

A REVIEW ON REQUIREMENT ELICITATION EFFORT IN SOFTWARE DEVELOPMENT PROCESS

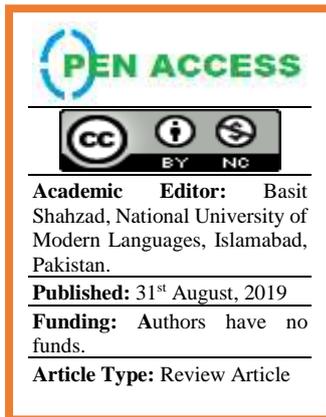
GBEFA PEREMOBOERE MAUREEN, SABRINA AHMAD, OLUYOMI AREMU

*Universiti Teknikal Malaysia Melaka, Melaka, Malaysia.
Twintech International University College of Technology (Twintech)
Persiaran Industri, Bandar Sri Damansara, 52200 Kuala Lumpur, Malaysia.
Email: maureenupm@gmail.com*

ABSTRACT

Requirement gathering is one of the main activities in software management process. In general, requirement elicitation is referred to as the essential process to aid software team members to develop new systems and user functionalities. Many studies have been accompanied for viewing the issues from different aspects. This study investigates the practice of requirement gathering in software management process by adopting literature review approach. This paper presents the results of the reviews on an evident of the current issues related to requirement elicitation, the need for requirement elicitation in software development and existing works aimed at requirement elicitation in software processes. The findings of this review indicate that requirement elicitation requires consideration to seek on numerous limitations and issues.

Keywords: requirement gathering, software management, literature review; software development process; software process;



1. INTRODUCTION

Requirement gathering is one of the activities of software management process, and it is intended to establish what services are required from the system and the constraints on the system's operation and development [1]. Software management guides software managers to create plans for software development. Software management ensure that all of the project activities follow a certain predefined process, the activities are usually organized in distinct phases, and the process the activities are usually organized in distinct phases, and the process specifies what artefacts should be developed and delivered in each phase [2].

Information technology has been involved almost in all aspects of life. Technology is built to alleviate human complex jobs to become much simpler and due to this purpose, most organizations use technology as a facility in doing their jobs and achieving organization-al goals [3, 4]. Due to several research studies, the information system has an accurate data accessibility and efficient run-time [5], high accuracy [6], and to support a proper decision [7], low cost [1], extended accessibility [3], intensify user knowledge [8], increase productivity [9], provide a better data and information [10], and used as data storage [11].

Requirements gathering in software management is a difficult process, mainly due to incomplete requirements and continuously changing requirements are major challenges in requirements gathering that can also be a contributor to software project failures. A major reason for this is that software developers cared less about user perspectives on the new software and rather focused on implementing what the management thought is appropriate. This appears to be a potential concern with open source software development as well, where requirements are often simply asserted by software practitioners [12]. The purpose of requirement elicitation in software management is to ensure that a software practitioner develops a system that satisfies user needs and customer. The needs of requirement elicitation practices in software development aims to keep changing based on the revolution of IT Infrastructure, software system and the business process. Requirement elicitation in software development is based mostly on best practice, knowledge, know-how or experience that has proven to be valuable or effective among software team members [13].

However current requirements elicitation data submitted by software end users and clients are often ignored by software practitioners. This is due to incompleteness and invalidity which in turn lead to problems such as software bugs not being able to be fixed and main system features requests not being executed. The origin of software projects failure appears to include software management domain as well. An example of a requirement elicitation system is Source forge, which is a web-based development environment that hosts software projects,

most of which become inactive soon after registration or within the first year. Improving requirements elicitation process within software development could have many potential benefits [12].

Software practitioners can remain active and have continuous developmental activities if they can get good ideas from users and developers since it can broaden the scope, purpose and functionalities of the software projects. Therefore, for software development process to be successful, software practitioners must communicate with end users. The first step in improving requirements elicitation process in software development process is aimed at ideally improving the requirements elicitation interface used in software project development environments.

