

## A review on automatic extraction and classification of non-functional requirements



Rahmat Ullah Khan\*, M. N. A. Khan

*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan*

### ARTICLE INFO

#### Article history:

Received 4 October 2016

Received in revised form

10 April 2017

Accepted 2 May 2017

#### Keywords:

Requirement engineering

Non-functional requirements

Classification of NF

### ABSTRACT

Extraction and classification of NFRs plays a vital role in the software development process. NFRs are often misunderstood and ignored due to which certain important aspect of the software such as robustness availability usability performance and security are compromised. Several different extraction and classification techniques proposed in the literature. In this study, we have made an effort to critically analyze different NFRs extraction and classification techniques in order to see their merits and demerits, as well as the scenarios in which these techniques can be helpful. Specifically we have identified the limitations of these techniques, their significance and the possible improvements that can be made to make these techniques more useful. In this paper, we have critically evaluated various techniques and methods proposed in the literature on the automatic extraction and classification of NFRs. The findings of the existing techniques are provided in the paper. The critical analysis tables are provided in this study to highlight the significance of NFRs extraction and classification technique.

© 2017 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### 1. Introduction

Requirements are considered as one of the most important entity in software development. It plays an important role in the success or the failure of the project. Requirement engineering is crucial in the development of software process but often requirements are ignored in different projects (Li et al., 2015). Requirements Analysis is a part of requirement engineering which helps in determining the expectations of the user related to the newly or modified product. Requirement analysis is a lengthy process which determines whether these stated requirements are complete or incomplete. Requirements are categorized into functional and non-functional requirements. Functional requirements specify a particular function to be provided by the developed software. It deals with input and output behavior of a system. Non-functional requirements are described as the quality attributes of the system. These, requirements are also called quality requirements. Non-functional requirements are known to be the most difficult to deal

with in software development (Umar and Khan, 2011). These requirements describe how the system should operate in certain environment. It covers a variety of quality attributes that include performance, reliability, security, operability, usability of the system. Extensive knowledge is required regarding requirement specification of software both in non-functional requirements and functional requirements point of view (Odeh and Odeh, 2011).

Non-functional requirements (NFRs) include usability requirements that consider user factors as one of the important factor in system development. Easy system should be easy to use, also complete user documentation and help wizard explains how to achieve the tasks. Reliability is another important requirement which deals with the accuracy of the system. A system would be reliable if its predictability and recoverability are high. Performance of the system depends upon throughput, speed, turnaround time and efficiency of the system. Supportability is another category of NFRs which relates to the monitoring and maintaining of the system and ease of system installation.

Automatic extractions of NFRs from the large document at early stages are important. It is a difficult process to analyze and implement NFRs from variety of sources such as requirement

\* Corresponding Author.

Email Address: [rahmatnazi88@gmail.com](mailto:rahmatnazi88@gmail.com) (R. U. Khan)

<https://doi.org/10.21833/ijaas.2017.06.005>

2313-626X/© 2017 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

document, SRS, guidelines, SOPs, brief notes etc. The benefit of extracting the non-functional requirements automatically is to aid the analysts in extracting these requirements effectively so that it reduce the costly re-work and time consuming issues.

Classification of NFRs is important because they are often mixed with each other, it is important to identify these requirements separately. Methods that are used for the classification of NFRs are i) Elicitation techniques ii) Detection techniques. Elicitation technique relies heavily on brainstorming; checklists and NFRs template are used to take the inputs from the stake holders (Cleland-Huang et al., 2006). NFRs catalogue helps the analysts to identify the quality requirements to reduce the conflicts. Detection techniques detect the low-level early aspects in design and code from different requirement specification document. The early aspect detection techniques used for identification and classification of NFRs requires a lot of interaction with the user, so it is important to for the identification and extraction of different non-functional requirements to use an automatic approach.

Non-Functional requirements can be split into main categories and sub categories. Identification of both categories plays a key role in software development. Table 1 outlines Non-functional requirements main categories and sub-categories.

