

Non-Functional Requirements Elicitation Guideline for Agile Methods

M. Younas^{1,2}, D. N. A. Jawawi¹, I. Ghani³ and R. Kazmi¹

¹Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.

²Government College University Faisalabad, Pakistan.

³Monash University Malaysia Campus, Subang Jaya, Selangor, Malaysia.
younas.76@gmail.com

Abstract—One of the essential activities in software development is elicitation of requirement. Majority of the studies has pointed out that less attention is given to the Non-Functional Requirement (NFR). The negligence of NFR elicitation is due to lack of knowledge of the user and developer about NFR. Our study presents elicitation guidelines for NFRs in agile methods. This guideline will help developers as well as users in agile methods. A case study is conducted on the group of master students for eliciting NFR with the help of elicitation guidelines. In addition, the initial results were obtained by extracting NFRs from eProcurement document that carries requirements of major European Union projects. The result of the case study is positive and encouraging for the new developers and users having less awareness about NFRs. Furthermore, the study describes the role of cloud computing in agile methods, especially in elicitation activity.

Index Terms—Agile; Cloud Computing; Elicitation; Guideline; Non-Functional Requirement.

I. INTRODUCTION

Agile methods are popular due to improve customer satisfaction, accommodation of requirement change at any stage of development, frequent delivery of software modules and close interaction with the client. In agile methods, software requirement is evolved with the progress of the project [1, 2].

In the industry, functional requirements are treated as primary requirements while Non Functional requirements (NFR) are ignored [3] or only catered at design and implementation level. In Agile methods, NFR is ignored due to unawareness of user about NFR [4-6] and nature of agile methods [7]. NFR is ill-defined in agile software development [8]. Incorporating NFR into different phases of software development process is still a difficult job [9]. The failure of the system is often due to ignorance of NFR [10] that is London Ambulance System [9]. Another study described that the failure in defining NFR is due to lack of technical and financial capacity of the organization to comply with NFR [11].

The NFRs are important in the early stage of development because it identifies the selection of technology, allocation of hardware, and the standards adopted in software development. Furthermore, NFR helps to determine the mechanism for the security of the software, license, and distribution of software [12]. Problems have been reported in the area of agile requirements elicitation particularly on the lack of elicitation guidelines [13, 14]. In software development methods, there is a lack of detail about the procedure to incorporate NFR.

Some existing studies [15-18] provide knowledge for identifying NFR types. Furthermore, other sources to describe NFR are international standards for software engineering artefacts, for example, ISO 25010 [17] and ISO/IEC 9126 quality standard [19]. The quality standard describes the level for quality requirement but does not explain how to perform actually or elicit the quality requirement. A Chung et al. [20] described the Software quality tree as shown in Figure 1. The tree represents the classification of quality attributes in software engineering. In addition, there is no established written standard. A few studies [3, 21] present NFR elicitation for the Agile methods. There are some studies [2, 13, 22] that focus on Agile requirement elicitation in general but not particularly for NFR. And also there are several studies [6, 9, 12, 18, 23-26] for NFR elicitation in non-agile methods.