Currently used requirements elicitation interfaces in software development environment such as Source forge are generally plain textboxes through which users and stakeholders can submit requirements information as text. As mentioned earlier, the requirements information submitted through such interfaces often suffers from issues such as invalid information and incomplete information [12]. The overall objective of this work is to carry out a review and synthesis of the current research, practices, issues and challenges of requirement gathering in software development based on the prior work done by many researchers. Though, all studies agree on the importance of requirement elicitation in software development process but there is no consensus on the framework, model and processes involved in their implementation.

For this reason, this paper researched existing work on requirement elicitation from 2004 until 2018 including their framework, models, techniques, and challenges. The structure of this paper is organized as follows: section 2 presents the method applied in this study which is literature review. Section 3 describes the findings and results of the study regarding the current approaches and issues related to challenges of requirement elicitation. Section 4 is the discussion of the paper. Finally, the study is concluded in the conclusion and future works section.

2. RESEARCH PROCESS

The research method used in this study is a systematic literature review (SLR) [14]. An SLR is a well-defined approach to identify, evaluate and interpret all relevant studies regarding a particular research question, topic area or phenomenon of interest [15]. The study aims to identify existing requirement elicitation works, requirement elicitation techniques in software development and identifies the issues that influence requirement elicitation in software development.

2.1 SLR Process

The review process to be followed is according to [15]. The study started with starting the research questions, review protocol and result presentation.

2.1.1 Research questions

The review was conducted following the guidelines for performing SLR in SE proposed by [15]. The following research questions are defined to understand the practice of requirement elicitation in software development;

RQ1: What are the techniques involved for requirement elicitation and the existing works in software development?

RQ2: What are issues faced by software practitioners in software development and requirement elicitation?

2.1.2 Review protocol

A review protocol is essential to any review. Driven by the research questions, the protocol defines inclusion/exclusion criteria to select primary studies, a search strategy, search scope, search term and the data items that will be collected to answer the research questions, and finally the approach that will be used for data analysis. In the following sections, we explain in more detail how we have applied the different steps of the protocol.

2.1.3 Inclusion and exclusion criteria

Table 1 below shows the inclusion and exclusive criteria's

Table 1 Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
The study concerns Requirement elicitation, Requirement engineering, Software development.	The patterns are not described in detail, or a structured template is lacking
Were published in, or submitted to, a conference or journal or were technical	A newer study exists that documents the same patterns

reports or book chapters and is published between 2004 and 2016	
The abstract and content are written in English.	The paper concerns a review or evaluation of existing patterns for requirement elicitation.
Reported SLRs or meta-analyses in Requirement elicitation, Software development	

2.1.4 Search scope

The search scope of the review is time period. We specify the time period of published studies for this SLR from Jan. 2004 to April. 2016. Electronic Databases: According to the suggestion by [15]. The following databases are selected as the primary study sources (Table 2).

Table 2 Electronic database included for this review

Database #	Electronic databases
DB1	IEEE Xplore
DB2	ACM Digital library
DB3	ScienceDirect
DB4	EI Compendex
DB5	ISI Web of Science
DB6	SpringerLink
DB7	Wiley InterScience
DB8	Google Scholar

Requirement elicitation, Software development are used as criteria to define the search terms for database search in this SLR based on the SLR guidelines by [15].

Table 5 List of search terms for this review

Search terms	
Requirement	Software requirement, Software gathering, Software development, software practitioners, Review in Software management, Review in requirement elicitation

The search strategy, describing how to combine the search terms, is used. This included an automatic search based on a list of keywords in the electronic databases in Table 2 and then the following Boolean search strings were used; “Requirement elicitation” or “Software management” or “Software process” or “Software development process” or “Software gathering”.

3. RESULTS AND FINDINGS

A total of 12 studies discuss requirement elicitation in software development and the 12 papers were selected for this review. Citations for the papers and other relevant papers are included in the reference for further reading. The identified papers were synthesized and extracted for secondary information based on the review of the papers to answer the research question mention in method of study Section 2 of this paper.