**Table 1:** Main categories and sub-categories of Non-Functional Requirements

Main Categories	Sub Categories
Performance	Speed, Response Time, Throughput, Availability
Security	Confidentiality, Authentication, Integrity, Accountability
Reliability	Completeness, Robustness, Consistency, Maturity, Fault tolerance, Accuracy
Usability	Accessibility, User error, Operability, Learnability
Maintainability	Modularity, Testability, Modifiability, Analyzability
Portability	Compliance, Adaptability, Co-existence, Replace ability
Efficiency	Time behavior, Resource behavior

In view of significance importance of non-functional requirements, we conducted a literature review on the study of different techniques and methods used to extract, classify and analyze non-functional requirements.

This paper is structured as follows. Section 2 discusses strengths and weaknesses of existing work. Section 3 covers critical evaluation of existing approaches and section 4 covers conclusion and the future work.

## 2. Methodology

In this research study, a systematic literature review has been conducted in accordance with the guidelines recommended by (Kitchenham and Charters, 2007) within the ambit of software

engineering. This study based on identifying, evaluating and interpreting the existing research inputs contribution relevant to the automated extraction and classification of NFRs in particular, the study focuses on the following three main aspects:

- Summarize the existing evidence of the benefits and limitations of automated extraction and classification of NFRs techniques.
- Identify gaps in the current research in order to discover potential areas for further investigations during the course of MS thesis research.
- Provide a comprehensive research background in order to appropriately position expected future research activities in the domain of NFRs extraction and classification.

## 3. Literature review

In this section literature review of existing techniques for automatic identification, classification and analysis of NFRs is conducted.

Casamayor et al. (2010) proposed a semi-supervised text categorization technique for identifying NFRs from a textual document, the supervised text categorization technique proposed earlier lot of pre categorize requirements are required to train a classifier before finding an accurate NFRs, with supervised technique it required manually categorization of numerous requirements by the analyst. This study has tried to automate this process. The learning method in the classification process used reduce number of categorize requirements as compared to the supervised approach. The benefit of this approach is that it is successfully used during the requirement analysis process and reduced the effort needed for manual identification and classification. The semi-supervised approach shows accuracy results that are above 70% which is higher than the supervised learning results using the same standards for the collection of documents.

Rahimi et al. (2014) proposed a data mining technique for extracting non-functional requirements. The proposed technique captures the quality concerns such as usability, performance and security of the system from the document .A hierarchy is developed which helps the extracted NFRs to model them according to the quality concerns. The proposed data mining technique is helpful in extracting the quality attributes in automatic extraction of non-functional requirements. Sequence of machine learning and data mining techniques are used in the paper to automatically detect different quality concerns from the document. A meaningful hierarchy is proposed to organize these quality concern, some are related to each other so at different stages of the hierarchy some relevant attributes are neglected to improve the performance of the model.

Slankas and Williams (2013) developed a tool based

approach called NFR Locator. The proposed tool helps the analyst in extracting non-functional requirements effectively from natural language documents. This tool is used to identify different NFRs according to their categories from available natural language documents. A k-NN classifier is used to identify the similar types of sentences in documents. Classification of the sentences is made on the basis of different categories of NFRs. It helps the analyst to extract those non-functional requirements that are relevant. Multiple types of classifiers are used in the paper and it is resulted that k-NN classifier achieve the maximum result in identifying non-functional requirements.

Ramadhani et al. (2015) proposed an automated system for the identification of NFRs taking account of an algorithm FSKNN (Fuzzy similarity based K-nearest neighbor) a requirement sentences-based classification algorithm. In FSKNN algorithm semantic factors and semantic relatedness measurement are not considered. The propose system classify different non-functional requirements from text documents. The system works on labeling of training data, classification of the data, measure the semantic relatedness between different classes and used terms. Automated process of labeling training data save the time than labeling the data manually. HSO method is used for measuring the semantic relatedness between the words. The method checks the semantic relatedness between every class and the term that is processed. The result show that with the addition of semantic factors improves the accuracy by 43.7% comparing to the Fuzzy similarity based K-nearest neighbor algorithm which is 41.4%.

