practices or the required knowledge for elicitation. Wagner *et al.* [14] added to our empirical understanding of the current practices and challenges in Agile requirements elicitation. Their survey spanned North America, South America, Central Europe, and Northern Europe, investigating Agile practices and issues. The survey provided insights into the methods of eliciting and documenting requirements, as well as common problems in Agile RE. However, the emphasis of the survey was predominantly on FR elicitation, with comparatively less attention given to NFR elicitation.

Furthermore, Kopczyńska *et al.* [18] conducted a survey exploring the viewpoints of Agile software development practitioners concerning the importance of delineating NFRs in their projects. Their findings revealed that more than 77% of respondents consider having NFRs defined in Agile software projects to be, at the very least, important, with 30% emphasizing their criticality. Additionally, the study observed a positive correlation between the perceived importance of NFRs and the respondents' level of experience. It is noteworthy. However, the study primarily addressed the significance of defining NFRs in Agile software projects and did not delve into the current practices of NFRs elicitation or the requisite knowledge for the elicitation process in the Agile environment.

Kotze [19] surveyed employees in software development organizations in South Africa to assess stakeholders' knowledge of FR elicitation using traditional methods. Similarly, the study did not delve into the current practices and required knowledge for Agile NFR elicitation. Additionally, Ochodek *et al.* [20] performed an online survey within social network groups to gather insights into Agile RE practices and their perceived importance. The study established a seven-tier ranking of these practices but primarily focused on the broader RE process, with less emphasis on NFR elicitation practices and the required knowledge for software practitioners, especially concerning SRs elicitation.

It seems that the majority of existing studies have concentrated on Agile RE in a general sense, with minimal consideration given to current practices and elicitation knowledge of NFRs, especially those related to security. Furthermore, some studies focused on traditional methods or FR elicitation, but they did not specifically address the Agile context. Furthermore, the bulk of these studies have been conducted in Europe, South America, and North America, with limited research originating from Asian countries [16].

Notably, Jordan, among these Asian nations, has witnessed expansion in its software development industry, propelled by incentive measures instituted by the Jordanian government [21]. In addition, there are more than 300 companies related to Agile software development mentioned by the Jordanian Ministry of Trade and Industry. This makes Jordan a fertile environment for conducting such a survey in this field. Therefore, this survey was conducted on Jordanian Agile software development companies.

3. RESEARCH METHOD

The data were collected through a questionnaire based on the guidelines provided by Mills *et al.* [20] and Zikmund et al. [23]. The main purpose of selecting the questionnaire is to collect the data because it covers wide-access samples with minimum cost [24]. In addition, the use of questionnaires facilitates data analysis as well as sustaining a high degree of privacy [25], [26], and [27]. There were five (5) phases involved in this study, as elaborated next.

3.1 Questionnaire Construction

The first phase is questionnaire construction. After reviewing the related works on NFR elicitation in the Agile context, four research questions were outlined, as follows:

RQ1: What are the current practices of NFR elicitation in an Agile context?

RQ2: What are the current practices of SRs elicitation in an agile context?

RQ3: What is the required knowledge of elicitation in an Agile context?

RQ4: What is the software practitioners' opinion on the importance of the guideline for NFR elicitation in an Agile context?

These research questions served as the foundation for constructing the questionnaire. The questionnaire encompasses five (5) primary sections: demographic information, current practices for eliciting NFRs, current practices for eliciting SRs, the required knowledge of software practitioners for NFRs elicitation, and the implementation of NFRs elicitation guidelines. These sections comprise a total of 46 questions, including both multiple-choice and Likert-scale questions [28]. These types of questions are utilized for many purposes, such as being easy to adapt, economical, and fast in the results [28]. The summary of each section is provided next.

Section I: The demographic information and the organization's background

This section contains eight (8) questions to assess the qualifications of respondents, such as the current position, the activities of the position, and years of experience. Also, this section assesses the organization's background, such as the type of organization, size, and the Agile methods used in these organizations.

Section II: The current practices for NFR elicitation

This section is intended to investigate the current practices for NFR elicitation in an Agile context. It consists of seventeen (17) questions, such as NFR elicitation techniques, software types, historical data, detached user stories, NFR sources, and NFR validation.

Section III: The current practices for SRs elicitation

This section investigates the current practices of SRs elicitation in Agile Software Development (ASD). It comprises ten (10) questions such as the responsibility of eliciting the SRs from the customer, eliciting and documenting SRs explicitly, the notation used to represent the SRs, and the importance of considering these requirements.

Section IV: The required knowledge of software practitioners for NFR elicitation

This section examines the required knowledge for software practitioners for NFR elicitation. It contains seven questions related to the domain, communication, and system of NFR elicitation in an Agile context.

Section V: The Implementation of NFR elicitation guideline

This section comprises four (4) questions designed to capture the perspectives of software practitioners regarding the importance and benefits of NFRs elicitation guidelines within the Agile context. Additionally, it seeks to identify the names and elements of the NFR guidelines employed by respondents.

3.2 Sampling

This study applied purposive sampling (judgment) that concerned choosing a unique sample with a certain feature [29]. The sample was selected among the Agile software practitioners in the biggest city in Jordan, which is the capital, Amman.

The Agile software companies were identified through three (3) means: 1) the compilation of companies listed by the Jordanian Ministry of Trade and Industry and the Jordan Business Directory Website, 2) information gathered from government and private organizations through the Internet and social media, and 3) reaching out to friends with relations in the Agile software industry. The sample size of this study is 121 software practitioners.

According to Taasoobshirazi, Gita, and Wang [30], a sample size of 100 is sufficient. Besides, the minimum sample size of 30 is acceptable for statistical analysis [31]. Hence, the survey's sample size is deemed adequate.

3.3 Pilot Study

Before the real survey, a pilot study was conducted. The pilot study was performed to examine the respondents' understandability of the questions and the time taken to answer the questionnaire. Besides, it also ensures the validity of the questionnaire, completeness of the included items, and readability

[32], [33]. Crucially, the pilot test serves to preempt any ambiguities, obstacles, or errors that may emerge when respondents interact with the questionnaire. A total of 34 participants took part in the pilot study, a suitable size as suggested by Cooper et al. [30], where a pilot group between 25 and 100 is considered sufficient. The pilot study employed both face-to-face meetings and online interviews. Initially, face-to-face meetings were utilized. However, when challenges arose, such as scheduling appointments or assessing the distance from the respondent's location, the study seamlessly transitioned to online interviews. The questionnaire was answered by Agile software practitioners in Jordan who have at least six years of working experience, such as PO and Agile Analyst (AA). They have the responsibility for eliciting the software requirements in an Agile context [34].

The results of the pilot study are discussed in Saleh, et al. [35]. The suggestions obtained from this pilot study were used to improve the questionnaire. Among the suggestions are making the questions simpler for better understanding and readability and adding conditional questions about applying several practices to allow only whoever applied these practices to answer the questions related to them. Based on that, the questionnaire was improved.

