

Review Article

Effective Requirement Engineering Process Model in Software Engineering

Isonkobong Udousoro

Department of Information Technology, School of Computing and Information Technology, Federal University of Technology, Owerri, Nigeria

Email address:

isonkobongudousoro@gmail.com

To cite this article:Isonkobong Udousoro. Effective Requirement Engineering Process Model in Software Engineering. *Software Engineering*. Vol. 8, No. 1, 2020, pp. 1-5. doi: 10.11648/j.se.20200801.11**Received:** December 16, 2019; **Accepted:** March 27, 2020; **Published:** April 7, 2020

Abstract: Requirement Engineering is regarded as one of the major stages in software development and management. The aim of requirement engineering is to analyse, investigate, document and check the services and also the constraints of the software system that is being developed. The world of requirement management has increasingly developed over the years and has become the cornerstone for any software development to be successful, therefore it is vital for every organization to consider and pay more attention to the requirement engineering if they intend to build a quality software product that will satisfy the users. Different viewpoints, objectives, roles and responsibilities are all incorporated into requirements engineering which makes it a difficult and complex process in software engineering. Many researchers have come up with different optimized approaches to the requirement engineering. This paper reviewed relevant literature from Elsevier, Emerald, IEEE, ProQuest and Google Scholar databases. Requirement engineering processes are further outlined and explained which include requirement elicitation and development, validation and verification etc. In conclusion, the paper recommends effective requirements engineering process to adopt depending on the industry's goal.

Keywords: Requirement Engineering, Requirement Analysis, Requirement Elicitation, Requirement Development, Requirement Management

1. Introduction

Developing a software product requires different processes and stages. Requirement engineering is one of the major phases that involves every stakeholder in the software product development [4]. Requirements in the literal sense can be referred to as anything that is discovered in order to develop products or services. A system's requirement includes services and constraints of the operation [1]. In order to meet customers' needs, a well defined requirement must include statements of system functionality that satisfies user requirement and system requirements [5]. Software system requirements are divided into functional and non-functional requirement. Functional requirements refer to statements of services that product or system is required to provide. These include reaction of the system to different inputs, it also involves what the system is not required to do [25]. Non-functional requirements on the other hand refers to the

constraints on the services offered by the system, these constraints could include time, process and/or standards [3]. Requirements engineering is a very complex and difficult phase in software development because it has the most impact on the results that will be gotten from the product and it involves every stakeholder including the users [9].

Requirement engineering is an approach or process through which a software engineer collects, analyzes and implement requirements from all stakeholder from all stakeholders into the software development processes. Requirements lifecycle of the software development [14].

In a typical software development environment, requirement engineering is only performed at the beginning phase of the project but in large and complex system development, requirement engineering is a continuous and iterative process as changes occur during the development so as to meet customer needs and quality [27]. Requirement engineering contains sets of activities for analysing, documenting, reviewing, maintaining and validating

requirements for a system. Requirement engineering can be further broken down into four groups which are requirement elicitation and development, documentation of requirement, requirement verification and validation and requirement management/planning [28].

This paper reviews an effective model of requirement engineering process for software development. This paper is organized as follows, section 2 describes various requirement engineering models being reviewed based on software development and also highlight the number of journals and their sources being used for this paper. Section 3 introduces and highlights a typical effective model for requirement engineering and its processes. Finally section 4 describes the concluding remarks and future research work.

2. Related Works

In the past years, researchers have focused more on the requirements engineering part of software engineering being an integral part of the discipline.

Requirements traceability is an important part of requirements management which is an integral part of requirements engineering [7]. Requirements traceability enables the developers to be able to trace the software requirements from the beginning to the end of the development. This includes definitions, techniques, challenges and tools using a systematic analysis [29].

Torkar, Gorschek, Feldt, Raja & Kamran in 2009 proposed a model for effective requirements engineering process thereby producing quality requirements for software development. They went on to discuss the requirement engineering as being the most essential stage of software development life cycle. It is important to apply requirements engineering practices to every stage of the life cycle for software development [12]. Some stages in requirements engineering is iterative and therefore must be repeated at every stage. The model showed that requirement engineering and management has a good impact on quality software production.

A proposed model for requirements elicitation, specification, analysis and management of requirements is being linked with software change management process [31]. This is used to help the development team without an adopted process methodology.

Mogk N. W. in 2014 defines engineering design process as measurable objective functions modelled as an improvement issue where goals, results and first choice of users are being incorporated, this is described in their article; Requirement management system based on optimization model of the design process. The main goal of this design process is to present a method of viewing the requirements that enhance communication with the user and what the user wants which combines decision making techniques with the requirements engineering.