Slankas and Williams (2013) proposed a tool assisted process security discoverer to identify the security requirements and classify the requirement sentences according to their relevant security objectives. A set of categories for security objective are created that requirement engineer consider during requirement engineering process. A tool is used to identify security related sentences in terms of security objectives. Context specific template is used in the paper which identify which are the requirements that meet their security objectives, k-NN and naïve Bayes classifiers are used and the study reported 82% prediction and 79% identification of security attributes from the document has been achieved. Classification approach identifies high precision of security objectives from classified sentences.

Sharma et al. (2014) proposed a framework for identifying and analyzing different non-functional requirements from the text document. A textual pattern identification technique is propose to identify terms that are related to non-functional requirements attributes from a natural language text and on the basis of applying different set of rules it identify different categories of NFRs. The proposed rule based approach for detecting and classifying different NFRs in natural language uses rules instead of identifying keywords approach like other machine

language techniques. Developed approach is analyzed against different manual categorize approaches for identifying non-functional requirements from the sentences.

Rahman and Ripon (2014) proposed a UML model on the basis of a questionnaire technique to elicit non-functional requirements in software development at early stages. UML use case is used to represent functional requirements, that are gathered and integrate through a questionnaire to extract NFRs on the basis of answers taken from list of questions include in the questionnaire. Stakeholder contribution is important in answering different questions to elicit non-functional requirements from functional requirements. The case study is applied on a Point of Sale (PoS) system. The proposed approach categorized the set of elicit non-functional requirements into a set of categories of well-defined non-functional requirements which is useful for tracking these non-functional requirements at different stages of the software development process. A tabular form representation is given to track the different NFRs at different levels to help the developers and customers in a cost effective purpose.

Gazi et al. (2015) proposed a classification scheme of NFRs for Information systems (IS). Many classification schemes are proposed for NFRs but they do not classify requirements for IS, web base system, real time system. A tree like structure is proposed for classifying NFRs. In this classification scheme, similar NFRs are identified for both real systems and web based systems. It is important that the NFRs that are included in classification scheme for IS are included in the software requirements specification document. Reliability and availability are two important NFRs for information system. In the classification scheme reliability requirement is further decomposed into accuracy, maturity, completeness. Identification of different NFRs in this paper are based on their similarity, accuracy and confidentiality are those non-functional requirements that are similar in IS as well as real time systems, interoperability and privacy are those non-functional requirements that are similar in web base systems and information systems, security, performance and usability requirements are similar in real time systems and web base systems.

Mahmoud and Williams (2016) used a multi-step unsupervised approach for detecting and classifying the non-functional requirements. The early methods used for classification and detection of non-functional requirements use manually classified data to train the model, classifier needs large training data set but for achieving high accuracy large data is not always available. A technique is used for extracting natural language content of source code to support NFRs traceability. Words semantic similarity methods are used in context to software requirements. Cluster configuration is used to generate the most logical clusters of requirements words. The proposed approach shows a modest complexity that helps it to scale it on larger systems

without wearing issues of time and space requirements. The proposed approach is unsupervised so it cannot require any data set so it can operate with minimum adjustment. The paper highlights long term benefit for software development process.

Sadiq et al. (2011) proposed a mechanism to identify NFRs in service oriented systems. NFRs such as availability, security and usability are not focused in service oriented systems. A quality model is proposed based on software quality standards. Different high priority NFRs are selected for service oriented systems domain. Quality requirements are gathered early and verified through quality model. Quality model are linked with Service Level Agreement (SLA) help customers to specify quality requirement. The evaluation model helps the developers and the customers to check the quality requirement at any time during service is in operation. Proposed model can be applied on different software development paradigms. Non-functional requirements are categorized and sub-categorized according to their different attributes. With quality and evaluation process almost all analyzed requirements are found to be correct which also reduce the time and cost in the phase of requirement engineering.

