AGILE DEVELOPMENT IN CLOUD COMPUTING FOR ELICITING NON-FUNCTIONAL REQUIREMENTS

MUHAMMAD YOUNAS

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

MARCH 2019

DEDICATION

This thesis is dedicated to my beloved grandfather (late) for her love, concern and support to make sure I achieve higher targets.

This thesis is dedicated to my father Rana Muhammad Latif, and their words of encouragement, prays and support for completion of the PhD

A special feeling of gratitude to my loving Wife for her support, encouragement, prayers and dua throughout the process and give me time to complete PhD. I will always appreciate all they have done.

I also dedicate this to my kids Abd-ur-Raqib, Hayyab Younas, Muhammad Ismaeel, Abdul Basit and Fatima**,** *who* have a suffer a lot due living away from me.

No words can express my gratitude for my beloved brothers, sisters and my loving Khala. They have always been in the forefront of love, concern, dua, advice and guidance to make sure I achieve the best of both worlds.

ACKNOWLEDGEMENT

All praise and glory to the Almighty ALLAH (Subhanahu Wa Ta'ala) who gave me the courage and patience to carry out this work. Peace and blessings of ALLAH be upon His last Prophet MUHAMMAD (Sallulaho-Alaihe-Wassalam).

I would like to express heartfelt gratitude to my supervisor, **Assoc. Prof. Dr. Dayang Norhayati Abang Jawawi** for his constant support during my study at UTM. He listened to my problem and helped me in solving them. He pushes me to work freely on this thesis. His willingness to motivate me contributed tremendously to completion of this work. I have learned a lot from him. I am also very thankful to Universiti Teknologi Malaysia (UTM) I have learned a lot, theoretically, practically, and improving the research capability since I entered the university. I am also thankful to all the staff at UTM for their support and help.

I am also grateful for the support of Prof. **Dr. Imran Ghani** for his guidance and help. I would also like to thank the authority of Universiti Teknologi Malaysia (UTM) for providing me with an excellent environment and facilities such as Computer laboratory to complete this project with software which I need during the process.

Many thanks to my friends and colleagues especially, Syed Hamid Hussain Madni, Muhammad Murad Khan, Ahmad Mustafa, Syed Arif Shah, Sami Ullah, Dr. Shahid Kamal, Dr. Rafaqut Kazmi and Dr. Ali Shahid for all their inputs during the numerous discussions. Also, Hafiz Faizan Akhtar, and Hafiz Faisal who have been a source of moral support through their fervent prayers. I am lucky and glad to have them in my life. Finally, I would like to thank all the persons who are not mentioned here but have been a great help during my period of study.

ABSTRACT

Agile is a popular and growing software development methodology. In the agile methodology, requirements are refined based on collaborations with customers and team members. However, the agile process faces a lack of visibility across the development and delivery processes, has complex and disjointed development processes and lacks communication agility between disconnected owners, development teams, and users. Furthermore, Non-Functional Requirements (NFR) are ignored due to the nature of agile development that lacks knowledge of the user and developer about NFR. In addition, extraction of the NFR is difficult and this difficulty is increased because the agile methodology promotes change in requirement at any stage of the development. Cloud computing services have helped solve some of the issues in the agile process. However, to address the issues in agile development, this research developed a framework for Agile Development in Cloud Computing (ADCC) that uses the facilitation of cloud computing to solve the above-mentioned issues. An Automated NFR eXtraction (ANFRX) method was developed to extract NFR from the software requirement documents and interview notes wrote during requirement gathering. The ANFRX method exploited the semantic knowledge of words in the requirement to classify and extract the NFR. Furthermore, an NFR Elicitation (NFRElicit) approach was developed to help users and development teams in elicitation of NFR in cloud computing. NFRElicit approach used components such as an organization's projects history, ANFRX method, software quality standards, and templates. The ADCC framework was evaluated by conducting a case study and industrial survey. The results of the case study showed that the use of ADCC framework facilitated the agile development process. In addition, the industrial survey results revealed that the ADCC framework had a positive significant impact on communication, development infrastructure provision, scalability, transparency and requirement engineering activities in agile development. The ANFRX method was evaluated by applying it on PROMISE-NFR dataset. ANFRX method improved 40% and 26% in terms of f-measure from the Cleland and Slankas studies, respectively. The NFRElicit approach was applied to eProcurement dataset and evaluated in terms of more "Successful", less "Partial Success" and "Failure" to identify NFR in requirement sentences. The NFRElicit approach improved 11.36% and 2.27% in terms of increase in "Successful" NFR, decrease of 5.68% and 1.14% in terms of "Partial success" and decrease of 5.68% and 1.13% in terms of "Failure" from the Non-functional requirement, Elicitation, Reasoning and Validation (NERV) and Capturing, Eliciting and Predicting (CEP) methodologies, respectively. The findings have shown the process was able to elicit and extract NFR for agile development in cloud computing.

ABSTRAK

Agil adalah metodologi pembangunan perisian yang popular dan berkembang. Dalam metodologi agil, keperluan disempurnakan berdasarkan kerjasama dengan pelanggan dan ahli pasukan. Walau bagaimanapun, proses agil menghadapi kekurangan dari segi kejelasan proses pembangunan dan penghantaran, proses pembangunan yang kompleks dan terputus, dan kekurangan komunikasi agil antara pemilik, pasukan pembangunan, dan pengguna. Seterusnya, Keperluan Bukan Fungsi (NFR) diabaikan kerana sifat semula jadi pembangunan agil yang kurang pengetahuan pengguna dan pemaju berkaitan NFR. Di samping itu, pengekstrakan NFR adalah tugas yang sukar dan kesukaran ini meningkat kerana metodologi agil menggalakkan perubahan keperluan di mana-mana peringkat pembangunan. Perkhidmatan pengkomputeran awan telah membantu untuk menyelesaikan beberapa isu dalam proses agil. Walau bagaimanapun untuk menangani isu-isu dalam pembangunan agil, penyelidikan ini membangunkan rangka kerja Pembangunan Agil dalam Pengkomputeran Awan (ADCC) yang menggunakan pemudahan pengkomputeran awan untuk menyelesaikan isu-isu yang disebutkan di atas. Kaedah Automasi pengekstrakan NFR (ANFRX) telah dibangunkan untuk mengektrak NFR dari dokumen keperluan perisian dan nota temu bual yang ditulis semasa pengumpulan data. Kaedah ANFRX memanfaatkan pengetahuan semantik perkataan dalam keperluan untuk mengklasifikasi dan mengektrak NFR. Seterusnya, pendekatan elisitasi NFR (NFRElicit) telah dibangunkan untuk membantu pengguna dan pasukan pembangunan dalam elisitasi NFR dalam pengkomputeran awan. Pendekatan NFRElicit menggunakan komponen seperti sejarah projek organisasi, kaedah ANFRX, piawaian kualiti perisian, dan templat. Rangka kerja ADCC dinilai dengan menjalankan kajian kes dan tinjauan industri. Hasil kajian kes menunjukkan penggunaan kerangka ADCC memudahkan proses pembangunan agil. Di samping itu, hasil tinjauan industri menunjukkan bahawa rangka kerja ADCC mempunyai kesan positif terhadap komunikasi, penyediaan infrastruktur pembangunan, penskalaan, ketelusan dan aktiviti keperluan kejuruteraan dalam pembangunan Agil. Kaedah ANFRX dinilai dengan menggunakan set data PROMISE NFR. Kaedah ANFRX meningkat 40% dan 26% dari segi ukuran-f masing-masing dari kajian Cleland dan Slankas. Pendekatan NFRElicit diterapkan pada set data eProcurement dan dinilai dari segi lebih "Berjaya" dan kurang "Separa Berjaya" dan "Gagal" untuk mengenal pasti NFR dalam ayat keperluan. Pendekatan NFRElicit meningkatkan 11.36% dan 2.27% dari segi " Berjaya" NFR dan penurunan sebanyak 5.68% dan 1.14% dari segi "Separa Berjaya" dan penurunan sebanyak 5.68% dan 1.13% dari segi "Gagal" masing-masing dari kaedah NERV dan CEP. Dapatan menunjukkan proses itu dapat membantu untuk mendapatkan dan mengektrak NFR untuk pembangunan agil dalam pengkomputeran awan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADCC	-	Agile Development using Cloud Computing
AGSD	-	Agile Global Software Development
ANFRX	-	Automated Non-Functional Requirement Extraction
CC	-	Cloud Computing
CD	-	Continuous Delivery
DSDM	-	Dynamic Systems Development Method
EHR	-	Electronic Health Record
FDD	-	Feature Driven Development
IaaS	-	Infrastructure as a Service
IEEE	-	Institute of Electrical and Electronics Engineers
ISO	-	International Standards Organization
ISP	-	Input Space Profile
MSE	-	Mean Square Error
MyRen	-	Malaysia Research and education network
NFR	-	Non-Functional Requirement
NFRElicit	-	NFR elicitation
NL	-	Natural Language
PaaS	-	Platform as a Service
POS	-	Part of Speech
SaaS	-	Software as a Service
SDLC	-	Software Development Life Cycle
SLR	-	Systematic Literature Review
SRS	-	Software Requirement Specification
RE	-	Requirement Engineering
RAD	-	Rapid Application Development
XP	-	eXtreme Programming

