

An Improved Model for Requirement Management System

Hafeez MS^{1*}, Farhan Rasheed¹ and Khan MR²

¹Department of Software Engineering, Foundation University, Islamabad

²Department of Computer Sciences, Comsats Institute of Information Technology, Attock

Abstract

Requirement engineering is the most important part of the modern software engineering, which evolved over decades and become a science. It probably plays the most vital part in success of the software system. The requirements management world has developed considerably in the last decade and has increasingly become one of the bases of successful software and systems engineering projects. Managing requirements is not an easy task; in order to record good requirements and manage the changes afterwards is the trickiest part. Many researchers around the world have worked on optimization of the requirement management process. In this paper we propose a model which is an improved version of the model by Nathan W Mogk. To overcome the limitations of that model we proposed a comprehensive model with focus on the limitation of this model and provide an enhanced model. This paper is the extension of the requirements management system based on an optimization model of the design process.

Keywords: Software requirements; Requirement management; Stakeholders; Requirement engineering; Requirement traceability

Introduction

Mostly the requirements are gathered from the customers by using some elicitation technique. There are many elicitation techniques currently in practice in the software development industry. The first and foremost thing to choose the elicitation technique which best suits the software, as the best requirement elicitation technique will help getting the good requirements and it will accelerate the development process. Once we have the requirements the changes are inevitable. Changes keep on coming till the end it is the responsibility of the requirement management team leader to decide which change to accommodate which change request needed to be ignored. Every software development company usually has the criteria for change management, usually not fixed as well. Requirements management is the process of documenting, analyzing, tracing, prioritizing, negotiating, agreeing on requirements and then controlling change and communicating to relevant stakeholders again negotiating. It is a process which continues throughout the software development life cycle. Without traceability the change management is not possible and it is nearly impossible to comprehend the effect that a configuration change has on the customer requirements and it is hard to survey the effect that a change to requirements has on plan at the beginning.

Overseeing requirements in an exceptional case the requirement management model makes it hard to gather requirements with change request and makes it even harder to survey the effect of changes. Even though where requirements are overseen, it is critical to get the “right” requirements from customer, and to get the requirements “right” in your software development plan, as the success of the software purely based on customer satisfaction. A recent survey depicts that the 46% of software development teams complains ‘Product does not meet customer needs’ as the reason product launch is either delayed or failed. What the customer wants is the question which seems simple but very difficult to answer [1].

Requirements Management is considered as management of requirements it basically starts when a customer provides his/her needs or a process of software development is started. It consists of managing the requirements definition, explanation and changing requirements during the development cycle and systems development. A requirements management expert describes the following key steps in requirements management process.

First of all requirement management plan is established, the

plan includes the criteria of requirement change. In the second step requirements are elicited. 3rd step is the development of the vision document. In the 4th step cases are developed. The 5th step describes the supplementary specifications. Test cases from the uses cases are developed in 6th step. The 7th step comprises of the test cases from the supplementary specification. In the last step system design is created [2].

Requirements traceability is apprehensive with documenting the life of a requirement from idea to development and providing bi-directional traceability b/w a variety of requirements associated. It allows users to identify the root of each and every requirement and keep track of each and every change to that particular requirement. For keep track of every change made to the requirement it is required to document each and every change made to the requirement.

Requirement traceability comprises of three steps:

- Following the life of a requirement from idea to implementation, from root to the end of the requirement.
- How requirements impact each other, and how requirements impact other development lifecycle artifacts.
- The decomposition of requirements from high level customer, market needs to system, hardware component requirements and its transformation into design [3].

This paper proposes a model which is the improved version of the model proposed by Nathan W Mogk a model based on an optimized approach for design process for requirement management. The limitations of that model are discussed in the methodology section and the new model is proposed.

Literature Review/ Related Work

The software engineering disciplined rapidly evolved over the last

***Corresponding author:** Mohammad Salman Hafeez, Department of Software Engineering, Foundation University, Islamabad, Jordan, Tel: + 923009769689; E-mail: salmaan.hafeez@gmail.com

Received December 21, 2016; **Accepted** January 20, 2017; **Published** January 27, 2017

Citation: Hafeez MS, Farhan Rasheed, Khan MR (2017) An Improved Model for Requirement Management System. J Inform Tech Softw Eng 7: 196. doi: [10.4172/2165-7866.1000196](https://doi.org/10.4172/2165-7866.1000196)

Copyright: © 2017 Hafeez MS, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

two decade. Many researchers around the world have focused their research for the improvement of this branch of study. Requirement engineering being the integral part of software engineering discipline is area of research of many all around the world.