Mahmoud and Niu (2015) performed an experimental analysis to evaluate the performance of different semantic Information Retrieval (IR) methods for automated requirement tracing and investigate the potential of natural language semantics in automated tracing. Objective behind this research is to get an insight of different operations of IR methods for the identification of NFRs and provide a guideline for the effective requirement tracing techniques and their management tools. A systematic analysis is performed on information retrieval methods on the basis of identifying functional and non-functional requirements in software systems. Semantic augmented methods and their sub categorize thesaurus support (VSM-T) and vector space model with Part-of-Speech tagging (VSM-POS) are explain in information retrieval method.

An experimental analysis is conducted on the performance of different IR methods that includes the methods of latent semantic, semantic relatedness and segmented-augmented methods. The result shows that higher semantic relation is not necessary in improving the performance of retrieval process whereas a focused explicit semantic in domain specific thesaurus able to achieve a higher performance.

Meth et al. (2013) proposed a framework to capture the current state of the art in automated requirements elicitation and extract different future research directions by identifying the possible gaps in the existing domains and through relating existing works. A systematic method is use to literature how different automation elicitation processes are performed by different requirement elicitation techniques.

Identified works are than categorized using an analysis framework that compares tool categories, evaluation approaches and technological concepts. The authors intend to contribute to the body of knowledge from requirement engineering as conceptualizing an analysis framework that works in the area of automated requirements elicitation.

The authors propose future research to different areas include comparison of different types of knowledge on elicitation process results. The proposed work in this paper helps in classifying the different tools in state of the art in automation requirement process on the basis of degree of tool automation, knowledge reused and evaluation of concepts.

Mizouni and Salah (2010) proposed a framework to estimate the system non-functional requirements on behavior model. NFRs are not handled properly in behavior models at early stages cause in failure of the system. A framework proposed that handles non-functional requirements in behavior model that helps analysts in non-functional requirements verification and also generate accurate estimation to enhance the success rate of the system.

It resolves different NFRs conflicts to reduce the factor of incorrect estimation and prevent the system from failure in the development phase. The approach is helpful for understanding and extracting the most beneficial non-functional requirements and providing the base for the better model selection.

Ouchani and Debbabi (2015) analyze the state of the art related to security requirement specification of software systems that are modeled upon UML languages the related work is in state of the art for security requirements at different design levels are compared. The approaches used for verification of security requirements include model checking, theorem proving. Authors review different techniques used for specification of security and verification techniques use for security of model-based systems and proposed different research areas on the basis of different comparison of automatic techniques in security requirements. The benefit of propose work is to avoid security experts from lengthy process of writing requirements manually.

Thakurta (2013) proposed a framework which decides that which NFRs should be considered for the project at early stages. The objective of this research is to produce a quantitative framework in order to effectively decide which NFRs to be considered in software development process. The assessment process has high control on the cost of the project, framework identify quality requirements early and helps organizations in reducing the cost of the project. The work differentiates from others by explicitly considering the dependencies among different NFRs in the evaluation process. The final results are expected to be valuable for the business and the project organization point of view by identifying NFRs than implementing the desired NFRs that contribute to the business value in a cost-effective manner. The proposed framework helps in

clarity of business objectives, and also improves the understanding about NFRs and reduces the association among NFRs.

**4. Critical evaluation**

Based on the literature review describe in section 2, we critically analyzed the merits and demerits of the various NFR identification and analysis techniques. We also describe the suggestive improvements for each technique proposed in the reviewed literature.

The critical analysis is described in Table 2. To identify comparison criterion among different extraction and classification techniques for non-functional requirements, we have studied the existing techniques and found characteristics considered to be important with respect to tool/technique used for non-functional requirements extraction, source of document, method use for the classification, data set used, validation of parameters, model validation method and accuracy achieved in that technique. Evaluation criteria for Table 2 is define as follows.