LIST OF SYMBOLS

α	-	cost of resource i per unit time
ʎ	-	energy is consumed by the resource i
fr(dQ, t):	-	finishing time of task
PQ(t):	-	starting time of task
SQ	-	submission time of task

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

This research is about the extraction and elicitation of the non-functional requirement for Agile Development in Cloud Computing (ADCC). The goal of this research is to enhance the agile software development through employing cloud computing platform, and NFR elicitation and extraction. The enhancement in NFR extraction and elicitation ultimately enhance the agile development methodology. Therefore, this thesis has three components such as agile development methodology, cloud computing and non-functional requirements. There are separate studies which incorporate agile software development methodology, cloud computing environment and eliciting and extracting non-functional requirement. In addition, there are studies which have more than one combination, however; there is a lack of studies that cover all these components together in one study. The thesis has focused on the environment for Requirement Engineering (RE) activities especially NFR extraction and elicitation in ADCC.

Agile software development methods are based on frequent delivery of software, improved customer satisfaction, closed interaction with the clients and accommodation of requirement change at any stage of development. However, these agile guidelines are not practiced completely due to rapid change in market demand and while teams are working in the distributed environment. Several challenges exist such as scalability, transparency (Tuli *et al*., 2014), face-to-face communication (Qureshi and Sayid, 2015), availability of experts (Nazir *et al*., 2013), smooth control of development, ability to build applications from distributed locations (Wang, 2011) and resource management (Dumbre *et al*., 2011; Tuli *et al*., 2014). The changing

demands require an environment to develop and test innovative ideas. The provision of resources for testing innovative ideas increases the development cost.

In order to deal with these challenges, cloud computing provides an environment to quickly test new ideas in the marketplace (Rathod and Surve, 2015). Cloud Computing has the potential to reduce the cost of agile development through data sharing, distributed application, prioritizing tasks and by providing infrastructure (hardware and software). Cloud computing enhances the development process by eliminating the need for installations procedures, software patches, and re-installation. Cloud services provide storage and computing resource based on pay per use (Franken *et al*., 2015; Peter Mell, 2011). Cloud computing extends the existing agile process through fast delivery, lowering cost and increasing software quality (Tuli *et al*., 2014; Wei-Tek *et al*., 2014).

Cloud computing comprises five resources, three service and four deployment models (Peter Mell, 2011). Five resources are networks, servers, storage, applications, and services. The services provided by cloud computing are platform, software and infrastructure as a service (PaaS, SaaS and IaaS) and deployment model are private, public, hybrid and community.

Functional and Non-functional requirements are both important in agile software development (Slankas and Williams, 2013). Functional requirement stated as "*what the system does*" and Non-Functional Requirement deals "*how the system performs or exhibits the behavior*" (Danylenko and Löwe, 2012). NFR also represents the observable and available qualities of the system. The example of available qualities is portability and maintainability whereas the example of observable characteristics is performance, dependability and availability (Ameller *et al.*, 2012).

In agile methods, Functional Requirements (FR) are considered important as compare to Non-Functional Requirements (NFR) (Gregorio, 2012). NFRs are ill-defined or ignored in the early stages of development or overlooked until the later stages of software development. The NFR is ignored due to the user's lack of

knowledge about NFR, lack of detail of the procedures to incorporate NFRs (Silva *et al*., 2016). and the nature of agile methods. A project cost and wasted effort increase if the NFR is detected in the later stages of software development (Silva *et al*., 2016).

1.2 Problem Background

Agile development methodology is based on twelve principles. However, these principles are not fully practiced due to difficulties in distributed agile development. The difficulties in distributed agile development are scalability, transparency (Tuli *et al.,* 2014), face-to-face communication (Qureshi and Sayid, 2015), availability of experts (Nazir *et al.,* 2013), smooth control of development, ability to build applications from distributed locations (Wang, 2011) and resource management (Dumbre *et al.,* 2011; Tuli *et al.,* 2014). For overcoming these difficulties cloud computing performs an important role. There are some studies that proposed solution to solve these difficulties in agile development. However, there is lack of evaluation and validation of approaches proposed by existing studies (Abhishek and Frank, 2014; Akerele *et al.*, 2013; Hasaba and Faraahib, 2014). There need more studies to propose a solution for managing agile development in cloud computing to overcome these difficulties.

The bottleneck in agile software development is the maturity of the requirement that is implicitly improved with the involvement of client (Haig-Smith and Tanner, 2016). The NFR in agile software development is neglected due to the nature of agile software process (Jeon *et al.*, 2011) and also neglected by the client (Gregorio, 2012) and industry practitioners (Domah and Mitropoulos, 2015). The improvement in extraction and elicitation of NFR would enhance the software quality in the agile software development process (Farid, 2012a).

Usually, FRs are treated as primary requirements while NFR is ignored or only treated at design and implementation level. NFRs such as performance,

reliability, scalability, security, and usability are handled later in an ad-hoc way during the later phases of software development (Nguyen, 2009). In software architecture, NFR has a critical role (Silva *et al.*, 2013). Business analyst, software architect, software integrators and customers depend on the identification of NFR in order to implement their jobs (Nuseibeh, 2001; Slankas and Williams, 2013). The mishandling of NFR in late stages of development leads to project failure (Saadatmand *et al.*, 2012). Security, load and usability are so important for software systems, this improper dealing lead to detrimental of the whole system (Yin and Jin, 2012).

