

# Software Change Management: A Note on Significance, Tools and Support

Sunbal Siddique  
Department of Computer Science  
University of Agriculture  
Faisalabad, Pakistan

Ali Siddique  
Department of ECE  
University of Macao  
Macao

## ABSTRACT

Requirements management is an essential part of software engineering. The changes occurring in the requirements of the software, if not handled properly, can lead to catastrophic failures. It is due to this reason that the proper management of requirements change is considered a critical part of software development. In this paper, a brief review on change management in requirements is presented. Moreover, general concepts of requirements change management are elaborated along with the information on related tools that are available for different aspects of requirements change management activities in the market. Last but not least, various tools proposed by different researchers on several categories of requirements change management are compared with each other. It has been observed that 'requirements traceability' tools are discussed more frequently in the literature than others, and find more support than other classes of requirements change management.

## General Terms

Requirements Change Management, Software Development

## Keywords

Change Impact Analysis, Distributed Software Development, Features Tree, Global Software Development, Requirements Traceability

## 1. INTRODUCTION

In modern software engineering, due to increased complexity of software projects, requirements management has evolved as an important subject for the sake of better and successful software development [16]. The changes made in the requirements of a software project may cause some serious troubles in the later stages of development if proper planning is not done beforehand [4]. Moreover, in order to achieve better software development efficiency, and to optimize the entire software development process, the development of new tools and technologies has become the need of the hour [13]. Requirements Change Management (RCM) can be defined as, "the process of understanding, controlling, tracing and documenting changes in requirements" [11]. The efficient handling of continuously evolving requirements involves tracking the requirement changes over the entire development lifecycle and then identifying the impacts of these changes [15]. As it can be seen from

the definition and the description, the important aspects of requirements change management include change control, requirements traceability, and impact analysis. In this note, various tools on all these aspects of requirements management are discussed in detail after a careful review of the available literature.

The remainder of this paper is organized as follows: Section 2 discusses and presents the related work, Section 3 describes and discusses the aim of research, enlists the research questions, and explains the significance and ramifications of this research. Finally, Section 4 details the results, concludes the work, and discusses future directions.

## 2. LITERATURE REVIEW

There has been done some work related to this area of research. For instance, Aziz, R. A. has argued that knowledge of requirements relationships puts a great impact on change management and the successful implementation of change management can lead a software project towards success [1]. Minhas, N. M. has proposed a Requirements Change Management Tool for Global Software Development RCM\_GSD [11].

Lloyd, D. has used the concept of features model to make a 'features tree' that will help in making a tool for change management in Global Agile Development (GAD). It helps assign features (such as feature id, feature name, parent feature, feature type) to teams in different parts of the globe and then traces the features. When a change is made in one feature, the other affected feature(s) will spontaneously be tracked [9].

Garcia, J. E. has proposed a tool to map the functional requirements of a web based application with its webpages/HTML elements. The author states that manual approaches to make requirements traceability matrix are prone to errors and are difficult to maintain for large systems due to which this tool can be useful in case of web based applications to understand the link functional requirements with different parts of the web application [3]. M. Lang has proposed a web-based tool for collaborative software requirements management [8]. G. Huang has proposed a web-based system for engineering change management [6].

Mohan, K. has proposed an approach to make use of process and product knowledge to integrate traceability with software configuration management for better management of requirements [12]. S. Hallerstede has proposed a tool for tracing requirements to specifications so that the specifications are different from the stakeholder requirements [5]. Jokinen, L. has discussed the possible

causes of changes in requirements and has used a case study of an organization to understand the factors responsible for acceptance or rejection of a proposed change [7].

H. Saiedian has proposed a cost effective tool for requirements traceability management. Though the tool takes more time to develop, it offers several advantages: lesser time in release, minimum bugs, and low cost as compared to other tools available in market [14]. Gea, J. M. has assessed the various tools available in market for requirement engineering process by conducting a survey of different vendors of the tools [2]. M. G. Violante has presented a strategy to support the design of requirements management tools by considering various features that should be available in a requirements management tool to improve user satisfaction [16].

### 3. DISCUSSION

In this section, the significance of requirements change management is discussed in detail. Furthermore, information on various tools and support for change management, and the works of some qualified researchers are presented.

#### 3.1 Importance of Requirements Change Management

The following advantages are often associated with the management of changes in requirements.

**3.1.1 Global Software Development.** When software projects undergo Global Software Development/Distributed Software Development, lack of face-to-face communication, different cultures, different languages, different understanding of problem statement can lead to a chaos in requirements engineering process. Therefore the requirements need to be formally written and managed at a common repository, and changes in requirements need to be updated in the common database and notified to each and every member of the concerning party so that confusion may be avoided [11].