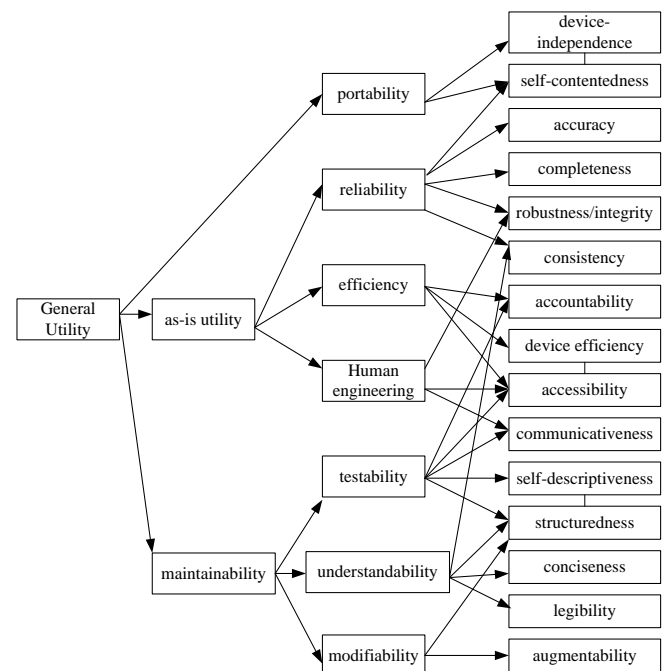


Figure 1: Software quality tree [20]

The Boehm [10] describes that a good customer should be collaborative, capable, knowledgeable, available, and good representative. These characteristics help development team in gathering the correct and precise requirements from the customer. However, lack of knowledge of developer and user in elicitation process is a big problem [4-6]. A survey [11] was conducted in order to find the need of a guide in eliciting

NFR. 90% of the participants suggest that guide would be useful to identify NFRs.

This study proposes an elicitation guideline for NFR in agile methods. The guidelines will help the software industry in determining and eliciting requirement for all type of software development and also guiding both user and developer. The elicitation guideline can help in two perspectives 1) when the project is going to start, and developers have to interact with the customer or user for requirement elicitation. 2) The team received predefined requirements in text form and has to extract or elicit NFR from the document. In both perspectives, the elicitation guidelines work effectively. In addition, the study identifies the role of cloud computing in Agile development activities.

The paper is organized as the next section describes the requirement elicitation concept in agile method and helping techniques for elicitation. Section III proposes the elicitation guideline for Agile NFR elicitation and Section IV describes the execution of the elicitation guideline. The conclusion of the study is presented in the last section.

II. AGILE REQUIREMENT ELICITATION AND CLOUD COMPUTING

According to Somerville [27], Elicitation, Negotiation, Analysis, Documentation, Validation, and Management are the core activities for Requirement Engineering (RE). Traditional methods use documentation for knowledge sharing whereas Agile methods rely on face-to-face communication between customers and developers. The essential requirements elicitation activities in non-agile methods are Interview, Focus groups, Observation and Social Analysis, Brainstorming and Prototyping. The detail of elicitation activities is as follows:

A. Interview

The interview is a way to collect facts and opinions of the users and other stakeholders of the project. It also eliminates or minimizes the mistakes and misunderstandings regarding requirements [13]. The interview is used in almost all agile methods, for seeking Requirements.

B. Observation and social Analysis

Observation with the customer may be a face-to-face meeting, audio recording or video recording [8]. It is an investigation for observing the user’s activities. Observation and social analysis are not explicitly used in agile methods [8]

C. Focus Groups

Focus groups are a way to find the will and the perception of the customer regarding the matter under investigation. It also helps in identifying the things important for the customer and what they expect from the system.

D. Brainstorming

Brainstorming is a way to tune the mind of the customer regarding requirements. It has two phases. The first phase is a collection of the ideas, and the second phase is a discussion on received ideas.

E. Prototype

An incomplete version of the software is called a prototype. There are two types of prototypes. One is throwaway

prototypes, used only to understand the requirements and perception of the user. The second one is evolutionary prototypes. It is workable software delivered to the client, and it provides a base for the final version of the software.

F. JAD Session

Dynamic systems development method (DSDM) uses Joint application design (JAD) session for understanding the new System in the beginning of the project [8] and use of prototyping. eXtrem Programming (XP) is more relying on on-site customer involvement.

How the elicitation technique (used in traditional methods) could be used in agile methods is explained in Table 1. The first row in Table 1 represents the different agile methods and their associated activity, and the first column represents the elicitation techniques used in traditional methods. Table 1 also describes the relationship between the various agile methods and their activities mapped on the requirement elicitation activities. For example in Agile method XP, User Story is used for requirement elicitation, and it covers or resembles the interview and brainstorming in traditional methods. Prioritization of Requirements is a common practice used in all agile methods.