3.1 Techniques involved for requirement elicitation in software development

To answer research question 1, we relate to research carried out by [16] on the techniques for requirements elicitation in software development which includes interview, joint application development (JAD), brainstorming, use cases, and observation. Interviews are a natural way of communication among people. Several researchers and academicians consider interview as an appropriate requirement elicitation tool. Interview involves three steps: the preparation of the interview, the interview itself, then the analysis of the results.

The Joint Application Development (JAD) usually provides good results; however, it takes too much time and too many people. The brainstorming is widely used because it is very easy to implement and ideas are generated easily. Use Cases are traditionally used in Software Engineering especially in Unified Modelling Language (UML) for specifying the required usages of a system. Observations, study of documentation, questionnaires, immersion, are some other common techniques. Other techniques used by software practitioner

include several steps to gather the requirements of a system. First software practitioners collect information regarding the problem domain in the current system or in the current context. Then the systems objectives are identified and described by means of specific require gathering templates. Such system objectives come from the division of the main problem in several sub-problems by following the divide-and-conquer technique. Once the main problem has been divided, functional requirements, non-functional requirements and information requirements are identified in every system objective. Such requirements are also described by way of specific templates whose metadata specify every requirement. Finally, requirements and objectives are organized and prioritized [20].

3.2 Issues faced by software practitioners in software development

To answer research question 2 [17] suggested that the issues and challenges faced by software practitioners in software development are stated as; the absence of face-to-face communication and interaction in OSS, but the organization or stakeholders want to communicate with development team directly. Increase the risks of code quality is second issue because some negative programmer involved in the process generate harmful code that effect the market values of OSS. This makes negative effect on the efficiency of software as well as hardware.

Modularity increases the risks of common coupling that makes it difficult to maintain the software that's why it is impossible to long run the OSS. Documentation, testing, internationalization/localization and field supports are tedious task in OSS. Because non-technical Free and Open Source Software (FOSS) contributor and user may not fill the gap to the extent originally predicted.

Continued support is not provided in FOSS. Interoperability and compatibility problems arise due to many different commercial versions of same Open Source product.

3.3 Issues in requirement elicitation

To answer research question 1, require elicitation practices in software development is faced with many challenges such as;

- a. Firstly, resistance to change is a widely recognized problem and is considered to be one of the biggest challenges in software process improvement. The scope of change needs to be clear so that it can easily be motivated and explained [13].
- b. Second, smaller software industries often lack the requirement elicitation knowledge. It will involve cost if they want to employ a consultant. Thus it takes longer time if they want to done it itself, because the required knowledge. The standard model sometimes not justified, some of the models focus on the overall process, and are not detailed enough helping a company with specific requirements engineering problems [13].
- c. Third, there is a gap between the theoretical application and tailoring of standard frameworks and the ability and acceptance to apply them in real-life projects. The understanding and mapping of such models is difficult and needs a lot of experience.
- d. Fourth, this is a problem, because the expectations on product quality are as high. There is a need for a simple framework that is easy to understand and limits the scope of change to the particular problems. It is necessary to have a small set of practices that can be introduced in a single step. Such a framework has to be easily available to the firms. It is also important that it is easily understandable to all stakeholders, not just to the person responsible for the development process.
- e. Lastly, requirements selection is often done too late. Stakeholder and developer discussed the need of requirement during requirement gathering. The stakeholder suggested the requirement based on the business process while the developer mapping the requirement based on their experienced. This process will take longer time until the both part agreed. Sometimes the developer does the prototype to make sure the stakeholder understand what they want.

3.4 Related works on requirement elicitation

To answer research question 2, requirement elicitation is essential to effectively develop quality software on time and at a reduced cost, thus academicians and researcher are trying to develop a model or framework to assist software practitioners in requirement gathering. In [13] authors researched on the development of requirement engineering practice for software project. They presented the development of requirement engineering practice for software project in Malaysian Public Sector and made a comparison among the requirement engineering practice that was implemented. They finally developed a framework that will act as the guideline for the organization or agency during the requirement elicitation