Hofman and Lehner in 2001 studied 15 requirements engineering process in various industries and came up with the fact that some participants saw requirements engineering process as an ad hoc process with a few customization of company’s requirement engineering processes or using

predefined requirement engineering processes.

3. Method of Investigation

This study took into consideration a review of two hundred and fifteen journal papers that is published in reputable academic journals ranging from 2001 to 2018. During the research, the keywords that were considered are “Software engineering”, “Requirements engineering” and “Software development” and these papers were reviewed. The search was carried out in databases which include IEEE, Elsevier, ProQuest, and Emerald publishing. Table 1 below shows a table detailing the distribution of the search, number of journals and its percentage distribution.

Table 1. Distribution of reviewed papers.

S/no.	Database	Number of papers	Percentage
1	Google Scholar	81	37%
2	IEEE	32	15%
3	ProQuest	32	15%
4	Elsevier	38	18%
5	Emerald	15	15%

4. Requirements Engineering Process

Requirements engineering is mainly implemented in software development so as to discover quality and important requirements necessary for the development of the software [11]. These requirements must be clear enough, consistent, editable and précised to enable the production of quality software. This paper reviews a proposed model for effective requirement engineering process and it consists of four phases which are requirements elicitation and development, documentation of requirements, validation and verification of requirements and requirements management and allocation as shown in the figure below.

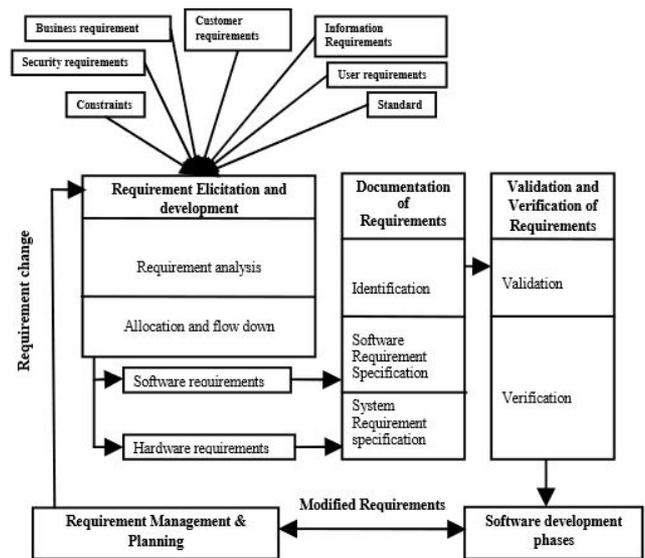


Figure 1. Requirements Engineering Process Model (Pandey, Suman, Ramani 2010).

Each of these phases is being explained in detail in the sections below. This model explains requirements engineering as part of the software development.

4.1. Requirements Elicitation and Development

This phase of requirements engineering focuses on collecting and gathering of data and information from respective stakeholders which include customers, users, constraints, environment, standards etc. It also examines these information and objectives of requirements for the system to be developed. First of all, the stakeholders must be identified with respect to the system and raw requirements are being gathered from various viewpoints. These raw information and data have not been analysed or documented in a well-formed requirement notation [15]. System requirements, business requirements, user requirements, customer requirements, security requirements, standards are some of the viewpoints collected at this phase which could be done either by observing or by interviewing stakeholders. User needs may not be understood which makes it difficult for the system analyst to get the user requirements, therefore, the system analyst will have to collect raw requirements from the users and perform analysis on them where feedback is received before the results is being used to develop a quality software based on the user requirements. A contextual model for requirements development is shown in the figure below.

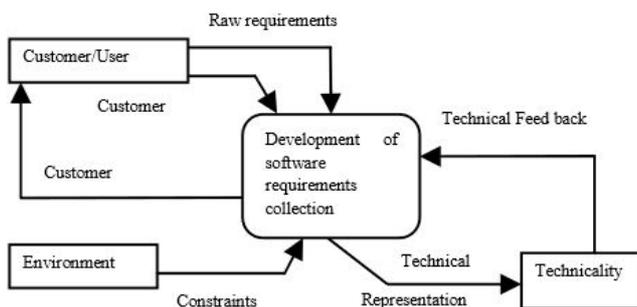


Figure 2. Development of Requirements (Pandey, Suman, Ramani 2010).

Requirements Analysis: The major activity in the development of any quality software is the analysis of collected requirements. Gathering requirements is one but analyzing these requirements is critical which can be conflicting and difficult all in the view of meeting business requirements [18]. Some of the activities in requirements analysis include communication, prioritisation, negotiation etc. These analyzed requirements must be documented so as to aid future improvements and maintenance of the system.