The ignorance of NFR in projects leads to an increase in the cost of a software system. United States army's project for intelligence sharing with troops costing $2.7 billion was declared as unworkable, due to performance and usability issues (Hoskinson, 2011). Another system named Electronic Health Record (EHR) was criticized for the lack of usability and eventually was not adopted by the medical community (Bertman *et al.*, 2010). A survey observed the failure rate of more than 60% of the projects were due to ignorance of NFR (Bajpai and Gorthi, 2012). The software development without considering the proper NFR and their implementation are vulnerable to failure (Boehm and In, 1996). The identification of NFR in the early stage of development helps in the selection of technology, allocation of hardware, selection of standards and licensing issues in software development.

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 a need for elicitation approach (Farid, 2012a). In agile methods, NFR is criticized for not having a method for NFR elicitation (Paetsch *et al.*, 2003). Another study (Maiti, 2016) suggests that there would be an NFR gathering technique which uses historical data of projects to predict additional NFR and FR in agile software development. Furthermore, the extraction method in cloud computing environment helps in extraction of NFR by sharing information from agile members located at different locations. Maiti (2016) study further describes that cloud storage can be used to gather historical NFR from other agile team members.

The requirements are expressed in Natural Languages (NL) (Chung and do Prado Leite, 2009b; Luisa *et al.*, 2004). In software requirement documents (SRS), FR and NFR are written in mixed. It is hard for a human analyst to identify NFR from the functional requirements. There are techniques for identification of NFR. However, most of the techniques are not automated or manual. In agile the process is fast and there is always stress of in time delivery (Domah and Mitropoulos, 2015). So, to analyze and implement NFR in software projects timely, the analyst needs some automated system to identify and classify NFRs quickly from the SRS documents (Maiti, 2016; Saadatmand *et al.*, 2012). The process-oriented and non-automated approaches cannot identify NFR quickly to meet the speed of agile methods. Furthermore, the requirement is taken on a user story card in agile methods. The NFR is defined with a high level of abstraction in later stages of development, there need some tools and approach to deal with FR and NFR elicitation and extraction in early stages of development (Saadatmand *et al.*, 2012).

There are several techniques for extraction of NFR (Casamayor *et al.*, 2010; Cleland-Huang *et al.*, 2006; Kaiya *et al.*, 2004; Mahmoud, 2015; Slankas and Williams, 2013). The review of the literature shows that most of the techniques for extraction of NFR are supervised (Cleland-Huang *et al.*, 2006, 2007). However, the supervised learning methods are labor-intensive and have the overhead of training the model (Mahmoud and Williams, 2016). If the training data is available, then train the model or algorithm otherwise training data will be prepared manually. The analyst reads the requirement document and classifies NFR manually. In the case of a large dataset, a lot of effort is required to train the data, to get acceptable results. Manual extraction of NFR from SRS document is a hard job. Another limitation is this the existing extraction approaches has more false positives and do not consider the historical data for additional prediction of NFR (Slankas and Williams, 2013).

NFRs are ignored due to unawareness of user about NFR and nature of agile methods. There are quality standards to describe NFR terminologies and their classifications, but these quality standards do not explain the procedure to incorporate NFR elicitation. There are several NFR elicitation methodologies, processes (Boehm and In, 1996; Maiti and Mitropoulos, 2015), templates (Kopczyńska and Nawrocki,

2014), frameworks (Waleed Helmy, 2012) and elicitation guidelines (Gregorio, 2012; Silva *et al.*, 2016) to incorporate NFR. The majority of these NFR elicitation solutions are for traditional software development. However, some studies (Haig-Smith and Tanner, 2016; Karunakaran, 2013) describes requirement engineering activities for distributed agile software development in cloud computing environment through Team Foundation Server (TFS), Microsoft share point and Pivotal Tracker. However, there is a lack of studies that incorporate NFR elicitation in ADCC.

The key points of the problems discussed above are: more consideration is given to FR as compared to NFR, the ignores leads to budget and time overrun and sometimes even failure of the system (Affleck and Krishna, 2012). The NFR is subjective nature and stated with a high level of abstraction. So, NFR requires quick attention in early stages of development. Furthermore, due to agile nature (accommodates rapid change in requirements), there needs an automated method for NFR extraction. The recent approaches are solutions are not up to mark and has some limitation. The existing studies have less attention on the elicitation process, knowledge base, the role of expert, project history, use of NFR extraction method, use of glossary resources and user involvement. Due to lack of knowledge of user and requirement engineers, there is a need for elicitation approach. There is a lack of elicitation approach for agile methodology. Cloud computing storage helps in elicitation process for distributed agile development environments.

1.3 Problem Statement

The review of the literature in Chapter 2 shows that the non-functional requirement for agile software development in the cloud computing environment has not yet reached the desired satisfaction level of the practitioners and researchers. Although intensive research is gaining attention, there needs more research to extract and elicit NFR for agile software development in the cloud computing environment. There are some studies that provide a solution for agile development in cloud computing. However, these studies have less discussion about RE activities and NFR.

For NFR extraction, there are several studies for extraction of NFR, but most of them are supervised learning based. The supervised learning approaches are labor-intensive and have the overhead of training the model. A lot of manual work must be done to train the model. The approaches other than supervised learning based, have low performance in term of precision-recall. As discussed in the previous section, most of the approaches are for FR elicitation, some of them for NFR elicitation and very few elicitation approaches for agile development. Farid (2012a) proposes a NORMAP methodology for extraction of FR and NFR from a text document. The NORMAP methodology links FR with W_8 story card. Another study (Domah and Mitropoulos, 2015) extends the work of NORMAP methodology and include two more story cards.

A study (Maiti, 2016) proposed tool-based Capturing, Eliciting, Predicting and prioritization (CEP) methodology for elicitation of NFR. The CEP enhances the NFR Locator tool to extract NFR from the text in the images. However, the CEP methodology does not consider the project history. Furthermore, CEP methodology suggests that the elicitation process should be enhanced by extracting information from other agile team members through cloud storage in distributed agile development. However, there needs an environment that focuses on RE activities especially NFR for agile development in cloud computing. More automated extraction methods are required to extract NFR from text documents with high precision and more recall (Mahmoud and Williams, 2016; Slankas and Williams, 2013). The semantic similarity measures and preprocessing methods have an important role in NFR extraction (Farid and Mitropoulos, 2012). There need elicitation methodologies to help user and developers because the quality standards are not enough for requirement elicitation process. The historical data and use of automated extraction tools increase the efficiency of the elicitation process to meet the changing demand of the market. In short, the goal of this thesis is to enhance agile software development. This research focuses on the role of cloud computing to enhance the agile software development methodology and the improvement in NFR elicitation and extraction in ADCC. The main research statement to solve the described research problems is as follows:

How to elicit and extract non-functional requirements for agile software development in cloud computing?

1.4 Research Questions

The main research question is broken down into the following research questions:

i. What is the necessary setup required to establish an environment for agile software development in cloud computing?
ii. How can non-functional requirements be extracted for agile software development in cloud computing?
iii. How to elicit NFR for agile software development in cloud computing environment?

1.5 Research Objective

The research focuses mainly on how to extract and elicit the non-functional requirement for agile development in cloud computing. The objective of the research thesis is organized as follows:

i. To design a framework for agile software development in the cloud computing environment.
ii. To propose an automatic method for extracting non-functional requirement in the requirements document.
iii. To propose an NFR elicitation approach for agile development in a cloud computing environment.

