

Review of Requirements Management Issues in Software Development

Muhammad Naeem Ahmed Khan
Department of Computing
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST),
Islamabad, Pakistan.
mnak2010@gmail.com

Muhammad Khalid
Department of Computing
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST),
Islamabad, Pakistan.
successcodes2009@gmail.com

Sami ul Haq
Department of Computing
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST),
Islamabad, Pakistan.
uomiansame007@gmail.com

Abstract— A requirement is a capability to which a product or service should conform to. A meticulous consideration to requirements engineering acts as a backbone of software projects. Ambiguous and unrealistic requirements are major source of failure in the software-intensive systems. Requirements engineering processes are complex as most of the requirements engineering documentation is written in natural languages which are less formal and often distract the designers and developers. Requirements management is a continuous process throughout the project lifecycle and relates to documenting, analyzing, tracing and prioritizing requirements and then finally controlling changes. The main issues related to requirements management are usually social, political and cultural. Software requirement engineers who gather the requirements generally consider that such issues are beyond the scope of their profession as they deem them within the project management ambit. In this study, we highlight the management issues that arise in the requirements engineering process and explore the possibilities to tackle them amicably. The study is supplemented with a critical review of the existing methodologies for resolving and managing software requirements.

Index Terms— Requirements Engineering, Requirements Management, Extreme Programming, Software Engineering

I. INTRODUCTION

Software requirements serve as a baseline for any software project and are considered as a capability or condition to which a system must conform to. In other words, requirements elicitation is the ability or function

required by a customer or consumer to achieve goals and objectives for a system like industry standards, government directives or restriction imposed by the stakeholders. Alternately, requirements can also be defined as a set of guidelines, condition, capability or ability that a system must provide or be consistent with.

Requirements are actually customer's statements of scope. In requirements finalization process, the stakeholders play an important role. A stakeholder can be defined as anyone who is directly or indirectly affected by the system being developed or deployed. Stakeholders are broadly categorized into two major classes — user and customer. User ordinarily uses the system and customer refers to those persons who have requested for the development of the system and are responsible for approving it. There may be a number of people who participate in the development of a system like business analysts, designers, coders, testers, project managers, deployment managers, use case designers, graphic designers etc. and are customarily considered as stakeholders.

Requirements engineering (RE) is a group of activities to elicit, analyze, specify, verify, validate and manage requirements.

Defining Requirements: Requirements are actually customer's statement of scope or interest. One peculiar aspect of human nature is that we fail to understand each other completely since the human communications are imprecise intrinsically. Requirements gathering is also a form of human communications in which an attempt is made to pass on complex ideas from one mind to another. Requirements are also a sparse form of communications, using bare written words to strive for precision [1]. Requirements engineering inflicts a direct elevated impact on the product's quality.

Requirements Elicitation: Elicitation is one of the first phases in requirements engineering and its purpose is to discover requirements for the system being developed. Requirements are elicited from customers, end-users and other stakeholders such as system developers [2].

Requirements Analysis and Specification: Requirements analysis is one of the first phases in requirements engineering and its purpose is to analyze the elicited requirements. Once the requirements have been gathered, then the conflicts, overlaps, omissions and inconsistencies need to be analyzed [2,3].

Requirements Specification: In this process, the requirements (both functional and non-functional) are documented. On the basis of the accumulated requirements, SRS document is created to which both the parties should agree upon.

Requirements Validation: This phase relates to the process of examining the requirements document to ensure that it pertains to the intended development of the right system (i.e., the system that the users expect) [2]. In this sense, validation processes purely commensurate to the functional requirements (FR).

Requirements Verification: It is the process of ensuring that the requirements' statements are accurate and complete as well as demonstrate the desired quality characteristics [4]. Hence, it pertains to non-functional requirements (NFR).

Requirements Management: Requirements management process administers changes to the agreed requirements, relationships between requirements and dependences between the requirements document as well as other documents produced during the entire system and software engineering process [2].

Requirements engineering plays an important role in the software development projects. If, somehow, this phase goes erroneous then the entire project will fail. A number of studies conducted on this issue have reported that the foremost reason of project failure is due to employing bad requirements engineering practices. In [5], it has been reported that 70% of the systems errors are ordinarily due to the improper requirements and remaining 30% are because of the design faults.

This paper is structured into four sections. An introduction to the requirements management issues has been provided in this section. The next section summarizes literature review of the current requirements engineering issues and practices. A critical analysis and comparison of the discussed techniques is enunciated in section III and, finally, we conclude in the last section.