Table 1
Agile Methods and Elicitation Techniques

Agile method activity \ Elicitation Techniques	FDD Feature List	Scrum Product backlog	XP User story	Agile Modeling	DSDM
Interview	✓	✓	✓	×	×
Observation and Social analysis	×	×	×	×	×
Focus group	×	×	×	×	×
Brainstorming	×	×	✓	✓	×
Prototype	×	×	×	×	✓
JAD Session	×	×	✓	×	✓
Prioritization	✓	✓	✓	✓	×

In agile methods, developer and customer work together. Customer elicits the requirement and the developer distributes the Requirement into User stories and then develops the part of the software and confirms it with the customer if OK then commences next iteration and so on. The requirement elicitation process is completed during all phases of software development as shown in Figure 2. It starts with collecting information from the user in the form of story cards. In agile methods, user story consists of high-level definition of requirements in the form of short and abstract descriptions. Requirements are discussed in detail with the customers during all phases of development.

The agile methods depend on the interactive communication between developers and customers. On-premises, it is easy to establish communication and interaction, however, in distributed environment it is difficult. Cloud computing helps by providing different means of communication between user and software team such as file sharing, idea sharing, and discussion forums, wikis , real-time reports and code sharing [28]. Project management tools, code management and testing tools are provided as Software as a Service (SaaS). For project development and deployment different IDEs and platforms are provided through Platform as a service (PaaS) in the cloud computing. Several studies claim that cloud computing helps in agile methods. In addition, Emails [1], Skype chat [29-31], and video conferencing, cloud telephony by Amazon Web Service

(AWS) [32] are also used for communication. NFRs elicitation can be improved by enabling Agile member to load requirement document from different locations using cloud [33].

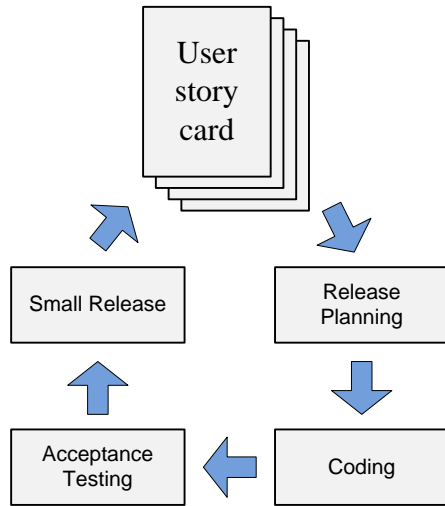


Figure 2: Agile process in general

On the other hand, if the customer or its representative is not available to the developer then product owner or business analyst acts as surrogates. These surrogates help the developer in changing the requirement and decision making. The customer and project lead help the development team in quick learning of Requirement [10].

III. PROPOSED ELICITATION GUIDELINE

By reviewing the NFR elicitation approaches, process [10, 21], templates [34], frameworks [13] and elicitation guidelines presented by previous studies[4], we propose elicitation guideline for agile methods. The previous Guideline is for the non-agile environment, and our Guideline is enhanced and for agile methods. In proposed elicitation guideline we adopt the method of reusing existing knowledge and method of elicitation and structured meeting techniques for NFR elicitation described by the Kopczyńska et al. [35]. The proposed elicitation guideline is also inspired by the suggestion given by the Too et al. [36] for improving elicitation of NFR. Question answering used in Guideline is adopted by Zachman framework [37] for elicitation. The Guideline applies the various artefacts to support elicitation process. Historical data can be used in order to predict the new NFRs, based on requirement given by Maiti [33]. Our proposed elicitation Guideline includes the role of the expert in artefacts and the previous data of organization used for addition NFR prediction.

In agile methods user story card is used for elicitation of FRs, here the paper augments another user story card for eliciting NFRs. Separate NFR cards are used by Song et al. [24] in their proposed solution for NFR elicitation. In our proposed solution NFR user story card includes the following features as depicted in Table 2.