**Table 2: Critical evaluation**

Ref.	Focused Area	Merits	Demerits	Possible Improvement
(Casamayor et al., 2010)	Classification of NFRs	Accurate classification with less effort	Vocabulary issues from different projects	Label informative example than top ranked
(Rahimi et al., 2014)	Extraction of quality requirements	Addition of quality feature improve product result	No generalizability in dataset	Broader set of quality concerns for other domains
(Slankas and Williams, 2013)	Extraction of NFRs from natural language text	Classification of sentence according to NFRs category	No information extraction from images, tables	Performance of classifiers other than health domain
(Ramadhani et al., 2015)	Automated NFRs extraction through semantic factors	Improve accuracy and reduce error rate and hamming loss	Additional search for semantic relatedness reduce recall	-
(Slankas and Williams, 2013)	Extraction of Security requirements	Set of security sub categories to achieve security objectives.	Manual classification of security sentences	Generalizability in other domains to check different security objectives
(Sharma et al., 2014)	NFRs identification through textual pattern	Rule based approach to identify NFRs, classification of text in NFR categories	Reduce the time of classification activity, achieve well categorize requirements	New sub-categories and detecting rules for those sub-categories of NFRs
(Rahman and Ripon, 2014)	Classification of NFRs through UML use-case approach	NFRs categories, elicit NFRs measurement and tracing	Scalability and applicability of the approach not clear	Elicit NFRs from other domain than Point of sale
(Gazi et al., 2015)	Classification of NFRs from Information system	Extracting NFRs in IS through classification scheme	Classification not include quality attribute e. g FURPS	Apply scheme in any IS to make it fast, reliable through star UML
(Mahmoud and Williams, 2016)	NFRs detection through word semantic similarity	Traceability improves the maintainability process	Experimental systems all developed in Java.	Prototype development for developer ease of software practices
(Gazi et al., 2015)	Identification of NFRs in service oriented system	Results of quality model matched with quality requirements	Validation of model in any domain is missing	Comparison of quality model with other technique
(Mahmoud and Niu, 2015)	Semantic IR methods for requirement extraction	Semantic method enhance understanding of requirement	-	Semantic method other than VSM for different semantic scheme
(Meth et al., 2013)	State of the art in automatic NFRs	Classification of proposed tools in state of the art	Completeness not guaranteed	Framework can be used as a start future research point
(Mizouni and Salah, 2010)	Important NFRs extraction for better behavior model	Early NFRs improve the design phase of the system	Relation among NFRs goals is missing	Better correlation between NFRs improve estimation
(Ouchani and Debbabi, 2015)	Security requirements in model-based systems	Model checker technique, for verification in automatic way	Comparison of the techniques not given.	Automatic technique to explore security requirements
(Thakurta, 2013)	Prioritizing NFRs in a software	Prioritization algorithm identify set of NFRs for a project	Non-optimal decision on mutual NFRs because of respondent views	Gap analysis between business and project organizations

**4.1. NFRs extraction tool/technique**

Extraction of NFRs from the document is done through different extraction techniques. The frequently used technique is semantic word similarity technique and textual pattern identification from a text document.

Semantic relatedness between two words on the basis of their similarity relationship is proposed in the literature work. Use case model based techniques are proposed on the basis of questioner data collected from the stakeholders. The non-

functional requirements extraction technique is given in our Table 3.

**4.2. Model validation**

NFRs extraction technique and classification technique are validated on the basis of experimental/empirical results, some techniques result are validated on the base of case studies to. Experimental results are used most frequently to validate the results of proposed techniques

### 4.3. Source document

Different source document are taken with includes software text documents, user manuals, software requirement specification document. Similar types of documents are used in different projects e.g. software text document, software requirement document.