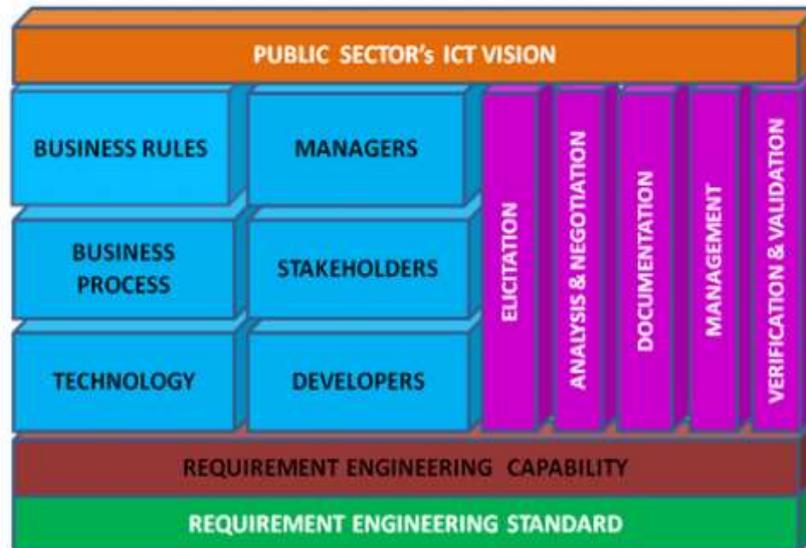


Figure. 1 Proposed requirement engineering framework [13]

Figure 1 show the developed requirement engineering framework [17] on the requirement elicitation for open source software using SCRUM and feature driven development. The researchers discuss the question, whether open source software development is in accordance with agile software development principles and therefore well within the planning spectrum?

A comparative study is discussed about small scale open source projects by implementing Scrum and Feature Driven Development (FDD). They later compare open source software (OSS), based on which methodology is good (SCRUM or FDD) for requirement elicitation of OSS projects and lastly compare which of the two methodologies (SCRUM or FDD) is good for requirements elicitation of OSS projects.

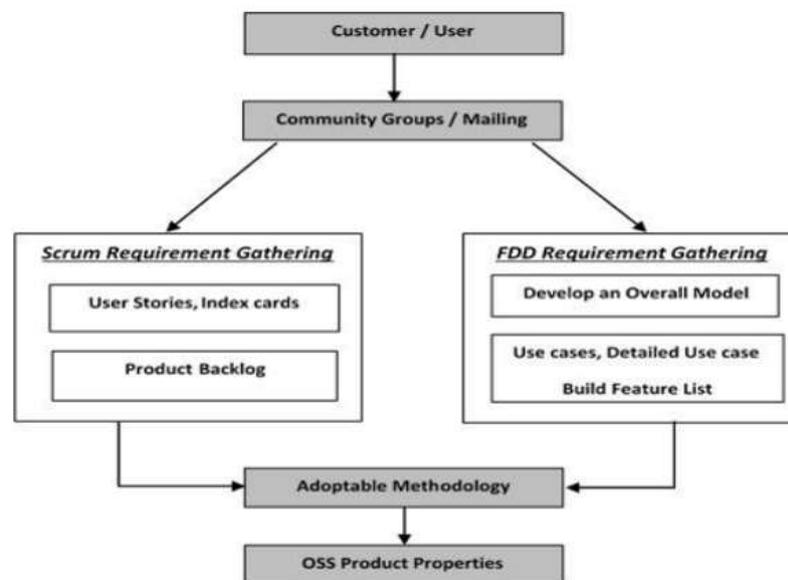


Figure. 2 Proposed requirement elicitation model [18]

Figure 2 show the developed requirement elicitation model aimed to assist organisations in gathering requirements. Other related works includes research by [18] implemented a Simulation Based Decision Support Systems: An Industrial End-user Based Requirement Elicitation Process aimed to capture requirement of an industry. They developed a decision support system to assist software practitioners in eliciting system and user requirements for software applications.