3.4 Data Collection

For the real survey, the potential respondents were contacted from the lists of organizations that were obtained by email, telephone, and social media, and they were asked for their willingness to participate in the survey (238 of them). Google Forms was used to create an online survey.

The Google Forms link was sent via email to the respective respondents who had consented to take part in the survey. The online questionnaire remained accessible for a duration of two months. A total of 121 responses were usable for this study. Regrettably, 98 were rejected due to incomplete answers. Also, there were nineteen (19) unreturned questionnaires. Table 1 provides an overview of the responses.

Table 1: *Overview of respondents.*

Details	Number	Percentage
Potential respondents	238	100%
Complete online survey	121	51%
Incomplete online survey	98	41%
Rejected/ Unreturned questionnaires	19	8%
Total usable	121	51%

3.5 Data Analysis

The gathered data was input into Statistical Package for the Social Sciences (SPSS) version 14.0 for analysis. The primary objective of employing SPSS is to concentrate on describing the opinions of respondents or the frequency of specific events [36]. The analyses included in the study encompassed frequency, mean, and cross-tabulation.

4. FINDINGS

This section aims to report the findings of the survey. They are classified into five (5) subsections, which are: 1) demographic information (respondents' background and organization's background), 2) the current practices of NFRs elicitation, 3) the current practices of SRs elicitation, 4) the required knowledge of NFRs elicitation, and 5) the implementation of NFRs guidelines in an Agile context.

4.1 Respondents' Background

This section is intended to evaluate both the background of the respondents and their organization. To comprehend the respondents' background, they were requested to specify their position within the company and indicate the number of years of experience in Agile software development. Table 2 presents the frequency and percentages of respondents according to their positions. Most of the respondents are programmers (38%), and POs (37%), followed by security advisors (13 %), and team leaders (5%). The rest of them are project managers (4%) and system analysts (3%).

Table 2: *Respondents' position in the company.*

Positions	Frequency	Percentage
Programmers	46	38%
Product Owner	45	37%
Security Advisor	16	13%
Team Leader	6	5%
Project Manager	5	4%
System Analyst	3	3%
Total	121	100%

In addition, the respondents were categorized depending on their experience and position by using the cross-tabulation analysis, as depicted in Table 3. Out of the 121 respondents, 66 have experience between 6 and 10 years in Agile software development. 32 of the respondents have 1–6 years' experience, and 20 have experience of more than ten (10) years. Besides, only three (3) of the respondents have experience of less than one (1) year. This means that most of the respondents to this survey have more than six (6) years of experience in Agile software development.

Table 3: *Respondents' experience.*

Positions	< 1	1–5 Years	6-10 Years	11–20 Years	Total
Programmer	3	23	17	3	46
Product Owner	0	8	32	5	45
Security Advisor	0	1	10	5	16
Team leader	0	0	3	3	6
Project manager	0	0	2	3	5
System Analyst	0	0	2	1	3
Total	3	32	66	20	121

4.2 Organization's Background

The type of organizations for respondents was found as follows: the private sector is the highest-ranking (81%), followed by the government agency (19%). Moreover, the respondents are categorized based on the organization's size. Table 4 portrays the frequency and percentages of respondents according to their organization's size.

Table 4: *The organizations' size for the respondents.*

Organizations size	Frequency	Percentage
Less than 10 people	5	4.1%
10–30 people	32	26.5%
31–50 people	45	37.2%
More than 50 people	39	32.2%
Total	121	100%

The majority of organizations are between 31 and 50 people (37.2%), followed by organizations that have more than 50 people (32.2%), and those with 10–30 people (26.4%). The rest of the organizations consist of less than ten (10) people (4.1%). Figure 1 shows the Agile methods applied by the respondents in their organizations. The respondents had the option to choose more than one answer for this question.

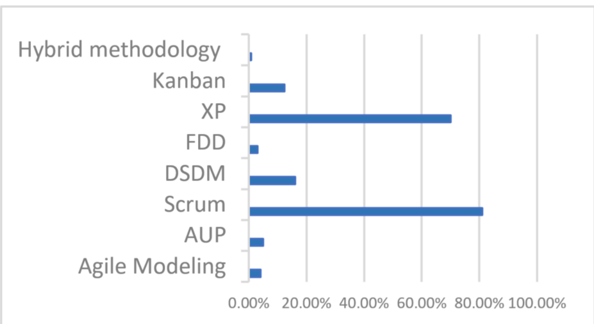

Fig.1: *The percentages of Agile methods applied by the respondents.*

Most of them used Scrum (81%), followed by XP (70%). The rest used DSDM (16%), Kanban (12.4%), FDD (3%), AUP (5%), and Agile Modeling (4%). Besides, only one respondent applies hybrid methodology (0.8%). The next subsections discuss the findings based on the outlined research questions.

4.3 The Current Practices of NFRs Elicitation

RQ1: What are the current practices of NFR elicitation in an Agile context?

This section addresses the software practitioners' opinions and experiences regarding the current practices of NFR elicitation in an Agile context by defining if they elicit NFRs in the early phase, who performs the elicitation of NFRs, the application domains and system types, the type of elicitation techniques commonly used for NFRs, the standards/theories used to define NFRs types, and who performs the validation of NFR. These are elaborated next.

4.3.1 Eliciting NFRs in the early phase

The respondents were asked whether their team elicited NFRs in the early phase or not. The respondents who answered "Yes" were allowed to continue answering the rest of the questionnaire questions 58%. Conversely, 42% of the respondents who answered "No" did not participate in the survey. This is to ensure that only those who have experience in NFR elicitation participate in the survey.

4.3.2 The Responsible Person for Performing NFRs Elicitation

The respondents were asked about the individual responsible for eliciting NFRs within their organizations. As depicted in Figure 2, it is evident that the majority of respondents identified the PO (93%), with smaller percentages indicating the Team Leader (4%), Programmer (2%), and Project Manager (1%). The responsible person is to perform NFRs elicitation.

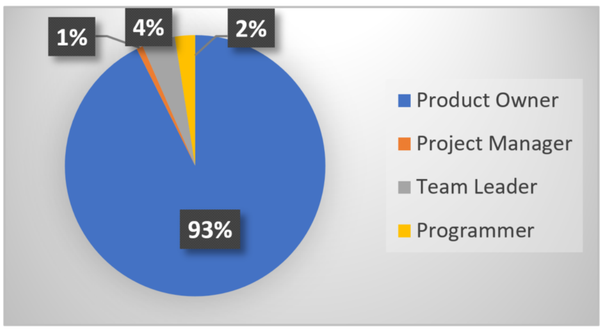

Fig.2: *The responsible person is to perform the NFRs elicitation.*

4.3.3 Types of Applications

The respondents were asked about the types of applications they are currently developing, with the option to choose more than one answer. The predominant responses included Web applications (89.3%) and Mobile applications (81.8%). Subsequent answers encompassed Telecommunication Services (46.3%), E-learning (42.1%), E-banking (29.8%), and E-commerce (19%). Figure 3 shows the results.