### 4.4. Classifier/clustering method

NFRs classification is done through different classifiers and different classification clustering methods in different techniques. Naïve Base, k-NN classifiers are frequently used classifiers in majority of the techniques. Hierarchal and partition clustering techniques are used in different proposed methods

### 4.5. Validation parameters

Different proposed methods are validated on the base of parameters highlighted in our work. Important parameters validated in many techniques are accuracy, performance and quality of the product

### 4.6. Dataset/test bed

Dataset are used to validate the study. Most of the techniques used healthcare domain. PROMISE

repository for software engineering which provides a collection of publically available tools used as a dataset.

### 4.7. Accuracy achieved

Different techniques and classification achieve different accuracy results. Some of the techniques achieve higher accuracy in predicting and identifying the accurate NFRs.

In semantic FSKNN technique achieved higher accuracy as compared to FSKNN technique. Identification of NFRs through semi-supervised learning achieves the highest accuracy of 75%.

### 4.8. No. of requirements

Different no. of functional and non-functional requirements considered in all datasets

### 4.9. Tables, figures and photographs

Table 1 describes the main categories and sub-categories of NFRs. In Table 2, we critically reviewed different techniques used for extraction of NFRs. In Table 3, we evaluate different requirement extraction and classification techniques on the basis of the above mentioned attributes.

**Table 3: Key Parameters Identified for Non-Functional Requirements analysis**

Ref.	Tool/ Technique for NFR extraction	Model Validation	Classifier/ Clustering methods	Source Document	Metrics Used	Validation/ Parameters	Accuracy achieved	Dataset/ Test Bed	No. of Requirements
(Casamayor et al., 2010)	Hierarchal clustering	Experimental	k-NN ,Naive Bayes	Text document	Precision, recall, F-measure	Quality metrics evolution metrics	75%	PROMISE	370 NFRs 255 FRs
(Rahimi et al., 2014)	Principal component analysis	Experimental	Incremental Diffusive clustering	SRS	Precision, Recall	Precision, Recall	Yes	CCHIT World Vista	235(CCHIT) 1736 (World Vista)
(Slankas and Williams, 2013)	NFR-Locator	Experimental	k-NN, Naive Bayes	System Related Documents	Distance metric Word similarity metric LSA word*document metric	Performance	k-NN=F1 measure(0.54) naive Bayes=F1 measure(0.32)	1.Health care domain 2.PROMISE	14 categories of NFRs
(Ramadhani et al., 2015)	FSKNN	Experimental	Hirst and S-Onge (HSO)	SRS, System documents Requirement document ,Feature Request; User stories	Accuracy Precision Recall Hamming loss	Accuracy Hamming loss	Accuracy=43.7% Hamming loss=21.9% FSKNN=41.4%(accuracy)	1.Itrust 2.CCHIT 3.World Vista US veteran health care health care 4.Online Project Marking System SRS 5.Archiving Facility SRS 6.Mars Express Processing	Requirement sentences(1342) Taken as training data=805 Taken as test data=537
(Riaz et al., 2014)	Security discoverer	Study oracle (set of procedures)	k-NN;SMO	Requirement document ,Feature Request; User stories	Distance metric	Accuracy	Prediction=82% Identification =79	Electronic Healthcare(HER)	Requirement sentences=10,963
(Sharma et al., 2014)	Domain specific language(textual pattern identification technique); NFR detection prototype	Experimental(case study)	Rule based	Requirement statement	Precision Recall	-	PERFORMAN CE=91%	PROMISE	Requirement statements=625
(Rahman and Ripon, 2014)	Use Case Diagram	Case study	Questionnaire approach	Software requirement document	-	Accuracy Cost effectiveness	Yes	Point of Sale	-
(Gazi et al., 2015)	Elicitation technique	Case Study	NFRs for IS, real time	Software requirement document	-	Accuracy (NFRs)	Yes	Information system	-