The priority of requirement based on the word “must have,” “should have” and “could have” or similar words used by the client in elicitation of requirements. Here user story ID and NFR ID has a link between FR and NFR. This link is essential as explained with an example. If there is an NFR “Every transaction must be performed under one (1) second”. Then

every user story having transaction must cater this requirement in order to achieve user satisfaction in the system. Dependency attribute also helps in this scenario.

Table 2
NFR card’s Attributes

Field	Description
NFR ID	Id for the non-functional requirement.
US ID	This id links FRs with identified NFRs
NFR	Description of primary NFR
Sub-NFR	Description of secondary NFR
Priority	Priority of NFR is set (High, medium, low) on the consensus of proactive stakeholder in project
Dependency	If NFR is dependent on some NFR

For NFR and the Sub-NFR description includes property name, property type, and value, for example, a sample statement:

“The response time on searching record must not be longer than 5 seconds.”

Here,

- Property name = Response time
- Subtype = Performance → Time-behavior
- Type = time in seconds and
- Value = 5.

The NFR story card is populated by utilizing the elicitation guideline depicts in Figure 3. The description of the activities used in elicitation guideline is as follows:

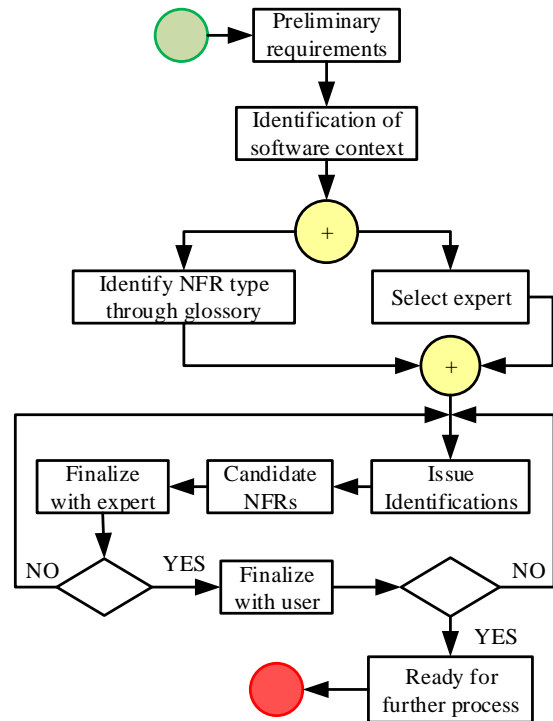


Figure 3: Agile NFR elicitation process

A. Preliminary Requirement

In this phase, the initial Requirements are collected from user or client during interview/face to face meeting. [8] In the case of distributed environment, the meeting is held with the help of cloud computing as discussed in Section 1. Then the type of software is identified on the basis of the preliminary requirement will discuss in next phase. In the case of a change in FR or NFR, it may affect the NFR; this effect can be

reviewed through the same process. Following roles and artefacts take part in this phase:

- i. Input: Not applicable
- ii. Output: Preliminary requirement
- iii. Responsible: Team conducting the process and User

B. Identification of software type

The purpose of this step is to identify the type of software on the basis of preliminary requirement. There is a different kind of software i.e. web-based software application or mobile based application or business application and so on. Details of various software types and corresponding NFRs are listed in Table 4 in the appendix. In addition, the software types classification is expressed by the studies[38, 39]. Furthermore, the software type can be searched and analyzed in the list of the project already developed in the organization. The software type is analyzed to design the possible applicable NFRs for the software. The developer brainstorms the user (if needed) regarding the non-functional requirements of the project being treated. Following roles and artefacts take part in this phase:

- i. Input: List of the project already developed by the organization.
- ii. Output: Definition of type of software being developed.
- iii. Responsible: Team conducting the process.

