Received June 14, 2020, accepted July 5, 2020, date of publication August 27, 2020, date of current version December 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014381

Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment

MUHAMMAD YOUNAS[1,2], (Member, IEEE),
DAYANG NORHAYATI ABANG JAWAWI[2], (Member, IEEE),
MUHAMMAD ARIF SHAH[2,3], AHMAD MUSTAFA[2], MUHAMMAD AWAIS[4],
MUHAMMAD KAMRAN ISHFAQ[5], AND KARZAN WAKIL[6], (Member, IEEE)
[1]Department of Computer Science, Government College University Faisalabad, Faisalabad 38000, Pakistan
[2]Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[3]Department of IT and Computer Science, Pak–Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur 22620, Pakistan
[4]Department of Software Engineering, Government College University Faisalabad, Faisalabad 38000, Pakistan
[5]Department of Electrical Engineering Technology, Government College University Faisalabad, Faisalabad 38000, Pakistan
[6]Research Center, Sulaimani Polytechnic University, Sulaimani 46001, Iraq

Corresponding author: Muhammad Arif Shah (arif.websol@gmail.com)

This work was supported by the Universiti Teknologi Malaysia for grant UTM-TDR Grant Vot No. 06G23.

ABSTRACT Nonfunctional requirements get less attention because functional requirements are considered more important in the domain of agile software methodologies. This is due to the lack of mature requirement elicitation methodologies and the nature of the software agile software development process. The less attention caused few solutions in the domain which lead to software project failure. Cloud computing helps to practice twelve (12) agile principles including nonfunctional requirement elicitation. This study proposed a semi-automated methodology which will help analyst and developers in eliciting nonfunctional requirements in agile development and cloud computing environment. The methodology used an NLP based automatic NFR extraction approach to fast the NFR elicitation process. The methodology is evaluated by applying on eProcurement dataset. The results are improved by 8.77% and 1.76% in terms of ''Successful'' NFR. It is decreased by 7.02% and 1.75% in term of ''Partial success'', and 1.76% to 0.0% in term of ''Failure'' as compared to existing studies.

INDEX TERMS Agile development, case study, cloud-based agile tools, cloud computing, software requirement elicitation.

I. INTRODUCTION

Agile methods are popular due to improved customer satisfaction, welcome change in requirement, frequent delivery of software modules and close interaction with the client. Cloud computing accelerates software development cycle by minimizing installations or re-installation of software and patches [1]. Unlimited storage and computing resource are facilitated by cloud services based on pay per use [2], [3]. Cloud services such as data sharing, providing infrastructure (hardware and software), distributed application and prioritizing tasks lower the development cost of software [4]. Non-Functional Requirements (NFR) are ignored in agile software development. Usually, NFR are ill-defined or not identified adequately in agile software development [5], [6]. Usually, Functional Requirements (FR) are treated as primary requirements while NFR are ignored or only catered at design and implementation level. In addition, NFR are ignored due to unawareness of user about NFR, Lack of detail about the procedure to incorporate NFR elicitation and Nature of agile methods.

A survey revealed that elicitation of NFR in agile projects is important [7] and negligence results to increase in the software cost. US defense project of worth $2.7 billion was rejected due to performance and usability issues [8]. A health project Electronic Health Record (EHR) was rejected due to poor usability in interface of software [9]. More than 60% of the projects are rejected due to ignoring of NFR [10]. The software development without considering the proper NFR and their implementation are vulnerable to failure [11], [12]. The NFR identification guides in the selection of technology, hardware specs, standards and licensing of software.

NFR are ignored due to unawareness of user about NFR and nature of agile methods. There are quality standards

The associate editor coordinating the review of this manuscript and approving it for publication was Liqun Fu.

to describe NFR terminologies and their classifications, but these quality standards do not explain about the procedure to incorporate NFR elicitation. There are several NFR elicitation approaches, process [11], [13], templates [12], [14], frameworks [15] and elicitation guidelines [16]–[18]. Majority of these NFR elicitation solutions are for traditional software development. Some studies [19], [20] perform requirement engineering activities in distributed agile software development using cloud computing environment through Team Foundation Server (TFS), Microsoft share point and Pivotal Tracker.

During requirement gathering, it is difficult to capture NFR and this difficulty increase if the requirement engineer has lack of deep technical knowledge about NFR. There is need for elicitation methodology [21]. In agile methods, NFR are criticized for not having a method for NFR elicitation [6]. Another study [22] suggest that there would be NFR gathering technique which use historical data of projects to predict additional NFR and FR in agile software development. Furthermore, NFR extraction techniques work on cloud computing environment for extraction of NFR by sharing information from agile members located at different locations. Maiti. study [22] further describe that the cloud storage such as google docs is used to access historical NFR data of different projects stored by other agile team members.

This study proposed NFR elicitation methodology for agile software development using the cloud computing environment. The methodology helps analysts and developers in elicitation of NFR in agile software development.

The rest of the paper is organized as Section II and Section III, explain the related studies and NFR Elicitation and Cloud Computing respectively. Section IV includes details of the proposed methodology. Evaluation criteria and dataset detail is presented in Section V. The results are shown in Section VI followed by the discussion and conclusion in Section VII and VIII, respectively.

II. RELATED STUDIES

Nonfunctional requirements elicitation has several approaches and are classified into the categories such as goal-based, UML-based, Use case based, User story card based, template based and pattern based. In agile software development, mostly user story card based and template based approaches are used. The study [23] proposed Non-functional Requirements Modelling for Agile Processes (NORMAP) Methodology. The model identifies, links, and models Agile Loose Cases (ALCs) with Agile Use Cases (AUCs) and Agile Choose Cases (ACCs) and historical data. It is an adapted version. They enhance three artifacts W8 User Story Card Model from their previous study, agile requirement taxonomy and third artifact Aspect-Oriented "pointcut" operators (Before, After, Override, and Wrap) for linking FRs and NFR.

Another study [24] proposed Non-Functional requirement Elicitation, Reasoning, and Validation (NERV) methodology to trace non-functional requirement in early phase of software development cycle. A project cost and wasted effort are increased if the NFR is traced in the later phases of software development cycles [17]. NERV methodology is implemented with the help of several artifacts. "Agile story card" and "non-functional requirement user story companion (NFRusCom) card" are used as a repository for storing information regarding FRs and NFR.

TABLE 1. Comparison of use case and card-based approaches.

Study	Artifacts used	Advantages	Limitations
[23]	W8 user story card	Visualization of NFR, semi-automated approach, quantifiable risk-driven approach,	More pre-processing is required for extraction of NFR, not focus on sentence parsing and identification, not consider the semantic similarity
[24]	W8 user story card, NERV agility index, NFR trigger card	Expand Chung's frameworks in term of considering 36 NFR, easy for the user and agile members have low knowledge	validated through the single dataset, total manual process
[13]	OCR, NFR-Locator	Extract the NFR from image text, automate tool based	The tool has low precision value, Part of speech not considered, supervised learning based

The study [13] improves the previous work NERV methodology and NORMAP methodology by considering the importance of NFR in the early phases of the software development cycle and also considers the role of FR's instability and versatility of software. The study extracts NFR from documents and images using OCR. They store metadata of NFR in the database and it helps in predicting and prioritizing the NFR. For elicitation, the CEP methodology utilizes NFRLocator tool. For extracting NFR from image text, enhance the NFRLocator and called NFRLocator Plus. The OCR is used to extract text from the image.