II. LITERATURE REVIEW

According to the IEEE glossary, the requirements engineering is the degree to which a system, component or process meets specified requirements (i.e., FR) as well as the customer/user needs or expectations (i.e., NFR). From ISO 8402 perspective, the requirements

engineering pertains to the totality of features and characteristics of a product or service that bears on its ability to satisfy specified (FR) or implied (NFR) needs. Whereas, the software quality is considered as *conformance to requirements* [6] which is a manufacturer's perspective and *fitness for use* [7] which is solely a user's perspective. As a result, the definitions of software product's quality also have a strong association with the requirements. In other words, it is fair enough to assume that software quality is actually the implementation of requirements in the real sense.

Mishra [8] highlights the importance of RE and stresses that selecting the right technique is the backbone for the project success. The author proposes some success factors for the RE process and defines some steps to be followed in the RE process. The first step is to hold interviews with the potential customers to identify the key requirements followed by the analysis of the information by the requirements engineering team in a brainstorming meeting to further clarify the formally defined key requirements. Workshops are also arranged for all the stakeholders to discuss the key requirements. The purpose of workshops is to further refine the requirements and to finalize only the specific and actual requirements. These key requirements then provide a base for the architectural design and maintain the requirements repository. Afterwards, the requirements are prioritized to identify the requirements that need to be accomplished earlier and an iterative process of development starts from these prioritized requirements. Each requirement is then tested by the customer and feedback is sought. The same techniques need to be applied in the real world projects to validate the requirements.

Separovic and Car [9] propose a requirements management process model for software based legacy systems. Main reason behind any failed project pertains to incomplete requirements specification and adopting improper requirements management techniques. It is also the main reason for cost and schedule overruns which ultimately induce conflicts with the clients. The success of a project heavily depends on effectively managing the five key factors — time, cost, performance, user satisfaction and its overall impact on the entire organization. To establish requirements elicitation process, building the requirements management team becomes necessary when effective project communication starts and roles and responsibilities are defined. A requirements management team should consist of system architects (to design the system architecture), end users, business domain experts, consultants (having strong knowledge of new technologies), management and developers.

To model requirements and elicitation process, the first step, known as preparing project activities, is to organize a kick off meeting at the start of a project to educate team members about the project scope, roles and responsibilities. In the next step, it necessitates to analyze current state for process examination, end user

interrogation, testing the application logic and interviewing the domain experts. Moreover, workflow diagrams, system architecture description, interface description and business process description are developed in this step. The third step is the elicitation of new requirements in which introspection or coordination of end user and experts is elaborated and documentation of desired requirements is prepared. The fourth step is the analysis phase which involves setting priorities for requirements verification and validation. The fifth step in this process is change management in which necessary documents are prepared with the help of end users and domain experts. This step iterates back with the third step. Whereas, to model the change management process for development based on legacy system functionalities, new change request can be made by the software engineers, end users and change control board. If the change request is within the scope then it will go to requirement implementation phase or otherwise it will be shifted to implementation phase.

Pandey *et al.* [10] emphasize that it is extremely important to execute planning phase independently to achieve an effective requirements management. Requirements engineering process pertains to analyzing the requirements, proper documentation of the requirements, finding or discovering the requirements and applying proper validation methods. The quality focused requirements engineering not only helps in the software development phase but also enhances software quality from both client and user perspective. Finally, the author proposes a requirements engineering process model for effective requirements gathering.

The requirements engineering is a continuous process. The requirements elicitation phase starts with identification of stakeholders. During the requirements elicitation and development phase, the target is to gather requirements from different sources of the system and examine them accordingly. The requirements are collected in multifaceted standpoint such as business point of view requirements, customer point of view requirements and system requirements. Besides customers and users, the requirements are also collected from the business environment. In other words, the requirements are also gathered from the other systems having similar functionality e.g., tools, required resources for the system and other environment factors. Finally, the gathered requirements are handed over to the technical department which checks the requirements from technical perspectives.

Cheng and Lui [11] present view-based requirements elicitation techniques to collect information from different stakeholders involved in the projects and a framework is sketched on the basis of merging of different viewpoints. The core philosophy of the idea is to present an effective and reasonable framework.