**3.1.2 Agile Development Technology.** Agile development method has grown to a large extent in software industry in the recent years. When software projects undergo agile development methodology, the requirements continuously change due to frequent meetings with customers and stakeholders. Therefore, the requirements need to be properly updated and stored [9].

**3.1.3 Correlation between Various Software Products in a Line.** A software product line means a family of products in a given application domain. The advancements in this trend of development demand great care for requirements change management because the requirement changes in one product will greatly impact other products in the application domain [10].

**3.1.4 Demands of Customers.** The demands of the customers keep changing [7]. Therefore, the updated version of requirements needs to be available all the time in order to prevent troubles.

**3.1.5 Competition.** For the companies to remain in the competition, the software projects need to be continuously changed and maintained even after deployment. Therefore the new requirements need to be properly analyzed, implemented and documented to keep the systems up to date [7].

**3.1.6 Increased Complexity.** Due to large size and increased complexity of the software projects in current era, it has become nearly impossible to understand and keep track of the changes without proper planning or taking preventive measures for change management. Therefore, it is of grave importance to plan and manage

requirements of the software projects from the very start of development [16].

#### 3.2 Tools and Support for Requirements Management

In this paper, seven works related to tools and frameworks for RCM are discussed. The suggested tools are given in Table 1. It can be seen from the table that Requirements Traceability Tools (RTTs) have been discussed extensively in the literature while Change Impact Analysis (CIA) has been neglected when it comes to proper calculation of change impact using automated tools.

Moreover Gea, J. M. also stated after a survey with the vendors of requirement engineering tools that Requirements Traceability has better tool support as compared to the other Requirements Management activities such as change control and impact analysis [2]. Therefore, it is concluded after a thorough study of the previous researches that 'requirements traceability' has more tool support as compared to 'change control' and 'impact analysis'. The results of the study are shown in Figure 1.

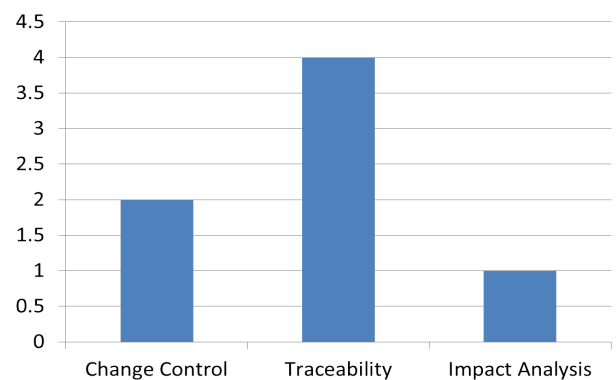


Fig. 1: Number of Tools for Different RCM Activities

However, it is to be informed that many of the tools suggested for implementation are only for research purposes and suffer from integration problems when implemented with commercial systems, and are very costly. Therefore, the use of such tools in industry would be very impractical [3, 14].

### 4. CONCLUSIONS

The RCM has become an important and necessary part of software development in the modern era. It is very important to know in detail the significance and impact of software change management. In this paper, the significance of software change management as well as various tools and support for different activities in RCM are briefly discussed.

It may be observed, however, that the set of samples of research works considered for this review is very small and therefore, the deduced results may not be claimed to be completely valid. Moreover, half of the tools suggested for automation have not been evaluated yet. In future it is intended to conduct an extensive research on this topic and provide better results in this area of study. Moreover, a higher level of focus on 'change impact analysis' is required in future so that the impact of each change may be critically examined and quantitatively calculated before execution.

Table 1. : Frameworks and Tools Presented in Literature for Different Activities of Change Management