1.6 Significance

Agile development is based on less documentation whereas requirement gathering activities involve requirement document, therefore, the requirement is ignored in agile methods. The primary purpose of this research is to enhance agile software development requirement engineering by eliciting and extracting non-functional requirement. Cloud computing is introduced in agile software development for overall enhancement of agile methods. A framework is proposed to provide an environment for agile development in cloud computing. The framework also describes how to incorporate RE activities in ADCC. The NFR extraction method helps analyst and developers in the early stages of development. Furthermore, the elicitation approach helps both users and requirement engineers in an agile elicitation process. This help in the form of extracting NFR in requirement document and guiding the elicitation process in ADCC.

1.7 Research Scope

The scope of the study is limited to the following aspects:

i. This thesis focuses on agile software development in the cloud computing environment in general and especially for RE activities.
ii. This study finds the impact of the proposed ADCC framework on scalability, transparency, communication and collaboration and RE activities.
iii. This study proposes an NFR extraction method and explores the role of indicator keywords and pre-processing in NFR extraction method
iv. This study proposes NFR elicitation approach and the role of experts, project history, and quality standards on the elicitation approach.
v. Furthermore, the role of automated extraction method in NFR elicitation approach is described.

1.8 Thesis Organization

This thesis is organized as follow: the background and literature review of this research is provided in Chapter 2. It reviews different approaches for agile development in cloud computing. This chapter also reviews the state-of-the-art approaches for extraction and elicitation of the non-functional requirement in agile development. The research methodology followed to conduct this research is presented in Chapter 3. It also introduces case studies and industrial survey that are used in this research. The framework for agile development in cloud computing is presented in Chapter 4. The evaluation of the framework is also presented in this chapter. In Chapter 5, an automated method for NFR extraction is introduced. Chapter 6 presents the approach to elicit NFR for agile software development in cloud computing. In Chapter 7, the evaluation of the extraction method and comparison with existing work is presented. The evaluation of NFR elicitation approach and comparison with existing methodologies is also presented in this chapter. Chapter 8 concludes the thesis whereas highlighting the findings, resolved issues and future work.

REFERENCES

Abad, Z. S. H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., and Schneider, K. (August, 2017). What Works Better? A Study of Classifying Requirements. In Proceeding of the *25th International Requirements Engineering Conference (RE), 2017*, IEEE, 496-501.

Abedi, V., Yeasin, M., and Zand, R. (2014). Empirical Study Using Network of Semantically Related Associations in Bridging the Knowledge Gap. *Journal of Translational Medicine,* 12(1), 324.

Abelló, A., Ferrarons, J., and Romero, O. (October, 2011). Building Cubes with Mapreduce. In Proceeding of the *14th International Workshop on Data Warehousing and OLAP*, ACM, 17-24.

Abhishek, S., and Frank, M. (2014). A Roadmap for Software Engineering for the Cloud: Results of a Systematic Review. *Software Design and Development: Concepts, Methodologies, Tools, and Applications*, Hershey, PA, USA: IGI Global, 1-16.

Affleck, A., and Krishna, A. (June, 2012). Supporting Quantitative Reasoning of Non-Functional Requirements: A Process-Oriented Approach. In Proceeding of the *International Conference on Software and System Process*, 88-92.

Akbari, F., and Sharafi, S. M. (August, 2012). A Review to the Usage of Concepts of Software Architecture in Agile Methods. In Proceeding of the *International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA),* 389-392.

Akerele, O., Ramachandran, M., and Dixon, M. (August, 2013). System Dynamics Modeling of Agile Continuous Delivery Process. In Proceeding of the *Agile Conference (AGILE),* 60-63.

Allen, B. K., Romney, G. W., Dey, P. P., and Romney, M. D. (2015). Collaborative Academic-Government Agile Development of a Cloud Prototype Fire Retardant Drop Log Application for Wildfire Management. *Journal of Research in Innovative Teaching, Publication of National University, 8*(1), 99-115.

Alliance, A. (2001). Agile Manifesto. Retrieved 25 Feb, 2018, from http://www.agilemanifesto.org

Almudarra, F., and Qureshi, B. (June, 2015). Issues in Adopting Agile Development Principles for Mobile Cloud Computing Applications. In Proceeding of the *6th International Conference on Ambient Systems, Networks and Technologies*, 52, 1133-1140.

Ameller, D., Ayala, C., Cabot, J., and Franch, X. (September, 2012). How Do Software Architects Consider Non-Functional Requirements: An Exploratory Study. In Proceeding of the *20th IEEE International Requirements Engineering Conference (RE)*, IEEE, 41-50.

Arbain, A. F. B., Ghani, I., and Kadir, W. M. N. W. (September, 2014). Agile Non Functional Requiremnents (Nfr) Traceability Metamodel. In Proceeding of the *8th Malaysian Software Engineering Conference (MySEC)*, 228-233.

Aveledo, M., and Moreno, A. M. (2008). Responsibilities in the Usability Requirements Elicitation Process. *Journal of Systemics, Cybernetics and Informatics, 6*(6), 54-60.

Bagiwa, I. L., Ghani, I., Younas, M., and Bello, M. (2016). Systematic Literature Review on Cloud Adoption. *International Journal of Internet, Broadcasting and Communication, 8*(2), 1-22.

Bajpai, V., and Gorthi, R. P. (March, 2012). On Non-Functional Requirements: A Survey. In Proceeding of the *IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1-4.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). Experience Factory. *Encyclopedia of software engineering*, John Wiley & Sons, 1, 469-476.

Benfenatki, H., Ferreira Da Silva, C., Benharkat, A.-N., and Ghodous, P. (June, 2014). Cloud-Based Business Applications Development Methodology. In Proceeding of the *23rd International WETICE Conference (WETICE)*, IEEE, 275-280.

Bertman, J., Skolnik, N., and Anderson, J. (2010). Ehrs Get a Failing Grade on Usability. *Internal Medicine News*, 43(11), 50.

Boehm, B.W., Brown, J.R. and Lipow, M. (October, 1976). Quantitative Evaluation of Software Quality. In Proceedings of the *2nd international conference on Software engineering*, IEEE Computer Society Press, 592-605.

Boehm, B. W., Brown, J. R., and Kaspar, H. (1978). Characteristics of Software Quality. North-Holland Pub. Company, 1-169.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., and Khalil, M. (2007). Lessons from Applying the Systematic Literature Review Process within the Software Engineering Domain. *Journal of systems and software, 80*(4), 571-583.

Brill, E. (April, 1992). A Simple Rule-Based Part of Speech Tagger. In Proceeding of the *third conference on Applied Natural Language Processing*, 152-155.

Budiu, R., Royer, C., and Pirolli, P. (May, 2007). Modeling Information Scent: A Comparison of Lsa, Pmi and Glsa Similarity Measures on Common Tests and Corpora. In Proceeding of the *Large scale semantic access to content (text, image, video, and sound)*, 314-332.

Burgess, C., Krishna, A., and Jiang, L. (August, 2009). Towards Optimising Non-Functional Requirements. In Proceeding of the *9th International Conference on Quality Software, QSIC'09*, 269-277.

Caithness, N., and Thurston, M. (2013). A Simple Drop and Compute Model for a Saas Cloud Infrastructure for Advanced Research Computing. Oxford e-Research Centre.

