Proc. of the Second Intl. Conf. on Advances in Computing and Networking — ACN 2013

ISBN: 978-981-07-5939-1

Software Changes:

Related Software Artifacts and their Relationships

Mohammad Hossein Abolghasem Zadeh
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia

Pourya Nikfard
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia

Mohd Nazri Kama
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia

Suhaimi bin Ibrahim
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia

***Abstract*—It is almost impossible to obtain and understand all software requirements at early stages of Software Development Life Cycle without changes. Technical requirement evolves after software specification phase by client organization or project team members frequently. This paper classified the software changes problems and list all related software artifacts to perform the requested software change. The relationships of various change software artifacts during applying change process are discussed in detail.**

***Keywords*—Software Changes, Change Software Artifacts, Requirement Change Management, Impact Analysis, Software Configuration Management, Traceability.**

I. Introduction

Software systems have evolved importantly over the past decades, and hence software modules have become the critical elements of any services or systems [1]. Developments of software systems are dynamic processes and because of this nature, the technical requirements may often change not only in maintenance phase but during Software Development Life Cycle (SDLC) processes as well [2-3]. It is almost impossible to obtain and understand software requirements specifications, designing and implementing the problem solutions without software changes. Especially, for complex and large software systems, it is difficult to understand all customers' requirements at early stage of SDLC phases [4].

Changes on specified software requirements may happen due to varied reasons; such as, changing the stakeholders' business needs and technically unfeasibility of initiated requirements by customers [4]. Even a little change in one requirement can bring ripple effects in other client requirements, if organization does not utilize a proper process for managing its requirements changes. The ripple effects eventually lead to inconsistency in requirements, customer dissatisfaction, and project failure [4-5]. Thus, the necessity of Requirements Change Management (RCM) process, which is a requirement management practice [6], is highlighted for software organizations.

Requirement management processes manage and handle all requirements development activities including eliciting, specifying, analyzing, validating and verifying requirements and also control the requirement changes [6-7]. The purposes of RCM process are managing all the related activities to perform a change in software projects and consequently helping the project completion [8]. Actually, the act of software requirement changes can be assumed as a mini cycle where a change in specific requirements needs some supplementary analysis, a little change to software design solution, adapting the project implementation to new design, and updating the test cases for regression testing [3]. Managing the change mini cycle indirectly leads to successful project completion, since unsuccessfully in changing of requirements is a highlighted reason of project failure.

Several approaches, models and frameworks proposed in literature for managing the requirement changes of software system organizations. A survey was conducted on ten present RCM approaches to compare them on the basis of activity, artifact and role [9]. The results reveal that there are extremely variation among descriptions, and also lack of a clear relationship between activity, artifact and actor. In this paper, we firstly focus on classification of software changes and special RCM techniques. Finally, all the related and affected software artifacts to perform a change mini cycle are listed, and how they are related to each other in RCM process are discussed.

II. Review of Literature

In accordance with CMMI that is a capability maturity model, all requirement changes must be sorted in a database, the impact of change need to be evaluated by existing requirements, and requirement changes effort and rationale for changes should be documented in a change history [6]. A

Proc. of the Second Intl. Conf. on Advances in Computing and Networking — ACN2013

ISBN: 978-981-07-5939-1

survey was conducted by Saffena Ramzan and Naveed Ikram for ten present RCM process models on the basis of activity, artifact and role. They compared the different process models based on 34 activities, 11 artifacts and 6 actors. The result shows that (1) there are various differences between descriptions, and (2) the relationship among activities, artifacts and roles are not clear [9].

The idea of lightweight requirement management was proposed by Delgadillo and Gotel [10]. Their model used the concept of Story-Wall which is composed of story cards. Any story card has a simple history which is useful to obtain the lightweight traceability of changes. Xiang Luo and Koushik Kar proposed an algorithmic way for improving the change management process of IT organizations [11]. They have tried to minimize the time of change implementation, and their approach was based on three steps such as, Change Requirement Determination, Change Approval, and Perform Change.

Mahmood Niazi and its research group introduced a change management model to support the requirement changes in software organizations based on five origin steps; such as, request, validate, implement, verify and update [2]. They mentioned that this model is compatible with level 2 of CMMI, but they did not evaluate the development of RCM process model. Kannan Mohan, Peng Xu, Lan Cao, and Balasubramaniam Ramesh proposed a RCM model [12]. They tried to improve the change management of software requirement changes by integrating the software traceability and software configuration management to their RCM model.