4.3.4 Types of Systems

The respondents were queried about the types of systems they are engaged in developing, with the option to select more than one answer. The responses included Real-time systems (83.5%), Information systems (76%), Safety-critical systems (74.4%), and controlled systems (72.7%). Refer to Figure 4 for an illustration of the results.

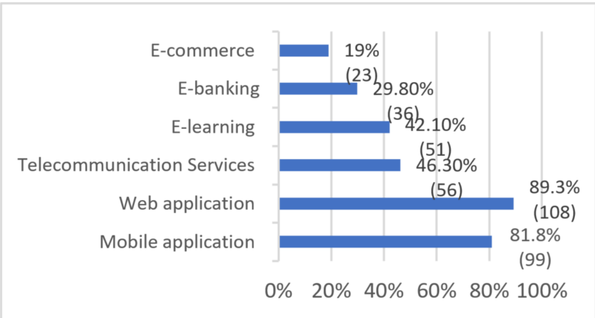

Fig.3: *The application domains.*

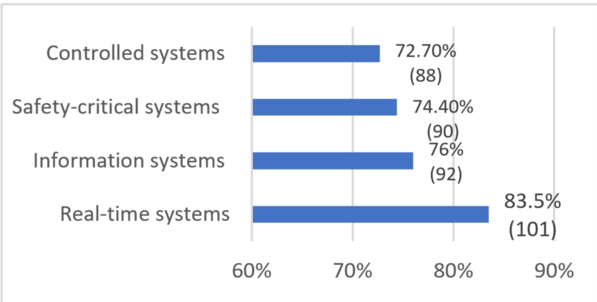

Fig.4: *The types of systems.*

4.3.5 The Standards and Theories Used for NFRs Classification

Figure 5 shows the standards and theories used to define the types of NFRs that are utilized by the respondents. For this question, the respondents were permitted to select more one answer. Most of them use ISO/IEC 25010 (92.5%) and ISO/IEC 9126 (25%) to define NFRs types, while the rest use the Chung framework (11.7%) and Somerville framework (18.3%).

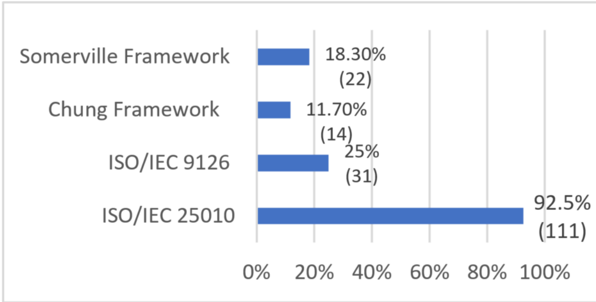

Fig.5: *The standard/theories utilized to define NFRs.*

4.3.6 The Validation Process of NFRs

Respondents were queried about whether their teams validated the NFRs after elicitation. Unfortunately, more than half of them (51%) do not conduct validation, while the remaining 49% perform validation after eliciting the NFRs. Figure 6 portrays these results.

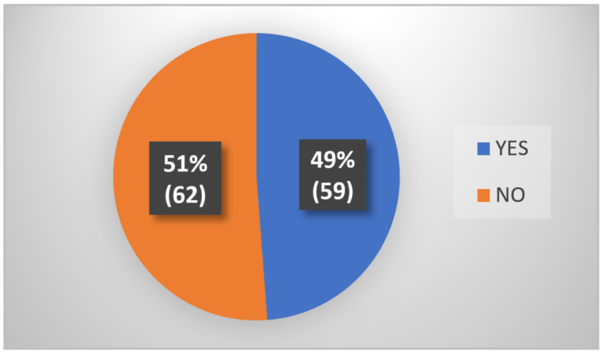

Fig.6: *The validation of NFRs after elicitation.*

4.3.7 The Responsible Person for NFRs Validations

The respondents who answered "Yes" to the previous question were permitted to answer this question. They are the ones who perform the NFR validation. Additionally, respondents had the option to choose more than one answer

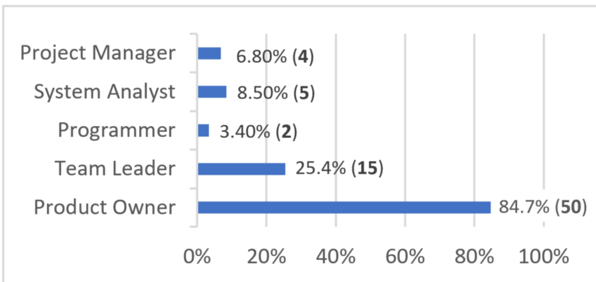

Fig.7: *Responsible for validating NFRs.*

As depicted in Figure 7, the majority of respondents (84.7%) indicated that the PO is responsible for validating NFRs after elicitation. On the other hand, 25.4% of the respondents answered team leader. Meanwhile, the rest chose system analyst, project manager, and programmer, 8.5%, 6.8%, and 3.4% respectively.

4.3.8 The Validation Methods Used for NFRs

Figure 8 shows the validation methods used for NFRs. For this question, the respondents were permitted to select more than one answer.

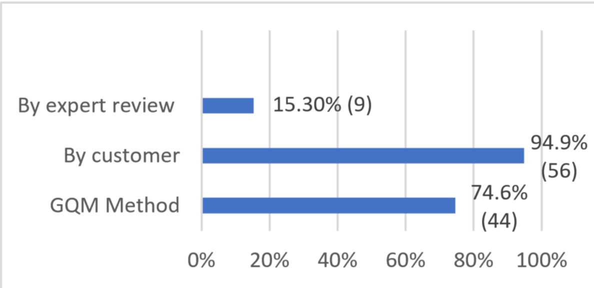

Fig.8: *The validation methods used for NFRs.*

The majority of the respondents validate NFR with their customers (94.9%), while 74.6% use the GQM method. The rest (15.3%) use expert reviews.

4.3.9 The Acceptance Degree of NFRs Elicitation Practices

Moreover, respondents were inquired about the practices employed for eliciting NFRs in an Agile context. They were requested to rank the level of acceptance regarding the application of these practices within their companies. Utilizing a Likert scale with five points, ranging from Strongly Disagree (value 1) to Strongly Agree (value 5), respondents provided their assessments of the degree of acceptance for each practice. The mean score was computed using a convenient interval that represents the actual mean for obtaining the results.

Table 5 illustrates the representation of the degree of acceptance for each practice.