Stakeholders differ from project to project as large projects have significant number of stakeholders involved in the execution of project. Different

stakeholders have different concerns and different viewpoints that must be managed for the successful execution of the project. A framework is required to merge different requirements of stakeholders into one single achievable goal. The proposed framework focuses on merging the different viewpoints and consists of different states to finally reach at the accepted requirements. Hierarchical transition system is used to combine state charts. Each stakeholder can rank its behaviors according to his/her preferences. For this purpose, a finite number of preferences are given to the stakeholders. The core of algorithm is M-operator which is used to combine the two state charts: disagreement state and non-determinism/redundancy state. Different stakeholders keep atypical expectations about the system that may not be realistic. For eliminating such requirements, refinement operator is applied to remove the undesired requirements. The proposed model is an extended framework based on merging that state models.

Lu *et al.* [12] proposed a novel technique to cope with the ambiguities in the requirements document and sketched a model called Model-based Object-oriented Requirement Engineering (MORE). The useful domain knowledge can be collected by using the modeling and object-oriented technologies for requirements engineering quality phases.

Jiang and Xuping [13] present RE process as a research object by proposing a formal method for describing RE processes and highlight different tools and technologies that are used during the RE processes. They also discuss three typical requirements engineering processes: Top-Down, Middle-Out and Bottom-Up. These processes are explained using the RE Process Model (REPM) framework. The research also presents eight formal definitions of RE process in general. These definitions are based on the mathematical concepts like set theory. The proposed method named as REPM is based on formalism and is an integrated approach consisting of three distinct aforesaid processes. However, the model lacks validation and the three RE processes are not explained in detail which becomes hard to properly understand and implement the proposed concept.

Requirements changes are very critical step in any of the software development process. Managing the requirements changes is an open issue in the literature for many decades. In literature, various requirements change approaches have been suggested but none of these completely cope with it. Lu and Yuan [14] suggest an algorithm which can accurately find out requirement changes. The proposed algorithm is called *LCS-NP++* and this approach generates different version of requirements changes. Based on accuracy measures, the version of requirements changes is selected which has maximum efficiency and accuracy. Not only this technique can perfectly identify the requirements changes, but also becomes handy when someone has to find the complex HTML structure.

Pavasasam and Subramaniam [15] propose a model for requirements inspection and state that the requirements change management and requirements analysis phases play an important role in this process. JDesigner tool is used for verification of different requirements engineering processes which consist of *requirement transformation* and *system requirement representation* phases. In the requirements transformation phase, the authors assign appropriate weights for amalgamation, abstraction and clarity factors. Afterwards, this information is assigned to system requirements which in turn assign 0.5%, 0.7% and 0.6% weights to consistency, completeness and correctness factors respectively. The functional requirements comes from system requirements. Finally, the membrane computing model computes the system requirements as a product of a constant multiplied by abstraction, amalgamation and clarity factors. To comprehend the working of the model, a strong mathematical background is required.

Risk is always a challenging issue that causes defects in the project execution as it inflicts at different phases of project with different level of effects. Eun-Ser and Je-Min [16] present an opportunity tree through which defects are managed and solutions are provided. A relationship is developed between the defects and their causes. Software processes are being studied for improvement since a long time. All the software requirements must be defined to develop high quality software. The basic technique used is requirements elicitation which is a critical part of the project and different techniques are required to determine the requirements. Fundamental way is to perform analysis in order to identify people, processes and different resources involved in the project. Defects are defined as a special type that represents the specific product in relation to the processes. Large scale projects have thousands of defects and are found at different stages of projects. There are many tools that are used to minimize the defects. The defect removal efficiency is used to represent total number of defects found at one stage. Analytic hierarchy process is also used for formalizing the decision making process. The concept of defect management is to estimate how the defects change during the development phase. For this purpose, opportunity tree is introduced to select main goals, sub goals, decision goals and eventually making questionnaire for verification, as well as suggesting solution to design the opportunity tree. Different ratings are used for defect data collection and identifications of items. Different defects are efficiently analyzed using the defect management opportunity tree. Defect removal efficiency analysis makes it easier to find the defects relative to the stages of the development of project.

According to Tchidi and He [17], Six Sigma defines set of practices to improve processes. Six Sigma can be used for requirements engineering process model. The

design thus evolve will be called Design For Six Sigma (DFSS). The DFSS focuses on the following key points:

- Continuous focus on quality.
- Defining solutions or goals that ensure adequate transformation of customer requirements.
- Constant improvement of processes. In the realm of requirements engineering, these processes are related to requirements definition.