Torkar et al. discusses the state of the art technique called requirement management. Requirements traceability facilitates software developers to trace a requirement from its inception to its completeness [4]. This paper discusses requirements traceability, its definitions, challenges it faces, techniques and tools, by the use of a systematic analysis performing an comprehensive research through the years 1997-2007. Concludes that requirement traceability is of utmost importance.

Pandey et al. proposed a model for software development and requirement management, it discusses that the Requirement engineering is the most essential stage of software development life cycle. The aim is to collect good requirements from key stakeholders in the precise way. It is vital for each organization to produce quality software products that can gratify what the user need. Requirements engineering in software development process is a composite work that considers product demands from viewpoints, positions, responsibilities, and purposes. Therefore, it's nearly mandatory to apply requirement engineering practices in every stage of software development process. This paper proposes an effective requirements engineering process model to produce quality requirements for software development. Requirement management and planning stage executed in separate for a useful management of requirements. It is iterative process which is for good for requirement engineering. The results implied that the successful implementation of proposed model for requirement engineering and management has a good impact on the production of quality software product [5].

Zagajsek et al. describes a process model for requirements elicitation, specification, analysis and management of requirements. The proposed process model is linked with the following software change management process. The connection b/w requirements and business process modeling is defined in business process artifacts which are obtained during requirement activities. The proposed process model is intended for a lately formed development team without an adopted process methodology [6].

Mogk et al. [7] in their research article a Requirements Management System based on an Optimization Model of the Design Process describes an engineering design process which is modeled as an improvement issue over the system design space, where the goals, results, and exchange first choices of the users are defined as measurable objective functions. In this kind of formulation, user requirements become constraints that define the viable region of all possible system designs. The execution of a requirement management process depends on the above mentioned reformulated design process. The ultimate goals of the reformulated design process is to present a method of viewing the requirements that enhanced communication with the what the user wants with the design team to build, to boost up the user participation in the system definition process, improve adherence what the user wants when conflicts arise, and to allow for combination of decision making techniques with the requirements discovery and management processes.

Following the proposed requirements management model, the article describes some possible methods of solving non-linear optimization issues that can be practical to this kind of formulation of the design process, and the conditions under which it can be applied. The model is named as BONC, binding constraints, objectives, non-binding constraints, and conflicts.

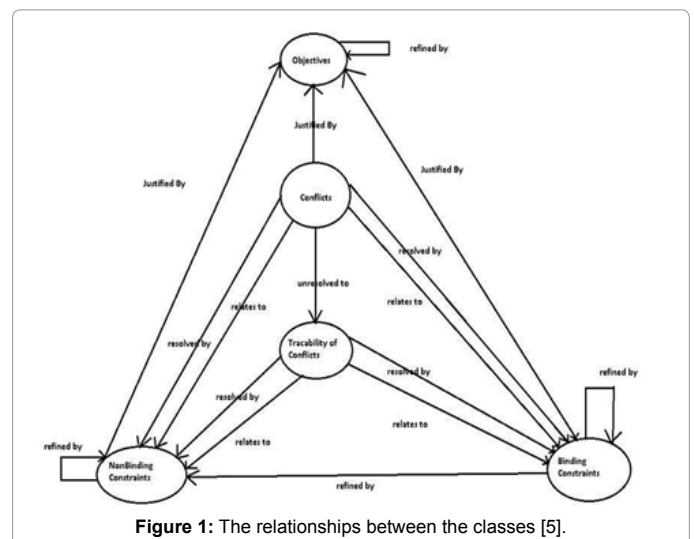
Proposed Model

The proposed model is the extension of the model proposed by Mogk et al. [7], for the optimization purposes. In this research article the proposed model does not mention any process which can be used to resolve conflicts b/w requirements in case of any change arrives, called them unresolved ones. They just proposed that requirement traceability will be done by using the traceability matrix, which might not be a good option in every situation and should not report with the body of current constraints and remove unsolved constraints as irrelevant.