Table 5: *The degree of acceptance and mean interval presentation.*

Mean interval presentation	Degree of acceptance
1 to 1.80	Strongly Disagree
1.81 to 2.60	Disagree
2.61 to 3.40	Neutral
3.41 to 4.20	Agree
4.21 to 5	Strongly Agree

The following equation (1) was used to calculate the interval [37], [38].

$$Appropriate\ interval = (n-1)/n \tag{1}$$

The maximum number on the scale is represented by N. In this study, the N value is five (5). Therefore, the appropriate interval for the study = (4/5) = 0.8. Table 6 presents the mean values of NFR elicitation practices and the degree of acceptance.

Table 6: *The mean values for practices of non-functional requirements elicitation.*

NFRs elicitation practices	Mean value	Degree of acceptance
Delineating the application domain and system type at the project's outset aids in the elicitation stage by defining potential non-functional requirements	4.34	Strongly Agree
Emphasizing face-to-face communication for the elicitation of non-functional requirements by the individual(s) responsible for the process.	4.67	Strongly Agree
Encouraging customer participation and face-to-face communication through natural language	4.34	Strongly Agree
Utilizing historical data from past projects to define and elicit the non-functional requirements for new software projects.	3.53	Agree
Upon determining the project's scope and potential non-functional requirements, a set of questions is formulated to extract these requirements from the customer.	3.52	Agree
Separating non-functional requirements by detaching story cards that provide a detailed and clear explanation of these requirements.	3.53	Agree
It is crucial to engage in the validation of non-functional requirements in natural language with the customer, ensuring these requirements are comprehensible to them	3.53	Agree
Validating non-functional requirements through a third party, such as an expert reviewer, is imperative to ensuring project quality.	2.31	Disagree

Based on Table 6, the findings indicate the level of acceptance for NFR elicitation practices. Specifically, three (3) practices received a Strongly Agree rating, five (5) practices garnered an Agree rating, and one (1) practice was rated as Disagree.

4.4 The Current Practices of SRs Elicitation

RQ2: What are the current practices of SRs elicitation in an Agile context?

This section addresses the software practitioners' opinions and experiences regarding the current practices for SRs elicitation in an Agile context. They were asked whether SRs are elicited and documented explicitly, the responsible person for performing the elicitation of SRs, the notation used to represent the SRs, the importance of considering and eliciting these requirements from the early phases, and the acceptance degree of SRs elicitation practices. These are elaborated next.

4.4.1 Explicit SRs Elicitation And Documentation

The respondents are queried whether they elicit the SR explicitly and document it. The majority of respondents answered that they only dealt with the security issues during the implementation stage or after developing the system (63%), while 35% of them answered that they elicit and document SRs explicitly.

Besides, 1% of the respondents discuss SRs from the early stages, but they are not documented. On the other hand, 1% of the respondents did not deal with these requirements at all, as shown in Figure 9.

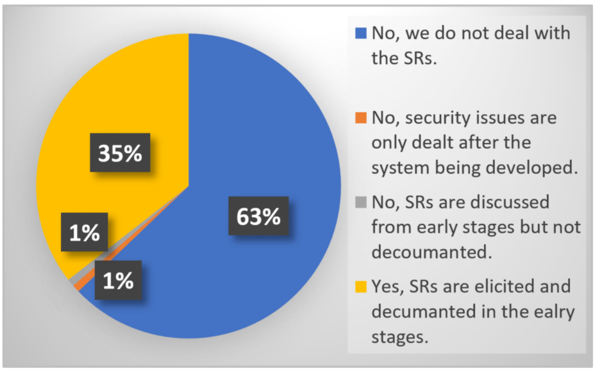

Fig.9: *Explicit SRs elicitation and documentation.*

4.4.2 The Responsible Person To Perform The Elicitation Of SRs

The respondents were asked about the person responsible for performing the elicitation of SR. It is apparent from Figure 10 that the majority of respondents answered that the PO has the responsibility to elicit SR 89.3%, followed by the team leader at 4.1%. Subsequently, the security advisor, project manager, and programmer each represent 1.7%. Another 1.7% mentioned no specific person, while 0.8% mentioned a system analyst.

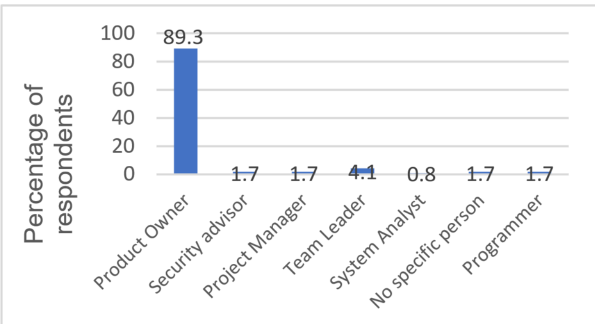

Fig.10: *The responsible person who performs the elicitation of SRs.*

4.4.3 The Notation Used to Represent SRs

The respondents who elicit SRs were also asked about the notation that is used to represent the SRs. 36% of respondents do not use a specific notation, besides, 6% of respondents do not document the SRs. 30% use misuser stories to represent these requirements, 20% use misuse cases, 4% use attack trees, and 4% use Abuse cases, as shown in Figure 11.

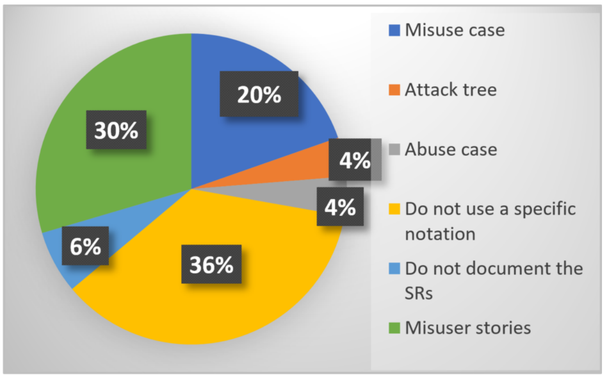

Fig.11: *The notation is used to represent the security requirements.*

4.4.4 The importance of Considering SRs from the early phases

The respondents were asked about the importance of considering and eliciting SRs from the early phases of ASD to ensure software quality. 98% of respondents agree with the importance of considering and eliciting the SRs in the early phases. Conversely, only 2% of respondents do not correspond with that, as shown in Figure 12.