C. Identification of requirement type through glossary

In this phase, bibliographic sources are searched out to find the NFR types and their subtypes. In a study [15], Chung described the classification of requirement types. The study defines more than hundred requirement types. Somerville presents different types of the requirement in their studies [16, 27]. Rojo et al. explore thirty-six NFRs in web-based applications; out of which fifteen quality types and twenty-one of restriction types. Apparently, this study will be helping in identifying NFR in web based applications. The quality concerns of different stakeholders and associated quality attributes are described by the Boehm's study [10]. In addition, NFR classification is defined in ISO 25010 [17] and ISO/IEC 9126 quality standard [10, 19]. All these resources help in identifying the type of NFR. Following roles and artefacts take part in this phase:

- i. Input: Bibliographic source and standards.
- ii. Output: Definition of NFRs, bibliographic source and standards related to NFRs treated in the project.
- iii. Responsible: Team conducting the process.

D. Selection of Expert

Agile software process promotes self-organizing and team cooperation behavior. In this phase, the technical expert is rectified (if needed) in the area related to NFR type. The expert may help in the previous analysis of the Requirement type in bibliographic sources. The experts involved in this activity until the selection of Requirements to be treated in the project. The developer can take help from the concerned specialist regarding question (asked the user) in order to elicit NFR in the software. In the Agile method, the selection of maybe during scrum meeting or the developer can take help from the team leader in identifying the relevant expert. Following roles and artefacts involved in this phase:

- i. Input: List of employees and their expertise.
- ii. Output: Selected list of expert related to each NFR tends to be treated in the project.

- iii. Responsible: Team conducting the process.

E. Issue Identification

After identifying the type of project and type of expected NFR, the developer prepares the list of questions in order to extract requirement from the user, and find the issues and the related NFR. The developer can take help from an expert in preparing the Requirement questions for negotiating with the user. The purpose of this activity is to define the list of the issues against the particular NFR so that a real questioning can occur to refine the NFR. On the completion of this activity, a set of questions should be prepared against the all expected NFR. Following roles and artefacts take part in this phase:

- i. Input: List of NFRs tends to be treated and the corresponding expert, Project type identified
- ii. Output: List NFRs and corresponding questions asked from the user.
- iii. Responsible: Process team and expert.

F. Candidate NFR

For each requirement question identify the Requirement model, which clearly explains the need of the customer. For the analyst, the attributes required to ask the user regarding NFR should be defined i.e. Requirement type and its base class, dependency, and priority. Following roles and artefacts take part in this phase:

- i. Input: Identified Issues and a corresponding list of questions.
- ii. Output: Candidate NFRs and a list of the question asked having a positive response from the user.
- iii. Responsible: Team conducting the process.

G. Finalizing with Expert

After estimating default/candidate NFR i.e. the complete set of questions against the expected NFRs, validate it and confirm with an expert regarding the quality of NFRs and Interdependency among NFRs. If not up to the standard then send it back to Issue Identification phase as shown in Figure 3. Following roles and artefacts take part in this phase:

- i. Input: Software type, candidate NFRs and a corresponding list of questions.
- ii. Output: Approved NFRs and quality standard certificate.
- iii. Responsible: Experts

H. Finalizing with the User

After validation from an expert it is sent to the user. Validation of the user is essential in order to confirm NFR in natural language is understandable to the user. It is to ensure that the NFR is suitable in customer language. The validation with an expert was from the technical point of view. If there are some changes in NFRs, then it is sent back as shown in Figure 3.

- i. Input: List of quality candidate NFRs and corresponding questions
- ii. Output: Finalized list of NFRs and elicitation guideline for future use for the particular type of project.
- iii. Responsible: Team conducting the process and User.

I. Ready for further process

After completing all activities, NFRs are finalized for further use. In addition, the developer finds NFRs complete

guideline to elicit NFR for certain project that can be used in future. The developer has information about the expected NFRs and appropriate expert to help in elicitation process. The guideline act as support tools for elicitation process.

After identifying the NFR, a checklist is maintained with the elicited NFR against FR [6]. The check-list table contains all functional requirements and their corresponding non-function requirements. This checklist table further helps in case of changing Requirement in first activity “New or change in Requirement” in order to update the effect of a change in FR onto NFR.