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.

Chen, D., and Manning, C. (October, 2014). A Fast and Accurate Dependency Parser Using Neural Networks. in Proceeding of the *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 740-750.

Chen, P.-I., and Lin, S.-J. (2010). Automatic Keyword Prediction Using Google Similarity Distance. *Expert Systems with Applications, 37*(3), 1928-1938.

Chung, L., and do Prado Leite, J. C. S. (2009). On Non-Functional Requirements in Software Engineering. In *Conceptual Modeling: Foundations and Applications*, Springer, 363-379.

Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (2012). *Non-Functional Requirements in Software Engineering,* Springer Science & Business Media. 5, 153-160.

Cleland-Huang, J., Settimi, R., Zou, X., and Solc, P. (September, 2006). The Detection and Classification of Non-Functional Requirements with Application to Early Aspects. In Proceeding of the *14th IEEE International Conference on Requirements Engineering*, 39-48.

Cleland-Huang, J., Settimi, R., Zou, X., and Solc, P. (2007). Automated Classification of Non-Functional Requirements. *Requirements Engineering, 12*(2), 103-120.

Cocco, L., Mannaro, K., and Concas, G. (September, 2012). A Model for Global Software Development with Cloud Platforms. In Proceeding of the *38th Euromicro Conference on Software Engineering and Advanced Applications*, 446-452.

Cohen, B. (2013). Paas: New Opportunities for Cloud Application Development. *Computer, 46*(9), 0097-0100.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research,* 2493-2537.

Concas, G., Mannaro, K., and Cocco, L. (January, 2013). Managing a Global Software Project under an Agile and Cloud Perspective. In Proceeding of t *the International Conference on Software Engineering Research and Practice (SERP),* 1.

Danylenko, A., and Löwe, W. (June, 2012). Context-Aware Recommender Systems for Non-Functional Requirements. In Proceeding of the *Third International Workshop on Recommendation Systems for Software Engineering*, 80-84.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American society for information science, 41*(6), 391.

Domah, D., and Mitropoulos, F. J. (April, 2015). The Nerv Methodology: A Lightweight Process for Addressing Non-Functional Requirements in Agile Software Development. In Proceeding of the *SoutheastCon 2015*, 1-7.

Dörr, J., Kerkow, D., Von Knethen, A., and Paech, B. (June, 2003). Eliciting Efficiency Requirements with Use Cases. In Proceeding of the *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality*, 37-46

Dumais, S. T. (2004). Latent Semantic Analysis. *Annual Review of Information Science and Technology, 38*(1), 188-230.

Dumbre, A., Senthil, S. P., and Ghag, S. S. (2011). Practising Agile Software Development on the Windows Azure Platform. *Infosys Whitepaper*.

Durrani, U., Richardson, J., Lenarcic, J., and Pita, Z. (February, 2013). Lean Traceability Solution through Slam Model: A Case Study of a Hybrid Delivery

Team in a Hybrid Cloud Computing Environment. In Proceeding of the *22nd Australasian Software Engineering Conference: ASWEC 2013*, 19-26.

Ellis, T. J., and Levy, Y. (June, 2010). A Guide for Novice Researchers: Design and Development Research Methods. In Proceeding of the *Informing Science & IT Education Conference (InSITE)*, 107-118.

Ernst, N. A., and Mylopoulos, J. (June, 2010). On the Perception of Software Quality Requirements During the Project Lifecycle. In Proceeding of the *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 143-157.

Eugene, F. (1987). Taxicab Geometry. *Great Britain: Dover Publications*.

Farid, W. M. (2011). *The Normap Methodology: Non-Functional Requirements Modeling for Agile Processe,* Nova Southeastern University.

Farid, W. M. (December, 2012). The Normap Methodology: Lightweight Engineering of Non-Functional Requirements for Agile Processes. In Proceeding of the *19th Asia-Pacific Software Engineering Conference (APSEC)*, 322-325.

Farid, W. M., and Mitropoulos, F. J. (March, 2012). Normatic: A Visual Tool for Modeling Non-Functional Requirements in Agile Processes. In Proceeding of the *Southeastcon*, IEEE, 1-8.

Farroha, D., and Farroha, B. (October, 2012). An Agile Enabler Framework: Architecting Services in the Clouds. In Proceeding of the *Military Communications Conference (MILCOM) 2012*, 1-6.

Fauzi, N. A. A., Hassan, R., and Shah, Z. A. (February, 2017). Extraction of Non-Functional Requirement in Natural Language Document Using Keyword with Usability Quality Aspect. *International Journal of Software Engineering and Technology, 2*(2), 20-26

Feng, G., Guo, J., Jing, B.-Y., and Hao, L. (2012). A Bayesian Feature Selection Paradigm for Text Classification. *Information Processing & Management, 48*(2), 283-302.

Franch, X., and Carvallo, J. P. (2003). Using Quality Models in Software Package Selection. *IEEE software, 20*(1), 34-41.

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I., and Koussouris, S. (2015). Cloudteams: Bridging the Gap between Developers and Customers During Software Development Processes. *Procedia Computer Science, 68*, 188-195.

Ghosh, M. M. A., and Al Sarraj, W. F. (2016). A Case Study on Academic Services Application Using Agile Methodology for Mobile Cloud Computing. *International Journal of Research in Engineering and Science (IJRES)*, 4(2), 22-30

Gilb, T. (September, 2006). How to Quantify Quality: Finding Scales of Measure. In Proceeding of the *International Conference on Software and Data Technologies*, Springer, Berlin, Heidelberg, 27-36.

Gnaho, C., Semmak, F., and Laleau, R. (May, 2013). An Overview of a Sysml Extension for Goal-Oriented Nfr Modelling. In Proceeding of the *Seventh IEEE International Conference on Research Challenges in Information Science (RCIS)* , 1-2

Goldberg, Y., and Levy, O. (2014). Word2vec Explained: Deriving Mikolov Et Al.'S Negative-Sampling Word-Embedding Method. *arXiv preprint arXiv:1402.3722*.

Gopularam, B. P., Yogeesha, C. B., and Ieee. (December, 2012a). Mechanism for on Demand Tag-Based Software Testing in Virtualized Environments. In Proceeding of the *Fourth International Conference on Advanced Computing (Icoac)*, 1-5

Gopularam, B. P., Yogeesha, C. B., Periasamy, P., and Ieee. (December, 2012b). *Highly Scalable Model for Tests Execution in Cloud Environments*, In Proceeding of the *Advanced Computing and Communications (ADCOM)*, 54-58

Greer, D., and Hamon, Y. (2011). Agile Software Development. *Software: Practice and Experience, 41*(9), 943-944.

Gregorio, D. D. (March, 2012). How the Business Analyst Supports and Encourages Collaboration on Agile Projects. In Proceeding of the *Systems Conference (SysCon), IEEE International*, 1-4.

Gross, D., and Yu, E. (2001). From Non-Functional Requirements to Design through Patterns. *Requirements Engineering, 6*(1), 18-36.

Guha, R., and Al-Dabass, D. (December, 2010). Impact of Web 2.0 and Cloud Computing Platform on Software Engineering. In Proceeding of the *International Symposium on Electronic System Design (ISED)*, 213-218.