We proposed a model which is an extension of the above mentioned model focusing on the limitation of the model proposed by Mogk et al. As traceability the issue, our focus is to improve the process of traceability in the above mentioned model. We added a phase of traceability of unresolved requirement with a focused negotiation stage which will require user involvement with the aim to resolve the conflicts. It will definitely results in lesser conflicts plus the user will have some idea of what its product will look alike now having some of his/her changes incorporated and some of them discarded. One can't just simply ignore the conflicts at a development end, considering unresolved conflicts as irrelevant without user involvement because it might the case that the important ones left behind without the customer knowledge ultimately results in failure of the software. We proposed that traceability matrix should not be fixed there should be traceability techniques that should be used which best suits the situation (Figure 1). We suggest that user involvement along with any of the traceability techniques finding the conflicts should be there for resolving the conflicts.

1. Objective: What the customer wants?
2. Binding Constraints: A constraint for the solution exists in exact.
3. Non-Binding Constraints: A constraint for which the solution exceeding the demands of the constraint.
4. Conflict: When binding and non-binding constraints conflict with each other new conflict is occurs?
5. Traceability of Conflicts: Trace the conflict between two constraints and perform impact analysis and find out the importance of the constraint in view of user. If importance is paramount then don't let it unresolved. Negotiate with customer aiming to resolve it.

These are all relationships between phases:



- Resolved by
- Relates to
- Justified by
- Refined by
- Unresolved to

Comparitive Analysis

In this section comparison b/w the model proposed by Nathan W Mogk and our proposed model is discussed.

We added a new phase called traceability stage, will focus on the unresolved constraints, with user involvement being the center of attention. In our proposed model at first objectives are defined. Objectives are linked with the binding and non-binding constraints. Binding constraints and non-binding constraints are the stages which iteratively refines themselves and by each other. Conflicts occur which are related to both binding and non-binding constraints, these conflicts can be resolved by the binding and non-binding constraints stages.

The conflicts which cannot be resolved by binding and non-binding stages, called unresolved conflicts which moves in to the stage of traceability with user involvement in focus. These conflicts caused by binding and non-binding are negotiated with the user in order to resolve them. The ultimate goal is to try and resolve as many conflicts as possible, which will guarantee the successful requirement engineering process which ultimately help the developers in developing the right product.

The limitation of the earlier model is there is no user involvement in focus for resolving the conflicts simply ignoring unresolved conflicts which might be vital for the success/ failure of the system. Our aim was to focus those unresolved conflicts and finding out the importance of these requirements causing the conflicts in user view, and resolving them.

Conclusion

The requirement engineering is plays a vital role in success/failure of the software products. If the good requirements are gathered it will ultimately help in building good software. The user/customer should

be active in requirement engineering process; user involvement is of paramount importance for the success of the software product. The requirement management is very important phase of the requirement engineering process. The traceability is the technique which keeps tracks of the requirement from idea to execution, used for impact analysis which is one of the many uses of the traceability when a change of requirement is requested. There is no requirement management without traceability. We proposed a model which overcomes the limitations of the similar model based on optimization of the design process for requirement management system. Aim is to involve user whenever the conflicts occurs. The focus in on user involvement for the unresolved conflicts, which ultimately results in good requirement which accelerate the development process plus aid the success of the software product.

Acknowledgement

This research was supported by Department of Software Engineering, Foundation University Islamabad. We thank our colleagues from Comsats institute of information technology Attock, who provided insight and expertise that greatly assisted this research. We would also like to show our gratitude to Dr. Mohammad Shaheen of Foundation University Islamabad for sharing their pearls of wisdom with us during the course of this research, and we thank 2 reviewers from faculty of software engineering Foundation University Islamabad for their so-called insights.

References

1. Zielczynski P (2008) Requirements Management Using IBM® Rational® RequisitePro®. Pearson, InformIT.
2. Hoffman M, Kühn N, Weber M, Bittner M (2004) Requirements for Requirements Management Tools. IEEE International Requirements Engineering Conference.
3. Long D and Scott Z (2012) A Primer for Model Based Systems Engineering. Vitech Corporation.
4. Highsmith JA (2000) Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing, New York.
5. Torkar R, Gorschek T, Feldt R, Raja UA, Kamran K (2012) Requirements Traceability State-of-the-Art: A Systematic Review and Industry Case Study. International Journal of Software Engineering and Knowledge Engineering 22: 1-49.
6. Shehata M, Eberlein A, Fapojuwo A (2006) IRIS-TS: Detecting Interactions between Requirements in Doors. Journal of Computer Science 5: 34-43.
7. Mogka NW (2014) A Requirements Management System Based on an Optimization Model of the Design Process. Procedia Computer Science 28: 221-227.