Muhammad Wasim Bhatti with a group of five persons produced a methodology for managing the requirement changes of software projects [4]. They determined stakeholders and product works of different step of RCM model including “Initiation of Change Management Process”, “Receiving of Change Request”, “Evaluations of Change Request”, “Approval or Disapproval of Change Request”, “Implementation of Approved Change Request”, and “Configuration of Change Request”. M. W. Bhatti and I.A. Manarvi surveyed ten items of RCM processes like “acceptance of customer’s request for requirement changes” and “impact of requirement changes” [13]. The results of analysis reveal that these items can provide a valid tool for measuring the maturity of the RCM process.

III. Software Changes

Customer requirements are documented by software team members according to customer’s descriptions at earliest stage of SDLC. The initiated technical requirement may often change during or after the software development phases due to various reasons [2]. The causes of software changes can be categorized in external factors and internal factors. The external factor of software changes are requested by customer, while the internal one usually arises from software project teams [14]. There are various reasons for customer’s pressures to change his initially defined requirements [4]; such as those which are listed below:

- Changing the stakeholders’ business needs in client organization.
- Misunderstanding of the customer’s descriptions by analysts.
- Creating the new requirements by client organization.
- Changes depending on project nature.
- Incompletion of customer’s needs after software specification phase.

Software team members may request for changing technical requirements, as well [14]. These changes can be requested based on some software development problems [4] including the following items:

- Poor understanding of client business needs.
- Technically unfeasibility of initiated requirements by customers.
- Requirements exceed the project’s scope boundaries.

Performing the software changes and managing the change processes are two related significant activities, since changing to software requirement is an emphasized reason to project failure. The change problem understanding and analysis is a most time-consuming activity in change process [15], and the biggest challenge of RCM process is to keep all system parts consistent to one another simultaneously [16]. Specific techniques are used to facilitate change management and support change process like, impact analysis techniques that analyze the impacts of changes in other parts of system [17], change request tracking for supporting the management of change request, and reverse engineering techniques to derive descriptions of higher level from presentations of lower level [16].

IV. Change Software Artifacts

Performing the software changes on projects are completely complex tasks for software engineers, because applying a specific change is related to different software artifacts. Hence, project managers assign the experienced person as change manager and change developers. As figure 1 (Change Software Artifacts) shows, software traceability, requirement engineering, software configuration management, requirement change management, and impact analysis are the related and affected software artifacts to perform the software changes.

Change processes start when a customer or a project team member request for modification in one or more technical requirements. At this time, the needs of initial analysis for verifying change request, and impact analysis to find other affected requirements by performing the requested change, are created. The initial analysis shows that some of the requested changes are new requirements and must be referred to requirement development, which is a requirement engineering process. Impact analysis techniques use software traceability to clarify related links among various requirements. The new

Proc. of the Second Intl. Conf. on Advances in Computing and Networking — ACN 2013

ISBN: 978-981-07-5939-1

software versions that are software configuration activities are produced after applying change in software projects.

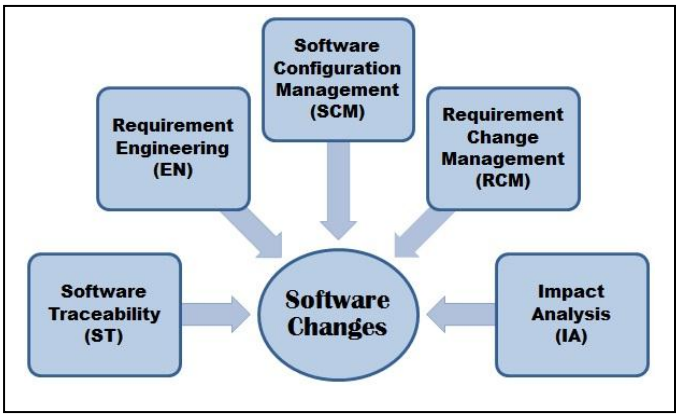

Figure 1. Change Software Artifacts

The different abstract level of affected software artifacts and their relationships to each other, are demonstrated in Figure 2 (Relationships of Change Software Artifacts). The explanations of listed software artifacts are expressed in detail as follows:

A. *Software Traceability*

Software requirement traceability emphasizes on two aspects of software artifact's relationships including (1) the capacity for following up the requirement's life throughout its elicitation, specification ,analysis, development, validation and verification [18], and (2) the ability to describe the association links among related requirements and their requirement's life [6]. The software traceability has two dimensions that are vertical traceability and horizontal traceability. The vertical traceability refers to dependences among different software artifacts at the same SDLC phase, while the horizontal traceability refers to relationships of different software artifacts across various software phases.