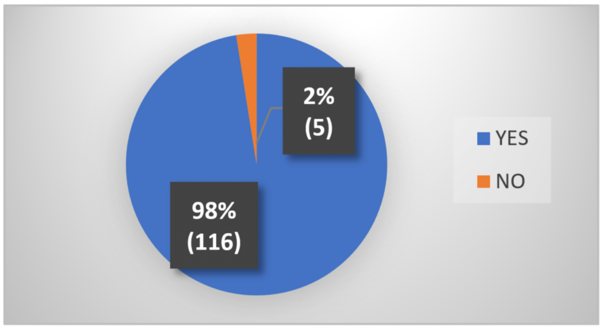

Fig.12: *The importance of considering and eliciting the SRs from the early phases.*

4.4.5 The Acceptance Degree Of SRs Elicitation Practices

The respondents were asked about the SRs elicitation practices to investigate the importance of each of them in the real environment. Thus, Five Likert scales ranging from Strongly Disagree (value 1) to Strongly Agree (value 5) were used. Table 7 presents the mean values of SRs elicitation and the degree of acceptance. It appears from Table 7 that most SR practices were rated as Strongly Agree and only two (2) practices were rated as Disagree.

Table 7: *The mean values for security requirements elicitation.*

Security Requirements and Elicitation Practices	Mean value	Degree of acceptance
Eliciting the security requirements explicitly.	4.57	Strongly Agree
Eliciting security requirements separately from other non-functional requirements by the person(s) responsible for elicitation.	2.31	Disagree
Documenting security requirements using a specific notation, such as a misuse case or attack tree.	4.47	Strongly Agreed
Utilizing the common attacks that occurred previously to elicit the security requirements for a new project.	4.31	Strongly Agreed
Taking into account the perspective of potential attackers during the elicitation of security requirements.	4.47	Strongly Agreed
Reaching a shared understanding of security requirements among those responsible for elicitation, including the customer.	2.40	Disagree

4.5 The Required Knowledge Of NFRs Elicitation

RQ3: What is the required knowledge of elicitation in an Agile context?

Respondents were inquired about the required knowledge for NFR elicitation, aiming to determine the necessary knowledge during the elicitation stage in an Agile context. They were tasked with ranking the level of application of these practices within their companies using a Likert scale consisting of five (5) points, ranging from Strongly Disagree (value 1) to Strongly Agree (value 5). Table 8 displays the mean values of the required knowledge and the level of acceptance. Table 8 reveals that two (2) aspects of required knowledge received a Strongly Agree rating, three (3) received an Agree rating, and the remaining two (2) received a Disagree rating.

Table 8: *The mean values for the required knowledge of elicitation.*

The Required Knowledge of NFRs Elicitation	Mean value	Degree of acceptance
Software practitioners engaged in elicitation need domain knowledge from diverse sources, such as standards, norms, or pertinent bibliographic references, to define non-functional requirements.	4.31	Strongly Agree
Providing basic domain knowledge for elicitation by training the individual(s) responsible for eliciting non-functional requirements under the guidance of specialists or experts.	3.53	Agree
Awareness of the customer background (profile) of the person who performs the elicitation will improve the collaboration and interaction between them during the non-functional requirements elicitation.	3.67	Agree
Ensuring non-functional requirements are comprehensible to the customer requires using straightforward language and expressions during the validation process.	3.93	Agree
Communication knowledge challenges often lead to the oversight of non-functional requirements (“assuming we discuss the same thing, but we’re not”).	2.17	Disagree
Insufficient understanding of the system can result in the neglect of essential and significant non-functional requirements.	2.17	Disagree
Software practitioners conducting elicitation must be familiar with notations associated with non-functional requirements and the elicitation process (notations knowledge).	4.43	Strongly Agree

4.6 The Implementation of NFR Guidelines in an Agile Context

This section addresses the software practitioners’ opinions and experiences regarding the guidelines for NFR elicitation in an Agile context. The respondents were asked about the importance of guidelines for NFR elicitation, whether they use a guideline for NFR elicitation and the elements of the guideline that they use.

RQ4: What is the software practitioners’ opinion on the importance of the guideline for NFR elicitation in an Agile context?

4.6.1 The Importance of NFRs Elicitation Guideline

Respondents were queried about the significance of guidelines for NFRs elicitation, and if it can help the software practitioners to enhance the understanding of the elicitation process for NFRs and avoid ignoring these requirements during the elicitation stage. 91% of respondents agree with the importance of guidelines for NFR elicitation in the early phases. Conversely, only 9% of respondents do not agree with that, as shown in Figure 13.

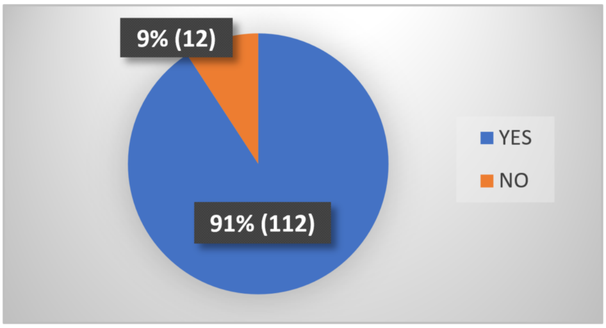

Fig.13: *The importance of guidelines for NFRs elicitation in the early phases of the Agile context.*

4.6.2 The Current Guidelines For NFRs Elicitation

The respondents were asked about the names of guidelines that they use to elicit the NFRs in the Agile context. Only three (3) of the respondents answered this question. One of them mentioned that they use the "Silva guidelines for eliciting NFRs" [48]. The rest of them did not mention a specific name for the guidelines that they use. Moreover, the respondents who used the guidelines to elicit the NFRs were asked about the elements of their guidelines. For this question, the respondents were permitted to select more than one answer. Figure 14 shows the elements of guidelines used by the respondents, which contain the activities, practices, and graphical notations. The principles are not chosen by the respondents.

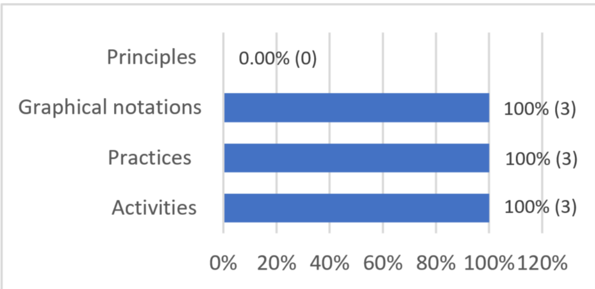

Fig.14: *The elements of guidelines for NFRs elicitation by the respondents.*

5. DISCUSSION

The majority of the respondents to this survey are programmers and POs. Also, most of the respondents have experience spanning more than six (6) years. Besides, the majority of the size of organizations in which respondents work among small to medium organizations is based on the definition of the organization's size provided by Akbar et al. [39]. In addition, the Agile methods that the respondents used in their organizations are Scrum and XP, which is a similar outcome as described by Anwer *et al.* [40].

Indeed, this questionnaire focuses on the software practitioners' opinions and experiences regarding the current practices of NFR elicitation in the early phase of the Agile context. Therefore, the respondents were asked whether their team elicited NFRs in the early phase or not. The respondents who answered "Yes" were allowed to continue answering the questionnaire. Unfortunately, around half of the respondents are not eliciting NFRs in the early phase, this result corresponds with the findings of studies by Curcio *et al.* [41] and Villamizar *et al.* [42]. Neglecting NFRs brings to light the erroneous impression that quality is not necessary or that it can be treated in the background or in later phases of software development.