Table 1 shows that there is a need to decrease false positives. Most of the approaches are manual or semi-automated. The use of the automated tool in approaches is appreciable in studies. The knowledgebase is used in all approaches and it is considered effective in agile methods.

The quality standards also help in elicitation and extraction of NFR. The study [25] extracts indicator keywords for classification of NFR as shown in Table 3. The keys identified are according to the ISO 9126 quality standard. The inclusion of quality standard in this process improves the quality assessment process [18].

III. NON-FUNCTIONAL REQUIREMENT ELICITATION AND CLOUD COMPUTING

Agile software development believes on teams' collaboration and rapid feedback with clients and other members. Collaboration and communication is in terms of sharing code, data and ideas. Collaboration and communication are executed with the help of cloud-based social media [19], [26]. Furthermore, different ways and tools are used for team

TABLE 2. Tools for sharing data and communication.

No	Description	Cloud services, tools and Studies
1	Sharing of Requirement documents	GoogleDocs, Drobox [19], Team Foundation Server (TFS) [19], [20], Pivotal Tracker [20]
2	Team Communication	Cockpit, Mingle [33], Skype [28], [29], [34], Basecamp, Microsoft share point [20], Bootstrap today and Confluences [20], [35]

collaboration as presented in Table 2. These communication [27] and collaboration means are equally applicable in distributed as well as local development environments [26]. For scrum meeting, Skype is used [28], [29], for team collaboration, wikis, discussion forums, real-time reports are used [30], AWS-EC2 instance provides databases for sharing data [31], [32].

Requirements documents are shared with cloud computing services. Cloud computing helps in instant file sharing, idea sharing, and discussion forums, wikis, real-time reports and code sharing [30]. Furthermore, Software as a Service (SaaS) is a way to provide project management, code management and software testing tools. In addition, Emails [36], Skype chat [28], [29], [34], and video conferencing, cloud telephony by Amazon Web Service (AWS) [32] are also used for communication as shown in Table 2. NFR elicitation can be improved by enabling the Agile member to load requirement document from different locations using the cloud [22]. In addition, the finalized requirement document can be uploaded on the team foundation server.

IV. PROPOSED NFR ELICITATION METHODOLOGY

The study proposed a methodology for NFR Elicitation (NFRElicit) for agile software development using cloud computing. The elicitation methodology is based on eight (8) steps as shown in Figure 1. The proposed elicitation methodology is the blend of multiple concepts, the method of reusing existing knowledge, method of elicitation and structured meeting techniques for NFR elicitation described by Kopczyóska *et al.* [37] are adopted. The proposed elicitation methodology is also inspired by the suggestion given by Too *et al.* [38] for improving elicitation of NFR. Question answering used in NFRElicit methodology is adopted by Zachman [39] for elicitation. The framework provides a structure to organize the comprehensive representation of information technology architecture data. The project historical data is used to predict the new NFR [13], [40], [41]. The proposed NFRElicit methodology includes the role of the experts and the previous data of organization for NFR elicitation. In agile methods, user story card is used for elicitation of functional requirements.

In step 1, initial requirements are collected from user or client during interview/face to face meeting [6]. In a distributed environment, the meeting is held with the help of services provided by Agile Development in Cloud Computing (ADCC) framework [26]. In this phase, Zachman framework helps in eliciting information from the user. The key questions of framework what, why, where, who, when and how help in questioning designing and asking. For example, “Who is the user against this requirement statement”, “What is the action performed”, “What are the keywords in requirement statements”. Another source that can help in the preliminary process of retirement elicitation is the use of a template for NFR presented by Kopczyńska *et al.* [37].

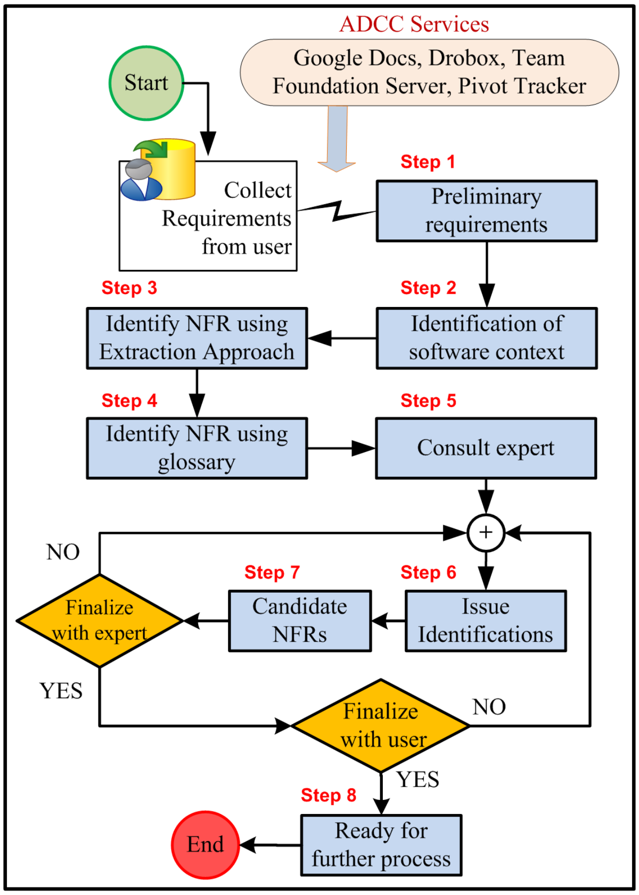

ADCC Services
Google Docs, Drobox, Team Foundation Server, Pivot Tracker
Start
Collect Requirements from user
Step 1
Preliminary requirements
Step 3
Identify NFR using Extraction Approach
Step 2
Identification of software context
Step 4
Identify NFR using glossary
Step 5
Consult expert
+
NO
Step 7
Step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

FIGURE 1. Proposed NFRElicit methodology.

The step 2 identifies the type of software based on 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. The software type helps to design possible applicable NFR for the software. The relevant NFR used in the context of software types are reported in the literature [42], [43].

In step 3, NFR are extracted from the requirement statements by using the NFR extraction approaches [44], [45], methodologies [23] and tools [46], if requirements are found in written form. These automated solutions for extraction of NFR will help to meet the needs of agile development. Use of NLP based tools speed up requirement elicitation process [47]. In this step, the study utilize ANFRX approach [45].

For further identification of NFR, the glossary is used in step 4. The glossary comprises of organizations data, the opinion of seniors and experts, bibliographic resources and quality standards [48]. The quality concerns of different stakeholders and associated quality attributes are described by the Boehm’s study [11]. In addition, NFR classification is defined in ISO 25010 [49] and ISO/IEC 9126 quality standard [11], [50]. For identification of NFR Zachman framework also help to analyze requirement statements analytically [39].