Haig-Smith, T., and Tanner, M. (January, 2016). Cloud Computing as an Enabler of Agile Global Software Development. In Proceeding of the *Issues in Informing Science & Information Technology, 13*, 121-144.

Handler, A. (2014). An Empirical Study of Semantic Similarity in Wordnet and Word2vec. *University of New Orleans Theses and Dissertations*, Retrived 12/01/2017, from https://scholarworks.uno.edu/td/1922.

Hasaba, S. A. M., and Faraahib, A. (2014). An Overview of Applying Cloud Computing in Addressing Agile Global Software Development Challenges. *Reef Resources Assessment and Management Technical Paper (RRAMT)*, 40(5), 184-191.

Hashmi, S. I., Clerc, V., Razavian, M., Manteli, C., Tamburri, D. A., Lago, P., et al. (August, 2011). Using the Cloud to Facilitate Global Software Development Challenges. In Proceeding of the *Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW)*, 70-77.

Hofmann, T. (July, 1999). Probabilistic Latent Semantic Analysis. In Proceeding of the *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 289-296.

Hoskinson, C. (2011). Army's Faulty Computer System Hurts Operations. *Politico*.

Hussain, I., Kosseim, L., and Ormandjieva, O. (June, 2008). Using Linguistic Knowledge to Classify Non-Functional Requirements in Srs Documents. In Proceeding of the *International Conference on Application of Natural Language to Information Systems*, 287-298.

ISMAIL, U., QADRI, S., and FAHAD, M. (2015). Requirement Elicitation for Open Source Software by Using Scrum and Feature Driven Development. *International Journal of Natural & Engineering Sciences*, 9(1). 38-43

ISO/IEC. (2008). Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (Square). Retrieved Oct 2016, from https://www.iso.org/obp/ui/fr/#iso:std:iso-iec:25010:ed-1:en

ISO/IEC:9126. (2011). Iso / Iec 9126. 2018, from https://en.wikipedia.org/wiki/ISO/IEC_9126

Jeon, S., Han, M., Lee, E., and Lee, K. (August, 2011). Quality Attribute Driven Agile Development. In Proceeding of the *9th International Conference on Software Engineering Research, Management and Applications (SERA)*, 203-210.

Ju, R., Zhou, P., Li, C. H., and Liu, L. (October, 2015). An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis.

In Proceeding of the *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, 2276-2283.

Jyothi, V. E., and Rao, K. N. (2014). Effective Implementation of Agile Practices–in Collaboration with Cloud Computing. *International Journal of Current Engineering and Technology*, *4*(3), 1690-1693.

Kaiya, H., Osada, A., and Kaijiri, K. (September, 2004). Identifying Stakeholders and Their Preferences About Nfr by Comparing Use Case Diagrams of Several Existing Systems. In Proceeding of the *12th IEEE International Requirements Engineering Conference*, 112-121.

Kalem, S., Donko, D., and Boskovic, D. (May, 2013). Agile Methods for Cloud Computing. In Proceeding of the *36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO)*, 1079-1083.

Karouw, S., and Wowor, H. (October, 2016). Using Cloud Computing for Building Das Tondano Mitigation Disaster Information System Prototype. In Proceeding of the *International Conference on Informatics and Computing (ICIC)*, 257-261.

Karunakaran, S. (2013). Impact of Cloud Adoption on Agile Software Development. In *Software Engineering Frameworks for the Cloud Computing Paradigm*, Springer, 213-234.

Keefe, K., and Dick, M. (January, 2004). Using Extreme Programming in a Capstone Project. In Proceeding of the *Sixth Australasian Conference on Computing Education, 30*, 151-160.

Kopczyńska, S., Maćkowiak, S., and Nawrocki, J. (2011). Structured Meetings for Non-Functional Requirements Elicitation. *Foundations of Computing and Decision Sciences, 36*, 35-56.

Kopczyńska, S., and Nawrocki, J. (August, 2014). Using Non-Functional Requirements Templates for Elicitation: A Case Study. In Proceeding of the *4th International Workshop on Requirements Patterns (RePa)*, IEEE, 47-54.

Kotonya, G., and Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*: Wiley Publishing.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (June, 2015). From Word Embeddings to Document Distances. In Proceeding of the *International Conference on Machine Learning*, 957-966.

Lázaro, M., and Marcos, E. (June, 2005). Research in Software Engineering: Paradigms and Methods. In Proceeding of the *CAiSE Workshops, (2)*, 517-522.

Lepatech. (2016). Agile Software Development Process. Retrieved September 2016, from http://www.leptatech.com/software-development-process/

Lilleberg, J., Zhu, Y., and Zhang, Y. (July, 2015). Support Vector Machines and Word2vec for Text Classification with Semantic Features. In Proceeding of the *14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 136-140.

Liu, Y., Ma, Z., Qiu, R., Chen, H., and Shao, W. (August, 2012). An Approach to Integrating Non-Functional Requirements into Uml Design Models Based on Nfr-Specific Patterns. In Proceeding of the *12th International Conference on,Quality Software (QSIC)*, 132-135.

Lu, M., and Liang, P. (June, 2017). Automatic Classification of Non-Functional Requirements from Augmented App User Reviews. In Proceeding of the *21st International Conference on Evaluation and Assessment in Software Engineering*, 344-353.

Luisa, M., Mariangela, F., and Pierluigi, N. I. (2004). Market Research for Requirements Analysis Using Linguistic Tools. *Requirements Engineering*, *9*(1), 40-56.

Mahmoud, A. (August, 2015). An Information Theoretic Approach for Extracting and Tracing Non-Functional Requirements. In Proceeding of the *23rd International Requirements Engineering Conference (RE)*, IEEE, 36-45.

Mahmoud, A., and Williams, G. (2016). Detecting, Classifying, and Tracing Non-Functional Software Requirements. *Requirements Engineering*, 1-25.

Mairiza, D., Zowghi, D., and Nurmuliani, N. (March, 2010). An Investigation into the Notion of Non-Functional Requirements. In Proceeding of the *Proceedings of the Symposium on Applied Computing, ACM*, 311-317.

Maiti, R. R. (2016). *Capturing, Eliciting, and Prioritizing (Cep) Non-Functional Requirements Metadata During the Early Stages of Agile Software Development.* Nova Southeastern University. Retrived on December, 2017 from https://nsuworks.nova.edu/gscis_etd/968.

Maiti, R. R., and Mitropoulos, F. J. (April, 2015). Capturing, Eliciting, Predicting and Prioritizing (Cepp) Non-Functional Requirements Metadata During the Early Stages of Agile Software Development. In Proceeding of the *SoutheastCon 2015*, 1-8.

Manuja, M., and Manisha. (February, 2014). Moving Agile Based Projects on Cloud. In Proceeding of the *International Advance Computing Conference (IACC)*, IEEE, 1392-1397.

Mazarura, J., and de Waal, A. (November, 2016). A Comparison of the Performance of Latent Dirichlet Allocation and the Dirichlet Multinomial Mixture Model on Short Text. In Proceeding of the *International Conference on Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 1-6.

Miglierina, M. (September, 2014). Application Deployment and Management in the Cloud. In Proceeding of the *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 422-428.

Miguel, J. P., Mauricio, D., and Rodríguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Products. *arXiv preprint arXiv:1412.2977*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Misra, S. C., and Mondal, A. (2011). Identification of a Company's Suitability for the Adoption of Cloud Computing and Modelling Its Corresponding Return on Investment. *Mathematical and Computer Modelling*, *53*(3), 504-521.