Change developers deal with both vertical traceability and horizontal traceability for applying requested change, since they must maintain bidirectional traceability across software artifacts during change mini cycle. The dashed line between different SDLC phases and traceability in Figure 2 (Relationships of Change Software Artifacts) shows both needed type dimensions for applying software changes among different SDLC phases.

B. *Requirement Engineering*

Requirement engineering is a branch of software engineering that cover five core activities include eliciting, analyzing, communicating, agreeing, and evolving requirements [7]. These activities are classified in two overall process areas including requirement development and requirement management. Performing the changes in software products are concerned with requirement management process area. According to CMMI [6], the requirement management has five specific practices that are shown in Figure 2 (Relationships of Change Software Artifacts). These practices are "understand requirements", "obtain commitment to requirements", "manage requirement changes", "maintain bidirectional traceability of requirements", and "ensure alignment between project work and requirements".

The third and fourth specific practices are used for establishing the connections of requirement engineering with other change software artifacts. The "manage requirement changes" practice refers to the requirement engineering activities to RCM processes, and the "maintain bidirectional traceability of requirements" practice highlights the association of requirement engineering activities and software traceability artifact.

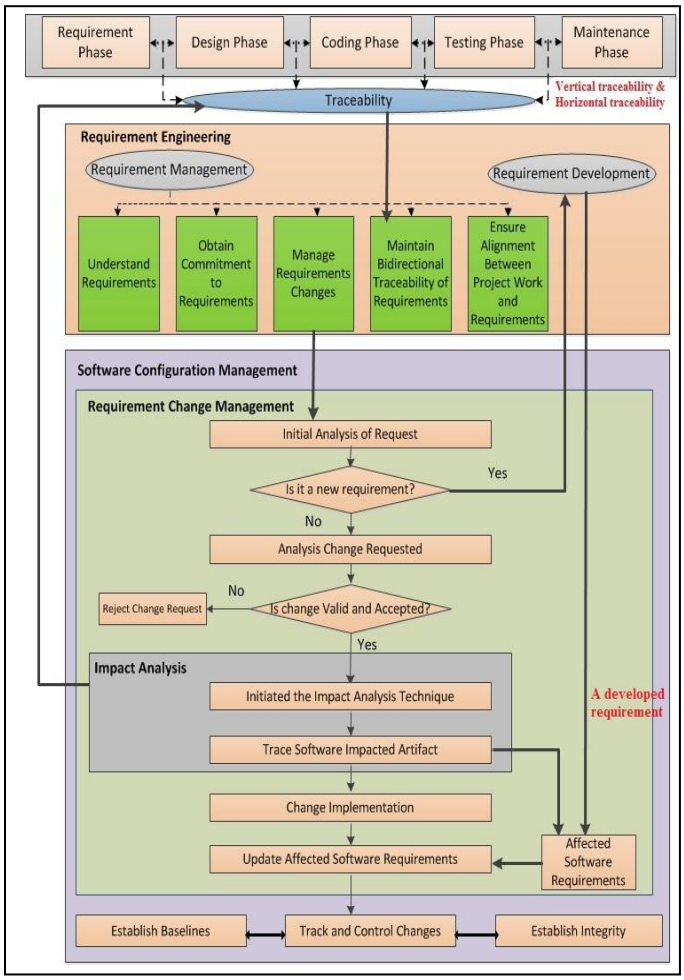

Figure 2. Relationships of Change Software Artifacts

C. *Software Configuration Management*

The aim of software configuration management is to set up and maintain integrity for software products [6]. As it is shown in Figure 2 (Relationships of Change Software Artifacts), the software configuration activities embrace the RCM and impact analysis artifacts. According to CMMI the configuration management process has three specific goals including "establish baselines", "track and control changes", and "establish integrity".