Agile software process promotes the self-organizing team and cooperative behavior. The technical expert is rectified (if needed) relevant to the project context and NFR type. The experts involved in this activity until the selection of requirements to be treated in the project. In step 5, the developer can take help from the concerned specialist regarding question (asked the user) to elicit NFR in the software. In the Agile method, the selection of maybe during a scrum meeting or the developer can take help from the team leader in identifying the relevant expert. The expert can have located in-house or in distribute environments. The help of expert can be acquired before and after elicitation while meeting client of user.

In step 6, the developer prepares the list of questions to gather requirement from the user and find the issues and the related NFR based on project type and expected NFR. The developer can take help from experts in preparing the requirement elicitation questions. The purpose of step 6 is to define the list of the issues against the 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.

For each question identify candidate NFR, which clearly explains the need of the customer. The attributes regarding NFR should be defined by the analyst to ask the user i.e. requirement type and its base class, dependency, and priority etc. The NFR keywords used in existing studies can be used to find candidate NFR.

In step 7, NFR are validated through expert and users. Share the candidate NFR along with complete set of questions against the expected NFR with an expert and discuss the quality of NFRs and interdependency between NFR. In case of change, send to Issue Identification phase. After validating with an expert, negotiate with the user. Validate and confirm NFR in natural language with the user. It is better to keep NFR in customer language. The validation through experts is from a technical point of view. If there are some changes, sent to issue identification phase to update process.

After completing all activities, NFR are ready for further use in other software development phases. In step 8, the developer finds a complete guideline to elicit NFR for certain project that can be used in the future. The developer has information about the expected NFR and appropriate expert to help in the elicitation process. The guideline act as supporting tools for the elicitation process.

V. EVALUATION CRITERIA AND DATASET

The evaluation of elicitation methodology was done by applying to the European Union (EU) electronic procurement (eProcurement), Volume 1 and 2 [51], [52]. In the eProcurement, 26 requirements from eProcurement document are taken as a baseline. These 26 requirement statements are manually classified into NFR list by NORMAP methodology and given in Appendix I in thesis document [53]. So, the NORMAP study is taken as a baseline. The EU eProcurement document is a real-life online system to manage procurement activities. The FR and NFR are described in mix within the requirement sentences. The Baseline study NORMAP identified 18 requirements types from total 26 requirements for validation. The NFR identified are Accessibility, Accuracy, Auditability, Availability, Compliance, Confidentiality, Documentation, Configuration, Efficiency, Interoperability, Legal, Multilingual, Performance, Reliability, Scalability, Security, Usability and User Interface. The same set of NFR is used in NERV methodology, CEP methodology and proposed NFRElicit methodology for validation. The list consists of NFR from EU eProcurement document given in Appendix I of NORMAP methodology report [53].

A. EVALUATION CRITERIA FOR NFR ELICITATION METHODOLOGY

The baseline utilized for validation of this study is a ‘‘Manual Classification’’ list generated by NORMAP study. The evaluation criteria are as follows:

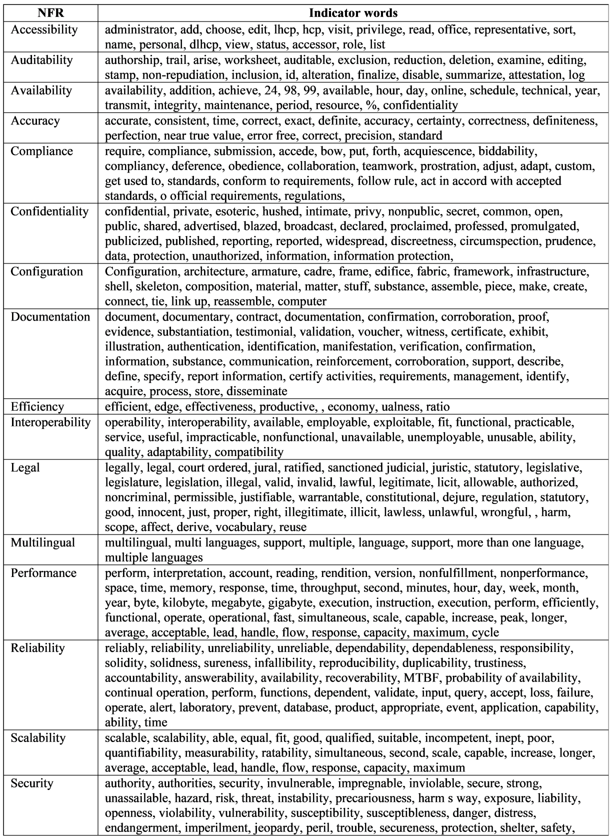
1) If the NFR detected by elicitation methodology is similar to the baseline, a ‘‘Success ’’ is declared in the result.
2) If all the NFR found by elicitation methodology is not similar to baseline i.e. partially found, a ‘‘Partial Success’’ is declared in the result.
3) If the baseline did not find some NFR and elicitation methodology find, a ‘‘Partial Success’’ is declared in the result.
4) If no NFR is identified, then a ‘‘Failure’’ is declared in the result.

B. EXECUTION OF ELICITATION METHODOLOGY

This Section explains the execution of elicitation process given in Figure 1. The execution starts with the preliminary requirement taken by the user or client. The medium used to extract the information is user story card or in case of agile distributed environment then use the environment provided in ADCC framework [26] for user communication and negotiation. After getting the initial requirement the requirement engineer or developer or analyst identifies the software context. At this stage, three artifacts help in identification of issues and NFR. The discussion with expert help to raise the issue. In case of mature requirements, the NFR can be extracted from the requirement statements through ANFRX approach [45] and further NFR would be drilled down by using glossary artifacts in the form of quality standards, projects history etc. In this paper for the evaluation, the mature requirement document (EU eProcurement document) is used, therefor user negotiation and preliminary requirement gathering phase are not used at this stage.

The requirement 1.2 for example, ‘‘The registration process must ensure the confidential transfer and storage of all personal information of users.’’, used the keywords registration, ensure, confidential, storage, transfer, storage, personal information and user. The ANFRX extraction approach

TABLE 3. NFR types and indicator keywords.