Furthermore, the majority of respondents answered that PO is the one who has to perform the NFR elicitation, and this result corresponds to the outcomes in the study of Asghar *et al.* [5] and Grangel *et al.* [43]. In addition, the majority of respondents defined the application domains and the types of systems in NFR elicitation. Defining the application domain and types of systems can help practitioners determine the possible NFRs to take into account during the elicitation stage [44]. Nonetheless, the previous studies failed to define this practice.

Besides, most of the respondents to this survey apply ISO/IEC 25010 to define NFR types. According to Estdale *et al.* [45], ISO 25010 is one of the most common quality models that determines the quality characteristics and should be taken into account when defining the properties of the software product. To our knowledge, none of the previous empirical studies have validated this practice.

Regarding the validation stage, the majority of respondents do not validate NFRs after elicitation. Indeed, NFRs validation is an essential phase to confirm that the NFRs meet the customer's needs [46]. Besides, the majority of respondents who are validating NFRs have used the GQM method and natural language to validate these requirements with customers. In addition, the respondents answered the PO, who has the responsibility to validate NFRs after elicitation, which corresponds with the findings in the study by Grangel *et al.* [43]. Indeed, the previous studies did not mention the most common validation method for NFRs in the Agile context.

Additionally, the respondents were asked to rank the degree of agreement on NFR elicitation practices'. The respondents strongly agreed that identifying the application domain and type of system at the beginning of the project will help in the elicitation stage to define the possible NFRs. According to Sherif *et al.* [44], identifying and analyzing the applications domain more frequently can help find NFRs that require more attention. Besides, the NFRs elicitation should be based on face-to-face communication by encouraging customer participation through natural language, which was also strongly agreed by respondents. This outcome corresponds with the study conducted by Younas *et al.* [12]. Moreover, the respondents agreed that using historical data from previous projects can help in defining and eliciting the NFRs of new software projects. To our knowledge, none of the previous empirical studies have defined this practice. According to Maiti *et al.* [47], utilizing historical data in the elicitation phase is beneficial to predicting future NFRs. After identifying the scope of the project and the possible NFRs, then a list of questions should be prepared to capture the candidate NFRs from the customer, which are agreed upon by respondents. This outcome corresponded with the outcomes of Silva *et al.* [48]. Also, the respondents agreed that the elicited NFRs should be separated by detaching story cards to explain them clearly. Indeed, NFRs include critical information that needs to be captured into separate stories from the FRs. This makes NFRs highly visible and easy to manage. This finding is aligned with the findings of Domah *et al.* [46].

In the case of changing requirements, the respondents agreed that they can use a checklist table that contains all requirements to update the effect of a change in FRs onto NFRs. In addition, the respondents agreed that validating NFRs in natural language with the customer is an essential activity to confirm that requirements are understandable by the customer. This result corresponds with the outcome of Silva *et al.* [48]. However, Silva *et al.* [48] didn't validate this practice through empirical study. On the other side, the respondents disagreed with validating NFRs by a third party such as an expert review, which contradicts the studies conducted by Younas et al. [12] and Silva *et al.* [48].

Regarding SRs, the respondents were asked if they were eliciting and documenting the SR explicitly. Most of the respondents are only dealing with security issues during the implementation phase or after the system is being developed. Indeed, SRs are critical requirements, and not a frivolous job, and should be taken into account from the early stages of Agile software development [42], [49].

In addition, the respondents who are dealing with SRs were asked about who performs the elicitation of SRs in their organizations. The majority of respondents answered that the PO and Security advisor. This outcome corresponds with the findings in the study by Grangel *et al.* [43]. Furthermore, the respondents who are dealing with SRs were also asked about the notation used to represent these requirements. A higher percentage of respondents who represent the SRs by notation are using misuser stories, followed by misuse cases. This outcome corresponds with the findings in a study by Saldanha [15]. However, Saldanha [15] did not empirically validate this practice. The misuser story functions as a lightweight method for representing and capturing SRs. It adopts the same structural format as a user story conventionally employed in Agile methods but conveys undesired stories. However, most of the respondents do not utilize a particular notation to represent the SRs. Indeed, the security aspects need to be explicitly represented to be visible and easy to manage [50].

Besides, the respondents were asked to rank their degree of agreement with the SRs elicitation practices. The respondents strongly agreed that the SRs should be elicited explicitly. Indeed, defining the common attacks that occurred previously to elicit the requirements security for a new project should utilize attackers' perspective while eliciting SRs should be considered, and these requirements should be documented in a particular notation as a misuse story. This result corresponds with the findings in the study by Saldanha [15]. Nevertheless, Saldanha [15] did not empirically validate this practice.

On the other hand, the respondents disagreed on two practices, which are reaching a common understanding of the security needs among the person(s) responsible for elicitation, including the customer, and eliciting SRs separately from the other NFRs by the person(s) responsible for elicitation. However, dealing with the SRs separately can lead to avoiding the potential higher costs related to fixing security flaws later on [50]. Ideally, the SRs should receive special attention from other NFRs in the early phases of elicitation [15]. On the other side, the respondents were asked to rank the degree of agreement for the required knowledge of the elicitation stage. The respondents strongly agreed that the person who performs NFRs elicitation should have domain knowledge of various sources, such as standards and norms or relevant bibliographic references, to identify the NFRs. Also, the respondents agreed that the person(s) responsible can be trained for eliciting NFRs by specialists to provide the basics of domain knowledge of the elicitation process. Besides, the respondents agreed that the customer needs simple words and expressions to ensure that the NFRs are understandable in their language, which is also similar to those found in Younas *et al.* [12]. However, there was no empirical validation of this knowledge by Younas et al. [12]. Moreover, the respondents agreed that those who perform the elicitation need to know notations related to NFRs and the elicitation process, and that corresponds with the outcomes of the study by Berghe *et al.* [50]. Nevertheless, Berghe *et al.* [50] did not validate this knowledge through empirical study.

On the other side, the respondents disagreed that NFRs are easily missed because of unsuccessful communication knowledge. However, conducting successful interviews for requirements elicitation often relies on their communication and technical knowledge [51]. Also, the respondents disagreed that the deficiency of the system knowledge would lead to the omission of necessary and important NFRs. System requirements outline services and features that must be delivered and adhered to and also must be fully understood by the development team to ensure the quality of the system [52].

Furthermore, the respondents were asked about the importance of guidelines for NFR elicitation in an Agile context. The vast majority of respondents agreed that the guidelines for NFR elicitation in the early phases are very significant and will help enhance the understanding of the stakeholders about the elicitation process and avoid ignoring these requirements during the elicitation stage in the Agile environment [12]. This result corresponds with findings from a survey conducted in a non-agile context by Silva et al. [48]. They recommend that guidelines for NFR elicitation are beneficial for both developers and customers to deal with these requirements.