			system, web based system	nt document	identification )				
(Mahmoud and Williams, 2016)	NLP (word semantic similarity method clustering method; traceability method)	Experimental	Hierarchal clustering Partition clustering	SRS	Distance metric Word similarity metric LSA word*document metric	Accuracy	-	1.SmartTrip 2.SafeDrink 3.BlueWallet	-
(Sadiq et al., 2011)	Quantification technique	Case study	Quality requirements	Service oriented systems requirement document and SLAs Requirement document, source code	Quality metrics, Evaluation metrics	Quality	-	Service oriented systems Service oriented systems	-
(Mahmoud and Niu, 2015)	Semantic methods Information Retrieval – methods	Experimental	Centroid base classifier	Degree of automation/ tool categories Non-functional requirements	Software requirement document	Precision Recall	Performance	VSM-T-TD IR method	iTrust, 2.eTour, 3.CM-1 iTrust=314req eTour=394req CM-1=235req
(Meth et al., 2013)	Comparison-based analysis	Case study	Degree of automation/ tool categories Non-functional requirements	Use case document	Metric for NFRs attributes	Estimated NFRs	-	Assessment of classification techniques used for NFR analysis	NFRs
(Mizouni and Salah, 2010)	Behavior model composition technique	Case Study	Proposed techniques classification based on tool used/scope of technique	Observed models and techniques	-	Reduce cost of security for software system	-	UML model based software security requirements techniques and tools.	Security requirements.
(Ouchani and Debbabi, 2015)	Security specification techniques/Security validation techniques	Case study	NFRs prioritization algorithm	Unstructured text documents	NFRs assessment matrix	Accuracy of quality requirements Cost effectiveness	Yes	software project(Exam assessment)	NFRs

Requirement document, SRS, Text Document

In Table 4, we identify different NFRs that have been extracted from literature work. These requirements are listed in the table. The symbol 'Y'

indicates those NFRs that are having been identified in the reviewed study; while blank box represents that the requirement is not identified by the study.

**Table 4:** Identified Non-Functional Requirements

NFRs	(Casamayor et al., 2010)	(Rahimi et al 2014)	(Williams, 2013)	(Santkas and Williams, 2015)	(Ramadhani et al 2015)	(Riaz et al., 2014)	(Sharma et al., 2014)	(Rahman and Ripon, 2014)	(Gazi et al., 2015)	(Williams, 2016)	(Mahmoud and Niu, 2015)	(Sadiq et al., 2011)	(Meth et al., 2013)	(Mizouni and Salah, 2010)	(Ouchani and Debbabi, 2015)	(Thakurta, 2013)
Availability		x			x	x		x	x				x	x		
Maintainability			X													
Performance	x	x	X	x		x	x	x	x		x		x			x
Scalability			x								x					
Security	x	x	x		x	x	x	x	x					x	x	
Usability			x			x	x									
Portability										x				x		
Recoverability			x													x
Privacy			x			x	x	x	x						X	
Access Control		x	x						x						x	
Accessibility					X					x					x	
Accuracy									x	x						x
Reliability									x	x						
Legal						x	x			x						
Flexibility								x								
Modifiability												x				

**5. Conclusion**

In this study, we critically analyzed various techniques used to automatically extract NFRs from the requirement documents. The purpose of this study is to reduce time and cost required for identifying NFRs for a software project. Automatic extraction of NFRs helps the analyst to identify different NFRs. It reduces the effort of manual

identification. As a future work we intend to propose a machine learning based model for automatic extraction of NFRs from a document and classifies them as per there predefined sub-categories.