Moher, D., Shamseer, L., Clarke, M., Ghersi, D., Liberati, A., Petticrew, M., et al. (2015). Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols (Prisma-P) 2015 Statement, *Systematic reviews*, *4*(1), 1.

Mwansa, G., and Mnkandla, E. (July, 2014). Migrating Agile Development into the Cloud Computing Environment. In Proceeding of the *7th International Conference on Cloud Computing (CLOUD)*, IEEE, 818-825.

Myren. (2017). Malaysian Research & Education Network. from https://myren.net.my/

Nazir, A., Raana, A., and Khan, M. F. (2013). Cloud Computing Ensembles Agile Development Methodologies for Successful Project Development. *International Journal of Modern Education and Computer Science (IJMECS)*, *5*(11), 28.

Nguyen, Q. L. (May, 2009). Non-Functional Requirements Analysis Modeling for Software Product Lines. In Proceeding of the *ICSE Workshop on Modeling in Software Engineering*, 56-61.

Nikitina, N., and Kajko-Mattsson, M. (May, 2011). Developer-Driven Big-Bang Process Transition from Scrum to Kanban. In Proceeding of the *International Conference on Software and Systems Process*, 159-168.

Normalización, O. I. d. (2011). Iso-Iec 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (Square)-System and Software Quality Models: ISO.

Nuseibeh, B. (2001). Weaving Together Requirements and Architectures. *Computer, 34*(3), 115-119.

Nuseibeh, B., and Easterbrook, S. (May, 2000). Requirements Engineering: A Roadmap. In Proceeding of the *Conference on the Future of Software Engineering*, 35-46.

Paasivaara, M., Behm, B., Lassenius, C., and Hallikainen, M. (August, 2014). Towards Rapid Releases in Large-Scale Xaas Development at Ericsson: A Case Study. In Proceeding of the *9th International Conference on Global Software Engineering (ICGSE)*, IEEE, 16-25.

Paetsch, F., Eberlein, A., and Maurer, F. (June, 2003). Requirements Engineering and Agile Software Development. In Proceeding of the *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WET ICE*, 308-313.

Pallant, J. (2013). *Spss Survival Manual*: McGraw-Hill Education (UK).

Palmer, S. R., and Felsing, M. (2001). *A Practical Guide to Feature-Driven Development*: Pearson Education.

Patidar, S., Rane, D., and Jain, P. (December, 2011). Challenges of Software Development on Cloud Platform. In Proceeding of the *World Congress on Information and Communication Technologies (WICT)*, 1009-1013.

Peter Mell, T. G. (2011). The Nist Definition of Cloud Computing. Retrived 20-02-2016, from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Pierdin, M., and Bulder, G. (2012). Dimensions and Types of Non-Functional Requirements Nfr Dimensiones Y Tipos De Requisitos No-Funcionales Rnf. 11-17

Polamuri, S. (2015). Five Most Popular Similarity Measures Implementation in Python. Retrived from http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

Portelli, B. (2010). The Beauty of Agile in the Cloud. Retrieved 09-05-2017, 2017, from https://www.agileconnection.com/article/beauty-agile-cloud

PROMISE. (2010). Promise Software Engineering Repository Data Set. Retrived from https://terapromise.csc.ncsu.edu/!/#repo/view/head/requirements/nfr/nfr.arff.

Python27. (2010). Python 2.7.0 Release. Retrived from https://www.python.org/download/releases/2.7/

Qureshi, M. R. J., and Sayid, I. (2015). Scheme of Global Scrum Management Software. *International Journal of Information Engineering and Electronic Business(IJIEEB), 2*, 1-7.

Raj, G., Yadav, K., and Jaiswal, A. (February, 2015). Emphasis on Testing Assimilation Using Cloud Computing for Improvised Agile Scrum Framework. In Proceeding of the *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, 219-225.

Raj, P., Venkatesh, V., and Amirtharajan, R. (2013). Envisioning the Cloud-Induced Transformations in the Software Engineering Discipline. *Software Engineering Frameworks for the Cloud Computing Paradigm*, Springer-Verlag London, 25-53.

Rashwan, A., Ormandjieva, O., and Witte, R. (July, 2013). Ontology-Based Classification of Non-Functional Requirements in Software Specifications: A New Corpus and Svm-Based Classifier. In Proceeding of the *37th Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, 381-386.

Rathod, N., and Surve, A. (January, 2015). Test Orchestration a Framework for Continuous Integration and Continuous Deployment. In Proceeding of the *International Conference on Pervasive Computing (ICPC)*, 1-5.

Rivero, J. M., Heil, S., Grigera, J., Gaedke, M., and Rossi, G. (2013). Mockapi: An Agile Approach Supporting Api-First Web Application Development. *Web Engineering* Springer, 7-21.

Rojo, S. D. V., and Oliveros, R. (2014). Elicitation and Specification Processes of Nfr for Web Applications. In Proceeding of the REFSQ Workshops, 141-142.

Rosenhainer, L. (October, 2004). Identifying Crosscutting Concerns in Requirements Specifications. In Proceeding of the *OOPSLA Early Aspects*, 24-28.

S.A., E. D. (2005a). Functional Requirements for Conducting Electronic Public Procurement under the Eu Framework Volume I Retrieved Oct 08 2016, from http://ec.europa.eu/idabc/servlets/Doc1ad3.pdf?id=22191

S.A., E. D. (2005b). Functional Requirements for Conducting Electronic Public Procurement under the Eu Framework Volume Ii Retrieved Oct 08 2016, from http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/functional-reguirements-vol2_en.pdf

Saadatmand, M., Cicchetti, A., and Sjodin, M. (September, 2012). Toward Model-Based Trade-Off Analysis of Non-Functional Requirements. In Proceeding of the *38th Euromicro Conference on Software Engineering and Advanced Applications*, 142-149.

Saeki, M., and Kaiya, H. (October, 2003). Transformation Based Approach for Weaving Use Case Models in Aspect-Oriented Requirements Analysis. In Proceeding of the *The 4th AOSD Modeling With UML Workshop*.

Saif, H., Fernández, M., He, Y., and Alani, H. (August, 2014). On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In Proceeding of the, *Ninth International Conference on Language Resources and Evaluation,* 810–817.

Shriver, R. (2012). Agile Cloud Development: The Future of Software. Retrieved 20/02/2016, from http://www.virtualizationpractice.com/agile-cloud-development-the-future-of-software-16226/.

Sillitti, A., and Succi, G. (2005). Requirements Engineering for Agile Methods. *Engineering and Managing Software Requirements,* Springer, 309-326.

Silva, A., Pinheiro, P., Albuquerque, A., and Barroso, J. (2016). A Process for Creating the Elicitation Guide of Non-Functional Requirements. *Software Engineering Perspectives and Application in Intelligent Systems,* Springer, 293-302.

Silva, C., and Ribeiro, B. (July, 2003). The Importance of Stop Word Removal on Recall Values in Text Categorization. In Proceeding of the *International Joint Conference on Neural Networks,* (3), 1661-1666.

Silva, F., Lucena, M., and Lucena, L. (July, 2013). Stream-Ap: A Process to Systematize Architectural Patterns Choice Based on Nfr. In Proceeding of the *3rd International Workshop on Twin Peaks of Requirements and Architecture (TwinPeaks)*, 27-34.