In [19] authors investigated user requirements for mobile educational application Impact of requirements elicitation on software development. The researchers understood the requirement of users and the environment in which they will use this app in the future. They collected data from teachers in three different languages using an

online questionnaire and printed questionnaire. Analysis of the results reveals that most of the teachers have the same preferences for application functional features, user interface and usability requirements. From analyses and literature review, they identified challenges that an application developer can face, studied the implications of requirement gathering on software development and users' expectation of application quality.

In [16] authors developed a requirement elicitation templates for groupware applications. The application is designed to be used by several users through a net of computers such as the Internet. Their approach may be integrated in a process model to identify the roles and tasks needed in the following stages of the development process starting from the new requirements specification.

In [12] authors proposed an enhanced requirements elicitation interface for open source software development environments used in open source software (OSS) development environments. Specifically they suggested embedding currently used interface with reusable requirement patterns and enhancement based on the result we obtained from an experiment on the availability of requirement patterns during requirements generation in OSS development. In [20] authors researched towards intermediate-agile model based on agile through requirement management and development enhancements. They suggested a new software development life cycle model called intermediate-agile based on agile model but with some changes in the life cycle, best practices, documentation templates and enhancements in the activities of requirement elicitation and management to cover these problems.

4. DISCUSSION

Many organizations are interested in improving their requirement elicitation practices and defining requirement elicitation process because of their confidence that requirement elicitation can be the key to developing successful systems. Before that, they have is to identify critical factors affecting to the organization during the software development processes [21]. Convincing organizations to improve their software project development process is even more difficult for smaller firms. This organization has small margins and most likely lack resources competence to make improvements [22]. Therefore, software practitioners have limited contact with end users and face many challenges. If the developed system does not meet the user requirement in terms of interface, features, pricing, app description, or user feedback it is failed. Since end users are from different countries, age group and gender have different behaviours and needs. Many organizations are interested in improving their requirement elicitation practices using selected requirement elicitation practice.

5. CONCLUSION AND FUTURE WORK

Most essential phases of a software management process are requirement elicitation, and for software practitioners end user involvement it is important for the development of useful and usable systems. The requirements elicitation for commonly developed software are derived from strategic business goals or from market opportunities. Software applications are becoming more and more usual every day. They are applications where users achieve common objectives by performing tasks through networks. They collaborate, cooperate, coordinate, and communicate with each other. Users are not considered as individual members but as members of groups which interact among them. Such applications have particular features that if they were taken into account explicitly from the beginning could improve the quality of the final system [1]. Requirements elicitation is the process of in-depth and thorough finding information from all stakeholders associated with the built software. The result of requirements elicitation is the explanation from every stakeholder with natural language related to the will and interest of the built software. This paper reviewed existing literatures on requirement elicitation and identified the practices of requirement elicitation in software development. The paper adopted secondary studies and finding from the paper identified the existing techniques for requirement elicitation, issues faced by software practitioners in software development, issues in requirement elicitation and reviewed related works in this domain. Future work will involve the development of a requirement elicitation model in software development using survey via questionnaire.

ACKNOWLEDGEMENT

We gratefully acknowledge supports by the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, University Twintech Malaysia and to our parents.

REFERENCES

1. Solemon, B., Sahibuddin, S., & Ghani, A. A. . *Requirements Engineering Process : A Review and Research Agenda* in *MySEC '07. The 3rd Malaysian Software Engineering Conference*. 2007.
2. PA, N.C. and B.A. JNR, *A REVIEW ON DECISION MAKING OF RISK MITIGATION FOR SOFTWARE MANAGEMENT*. *Journal of Theoretical & Applied Information Technology*, 2015. 76(3).
3. Slamet, C., et al., *Clustering the Verses of the Holy Qur'an using K-Means Algorithm*. *Asian Journal of Information Technology*, 2016. 15(24): p. 5159-5162.