NFR	Indicator words
Accessibility	administrator, add, choose, edit, lhcp, hcp, visit, privilege, read, office, representative, sort, name, personal, dlhcp, view, status, accessor, role, list
Auditability	authorship, trail, arise, worksheet, auditable, exclusion, reduction, deletion, examine, editing, stamp, non-repudiation, inclusion, id, alteration, finalize, disable, summarize, attestation, log
Availability	availability, addition, achieve, 24, 98, 99, available, hour, day, online, schedule, technical, year, transmit, integrity, maintenance, period, resource, %, confidentiality
Accuracy	accurate, consistent, time, correct, exact, definite, accuracy, certainty, correctness, definiteness, perfection, near true value, error free, correct, precision, standard
Compliance	require, compliance, submission, accede, bow, put, forth, acquiescence, biddability, compliancy, deference, obedience, collaboration, teamwork, prostration, adjust, adapt, custom, get used to, standards, conform to requirements, follow rule, act in accord with accepted standards, o official requirements, regulations,
Confidentiality	confidential, private, esoteric, hushed, intimate, privy, nonpublic, secret, common, open, public, shared, advertised, blazed, broadcast, declared, proclaimed, professed, promulgated, publicized, published, reporting, reported, widespread, discreetness, circumspection, prudence, data, protection, unauthorized, information, information protection,
Configuration	Configuration, architecture, armature, cadre, frame, edifice, fabric, framework, infrastructure, shell, skeleton, composition, material, matter, stuff, substance, assemble, piece, make, create, connect, tie, link up, reassemble, computer
Documentation	document, documentary, contract, documentation, confirmation, corroboration, proof, evidence, substantiation, testimonial, validation, voucher, witness, certificate, exhibit, illustration, authentication, identification, manifestation, verification, confirmation, information, substance, communication, reinforcement, corroboration, support, describe, define, specify, report information, certify activities, requirements, management, identify, acquire, process, store, disseminate
Efficiency	efficient, edge, effectiveness, productive, , economy, ualness, ratio
Interoperability	operability, interoperability, available, employable, exploitable, fit, functional, practicable, service, useful, impracticable, nonfunctional, unavailable, unemployable, unusable, ability, quality, adaptability, compatibility
Legal	legally, legal, court ordered, jural, ratified, sanctioned judicial, juristic, statutory, legislative, legislature, legislation, illegal, valid, invalid, lawful, legitimate, licit, allowable, authorized, noncriminal, permissible, justifiable, warrantable, constitutional, dejure, regulation, statutory, good, innocent, just, proper, right, illegitimate, illicit, lawless, unlawful, wrongful, , harm, scope, affect, derive, vocabulary, reuse
Multilingual	multilingual, multi languages, support, multiple, language, support, more than one language, multiple languages
Performance	perform, interpretation, account, reading, rendition, version, nonfulfillment, nonperformance, space, time, memory, response, time, throughput, second, minutes, hour, day, week, month, year, byte, kilobyte, megabyte, gigabyte, execution, instruction, execution, perform, efficiently, functional, operate, operational, fast, simultaneous, scale, capable, increase, peak, longer, average, acceptable, lead, handle, flow, response, capacity, maximum, cycle
Reliability	reliably, reliability, unreliability, unreliable, dependability, dependableness, responsibility, solidity, solidness, sureness, infallibility, reproducibility, duplicability, trustiness, accountability, answerability, availability, recoverability, MTBF, probability of availability, continual operation, perform, functions, dependent, validate, input, query, accept, loss, failure, operate, alert, laboratory, prevent, database, product, appropriate, event, application, capability, ability, time
Scalability	scalable, scalability, able, equal, fit, good, qualified, suitable, incompetent, inept, poor, quantifiability, measurability, ratability, simultaneous, second, scale, capable, increase, longer, average, acceptable, lead, handle, flow, response, capacity, maximum
Security	authority, authorities, security, invulnerable, impregnable, inviolable, secure, strong, unassailable, hazard, risk, threat, instability, precariousness, harm s way, exposure, liability, openness, violability, vulnerability, susceptibility, susceptibleness, danger, distress, endangerment, imperilment, jeopardy, peril, trouble, secureness, protection, shelter, safety,

with the help of the indicator keywords "registration" identify the Documentation, the "confidential" keyword identifies the confidentiality and the "user" keyword identifies the accessibility and user interface type. The detail of indicator keywords utilized in study through ANFRX approach is given in Table 3.

Further identification of NFR is done through the context of software developed and the expert experience. According to the Zachman framework, the question related to identified NFR by extraction approach are further drill down to identify more NFR types. The Zackman enterprise architecture is based to answer what, why, how, where,who, when and why. In requirements elicitation questions are explored in this way. Each keyword has a related word gives direction to identify more NFR.by zackman framework. After using all these artifacts discussed, the NFR in all requirements in eProcurement dataset are identified with flag of "Success", "Partial Success" and "Fail"; expels are as follows:

The requirement 7.1 is "The new Public Procurement Directives require contracting authorities to use the CPV to advertise their procurement needs". The elicitation methodology identified Security, Document, Usability and Auditability by using keywords authority, contract, use, directives, respectively. The security further drill down the Availability and Confidentiality by using Chung classification [54]. By using ISO/IEC25010 standard, Usability further drills down the Accessibility and User Interface whereas the baseline finds Usability. So, it is declared successful.

In requirement 18.2, "To "open" or "unlock" Tenders, two or more procurement officers need to perform simultaneous actions". The extraction tool identified Accessibility, Legal, Performance and Security using keywords open, perform and authorized. The IEC/ISO standard identify Confidentiality from Security and Efficiency from Performance. The expert identifies the Configuration and Availability from structure and semantics of requirement statement. All NFR identified by elicitation methodology are Accessibility, Availability, Confidentiality, Configuration, Efficiency, Legal, Performance and Security. The baseline study identifies the NFR are Performance and Compliance. The elicitation methodology cannot find the compliance. So, according to validation criteria the result declared is "Partial Success". Some examples of NFR elicitation procedures is explained here, however, the study applies NFRElicit methodology to all requirements in the eProcurement data set and the details are given in supplementary material of paper. Furthermore, the concise results are described in next section.

VI. RESULTS OF NFR ELICITATION METHODOLOGY

The elicitation methodology is evaluated by applying on eProcurement dataset. The results are achieved by following the procedure described in Section IV. The validation criteria based on "Success", "Partial Success" and "Failure" is described in Section V-A. The result of elicitation methodology on Baseline NFR are shown in Table 4.

Table 4. has 18 baseline NFR types which are used by existing studies in the evaluation, Number of actual NFR in the dataset and NFR identified in term "Success", "Partial Success" and "Failure". Out of 88 baseline NFR, 81 identified as completely successful, five (5) partial success and one (1) failure. In five (5) partial success, the result of three (3) requirement sentence was "Partial Success" due to NONE type in the baseline as shown in Table 4.

In addition, the elicitation methodology finds the extra NFR type other than baseline NFR. The details of the total number of NFR identified against the baseline NFR are shown in Figure 2 "Security" with a count of 17 and in NFRElicit methodology, the NFR "Accessibility" is with a maximum count of 39. In the baseline study, three sentences were not identified and called "NONE" whereas in NFRElicit all requirement sentences are identified with some NFR type.

A. COMPARISON OF NFR ELICITATION METHODOLOGY WITH EXISTING STUDIES

The NFRElicit methodology shows overall improvement with more success and less partial success and failures. The paper compares the performance with three aspects: 1) comparison of requirement sentences with NFR in the categories of success, partial success and fail, 2) comparison of NFR identify in the categories of success, partial success and fail and 3) comparison of artifacts and feature used in solutions.

TABLE 4. NFRElicit methodology results in term of number of NFR.