**References**

Casamayor A, Godoy D, and Campo M (2010). Identification of non-functional requirements in textual specifications: A semi-

- supervised learning approach. *Information and Software Technology*, 52(4): 436-445.
- Cleland-Huang J, Settini R, Zou X, and Solc P (2006). The detection and classification of non-functional requirements with application to early aspects. In the 14<sup>th</sup> IEEE International Conference on Requirements Engineering. IEEE, Saint Paul, USA: 39-48. <https://doi.org/10.1109/RE.2006.65>
- Gazi Y, Umar MS, and Sadiq M (2015). Classification of NFRs for information system. *International Journal of Computer Applications*, 115(22): 19-22.
- Kitchenham BA and Charters S (2007). Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report EBSE-2007-01.2007, Keele University and University of Durham.
- Li Y, Guzman E, Tsiamoura K, Schneider F, and Bruegge B (2015). Automated requirements extraction for scientific software. *Procedia Computer Science*, 51: 582-591.
- Mahmoud A and Niu N (2015). On the role of semantics in automated requirements tracing. *Requirements Engineering*, 20(3): 281-300.
- Mahmoud A and Williams G (2016). Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, 21(3): 357-381.
- Meth H, Brhel M, and Maedche A (2013). The state of the art in automated requirements elicitation. *Information and Software Technology*, 55(10): 1695-1709.
- Mizouni R and Salah A (2010). Towards a framework for estimating system NFRs on behavioral models. *Knowledge-Based Systems*, 23(7): 721-731.
- Odeh Y and Odeh M (2011). A new classification of non-functional requirements for service-oriented software engineering. *Naif Arab University for Security Sciences (NAUSS)*: 1-7. Available online at: [http://nauss.edu.sa/En/DigitalLibrary/Researches/Documents/2011/articles\\_2011\\_3149.pdf](http://nauss.edu.sa/En/DigitalLibrary/Researches/Documents/2011/articles_2011_3149.pdf)
- Ouchani S and Debbabi M (2015). Specification, verification, and quantification of security in model-based systems. *Computing*, 97(7): 691-711.
- Rahimi M, Mirakhorli M, and Cleland-Huang J (2014). Automated extraction and visualization of quality concerns from requirements specifications. In the IEEE 22<sup>nd</sup> International Requirements Engineering Conference (RE), IEEE, Karlskrona, Sweden: 253-262. <https://doi.org/10.1109/RE.2014.6912267>
- Rahman M and Ripon S (2014). Elicitation and modeling non-functional requirements-a POS case study. *International Journal of Future Computer and Communication*, 2(5): 485-489.
- Ramadhani DA, Rochimah S, and Yuhana UL (2015). Classification of non-functional requirements using semantic-FSKNN based ISO/IEC 9126. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 13(4): 1456-1465.
- Riaz M, King J, Slankas J, and Williams L (2014). Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. In the IEEE 22<sup>nd</sup> International Requirements Engineering Conference, IEEE: 183-192. <https://doi.org/10.1109/RE.2014.6912260>
- Sadiq J, Mohsin A, and Arif F (2011). Quantifying non-functional requirements in service oriented development. In the 9<sup>th</sup> International Conference Frontiers of Information Technology (FIT), IEEE, Islamabad, Pakistan: 224-229. <https://doi.org/10.1109/FIT.2011.48>
- Sharma VS, Ramnani RR, and Sengupta S (2014). A framework for identifying and analyzing non-functional requirements from text. In the 4<sup>th</sup> International Workshop Conference on Twin Peaks of Requirements and Architecture, ACM, Hyderabad, India: 1-8. <https://doi.org/10.1145/2593861.2593862>
- Slankas J and Williams L (2013). Automated extraction of non-functional requirements in available documentation. In the 1<sup>st</sup> International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE), IEEE: 9-16. <https://doi.org/10.1109/NaturaLiSE.2013.6611715>
- Thakurta R (2013). A framework for prioritization of quality requirements for inclusion in a software project. *Software Quality Journal*, 21(4): 573-597.
- Umar M and Khan NA (2011). Analyzing non-functional requirements (nfrs) for software development. In the IEEE 2<sup>nd</sup> International Conference on Software Engineering and Service Science, IEEE, Beijing, China: 675-678. <https://doi.org/10.1109/ICSESS.2011.5982328>