Singh, S., and Chana, I. (2013). Introducing Agility in Cloud Based Software Development through Asd. *International Journal of u-and e-Service, Science and Technology, 6*(5), 191-202.

Slankas, J., and Williams, L. (May, 2013). Automated Extraction of Non-Functional Requirements in Available Documentation. In Proceeding of the *1st International Workshop on,Natural Language Analysis in Software Engineering (NaturaLiSE),* 9-16.

Sommerville, I. (2011). *Software Engineering*: Pearson.

Song, X., Duan, Z., and Tian, C. (October, 2010). Non-Functional Requirements Elicitation and Incorporation into Class Diagrams. In Proceeding of the *International Conference on Intelligent Information Processing*, 72-81.

Std, I. (1990). Ieee Standard Glossary of Software Engineering Terminology. from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=159342

Sunner, D., and Bajaj, H. (2016). Classification of Functional and Non-Functional Requirements in Agile by Cluster Neuro-Genetic Approach. *International Journal of Software Engineering and Its Applications, 10*(10), 129-138.

Supakkul, S., and Chung, L. (September, 2010). Visualizing Non-Functional Requirements Patterns. In Proceeding of the *Fifth International Workshop on Requirements Engineering Visualization (REV)*, 25-34.

TextMiner. (2015). Training Word2vec Model on English Wikipedia by Gensim. Retrived from https://textminingonline.com/training-word2vec-model-on-english-wikipedia-by-gensim

Toews, E., Satchwill, B., Rankin, R., Shillington, J., and King, T. (May, 2011). An Internationally Distributed Cloud for Science: The Cloud-Enabled Space Weather Platform. In Proceeding of the *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing*, 1-7.

Too, C. W., Hassan, S. a., Din, J., Ghani, A., and Azim, A. (2013). Towards Improving Nfr Elicitation in Software Development. *International Journal of Information Technology and Computer Science, 7*(1), 33-44.

Tran, Q., and Chung, L. (1999). Nfr-Assistant: Tool Support for Achieving Quality. In Proceeding of the *EEE Symposium on Application-Specific Systems and Software Engineering and Technology ASSET'99*, 284-289.

Tuli, A., Hasteer, N., Sharma, M., and Bansal, A. (2014). Empirical Investigation of Agile Software Development: Cloud Perspective. *SIGSOFT Softw. Eng. Notes, 39*(4), 1-6.

Umar, M., and Khan, N. A. (July, 2011). Analyzing Non-Functional Requirements (Nfrs) for Software Development. In Proceeding of the *2nd International Conference on Software Engineering and Service Science (ICSESS),* IEEE, 675-678.

Uysal, A. K., and Gunal, S. (2014). The Impact of Preprocessing on Text Classification. *Information Processing & Management, 50*(1), 104-112.

Uzunov, A. V., Falkner, K., and Fernandez, E. B. (June, 2013). Decomposing Distributed Software Architectures for the Determination and Incorporation of Security and Other Non-Functional Requirements. In Proceeding of the *22nd Australian Software Engineering Conference (ASWEC),* 30-39.

versionone. (2015). 7th Annual State of Agile Versionone® Agile Made Easier Development Survey. Retrieved 12/01/2015, from http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf

Waleed Helmy, A. K., Osman Hegazy (2012). An Evaluation Framework for Requirements Elicitation in Agile Methods. In Proceeding of the *International Conference on Software Engineering Advances,* 588-593.

Wang, W. (2011). Reinforcing Agile Software Development in the Cloud. Retrieved 30 Jan 2016, from https://www.open.collab.net/media/pdfs/CollabNet%20Whitepaper_Reinforcing%20Agile%20Dev%20in%20the%20Cloud.pdf?_=d

Wei-Tek, T., Wenjun, W., and Huhns, M. N. (2014). Cloud-Based Software Crowdsourcing. *Internet Computing, IEEE, 18*(3), 78-83.

Wells, j. D. (2013). Xp Programming. 2018, from http://www.extremeprogramming.org/map/project.html

Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2006). Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion. *Requirements Engineering, 11*(1), 102-107.

wikipedia. (2018). Wikipedia:Database Download. from https://en.wikipedia.org/wiki/Wikipedia:Database_download

Wikitovec. (2018). Wiki2vec.

Yan, X., Guo, J., Lan, Y., and Cheng, X. (May, 2013). A Biterm Topic Model for Short Texts. In Proceeding of the *22nd international conference on World Wide Web*, 1445-1456.

Yin, B., and Jin, Z. (May, 2012). Extending the Problem Frames Approach for Capturing Non-Functional Requirements. In Proceeding of the *11th International Conference on Computer and Information Science (ICIS), IEEE/ACIS*, 432-437.

Younas, M., Ghani, I., Jawawi, D. N., and Khan, M. M. (2016). A Framework for Agile Development in Cloud Computing Environment. *Journal of Internet Computing and Services (JICS), 5*, 67-74.

Younas, M., Jawawi, D., Ghani, I., and Kazmi, R. (2017). Non-Functional Requirements Elicitation Guideline for Agile Methods. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9*(3-4), 137-142.

Yung, C., and Lin, Y.-T. (2015). Implementing Toast, a Tool for Agile Software Project Management in Cloud Computing Environments. *Journal of Software (JSW)*, 1310-1318.

Zachman, J. A. (1999). A Framework for Information Systems Architecture. *IBM Syst. J., 38*(2-3), 454-470.

Zarinah, I. I. S. S. S., and Kasirun, M. (2013). Agile-Based Software Product Development Using Cloud Computing Services: Findings from a Case Study. *Science International journal (Lahore)*, 1045-1052.

Zhang, W., Yang, Y., Wang, Q., and Shu, F. (2011). An Empirical Study on Classification of Non-Functional Requirements. In Proceeding of the *The Twenty-Third International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, 190-195.

LIST OF PUBLICATIONS

Journal with Impact Factor

1. **Younas, M**., Jawawi, D. N., Ghani, I., Fries, T., & Kazmi, R. (2018). Agile development in the cloud computing environment: A systematic review. *Information and Software Technology*, 103, 142-158. https://doi.org/10.1016/j.infsof.2018.06.014. **(Q1, IF:2.65)**

Indexed Journal

1. **Younas, M**., Jawawi, D. N., Ghani, I., Khan, M. M., (2016). A Framework for Agile Development in Cloud Computing Environment, *Journal of Internet Computing and Services (JICS).* 17(5), 67-74 http://dx.doi.org/10.7472/jksii.2016.17.5.67 **(Indexed by Web of Science)**

2. **Younas, M.,** Jawawi, D. N. A., Ghani, I., & Kazmi, R. (2017). Non-functional requirements elicitation guideline for agile methods. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* 9(3-4), 137-142.

Non-indexed Journal

1. **Younas**, **M**., Jawawi, D.N.A., Ghani, I., Khan, M.I., Ghani, I., A Survey of Agile Development Methods and Tools in Cloud Environment, *Journal of Software Engineering & Intelligent Systems* 2 (3), 236-242

Non-indexed Conference Proceeding

1. **Younas**, **M**., Ghani, I., Jeong, S.R., Kwon, D.Y., A survey of Cloud-based Agile Tools. IN PROC. *KSII, The 7th International Conference on Internet (ICONI)* 2015, Kuala Lumpur.