IV. EXECUTION OF GUIDELINES

In order to analyze the proposed elicitation guideline, a case study was conducted on the master students in Software Engineering department. The group of students elicits NFR by using proposed guideline and without using proposed guideline. The feedback shows the positive response in elicitation of NFR in agile methods by seeking help from our proposed guideline. The guidelines provide the baseline knowledge for identifying NFRs. In addition, the approach is validated by the initial results coming from using some requirements from EU eProcurement document [40]. The report contains 26 requirements for developing software in European Union projects. Example of NFR card, populated (given in Figure 5) according to the guideline using the requirement given in Figure 4.

User story card

User Story ID :18.1

Description: needs to ensure that **access** to Tenders cannot be obtained by anyone, until **authorised** procurement officers proceed to the opening of Tenders following the four-eye principle .

Figure 4: User Story card containing requirements

NFR card

User Story ID :18.1

NFR ID:1

NFR: Accessibility, Security

Sub-NFR:

NFR Priority: high

Dependency: N.A

Figure 5: NFR card populated

NFRs identified are further explained by Sub-NFRs. Possible Sub-NFRs can be Accessibility, Availability, Maintenance, Performance, Security and Testing. The case study is encouraging and needed to conduct more case studies, especially industrial case studies. On the basis of the guideline, a tool should be developed in future.

V. DISCUSSION

Our proposed elicitation guideline utilized previous studies as a baseline. Silva [11] described an approach to define elicitation guideline for non-agile methods. Franch and Carvalho [19] presents elicitation guideline for agile methods. The comparison of artefact used in previous studies and our study is given in Table 3. Separate NFR story card is not used

in Silva’s study. In the previous studies, they base of 36 types of NFR where in our study, we take more than 100 types of NFRs. Furthermore, our study helps in identifying project type and requirement type whereas prior study has focus on requirement type but not on project type. The project type helps in predicting NFR in the project under considerations as given in Table 4 in appendix. In order to observe change in requirement the checklist table is adopted in our solution. The comparison with prior studies shows that the good feature in previous study is added on proposed guideline and this will produce positive results.

Table 3
Comparison with prior studies

Artefacts	NERV methodology	Silva elicitation guideline	Our elicitation guideline
Separate NFR story card	✓	×	✓
Application context	×	✓	✓
Software glossary	✓	✓	✓
Expert involvement	×	✓	✓
Project history/ NFR trends	×	✓	✓
w8 story card dependency/ checklist table	✓	×	×
	×	✓	✓

VI. CONCLUSION

Agile Non-Functional requirement elicitation is described in few studies. In previous studies, elicitation process is for non-agile methods, or it describes the elicitation process by finding the NFR in the SRS document. Our study explains the Agile elicitation process before starting the project. It guides developer as well as user or client. The elicitation guidelines also cater and accommodate the change in requirement during any phase of software development. We analyze the proposed elicitation guideline by using some requirements written in EU eProcurement document and find the encouraging results. In addition, the study explains the role of cloud computing in Agile elicitation process.

APPENDIX

Table 4
Software Types and Relevant NFR [38]

Application Domain	Relevant NFRs
Banking and Finance	accuracy, confidentiality, performance, security, usability
Education	interoperability, performance, reliability, scalability, security, usability
Energy Resources	availability, performance, reliability, safety, usability
Government and Military	accuracy, confidentiality, performance, privacy, provability, reusability, security, standardizability, usability, verifiability, viability
Insurance	accuracy, confidentiality, integrity, interoperability, security, usability
Medical / Health Care	communicativeness, confidentiality, integrity, performance, privacy, reliability, safety, security, traceability, usability
Telecommunication Services	compatibility, conformance, dependability, installability, maintainability, performance, portability, reliability, usability
Transportation	accuracy, availability, compatibility, completeness, confidentiality, dependability, integrity, performance, safety, security, verifiability

ACKNOWLEDGMENT

We are thankful to Ministry of Science, Technology and Innovation (MOSTI) to support this research under eScience grant vote: 4S113. We are also thankful to Universiti Teknologi Malaysia (UTM) for providing us with the research facilities.