The respondents were asked if they were applying a guideline for NFR elicitation in an Agile context. Only three (3) respondents answered this question. This result corresponds with the results of studies conducted by Kumar *et al.* [1], Viviani *et al.* [11], and Younas *et al.* [12], which reported there is a lack of guidelines for NFRs elicitation in the area of Agile. One of three respondents mentioned that the "Silva guideline for eliciting NFRs" [48] is used in their software projects. However, this guideline involved heavy-weight processes with a lot of documentation, which is not compatible with the Agile context [12]. Thus, there is a necessary need to develop a guideline for NFR elicitation in the early stages of the Agile context. Furthermore, the participants also highlighted that the guidelines they employed encompassed elements such as activities, practices, and graphical notations.

6. CONCLUSION AND FUTURE WORK

This study has been conducted to investigate the current practices of NFR elicitation in Agile software development organizations, with a focus on SRs elicitation, and to explore the required knowledge of stakeholders in the elicitation stage. The survey, conducted among 121 respondents from Jordanian Agile software development firms, identified sixteen (16) practices of NFRs elicitation and ten (10) practices of SRs elicitation. Additionally, the study defined the necessary knowledge for elicitation in an Agile context. Notably, 91% of respondents acknowledged the importance of elicitation guidelines for NFRs in the early phases, highlighting the significance of this aspect in Agile development. These findings provide valuable insights for practitioners and can inform the development of new guidelines for NFR elicitation in an Agile context. However, it is crucial to recognize the limitations of our research, such as not accounting for specific contextual factors like team size or project complexity. Future research should address these limitations to enhance the applicability and generalizability of our findings.

AUTHOR CONTRIBUTIONS

Conceptualization, M.S. and M.S; methodology, M.S; software, M.S; validation, M.S.; formal analysis, M.S, F.B., and S.M.; investigation, M.S.; data curation, M.S; writing—original draft preparation, S.M.; writing—review and editing, M.S, F.B. and S.M.; visualization, M.S.; supervision, F.B. and S.M. All authors have read and agreed to the published version of the manuscript.

References

[1] D. Kumar, A. Kumar and L. Singh, "Non-functional requirements elicitation in agile base models," *Webology*, vol. 19, no. 1, pp. 1992-2018, 2022.

[2] S. Alam, S. N. Bhatti and S. Asim, "Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol.8, no. 4, 2017.

[3] V. T. Heikkila, D. Damian, C. Lassenius and M. Paasivaara, "A Mapping Study on Requirements Engineering in Agile Software Development," *Proc. - 41st Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2015*, pp. 199–207, 2015.

[4] K. Pohl and C. Rupp, "Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam," *IREB compliant*, pp. 184, 2015.

[5] A. R. Asghar, "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development," *International Journal of Advanced Computer Science and Applications(IJACSA)*,vol. 7, no. 12, pp. 300–306, 2016.

[6] S. Rahy and J. M. Bass, "Managing non-functional requirements in agile software development," *IET Software*, vol. 16, no. 1, pp. 60-72, 2022.

[7] R. R. Maiti, "Capturing, Eliciting, and Prioritizing (CEP) Non-Functional Requirements Metadata during the Early Stages of Agile Software Development," Doctoral dissertation, Nova Southeastern University, 2016.

[8] M. Shahid and K. A. Tasneem, "Impact of avoiding non-functional requirements in software development stage," *American Journal of Information Science and Computer Engineering*, vol. 3, no. 4, pp. 52-55, 2017.

[9] A. Eberlein and J. C. Sampaio do Prado Leite, "Agile Requirements Definition: A View from Requirements Engineering Julio Cesar Sampaio do Prado Leite," *Int. Work. Time-Constrained Requir. Eng.*, 2002.

[10] R. Veleda and L. M.Cysneiros, "Towards an ontology-based approach for eliciting possible

solutions to non-functional requirements," *Advanced Information Systems Engineering*, pp. 145-161, 2019.

[11] L. Viviani, E. Guerra, J. Melegati and X. Wang, "An Empirical Study About the Instability and Uncertainty of Non-functional Requirements," in *International Conference on Agile Software Development*, pp. 77-93, 2023.

[12] M. Younas, D. N. A. Jawawi, I. Ghani and R. Kazmi, "Non-functional requirements elicitation guideline for agile methods," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 3-4, pp. 137-142, 2017.

[13] A. Jarzębowicz and P. Weichbroth, "A Qualitative Study on Non-Functional Requirements in Agile Software Development," in *IEEE Access*, vol. 9, pp. 40458-40475, 2021.

[14] S. Wagner, D. M. Fernández, M. Kalinowski and M. Felderer, "Agile requirements engineering in practice: Status quo and critical problems," *CLEI Electronic Journal*, vol. 21, no. 1, pp. 6-1, 2018.

[15] L. R. Saldanha and A. Zorzo, "Security requirements in agile software development: a systematic mapping study," Pontifical Catholic University of Rio Grande Do Sul, 2019.

[16] I. Inayat, S. S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915–929, 2015.

[17] K. Elghariani and N. Kama, "Review on Agile requirements engineering challenges," *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, pp. 507-512, 2016.

[18] S. Kopczyńska, M. Ochodek and J. Nawrocki, "On the importance of non-functional requirements in agile software projects—a survey," *Integrating Research and Practice in Software Engineering*, 145-158, 2019.

[19] T. L. A. Kotze, "The role of knowledge capturing during the elicitation of system requirements in a high-reliability organization in South Africa," Doctoral dissertation, University of Pretoria, 2017.

[20] M. Ochodek, "Approximation of COSMIC Functional Size of Scenario-Based Requirements in Agile Based on Syntactic Linguistic Features—A Replication Study," *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, Berlin, Germany, pp. 201-211, 2016.

[21] H. Yaseen, K. Dingley and C. Adams, "Capturing the growth of e-commerce in Jordan using a novel research approach," *International Journal of Management and Commerce Innovations*, vol. 3, no. 2, pp. 811-827, 2016.

[22] G. E. Mills and L. R. Gay, *Educational research: Competencies for analysis and applications*, Pearson. One Lake Street, Upper Saddle River, New Jersey 07458, 2019.

[23] W. G. Zikmund, B. J. Babin, J. C. Carr and M. Griffin, *Business Research Method*, 8th ed., South Western Cengage Learning, 2010.

[24] C. Frankfort-Nachmias and D. Nachmias, *Research methods in the social sciences*, 5th ed., New York: St Martin's Press, 1996.

[25] C. Sudlow *et al.*, "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *PLoS medicine*, vol. 12, no. 3, e1001779, 2015.