Non Functional Requirement			NFR identified by NFRElicit Methodology					
Baseline NFR		# of actual requirements	Success		Partial Success		Failure	
			# of NFR	Percentage	# of NFR	Percentage	# of NFR	Percentage
1	Accessibility	10	10	11.36%	0	0	0	0
2	Accuracy	2	2	2.27%	0	0	0	0
3	Auditability	2	2	2.27%	0	0	0	0
4	Availability	8	8	9.09%	0	0	0	0
5	Compliance	9	7	7.95%	2	2.27%	0	0
6	Confidentiality	5	5	5.68%	0	0	0	0
7	Documentation	1	1	1.14%	0	0	0	0
8	Configuration	6	6	6.82%	0	0	0	0
9	Efficiency	1	1	1.14%	0	0	0	0
10	Interoperability	1	1	1.14%	0	0	0	0
11	Legal	1	1	1.14%	0	0	0	0
12	Multilingual	2	2	2.27%	0	0	0	0
13	Performance	5	5	5.68%	0	0	0	0
14	Reliability	1	1	1.14%	0	0	0	0
15	Scalability	1	0	0.00%	0	0	0	0
16	Security	17	16	18.18%	0	0	1	1.14%
17	Usability	7	7	7.95%	0	0	0	0
18	User Interface	6	6	6.82%	0	0	0	0
19	NONE	3	0	0	3	3.41%	0	0
	Total	88	81	92.04%	5	5.68%	1	1.14%

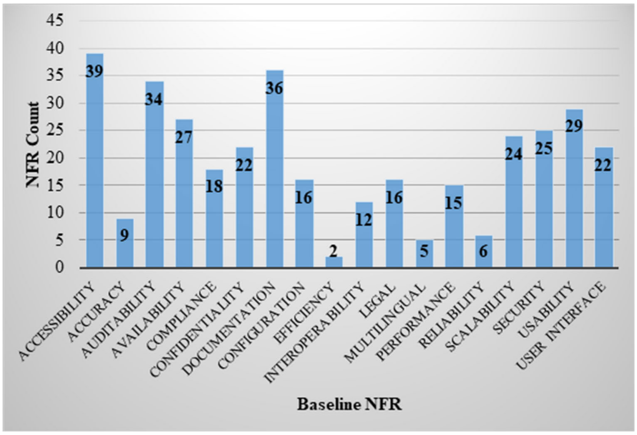


FIGURE 2. NFR distribution identified through elicitation methodology.

The comparison of complete success NFR identified in requirement sentences is given in Figure 3. The X-axis shows the methodologies and Y-axis is a number of complete success NFR identified in requirement sentences. With respect to complete successful identification, the elicitation methodology identifies 89.47% of the baseline NFR in comparison to NERV methodology of 80.70% and CEP methodology of 87.71% which is an improvement of 8.77% and 1.76% respectively.

The comparison of partial success NFR identified in requirement sentences is given in Figure 4. The X-axis shows the methodologies and left Y-axis is number requirement sentences containing partial success NFR identified by applying methodology and right Y-axis is the percentage. With respect to partial successful identification, the elicitation methodology identifies 8.77% of the baseline NFR in comparison to NERV methodology of 15.79% and CEP methodology of 10.52% which is an improvement in term of decreasing number of partial success NFR identifying in requirement sentences is 7.02% and 1.75% respectively.

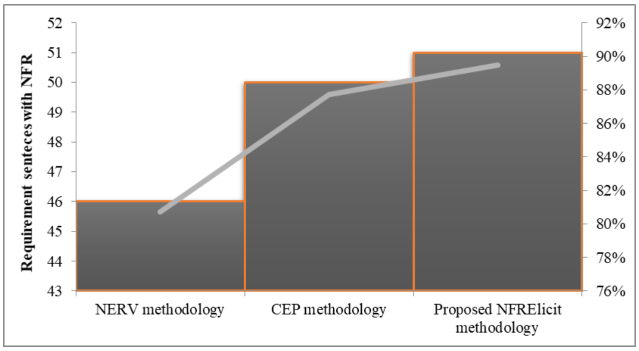


FIGURE 3. Comparison of success NFR in requirement sentences.

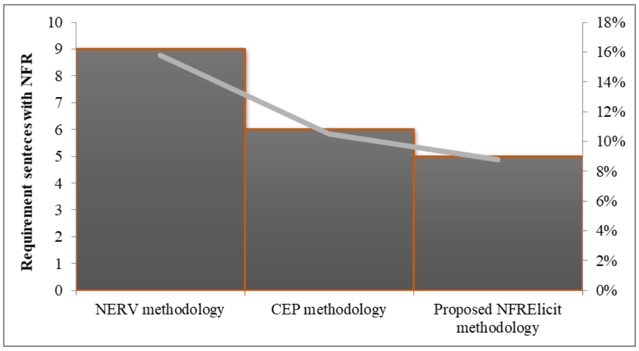


FIGURE 4. Comparison of partial success NFR in requirement sentences.

The comparison of failure to identify NFR in requirement sentences is given in Figure 5. The X-axis shows the methodologies for NFR elicitation and left Y-axis is a number of requirement sentences failed to identify NFR and right Y-axis is the percentage. With respect to the failure to identify, the elicitation methodology failed to identify 1.75% of the baseline NFR in comparison to NERV methodology of 3.51% and CEP methodology of 1.75% which is an improvement in term of decreasing number of failures to identifying NFR in requirement sentences is 1.76% and 0.0% respectively.

TABLE 5. Comparison with existing studies requirement sentences with NFR.

Criteria	NERV Methodology		CEP Methodology		Proposed NFRElicit Methodology	
	Requirement sentences with NFR	Percentage	Requirement sentences with NFR	Percentage	Requirement sentences with NFR	Percentage
Success	46	80.70%	50	87.71%	51	89.47%
Partial Success	9	15.79%	6	10.52%	5	8.77%
Failure	2	3.51%	1	1.75%	1	1.75%

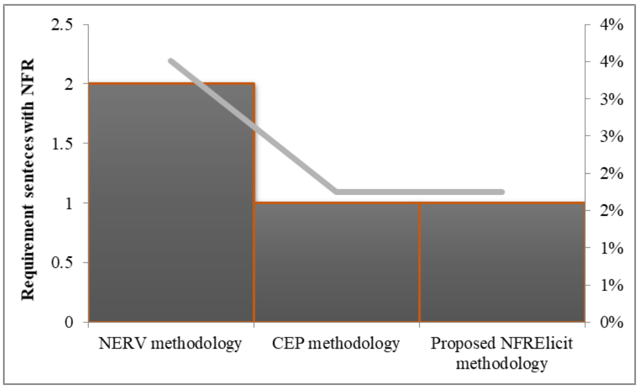


FIGURE 5. Comparison of failures to identify NFR in requirement sentences.

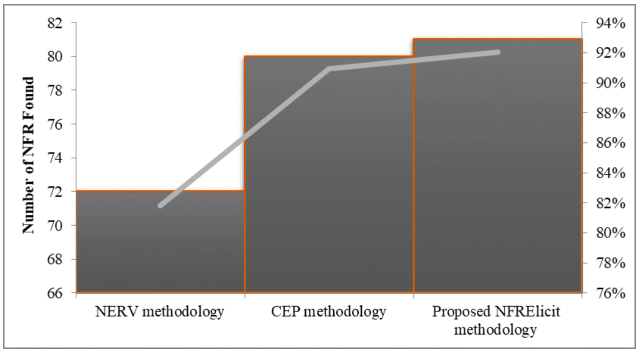


FIGURE 6. Comparison of NFR in requirement sentences with success.