Allocation and Flow-down of Requirements: The top-level requirements and low-level requirements are very different which brings about the allocation and flow-down of requirements. This is to make sure that the system requirements at both top-level and low-level will work together to reach the objectives of the software [19]. The top-level requirement is designed in an hierarchical form so as to manage information in all the levels. Detailing of the requirements tends to increase as design is being carried out

down to the low-level because system requirements at the low-level must be specific. Subsystems must carry the system requirements of the whole system which brings about allocation of requirements in order to design the structure of the system. Low-level requirements may resemble that of the top-level or might differ therefore, the flow-down of requirements consists of requirements meant for the low-level elements in response to the allocation. During this phase, requirements identification, traceability, verification and validation must be considered before moving on to the next phase.

4.2. Documentation of Requirements

After the gathering and collection of requirements for the system, it is being documented to contain a description of the behaviour of the software system. Functional and non-functional requirements are being gathered and documented with the aid and completion of allocation and flow-down of requirement processes [21]. Documentation of requirements include two major stages which are requirement identification and requirement specification.

Requirements Identification: In requirements identification, a unique identifier is being assigned to each requirement which is used to identify a requirement during the process of software development and management. There are three activities carried out during requirements identification which are basic numbering activity, identification activity and support/automation of managed items and requirements.

Requirement Specification: The requirements specification document is being made available after the identification of requirements has been successful. This document does not pay attention to the process of development but focuses on the software product that is delivered software requirement specification is a tool that describes the software and its behaviour including user interaction [27]. This requirement specification also describes the non-functional requirements which talks about the constraints of the design and implementation. Requirement specification describes the performance of the software which also helps in reducing development cost. Operating speed, availability, maintainability, security and recovery are amongst the features the requirements specification evaluates.

4.3. Requirement Verification & Validation

Different group of people are being put in place to confirm and affirm that the requirements from the software requirements specification are stated which is the validation process and also that they are correctly stated which is the verification process [28]. These activities has to do with checking the raw requirements against the system requirements and making sure the documentation is correct. There are various methods used for validating requirement but the ones most commonly used are prototyping and review of requirements with stakeholders. System level requirement need to be verified.

Traceability is important during validation and verification so as to introduce audit trail in checking correctness, consistency and understandability of the requirements.

4.4. Requirement Management and Planning

Requirement management and planning has to do with checking and following the changes in the requirements and dependencies of the document to make sure everything is in order during the software engineering process. Requirement management is an iterative process that must continue from beginning of the software production i.e. from planning stage to completion of software and during maintenance [17]. This happens because the requirements tend to change in the process. Requirements are bound to change and its changes can cause damage hence, the introduction of change management is vital. Requirement change management is the ability to manage the changes in requirements during the software development life cycle.

Requirement management is one of the difficult phases in the development lifecycle because reviewing and adopting a change to requirement will bring about a heavy cost and redevelopment work. Requirement management have tools dedicated to collecting system requirement and access information about the requirements. The development team is being brought together in a secure and concurrent cooperative standard by software requirement management tool. It is important to keep record of changes and create repository for requirements change requests and their documents must comply with standard industrial templates.

5. Conclusion

Requirement engineering is the beginning phase of software development life cycle where requirements such as user requirements and system requirements are collected, understood and analysed for the development of the software product [7]. This paper has reviewed a requirement engineering process model which is effective in the production of quality software. This model is more effective compared to others because of the introduction of requirements management and planning phase whereas other models are being limited to verification, validation, specification and elicitation. Because of the iterative nature of requirements, management and planning phase helps to put things in place for changes. For further research, software requirements specification can be looked into for modification to be implemented in software development.