REFERENCES

- [1] S. Franken, S. Kolvenbach, W. Prinz, I. Alvertis, "CloudTeams: Bridging the gap between developers and customers during software development processes," *Procedia Computer Science*, vol. 68, pp. 188-195, 2015.
- [2] L. Williams, "Agile requirements elicitation," 2004. Available at <http://agile.csc.ncsu.edu/SEMaterials/AgileRE.pdf>
- [3] D. Domah, and F. J. Mitropoulos, "The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development," in *SoutheastCon 2015*, 2015, pp. 1-7.
- [4] D. D. Gregorio, "How the business analyst supports and encourages collaboration on agile projects," in *2012 IEEE International Systems Conference SysCon 2012*, 2012, pp. 1-4.
- [5] M. Galster, and E. Bucherer, "A taxonomy for identifying and specifying non-functional requirements in service-oriented development," in *I 2008 IEEE Congress on Services - Part I*, 2008, pp. 345-352.
- [6] M. Rahman, and S. Ripon, "Elicitation and modeling non-functional requirements-A pos case study," *International Journal of Future Computer and Communication*, vol. 2, no. 5, pp. 485-489, 2014.
- [7] S. Jeon, M. Han, E. Lee, and K. Lee, "Quality attribute driven agile development," in *2011 9th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2011, pp. 203-210.
- [8] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," in *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2003*, 2003, pp. 308.
- [9] S. Ullah, M. Iqbal, and A. M. Khan, "A survey on issues in non-functional requirements elicitation," in *2011 International Conference on Computer Networks and Information Technology (ICCNIT)*, 2011, pp. 333-340.
- [10] B. Boehm, and H. In, "Identifying quality-requirement conflicts," *IEEE Software*, vol. 13, no. 2, pp. 25-35, 1996.
- [11] A. Silva, P.R. Pinheiro, A. Albuquerque, J. Barroso, "Approach to Define a Non-Functional Requirements Elicitation Guide Using a Customer Language," 2016.
- [12] A. Silva, P. Pinheiro, A. Albuquerque, and J. Barroso, "A process for creating the elicitation guide of non-functional requirements," in *Software Engineering Perspectives and Application in Intelligent Systems*, 2016, pp. 293-302.
- [13] W. Helmy, and A. K. O. Hegazy, "An evaluation framework for requirements elicitation in agile methods," in *The Seventh International Conference on Software Engineering Advances*, Lisbon, Portugal, 2012, pp. 588-593.
- [14] S. D. V. Rojo, and R. Oliveros, "Elicitation and Specification Processes of NFR for Web Applications," in *Joint Proceedings of Workshops, Doctoral Symposium, Empirical Track, and Posters, co-located with the 20th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ) 2014*, 2014, pp. 141-142.
- [15] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Springer US, 2000.
- [16] I. Sommerville, *Software Engineering*. Pearson, 2011.
- [17] ISO-IEC 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models, International Organization for Standardization 2011
- [18] D. Zowghi, and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools, in Engineering and managing software requirements," in *Engineering and Managing Software Requirements*, A. Aurum, and C. Wohlin, Eds. Berlin, Heidelberg: Springer, 2005, pp. 19-46.
- [19] X. Franch, and J. P. Carvalho, "Using quality models in software package selection," *IEEE Software*, vol. 20, no. 1, pp. 34-41, 2003.
- [20] L. Chung, and J. do Prado Leite, "On non-functional requirements in software engineering," in *Conceptual modeling: Foundations and applications*, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds. Berlin Heidelberg: Springer, 2009, pp. 363-379.
- [21] R. R. Maiti, and F. J. Mitropoulos, "Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development," in *SoutheastCon 2015*, 2015, pp. 1-8.