4. Gerhana, Y., et al. *Implementation of Nearest Neighbor using HSV to Identify Skin Disease*. in *IOP Conference Series: Materials Science and Engineering*. 2018: IOP Publishing.
5. Slamet, C., et al. *Social Media-Based Identifier for Natural Disaster*. in *IOP Conference Series: Materials Science and Engineering*. 2018: IOP Publishing.
6. Slamet, C., et al. *Web scraping and Naïve Bayes classification for job search engine*. in *IOP Conference Series: Materials Science and Engineering*. 2018: IOP Publishing.
7. Bustomi, Y., M.A. Ramdhani, and R. Cahyana, *Rancang Bangun Sistem Informasi Geografis Sebaran Tempat Riset Teknologi Informasi di Kota Garut*. *Jurnal Algoritma*, 2012. 9(01).
8. Maylawati, D.S.a., et al. *Expert system for predicting the early pregnancy with disorders using artificial neural network*. in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. 2017: IEEE.
9. Zulfikar, W., P. Prasetyo, and M. Ramdhani. *Implementation of mamdani fuzzy method in employee promotion system*. in *IOP Conference Series: Materials Science and Engineering*. 2018: IOP Publishing.
10. Maylawati, D.S.a., et al. *Incremental technique with set of frequent word item sets for mining large Indonesian text data*. in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. 2017: IEEE.
11. Taofik, A., et al. *Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition*. in *IOP Conference Series: Materials Science and Engineering*. 2018: IOP Publishing.
12. Kuriakose, J. and J. Parsons. *An enhanced requirements gathering interface for open source software development environments*. in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. 2015: IEEE.
13. Haron, A., et al. *Requirement engineering practice research framework for the public service organisation*. in *2011 Malaysian Conference in Software Engineering*. 2011: IEEE.
14. Ramdhani, A., M.A. Ramdhani, and A.S. Amin, *Writing a Literature Review Research Paper: A step-by-step approach*. *International Journal of Basic and Applied Science*, 2014. 3(1): p. 47-56.
15. Barbara, K. and S. Charters, *Guidelines for performing systematic literature reviews in software engineering*. Keele University, UK, 2007. 9.
16. Penichet, V.M., et al., *Requirement gathering templates for groupware applications*, in *New Trends on Human-Computer Interaction*. 2009, Springer. p. 141-150.
17. ISMAIL, U., S. QADRI, and M. FAHAD, *Requirement Elicitation for Open Source Software By using SCRUM and Feature Driven Development*. *International Journal of Natural and Engineering Sciences (IJNES)* E-ISSN: 2146-0086, 2015. 6(1): p. 38-43.
18. Lanning, I., et al., *Simulation based decision support systems: An industrial end-user based requirement gathering process*. *Procedia CIRP*, 2014. 25: p. 238-245.
19. CHOUDARY, S.A.R., et al., *Investigating user Requirements for Mobile Educational App Impact of Requirement Gathering on Software Development*. 2015.
20. Rekaby, A. and M. Soliman. *Towards Intermediate-Agile Model Based On Agile Through Requirement Management And Development Enhancements*. in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*. 2011: The Steering Committee of The World Congress in Computer Science, Computer
21. Olsson, T., et al., *A flexible and pragmatic requirements engineering framework for SME*. *Proceedings: Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes*, 2005: p. 1-12.
22. Kauppinen, M., et al., *Implementing requirements engineering processes throughout organizations: success factors and challenges*. *Information and Software Technology*, 2004. 46(14): p. 937-953.

AUTHORS PROFILE



Gbefa Peremoboere Maureen is a graduate and studied MSc in Computer Science (Software and Intelligence) at Faculty of Information and Communication Technology, Universiti Teknikal Malaysia. She completed her BSc in Software Engineering (Hons) from the University of East London (UEL) in 2012.



Sabrina Ahmad is currently a Senior Lecturer at the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia.

She completed her PhD in Computer Science from the University of Western Australia, 2012, Master in Science (Computer Science - Real time Software Engineering) from Universiti Teknologi Malaysia, 2002 and Bachelor of Information Technology (Hons) from Universiti Utara Malaysia, 2000. IREB CPRE-FL.



Oluyomi Olugboyega Aremu is currently a DBA Candidate at Twintech University IUCTT Bdr Sri Damansara, Malaysia. He completed his MIT at one of the prestigious universities in Malaysia, Multimedia University, Cyberjaya, and BSc at University of Ilorin, Kwara State, Nigeria.