References

- [1] Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: processes and techniques*. Wiley Publishing.
- [2] Macaulay, L. A. (2012). *Requirements engineering*. Springer Science & Business Media.
- [3] Glinz, M. (2007, October). On non-functional requirements. In *15th IEEE International Requirements Engineering Conference (RE 2007)* (pp. 21-26). IEEE.
- [4] De Lucia, A., & Qusef, A. (2010). Requirements engineering in agile software development. *Journal of emerging technologies in web intelligence*, 2 (3), 212-220.
- [5] de Gea, J. M. C., Nicolás, J., Alemán, J. L. F., Toval, A., Ebert, C., & Vizcaíno, A. (2011). Requirements engineering tools. *IEEE software*, 28 (4), 86-91.
- [6] De Gea, J. M. C., Nicolás, J., Alemán, J. L. F., Toval, A., Ebert, C., & Vizcaíno, A. (2012). Requirements engineering tools: Capabilities, survey and assessment. *Information and Software Technology*, 54 (10), 1142-1157.
- [7] Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2006). Requirements engineering for trust management: model, methodology, and reasoning. *International Journal of Information Security*, 5 (4), 257-274.
- [8] Schön, E. M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49, 79-91.
- [9] Garrido, J. L., Gea, M., & Rodríguez, M. L. (2005). Requirements engineering in cooperative systems. In *Requirements Engineering for Sociotechnical Systems* (pp. 226-244). IGI Global.
- [10] Massacci, F., Prest, M., & Zannone, N. (2005). Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation. *Computer Standards & Interfaces*, 27 (5), 445-455.
- [11] De Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M.,... & Weyns, D. (2013). Software engineering for self-adaptive systems: A second research roadmap. In *Software Engineering for Self-Adaptive Systems II* (pp. 1-32). Springer, Berlin, Heidelberg.
- [12] Van Vliet, H., Van Vliet, H., & Van Vliet, J. C. (2008). *Software engineering: principles and practice* (Vol. 13). Hoboken, NJ: John Wiley & Sons.
- [13] Wasserman, T. (2010). Software engineering issues for mobile application development. *FoSER 2010*.
- [14] Frank, U. (2013). Domain-specific modeling languages: requirements analysis and design guidelines. In *Domain Engineering* (pp. 133-157). Springer, Berlin, Heidelberg.
- [15] Bleistein, S. J., Cox, K., & Verner, J. (2005, March). Strategic alignment in requirements analysis for organizational IT: an integrated approach. In *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 1300-1307). ACM.
- [16] Verma, K., & Kass, A. (2008, October). Requirements analysis tool: A tool for automatically analyzing software requirements documents. In *International semantic web conference* (pp. 751-763). Springer, Berlin, Heidelberg.
- [17] Geisser, M., & Hildenbrand, T. (2006, August). A method for collaborative requirements elicitation and decision-supported requirements analysis. In *IFIP World Computer Congress, TC 2* (pp. 108-122). Springer, Boston, MA.
- [18] Sutcliffe, A., & Sawyer, P. (2013, July). Requirements elicitation: Towards the unknown unknowns. In *2013 21st IEEE International Requirements Engineering Conference (RE)* (pp. 92-104). IEEE.

- [19] Chakraborty, S., Sarker, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the association for information systems*, 11 (4), 1.
- [20] Fuentes-Fernández, R., Gómez-Sanz, J. J., & Pavón, J. (2010). Understanding the human context in requirements elicitation. *Requirements engineering*, 15 (3), 267-283.
- [21] Razali, R., & Anwar, F. (2011). Selecting the right stakeholders for requirements elicitation: a systematic approach. *Journal of Theoretical and Applied Information Technology*, 33 (2), 250-257.
- [22] Sharma, S., & Pandey, S. K. (2013). Revisiting requirements elicitation techniques. *International Journal of Computer Applications*, 75 (12).
- [23] Sharma, S., & Pandey, S. K. (2014, March). Requirements elicitation: Issues and challenges. In *2014 International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 151-155). IEEE.
- [24] Kiran, A. S., Kaplan, C., Cetinkaya, T., Bayiz, M., & Cameron, J. (2010). *U.S. Patent No. 7,650,293*. Washington, DC: U.S. Patent and Trademark Office.
- [25] Khan, M. N. A., Khalid, M., & ul Haq, S. (2013). Review of requirements management issues in software development. *International Journal of Modern Education and Computer Science*, 5 (1), 21.
- [26] Bernard, Y. (2012). Requirements management within a full model-based engineering approach. *Systems Engineering*, 15 (2), 119-139.
- [27] Dick, J., Hull, E., & Jackson, K. (2017). *Requirements engineering*. Springer.
- [28] Chemuturi, M. (2012). *Requirements engineering and management for software development projects*. Springer Science & Business Media.
- [29] Highsmith, J. R. (2013). *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley.
- [30] Torkar, R., Gorschek, T., Feldt, R., Raja, U. A., & Kamran, K. (2009). Requirements traceability state-of-the-art: A systematic review and industry case study. *IST Journal*, in submission.
- [31] Shehata, M., Eberlein, A., & Fapojuwo, A. (2006). IRIS-TS: Detecting Interactions between Requirements in Doors. *INFOCOMP*, 5 (4), 34-43.
- [32] Mogk, N. W. (2014). A requirements management system based on an optimization model of the design process. *Procedia Computer Science*, 28, 221-227.
- [33] Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE software*, (4), 58-66.
- [34] Pandey, D., Suman, U., & Ramani, A. K. (2010, October). An effective requirement engineering process model for software development and requirements management. In *2010 International Conference on Advances in Recent Technologies in Communication and Computing* (pp. 287-291). IEEE.