The number of failures to identify in CEP and the proposed methodology is the same as shown in Figure 5.

In the eProcurement document, there are 56 requirement sentences. According to the validation criteria described in Section V-A, there might be more than one NFR type exist in one requirement sentence. The detail of a number of requirement sentence containing or identified NFR through the elicitation methodology along with their percentage is given in Table 5.

As there can be more than one NFR in one requirement sentence. Therefore, with respect to the number of NFR identified within the 57 requirement sentences under the category of success, partial success and fail. A total number of complete successful NFR for the NFRElicit methodology is 92.04% compared to NERV methodology 81.82% and CEP methodology 90.91% which is an improvement of 10.22% and 1.13% respectively. In Figure 6, the X-axis shows the methodologies and left side Y-axis is a number of complete success NFR identified in requirement sentences. The right Y-axis in the percentage of NFR identified.

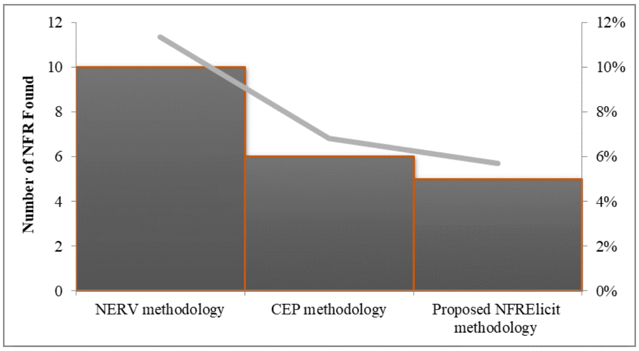


FIGURE 7. Comparison of NFR in requirement sentences with partial success.

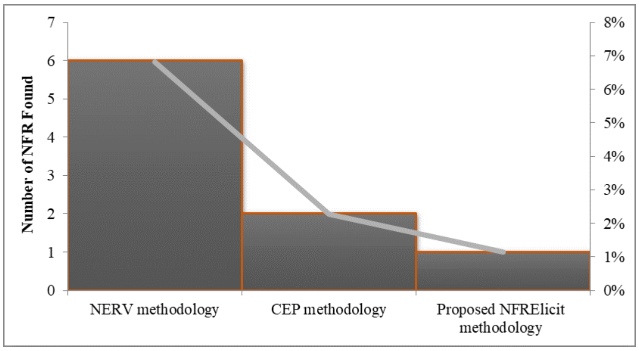


FIGURE 8. Comparison of NFR in requirement sentences with failure.

In Figure 7, the X-axis shows the methodologies and left side Y-axis is a number of partial success NFR identified in requirement sentences. The right Y-axis in the percentage of NFR identified. A total number of partial successful NFR for the NFRElicit methodology is 5.68% compared to NERV methodology 11.36% and CEP methodology 6.82% which is an improvement in term of decreasing number of partial success NFR are of 5.68% and 1.14% respectively.

In Figure 8, left side Y-axis is number of NFR identified in requirement sentences of category fail. The right Y-axis in the percentage of NFR identified the and X-axis shows the methodologies. A total number of NFR for NFRElicit methodology is 1.14% compared to NERV methodology 6.82% and CEP methodology 2.27% which is an improvement in term of decreasing number of failures NFR is of 4.58% and 1.13% respectively.

In eProcurement document, there is 88 NFR according to baseline NORMAP study. According to the validation criteria described in Section V-A, there might be more than one NFR type exist in one requirement sentence. The detail of a number of identified NFR through the elicitation methodology along

TABLE 6. Comparison with existing studies on number of NFR.

Evaluation Criteria	NERV Methodology		CEP Methodology		Proposed NFRElicit Methodology	
	Number of NFR detected	Percentage	Number of NFR detected	Percentage	Number of NFR detected	Percentage
Success	72	81.82%	80	6.82%	81	92.04%
Partial Success	10	11.36%	6	6.82%	5	5.68%
Failure	6	6.82%	2	2.27%	1	1.14%

TABLE 7. Feature wise comparison of existing studies.

Artifacts	NORMAP Methodology	NERV Methodology	CEP Methodology	NFRElicit Methodology
W8 Story Card	✓	✓	✓	×
W8 Story Card	✓	✓	✓	✓
Separate NFR story card	✓	✓	×	✓
Application Context	×	×	×	✓
Software glossary / Quality Standards	✓	✓	×	✓
Expert involvement	×	✓	×	✓
Project history / NFR trends	✓	×	✓	✓
Zachman framework	×	✓	×	✓
dependency/Prioritization	✓	✓	✓	✓
Short Note/ Pictures	×	×	✓	×
Use of Extraction Tool / approach	✓	×	✓	✓
Risk	✓	×	×	×
Part of Speech	✓	×	×	✓
Knowledge Based	✓	✓	×	×
Chung's Approach	✓	✓	×	×
FR to NFR linking	✓	✓	×	×
NFR visualization	✓	×	✓	✓

with their percentage in the three categories of success, partial success and failure is given in Table 6.

Another way to validate the elicitation methodology is by comparing the features or artifacts used in current and existing studies as shown in Table 7. The story card and prioritization are used in all studies. Project history, expert involvement and application context is only used in proposed NFRElicit elicitation methodology. NFR extraction from images is taken placed by only CEP methodology. The automated extraction is used in NFRElicit and CEP methodologies. Prioritization of NFR are corporate by NERV and CEP methodologies. The concept of separate NFR card is introduced in NERV and NFRElicit methodologies. The adoption of features shows that NFRElicit methodology not only corporates the feature of NERV or CEP as well as additional features such as application context, the role of expert and project history.

VII. DISCUSSION

The research presents a semi-automatic methodology for NFR elicitation. This methodology helps analyst, developer and user in eliciting NFR in agile software using cloud computing environment. The methodology is based on knowledge management and. This methodology incorporates an automatic approach for extracting NFR from requirement document, a glossary of software engineering, quality standards, history of project data, experts and seniors in the organization and bibliographic resources to implement the methodology. The previous study NERV methodology used Zackman framework, NFR trigger card, quality standards and glossary resources. The proposed NFRElicit methodology used all the artifacts used by NERV and also used the automatic NFR extraction approach ANFRX. A total number of complete successful NFR for the NFRElicit methodology is 97.73% compared to NERV methodology 81.82% which is an improvement of 10.22%.

Although, the other existing study CEP methodology used NFR Locator Tool for NFR extraction and also NFR Locator Plus for extracting NFR from the images by capturing only text used in images. However, NFRElicit methodology used the ANFRX approach which has better performance of extraction then NFR Locator. The extraction of NFR from images is covered by the experts in the proposed NFRElicit methodology. The improvement of NFRElicit methodology over NERV and CEP methodology in identifying NFR with the improvement of 10.22% and 1.13% respectively. NFRElicit methodology improves by identifying more Success and less Partial Success and Fail NFR.