As mentioned above, the DFSS focuses on process improvement to achieve quality and customer satisfaction. In light of this, DFSS methodology is defined in the following phases:

- Define goals for requirements engineering that are consistent with customer demands.
- Measure and identify the quality characteristics. In the realm of requirements engineering, this pertains to defining key quality goals that can help decide whether the desired quality is met or not. If the quality is not up to the mark, then it is deemed as a defect that needs to be rectified.
- Develop designs that meet the defined goals. Analyze the developed design based on the quality characteristics. Discard the designs that do not meet the predefined quality characteristics. If multiple designs are available which meet the quality characteristics, then select the best design.
- Optimize the best developed design. This may require constant probing of design using simulations or dry runs. The final output will be an optimized design solution.
- Verify the design by implementing system prototype and test runs.

According to Schummer and Lukosch [18], distributed pair programming is an agile software development methodology where two programmers located at different geographic locations jointly work using collaborative real time editor. The key difference between pair programming and distributed pair programming is that the programmers in the later technique are located at different geographic locations and various communication means are required to be made available for practicing this methodology. Distributed extreme programming, especially distributed pair programming, is destined to failure unless proper tools are employed that support social practices. Murphy and Phung [19] state that extreme programming is also a type of agile software development methodology that focuses on early software releases and is geared towards improving the software quality.

III. CRITICAL ANALYSIS

A critical review of the several requirements management techniques proposed in the contemporary literature is provided in Table I. The critical analysis is primarily based on highlighting scope and key features of the techniques discussed in the literature.

Table I. CRITICAL ANALYSIS OF REQUIREMENTS MANAGEMENT TECHNIQUES.

Author	Scope	Key Features Addressed	Critical Analysis
Mishra [8]	RE techniques and process.	The success factors for RE process are highlighted.	The proposed technique needs to be applied in real world projects to check its validity and performance.
Separovic and Car [9]	Requirements elicitation and Change management.	The need for establishing link between requirements and change management is emphasized.	This process model requires proper implementation before adaptation by the industry.
Pandey [10]	Requirement analysis, Requirement elicitation.	The strong relationship among the RE activates e.g., requirement elicitation, requirement validation, verification and requirement documentation are underlined.	The proposed RE process model is merely effective for large software projects.
Cheng and Lui [11]	Stakeholder preference.	The proposed model is an extended framework based on the merging state model viewpoints.	The model requires proper implementation and verification.
Lu <i>et al.</i> [12]	Object oriented approach for RE.	Technique to cope with ambiguities in the requirement document is proposed.	Relationship between MOORM and MORE is required to be clearly defined.
Jiang and Xuping [13]	RE Process Met model.	The proposed RE Process Met model (REPM) is based on formalism which offers an integrated approach.	The model lacks validation and the three RE processes needs detailed explanation for proper implementation.
Lu and Yuan [14]	RE, Requirement management.	The proposed algorithm is capable to accurately find out requirement changes.	This technique become handy while finding complex HTML structure.
Pavanasam and Subramaniam [15]	Requirement inspection.	The model for requirement inspection using JDesigner can be used for verification of different RE processes.	Ample comprehension of JDesigner tool is required to implement the proposed model.
Lee and Bae [16]	Requirement Management.	The proposed opportunity tree is in fact not a model, but can be used for defect analysis of elicited requirements.	A proper framework for analyzing defects.
Tchidi and He [17]	Process improvement	A structured process improvement strategy to reduce risk and achieve quality.	Authors emphasize more on quality improvement than the process improvement.
Schummer and Lukosch [18]	Agile methods, Pair Programming.	Distributed extreme programming with reference to distributed pair programming is destined to failure unless proper tools are used.	Distributed pair programming has a serious communication and coordination issues as the programmers are located at different geographic locations.
Murphy and Phung [19]	Distance learning.	A novel distance learning approach for teaching extreme programming.	Virtual presence is the key limitation as participants are not co-located.

IV. CONCLUSION

The requirement engineering is a continuous process. The requirement elicitation phase starts with identification of stakeholders. During the requirement elicitation and development phase, the target is to gather requirements from different sources of the system and examine them. Besides the client and the users, the requirements are also collected from the environment. In other words, the requirements are gathered from other systems that have similar functionality e.g., tools, required resources for the system and other environment

factors etc. The gathered requirements are handed over to technical department which checks the technicality of the requirements.

There are some prominent factors in software project management e.g., scope, cost, time and quality. Software engineering explores constructive and dynamic ways to manage the entire project lifecycle. A requirements change is a critical phase in any of the software development process. Managing the requirement changes is an open issue in the literature for many decades.

Software processes are being studied for improvement since long time. All the software requirements must be defined to develop high quality software. The basic techniques used in requirements elicitation are a critical part of the project and different techniques are required to determine the requirements of the client.