| Author's Name       | Proposed Frameworks and Tools  | Activities             | Evaluation Status |
|---------------------|--|------------------------|-------------------|
| N. M. Minhas [11]   | Proposed a framework for change management with multilingual support so that language may not become a hurdle for distributed software development. The change request (CR) form contains all the information on requested change and the initiator. The change is assessed and accepted if all the members approve of it, otherwise the suggested change is rejected. The change request and the updated changes are stored in a common RCM repository/database.          | Change Control         | Evaluated         |
| D. Lloyd [9]        | Proposed a traceability tool by making using of features tree. The tool will store each feature with a feature id, feature name, parent feature and feature type (mandatory, alternative, optional) to maintain links and connections between features.  | Traceability           | Evaluated         |
| J. E. Garcia [3]    | Proposed a tool for web applications that maps functional requirements of the web application to its HTML components. The XML file containing the specification is first loaded to the tool manually. Then, the user chooses the functional requirements from the checklist and maps them by pointing and clicking to the HTML components. The tool then generates the traceability matrix and lists of tracing links between functional requirements and HTML components. | Traceability           | Not Evaluated     |
| J.E. Huang [6]      | Proposed a web-based tool for change management which keeps the information of change originator, change request id, change request name, the status of change, deadline etc., but does not provide traceability information at all. Also, the impact analysis is only <i>qualitative</i> (i.e., high, medium, low)  | Change Control         | Not Evaluated     |
| S. Hallerstedte [5] | Proposed a tool that makes use of formal methods to trace the requirements of the stakeholders to specifications.  | Traceability           | Not Evaluated     |
| H. Saiedian [14]    | Proposed a tool to trace the requirements to other requirements, design, source code and test cases. It has a database that keeps the information of the artifacts by their identifiers only, and a software wrapper to the database that has buttons and custom menu for human analysts to operate and manage it.   | Traceability           | Evaluated         |
| J. M. Zoun [10]     | Proposed a method for change impact analysis in software product lines that makes use of features model. It will estimate the impact of change by calculating the number of features, classes, methods, attributes and associations that are affected by changing a particular feature.  | Change Impact Analysis | Evaluated         |

## 5. REFERENCES

- [1] Ruhaya Ab Aziz and Bernard Wong. The interplay between requirements relationships knowledge and requirements change towards software project success: an assessment using partial least square (PLS). *Procedia Computer Science*, 46:732–741, 2015.
- [2] Juan M Carrillo De Gea, Joaquín Nicolás, José L Fernández

- Alemán, Ambrosio Toval, Christof Ebert, and Aurora Vizcaíno. Requirements engineering tools: Capabilities, survey and assessment. *Information and Software Technology*, 54(10):1142–1157, 2012.
- [3] Jorge Esparteiro Garcia and Ana CR Paiva. A Requirements-to-Implementation Mapping Tool for Requirements Traceability. *JSW*, 11(2):193–200, 2016.
- [4] Manju Geogy and Andhe Dharani. A Scrutiny of the Software Requirement Engineering process. *Procedia Technology*, 25:405–410, 2016.
- [5] Stefan Hallerstede, Michael Jastram, and Lukas Ladenberger. A method and tool for tracing requirements into specifications. *Science of Computer Programming*, 82:2–21, 2014.
- [6] GQ Huang, WY Yee, and KL Mak. Development of a web-based system for engineering change management. *Robotics and Computer-Integrated Manufacturing*, 17(3):255–267, 2001.
- [7] Lauri Jokinen, Ville Vainio, and Antti Pulkkinen. Engineering Change Management Data Analysis from the Perspective of Information Quality. *Procedia Manufacturing*, 11:1626–1633, 2017.
- [8] Michael Lang and Jim Duggan. A tool to support collaborative software requirements management. *Requirements Engineering*, 6(3):161–172, 2001.
- [9] Domia Lloyd, Ramadan Moawad, and Mona Kadry. A supporting tool for requirements change management in distributed agile development. *Future Computing and Informatics Journal*, 2(1):1–9, 2017.
- [10] Jihen Maâzoun, Nadia Bouassida, and Hanène Ben-Abdallah. Change impact analysis for software product lines. *Journal of King Saud University-Computer and Information Sciences*, 28(4):364–380, 2016.
- [11] Nasir Mehmood Minhas, Atika Zulfiqar, et al. An improved framework for requirement change management in global software development. *Journal of Software Engineering and Applications*, 7(09):779, 2014.
- [12] Kannan Mohan, Peng Xu, Lan Cao, and Balasubramaniam Ramesh. Improving change management in software development: Integrating traceability and software configuration management. *Decision Support Systems*, 45(4):922–936, 2008.
- [13] David Raffo, Robert Ferguson, Siri-on Setamanit, and Bhuricha Sethanandha. Evaluating the impact of requirements analysis tools using simulation. *Software Process: Improvement and Practice*, 13(1):63–73, 2008.
- [14] Hossein Saiedian, Andrew Kannenberg, and Serhiy Morozov. A streamlined, cost-effective database approach to manage requirements traceability. *Software Quality Journal*, 21(1):23–38, 2013.
- [15] Günther Schuh, Henrique Rozenfeld, Dirk Assmus, and Eduardo Zancul. Process oriented framework to support PLM implementation. *Computers in industry*, 59(2-3):210–218, 2008.
- [16] Maria Grazia Violante and Enrico Vezzetti. A methodology for supporting requirement management tools (RMt) design in the PLM scenario: An user-based strategy. *Computers in industry*, 65(7):1065–1075, 2014.