VIII. CONCLUSION

This study investigates and traces the non-functional requirements in agile development using a cloud computing environment. The tracing of NFR with respect to eliciting NFR from the users and other stakeholders of a software project. The tracing with respect to identifying NFR from the software requirement documents in the form of interview notes, user stories, stakeholder messages, etc. The need of environment to perform requirement engineering activities in agile development using cloud computing.

The research study presents a solution to help the user and requirement engineers in the elicitation process. The methodology is based on human intervention and automated support. The methodology utilized known artifacts to incorporate NFR elicitation in ADCC. The artifacts are the project history regarding NFR, quality standards, questioning and reasoning through Zachman framework, automated NFR extraction artifacts and involvement of seniors and experts in the organization. The methodology is evaluated by applying in electronic procurement (eProcurement) dataset. The results are compared with the existing works. The proposed NFRElicit methodology improve in terms of more success NFR findings and a smaller number of partial success and failure NFR.

ACKNOWLEDGMENT

The authors are thankful to Universiti Teknologi Malaysia (UTM) and Government College University Faisalabad (GCUF), Pakistan for providing resources and also acknowledge Universiti Teknologi Malaysia for grant UTM-TDR Grant Vot No. 06G23.

REFERENCES

[1] S. Gochhait, S. A. Butt, T. Jamal, and A. Ali, "Cloud enhances agile software development," in *Cloud Computing Applications and Techniques for E-Commerce*. Hershey, PA, USA: IGI Global, 2020, pp. 28–49.

[2] H. Benfenatki, C. F. D. Silva, A.-N. Benharkat, and P. Ghodous, "Cloud-based business applications development methodology," in *Proc. IEEE 23rd Int. WETICE Conf.*, Jun. 2014, pp. 275–280.

[3] P. Mell. (2011). *The NIST Definition of Cloud Computing*. Accessed: Feb. 20, 2016. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[4] S. A. Butt, M. I. Tariq, T. Jamal, A. Ali, J. L. Diaz Martinez, and E. De-La-Hoz-Franco, "Predictive variables for agile development merging cloud computing services," *IEEE Access*, vol. 7, pp. 99273–99282, 2019.

[5] Q. L. Nguyen, "Non-functional requirements analysis modeling for software product lines," in *Proc. ICSE Workshop Modeling Softw. Eng.* Washington, DC, USA: IEEE Computer Society, May 2009, pp. 56–61.

[6] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," in *Proc. 12th IEEE Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises*, Jun. 2003, pp. 308–313.

[7] S. Kopczyńska, M. Ochodek, and J. Nawrocki, "On importance of non-functional requirements in agile software projects—A survey," in *Integrating Research and Practice in Software Engineering*. Cham, Switzerland: Springer, 2020, pp. 145–158.

[8] C. Hoskinson, "Army's faulty computer system hurts operations," Politico, Arlington, VA, USA, Tech. Rep., 2011. [Online]. Available: https://www.politico.com/story/2011/06/computer-bugs-hurt-army-ops-058051

[9] J. Bertman and N. Skolnik, "EHRs get a failing grade on usability," *Internal Med. News*, vol. 43, no. 11, p. 50, Jun. 2010.

[10] V. Bajpai and R. P. Gorthi, "On non-functional requirements: A survey," in *Proc. IEEE Students' Conf. Elect., Electron. Comput. Sci.*, Mar. 2012, pp. 1–4.

[11] B. Boehm and H. In, "Identifying quality-requirement conflicts," *IEEE Softw.*, vol. 13, no. 2, pp. 25–35, Mar. 1996.

[12] S. Kopczyńska, "Supporting non-functional requirements elicitation with templates," Ph.D. dissertation, Poznań Univ. Technol., Poznań Poland, 2019. [Online]. Available: https://sin.put.poznan.pl/dissertations/details/d58

[13] 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 *Proc. SoutheastCon*, Apr. 2015, pp. 1–8.

[14] S. Kopczyńska and J. Nawrocki, "Using non-functional requirements templates for elicitation: A case study," in *Proc. IEEE 4th Int. Workshop Requirements Patterns (RePa)*, Aug. 2014, pp. 47–54.

[15] W. Helmy and A. K. O. Hegazy, "An evaluation framework for requirements elicitation in agile methods," in *Proc. Int. Conf. Softw. Eng. Adv.*, Lispon, Portugal, 2012, pp. 588–593.

[16] D. D. Gregorio, "How the business analyst supports and encourages collaboration on agile projects," in *Proc. IEEE Int. Syst. Conf. SysCon*, Mar. 2012, pp. 1–4.

[17] 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*. Cham, Switzerland: Springer, 2016, pp. 293–302.

[18] M. Younas, D. N. Jawawi, I. Ghani, and R. Kazmi, "Non-functional requirements elicitation guideline for agile methods," *J. Telecommun., Electron. Comput. Eng.*, vol. 9, nos. 3–4, pp. 137–142, 2017.

[19] T. Haig-Smith and M. Tanner, "Cloud computing as an enabler of agile global software development," *Issues Informing Sci. Inf. Technol.*, vol. 13, no. 13, pp. 121–144, 2016. [Online]. Available: http://www.informingscience.org/Publications/3476

[20] S. Karunakaran, "Impact of cloud adoption on agile software development," in *Software Engineering Frameworks for the Cloud Computing Paradigm*. London, U.K.: Springer, 2013, pp. 213–234.

[21] A. Cockburn and J. Highsmith, "Agile software development: The people factor," *Computer*, vol. 34, no. 11, pp. 131–133, Nov. 2001.

[22] R. R. Maiti, "Capturing eliciting, and prioritizing (CEP) non-functional requirements metadata during the early stages of agile software development," College Eng. Comput., Nova Southeastern Univ., Fort Lauderdale, FL, USA, Tech. Rep., 2016. [Online]. Available: https://nsuworks.nova.edu/gscis_etd/968/

[23] W. M. Farid, "The NORMAP methodology: Lightweight engineering of non-functional requirements for agile processes," in *Proc. 19th Asia–Pacific Softw. Eng. Conf.*, Dec. 2012, pp. 322–325.

[24] D. Domah and F. J. Mitropoulos, "The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–7.

[25] N. A. A. Fauzi, R. Hassan, and Z. A. Shah, "Extraction of non-functional requirement in natural language document using keyword with usability quality aspect," *Int. J. Softw. Eng. Technol.*, vol. 2, no. 2, pp. 5–10, 2017. [Online]. Available: https://engineering.utm.my/computing/ijset/

[26] M. Younas, I. Ghani, D. N. A. Jawawi, and M. M. Khan, "A framework for agile development in cloud computing environment," *J. Internet Comput. Services*, vol. 17, no. 5, pp. 67–74, Oct. 2016.

[27] M. A. Shah, R. Hashim, A. A. Shah, and U. F. Khattak, "Communication management guidelines for software organizations in Pakistan with clients from Afghanistan," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, 2016, Art. no. 012100.

[28] I. Zarinah and M. Kasirun, "Agile-based software product development using cloud computing services: Findings from a case study," *Sci. Int. J. (Lahore)*, vol. 7, no. 5, pp. 67–74, 2013.