REFERENCES

- [1] R. R. Young, *The Requirements Engineering Handbook*, ARTECH HOUSE, INC. 685 Canton Street Norwood, MA 02062, 2004.
- [2] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, Chichester, UK: John Wiley & Sons, 1998.
- [3] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, New York: John Wiley & Sons, 1997.
- [4] K. E. Wiegers, *Software Requirements*, 2nd ed., Redmond, W A: Microsoft Press, 2003.
- [5] F. W. Beichter, O. Herzog and H. Petzsch, "SLAN-4 A Software Specification and Design Language", IEEE Transaction on Software Engineering, Volume 10, Number 1, pp 155-162, 1994.
- [6] P. B. Crosby, *Quality is Free*, New York: McGraw-Hill, 1979.
- [7] J. M. Juran and F. M. Gryna, *Quality planning and analysis: From product development through usage*, McGraw-Hill, New York, 1970.
- [8] D. Mishra, A. Mishra and A. Yazici, "Successful Requirement Elicitation by Combining Requirement Engineering Techniques", International Conference on the Applications of Digital Information and Web Technologies (ICADIWT'08), pp. 258-263, 2008.
- [9] B. Zagajsek, K. Separovic and Z. Car, "Requirements Management Process Model for Software Development Based on Legacy System Functionalities", 9th International Conference on Telecommunications (ConTEL 2007) pp. 115 - 122, 13-15 June 2007.
- [10] D. Pandey, U. Suman and A. K. Ramani, "An Effective Requirement Engineering Process Model for Software Development and Requirements Management", International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom, 2010), pp. 287 - 291, 16-17 October 2010.
- [11] J. Cheng and Q. Liu, "Using Stakeholder Analysis for Improving Statechart Merging in Software Requirement Management", The 9th International Conference for Young Computer Scientists (ICYCS, 2008), pp. 1217 - 1222, School of Software, Tsinghua University, Beijing, China, 18-21 November 2008.
- [12] C-W Lu, W. C. Chu and C-H Chang, "A Model-based Object-oriented Approach to Requirement Engineering (MORE)", 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), Hsiuping Institute of Technology, Taichung, Volume 1, pp. 153 - 156, 24-27 July 2007.
- [13] X. Jiang, "Modeling and Application of Requirements Engineering Process Metamodel", IEEE International Symposium on Knowledge Acquisition and Modeling Workshop (KAM Workshop, 2008), Department of Commanding Communication, PLA Commanding Communication Academy, Wuhan, China, pp. 998 - 1001, 21-22 December 2008.
- [14] G. Lu and F. Yuan, "Comparison of Requirement Items based on the Requirements Change Management System of QONE", Second World Congress on Software Engineering (WCSE), Institute of Computer Technology, Chinese Academy of Science, Beijing, China Volume 2, pp. 211 - 215, 19-20 December 2010.
- [15] V. Pavanagam, C. Subramaniam, T. Srinivasan, J. K. Jain, "Membrane Computing Model for Software Requirement Engineering", Second International Conference on Computer and Network Technology (ICCNT), Sathyabama University, Chennai, India, pp. 487 - 491, 23-25 April 2010.
- [16] E-S Lee and J-M Bae, "Design Opportunity Tree for Requirement Management and Software Process Improvement", International Conference on Multimedia and Ubiquitous Engineering (MUE'07), Soongsil University, Seoul, pp. 395 - 400, 26-28 April 2007..
- [17] M. F. Tchidi and Z. He, "The Requirements Engineering Process Model Based on Design for Six Sigma", The 2nd IEEE International Conference on Information Management and Engineering (ICIME, 2010), Tianjin University, Tianjin, China, pp. 287 - 290, 16-18 April 2010.
- [18] T. Schummer and S. Lukosch, "Supporting the Social Practices of Distributed Pair Programming", In Groupware: Design, Implementation, and Use, volume 5411/2008 of Lecture Notes in Computer Science, pages 83-98. Springer, Berlin / Heidelberg, March 2008.
- [19] C. Murphy, D. Phung and G. Kaiser, "A Distance Learning Approach to Teaching eXtreme Programming", ITiCSE, June 30-July 2, 2008, Madrid, Spain. ACM. 2008.

AUTHORS



Muhammad Naeem Ahmed Khan obtained degree in Computer Systems from the University of Sussex, UK. His research interests include software engineering, digital forensics and machine learning techniques.



Muhammad Khalid is presently pursuing his MS in Computing from SZABIST, Islamabad. His research interests include software engineering.



Sami Ulhaq is a student of Masters in Software Engineering at SZABIST, Islamabad. His focused research areas are software requirement engineering, software quality engineering and global software development.