[26] J. Kirakowski, "Questionnaire in usability engineering: A list of frequently asked questions," Ed. 3. Human Factors Research Group, Ireland, 2000.

[27] C. Robson, *Real world research: a resource for social scientists and practitioner-researchers*, Oxford: Blackwell Publishers Ltd, 1993.

[28] J. Dawes, "Do data characteristics change according to the number of scale points used?," *International Journal of Market Research*, vol. 50, no. 1, pp. 61–77, 2007.

[29] I. Etikan, S. A. Musa and R. S Alkassim, "Comparison of convenience sampling and purposive sampling," *American Journal of Theoretical and Applied Statistics*, vol. 5, no.1, pp. 1-4, 2016.

[30] G. Taasoobshirazi and S. Wang,, "The performance of the SRMR, RMSEA, CFI, and TLI: An examination of sample size, path size, and degrees of freedom," *Journal of Applied Quantitative Methods*, vol. 11, no. 3, pp. 31-39, 2016.

[31] U. Sekaran, "Determining sample size," *Sekaran. Uma (4th Eds.), Research methods for business: A skill building approach*, pp. 263-298, 2003.

[32] R. Umezawa, K. Takanami, N. Kadoya, Y. Nakajima, M. Saito, H. Ota and Y. Ishikawa, "Assessment of myocardial metabolic disorder associated with mediastinal radiotherapy for esophageal cancer-a pilot study," *Radiation Oncology*, vol. 10, no. 96, 2015.

[33] D. R. Cooper and P. S. Schindler, "Qualitative research. Business Research Methods," vol. 4, no. 1, pp. 160-182, 2011.

[34] A Przybyłek and M. E. Morales-Trujillo, "Advances in Agile and User-Centred Software Engineering," *Lecture Notes in Business Information Processing*, vol. 376, 2020.

[35] M. Saleh, F. Baharom, S. F. P. Mohamed and A. M. Din, "The Current Practices and Required Knowledge for Non-Functional Requirements Elicitation in Agile Context: A Pilot Study in Jordan," in *Proceedings of Knowledge*

Management International Conference (KMICe) 2021, pp. 265-270, 2021.

[36] A. N. Oppenheim, “Questionnaire design, interviewing, and attitude measurement,” London: Pinter Publishers, 1992.

[37] W. Ismail, N. Abedlazeez and Z. Hussin, “Epistemological beliefs of students at high schools: A survey study in Malaysia,” *OIDA International Journal of Sustainable Development*, vol. 2, no. 8, pp. 39-46, 2011.

[38] S. F. P. Mohamed, “A process-based approach software certification model for the agile and secure environment,” Ph.D. dissertation, Universiti Utara Malaysia, 2015.

[39] M. A. Akbar, S. Mahmood, A. Alsanad, M. Shafiq, A. Gumaei and A. A. -A. Alsanad, “Organization Type and Size Based Identification of Requirements Change Management Challenges in Global Software Development,” in *IEEE Access*, vol. 8, pp. 94089-94111, 2020.

[40] F. Anwer, S. Aftab, S. S. M. Shah and U. Waheed, “Comparative analysis of two popular agile process models: extreme programming and scrum,” *International Journal of Computer Science and Telecommunications*, vol. 8, no. 2, pp. 1-7, 2017.

[41] K. Curcio, T. Navarro, A. Malucelli and S. Reinehr, “The Journal of Systems and Software Requirements Engineering: A systematic mapping study in agile software development,” *Journal of Systems and Software*, vol. 139, pp. 32–50, 2018.

[42] H. Villamizar, M. Kalinowski, M. Viana and D. M. Fernández, “A Systematic Mapping Study on Security in Agile Requirements Engineering,” *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Prague, Czech Republic, pp. 454-461, 2018.

[43] R. Grangel, C. Campos, “Agile model-driven methodology to implement corporate social responsibility,” *Computers & Industrial Engineering*, vol. 127, pp. 116-128, 2019.

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

[45] J Estdale and E. Georgiadou, “Applying the ISO/IEC 25010 quality models to software product,” in *Systems, Software and Services Process Improvement*, Springer International Publishing, pp. 492-503, 2018.

[46] D. Domah and F. J. Mitropoulos, “The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development,” *SoutheastCon 2015*, Fort Lauderdale, FL, USA, pp. 1-7, 2015.

[47] R. R. Maiti, A. Krasnov and D. M. Wilborne, “Agile software engineering & the future of non-functional requirements,” *Journal of Software Engineering Practice*, vol. 2, no. 1, pp. 1-8, 2018.

[48] A. Silva, P. Pinheiro, A. Albuquerque and J. Barroso, “A process for creating the elicitation guide of non-functional requirements,” *Software Engineering Perspectives and Application in Intelligent Systems*, Springer International Publishing, pp. 293-302, 2016.

[49] V. Sachdeva and L. Chung, “Handling non-functional requirements for big data and IOT projects in Scrum,” *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, Noida, India, pp. 216-221, 2017.

[50] A. Berghe, R. Scandariato, K. Yskout and W. Joosen, “Design notations for secure software: a systematic literature review,” *Software & Systems Modeling*, vol. 16, pp. 809-831, 2017.

[51] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini and B. Donati, “Teaching requirements elicitation interviews: an empirical study of learning from mistakes,” *Requirements Engineering*, vol. 24, pp. 259-289, 2019.

[52] L. R. Wong and D. S. Mauricio, “Qualities that the activities of the elicitation process must meet to obtain a good requirement,” *Journal of Engineering Science and Technology*, vol. 14, no. 5, pp. 2883-2912, 2019.

Mohammed Saleh received his B.S. degree in computer science from Yarmouk University, Jordan, and his M.S. degree in computer science. He is currently pursuing a Ph.D. degree in computer science (software engineering) from the School of Computing at Universiti Utara Malaysia (UUM), Malaysia. He has presented numerous papers at international conferences and has authored several publications focusing on agile software engineering. His research interests encompass software engineering, agile software development, agile requirements engineering, and non-functional requirements elicitation.

Fauziah Baharom received a Ph.D. degree in computer science from Universiti Kebangsaan Malaysia. She is currently an Associate Professor with the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. She has been appointed as the Head of the Data Management and Software Solutions (DMSS) Research Laboratory. She is also an active researcher and managed to secure several research grants awarded by the Ministry of Higher Education, Malaysia. Her research interests include software engineering, big data quality, and governance.

Shafinah Farvin Packeer Mohamed holds a diploma from UNISHAMS in the year 2003 and a Bachelor's Degree from Universiti Utara Malaysia in the year 2008. Her Ph.D. was obtained from the same university in the year 2015. Her research interest is related to software process quality, software process assessment, and certification, Agile software process, and multicriteria decision-making. She is currently a lecturer in the School of Computing, Universiti Utara Malaysia.