[29] S. Kalem, D. Donko, and D. Boskovic, "Agile methods for cloud computing," in *Proc. 36th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2013, pp. 1079–1083.

[30] S. Singh and I. Chana, "Introducing agility in cloud based software development through ASD," *Int. J. U- E- Service, Sci. Technol.*, vol. 6, no. 5, pp. 191–202, Oct. 2013.

[31] F. Almudarra and B. Qureshi, "Issues in adopting agile development principles for mobile cloud computing applications," in *Proc. 6th Int. Conf. Ambient Syst., Netw. Technol.*, in Procedia Computer Science, E. Shakshuki, Ed. Amsterdam, The Netherlands: Elsevier, 2015, pp. 52 and 1133–1140.

[32] M. Manuja, "Moving agile based projects on cloud," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, 2014, pp. 1392–1397.

[33] P. Raj, V. Venkatesh, and R. Amirtharajan, "Envisioning the cloud-induced transformations in the software engineering discipline," in *Software Engineering Frameworks for the Cloud Computing Paradigm*. London, U.K.: Springer, 2013, pp. 25–53.

[34] M. R. J. Qureshi and I. Sayid, "Scheme of global scrum management software," *Int. J. Inf. Eng. Electron. Bus.*, vol. 7, no. 2, pp. 1–7, Mar. 2015.

[35] W.-T. Tsai, W. Wenjun, and M. N. Huhns, "Cloud-based software crowdsourcing," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 78–83, May/Jun. 2014.

[36] S. Franken, S. Kolvenbach, W. Prinz, I. Alvertis, and S. Koussouris, "CloudTeams: Bridging the gap between developers and customers during software development processes," *Procedia Comput. Sci.*, vol. 68, no. 68, pp. 188–195, 2015.

[37] S. Kopczyńska, S. Maćkowiak, and J. Nawrocki, "Structured meetings for non-functional requirements elicitation," *Found. Comput. Decis. Sci.*, vol. 36, no. 1, pp. 35–56, 2011.

[38] C. W. Too, S. Hassan, J. Din, A. A. Ghani, and A. Azim, "Towards improving NFR elicitation in software development," *Int. J. Inf. Technol. Comput. Sci.*, vol. 7, no. 1, pp. 33–44, 2013.

[39] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 38, nos. 2–3, pp. 454–470, 1999.

[40] M. Oriol, P. Seppänen, W. Behutiye, C. Farré, R. Kozik, S. Martínez-Fernández, P. Rodríguez, X. Franch, S. Aaramaa, A. Abhervé, M. Choras, and J. Partanen, "Data-driven elicitation of quality requirements in agile companies," in *Proc. Int. Conf. Qual. Inf. Commun. Technol.* Cham, Switzerland: Springer, 2019, pp. 49–63.

[41] M. Oriol, S. Martínez-Fernández, W. Behutiye, C. Farré, R. Kozik, P. Seppänen, A. M. Vollmer, P. Rodríguez, X. Franch, S. Aaramaa, A. Abhervé, M. Choras, and J. Partanen, "Data-driven and tool-supported elicitation of quality requirements in agile companies," *Softw. Qual. J.*, pp. 1–33, Mar. 2020, pp. 28 and 931–963.

[42] D. Mairiza, D. Zowghi, and N. Nurmuliani, "An investigation into the notion of non-functional requirements," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2010, pp. 311–317.

[43] M. Pierdin and G. Bulder, "Dimensions and types of non-functional requirements NFR dimensiones y tipos de requisitos no-funcionales RNF," *RNF, Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*, vol. 2, no. 1, pp. 11–17, 2012.

[44] A. Mahmoud, "An information theoretic approach for extracting and tracing non-functional requirements," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 36–45.

[45] M. Younas, D. N. A. Jawawi, I. Ghani, and M. A. Shah, "Extraction of non-functional requirement using semantic similarity distance," *Neural Comput. Appl.*, vol. 32, pp. 7383–7397, May 2019.

[46] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," in *Proc. 1st Int. Workshop Natural Lang. Anal. Softw. Eng. (NaturaLiSE)*, May 2013, pp. 9–16.

[47] E. C. Groen, J. Schowalter, S. Kopczynska, S. Polst, and S. Alvani, "Is there really a need for using NLP to elicit requirements? A benchmarking study to assess scalability of manual analysis," in *Proc. REFSQ Workshops*, 2018, pp. 1–10.

[48] L. Chung and J. C. S. do Prado Leite, "On non-functional requirements in software engineering," in *Conceptual Modeling: Foundations and Applications*. 2009, pp. 363–379.

[49] *Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models ISO*, Standard ISO-IEC 25010, Normalización, O.I.d, 2011.

[50] X. Franch and J. P. Carvallo, "Using quality models in software package selection," *IEEE Softw.*, vol. 20, no. 1, pp. 34–41, Jan. 2003.

[51] (2005). *S.A., E.D. Functional Requirements for Conducting Electronic Public Procurement Under the EU Framework Volume I*. Accessed: Oct. 8, 2016. [Online]. Available: http://ec.europa.eu/idabc/servlets/Doc1ad3.pdf?id=22191

[52] (2005). *S.A., E.D. Functional Requirements for Conducting Electronic Public Procurement Under the EU Framework Volume II*. Accessed: Oct. 8, 2016. [Online]. Available: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/functional-reguirements-vol2_en.pdf

[53] W. M. Farid, *The NORMAP Methodology: Non-Functional Requirements Modeling for Agile Processes*. Fort Lauderdale, FL, USA: Nova Southeastern University, 2011.

[54] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, vol. 5. New York, NY, USA: Springer, 2012.

MUHAMMAD YOUNAS (Member, IEEE) received the degree from the Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia. He is working as an Assistant Professor with the Department of Computer Science, Government College University Faisalabad. His research interests include software engineering, agile software development, cloud computing, and code clone detection.

DAYANG NORHAYATI ABANG JAWAWI (Member, IEEE) received the Ph.D. degree. She is an Associate Professor and the Deputy Dean (academic and student development) of the Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia, where she is also a member of the Department of Software Engineering. She is also a member of the Software Engineering Research Group (SERG).

MUHAMMAD ARIF SHAH received the M.Sc. degree in software engineering from UET (Mardan Campus), Peshawar, and the Ph.D. degree in software engineering and data science from University Technology Malaysia (UTM). He is an Assistant Professor in software engineering with the Pak–Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan. He received four awards from academia, industry, and research community.

AHMAD MUSTAFA is with the Department of Software Engineering, School of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia.

MUHAMMAD AWAIS received the Ph.D. degree. He is working as an Assistant Professor as well as the Incharge of the Department of Software Engineering, Government College University Faisalabad (GCUF).

MUHAMMAD KAMRAN ISHFAQ is working as a Researcher with the Research Center, Sulaimani Polytechnic University, Sulaimani, Iraq.

KARZAN WAKIL (Member, IEEE) received the Ph.D. degree. He is working as a Researcher with the Research Center, Sulaimani Polytechnic University, Sulaimani, Iraq.