- [22] R. Ankori, "Automatic requirements elicitation in agile processes," in *2005 Proceedings. IEEE International Conference on Software-Science, Technology and Engineering*, 2005, pp. 101-109.
- [23] T. H. Al Balushi, P. R. F. Sampaio, D. Dabhi, and P. Loucopoulos, "ElicitO: A quality ontology-guided NFR elicitation tool," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2007, pp. 306-319.
- [24] X. Song, Z. Duan, and C. Tian, "Non-functional requirements elicitation and incorporation into class diagrams," in *International Conference on Intelligent Information Processing*, 2010, pp. 72-81.
- [25] L. M. Cysneiros, and E. Yu, "Non-functional requirements elicitation, in Perspectives on software requirements," in *Perspectives on Software Requirements*, J. C. S. do Prado Leite, and J. H. Doorn, Berlin, Heidelberg: Springer, 2004, pp. 115-138.
- [26] L. M. Cysneiros, and J. C. S. do Prado Leite, "Nonfunctional requirements: From elicitation to conceptual models," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 328-350, 2004.
- [27] G. Kotonya, and I. Sommerville, *Requirements Engineering: Processes and Techniques*. Wiley Publishing, 1998.
- [28] S. Singh, and I. Chana, "Introducing agility in cloud based software development through ASD," *Int. J. u-and e-Service, Sci. Technol*, vol. 6, no. 5, pp. 191-202, 2013.
- [29] S. Kalem, D. Donko, and D. Boskovic, "Agile methods for cloud computing," in *2013 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO)*, 2013, pp. 1079-1083.
- [30] I. I. S. S. S. Zarinah, and M. Kasirun, "Agile-based software product development using cloud computing services: Findings from a case study," *Sci. Int. (Lahore)*, pp. 1045-1052, 2013.
- [31] M. R. J. Qureshi, and I. Sayid, "Scheme of global scrum management software," *International Journal of Information Engineering and Electronic Business(IJIEEB)*, vol. 7, no. 2, pp. 1-7, 2015.
- [32] M. Manuja, and Manisha, "Moving agile based projects on cloud," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014, pp. 1392-1397.
- [33] R. R. Maiti, *Capturing, Eliciting, and Prioritizing (CEP) Non-Functional Requirements Metadata During the Early Stages of Agile Software Development*, College of Engineering and Computing, Nova Southeastern University, 2016.
- [34] S. Kopczyńska, and J. Nawrocki, "Using non-functional requirements templates for elicitation: A case study," in *2014 IEEE 4th International Workshop on Requirements Patterns (RePa)*, 2014, pp. 47-54.
- [35] S. Kopczyńska, S. Maćkowiak, and J. Nawrocki, "Structured meetings for non-functional requirements elicitation," *Foundations of Computing and Decision Sciences*, vol. 36, pp. 35-56, 2011.
- [36] C. W. Too, S. Hassan, J. Din, A. A. Ghani, and A. Azim, "Towards improving NFR elicitation in software development," *International Journal of Information Technology and Computer Science*, vol. 7, no. 1, pp. 33-44, 2013.
- [37] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 38, no. 2-3, pp. 454-470, 1999.
- [38] D. Mairiza, D. Zowghi, and N. Nurmuliani, "An investigation into the notion of non-functional requirements," in *SAC '10 Proceedings of the 2010 ACM Symposium on Applied Computing*, 2010, pp. 311-317.
- [39] M. Pierdin, and G. Bulder, "Dimensions and Types of Non-Functional Requirements NFR Dimensiones y tipos de Requisitos No-Funcionales RNF," *RACCIS*, vol. 2, no. 1, pp. 11-17, 2012.
- [40] Functional Requirements For Conducting Electronic Public Procurement Under The Eu Framework, Volume I, European Communities, 2005, Retrieved: Oct 08 2016, Available at <http://ec.europa.eu/idabc/servlets/Doc1ad3.pdf?id=22191>.