ISBN: 978-981-07-5939-1

The second goal "track and control changes" emphasize the connection of software configuration management and RCM artifacts. Track of requested changes and control of change configuration are two configuration activities that are related to perform the software changes.

D. *Requirement Change Management*

The most stressed software area for applying the software changes is RCM process. As Figure 2 (Relationships of Change Software Artifacts) revealed, this artifact is related to three other software artifacts including requirement engineering, software configuration management and impact analysis. In some cases, the initial analysis of requested change specifies that it is a request for new requirement. Therefore, change analysts refer the new requirement to requirement development team. Once the implementation of new requirement is finished by requirement developers, some of requirements need to be updated for maintaining consistency.

In the case that the initial analysis determines the request as a software change request, the change analysts focus on validity of requested change. Sometimes the requested change is rejected due to various reasons like technically unfeasibility of change implementation. The valid requests are referred to impact analysts to list the affected requirements. After change implementation, the affected software requirement of software changes need to be updated for requirement consistency.

E. *Impact Analysis*

The purpose of impact analysis or change impact analysis is to manage the problem that happens during the apply of software changes. These problems can create unintended result, and costly and time-consuming effects [17]. As Figure 2 (Relationships of Change Software Artifacts) indicated, this software artifact is allocated inside the RCM process, and is related to RCM software traceability artifacts as well.

When change analysts ascertain the request as a valid change request, the impact analysts initiate the impact analysis techniques like cost and time estimations. The significant task of impact analysts are to list the affected requirements by performing specified change to software project. This list is going to be used by change developers to update other requirement for maintaining the consistency.

V. Conclusion and Future work

Software changes and RCM of projects can lead to troublesome and expensive errors in requirement management. Software changes that are requested by client organization may occur due to modification in stakeholders' business needs. Besides, software changes that are requested by project team members may happen because of technically unfeasibility of initiated requirements. This paper has listed change software artifacts and their relationships to each other, while applying change mini cycle process.

This helps the change manager and change developers in software organizations to have excellent understanding of change process activities, and also aids the software change researchers to observe clearly all affected software artifacts of software changes. Our future research is to propose a RCM framework in order to support requirement changes in software organization.

Acknowledgment

The authors would like to thank Lab of Advanced Informatics School for the helps offered, and all members of Lab for their useful discussions that guide for this research. The financial of this project is supported by Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under Vot No: 00K01.

References

[1] Pekka Abrahamsson, Jurgen Munch and Pasi Kuvaja, "Perspectives on Global Software Development: Special Issue on PROFES 2007", *Softw. Process Improve. Pract.* 2008; 13: 213– 215 Published online 3 March 2008 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/spip.377.

[2] Mahmood Niazi, Charles Hickman, Rashid Ahmad and Muhammad Ali Babar," A Model for Requirements Change Management: Implementation of CMMI Level 2 Specific Practice", A. Jedlitschka and O. Salo (Eds.): PROFES 2008, LNCS 5089, pp. 143 – 157, 2008. © Springer-Verlag Berlin Heidelberg 2008.

[3] S. M. Ghosh, H. R. Sharma and V. Mohabay, " Analysis and Modeling of Change Management Process Model", International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011.

[4] Muhammad Wasim Bhatti, Farah Hayat, Nadeem Ehsan, Sohail Ahmed, Azam Ishaque, Ebtisam Mirza, "A Methodology to Manage the Changing Requirements of a Software Project", 978-1-4244-7818-7/10/$26.00_c 2010 IEEE.

[5] IAN SOMMERVILLE and JANE RANSOM, "An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement", ACM Transactions on Software Engineering and Methodology, Vol. 14, No. 1, January 2005, Pages 85–117.

[6] CMMI Product Team, "CMMI® for Development, Version 1.3", CMU/SEI-2010-TR-033 ESC-TR-2010-033.

[7] Bashar Nuseibeh and Steve Easter Brook, "Requirement Engineering: A RoadMap", Future of Software Engineering Limerick Ireland, Copyright ACM 2000 1-58113-253-0/00/6

[8] Khaled El Emam, Dirk Höltje, Nazim H. Madhavji, "Causal Analysis of the Requirements Change Process for a Large System", International Software Engineering Research Network Technical Report ISERN-97-23

[9] Saffena Ramzan and Naveed Ikram, "Requirement Change Management Process Models: Activities, Artifacts and Roles", 1-4244-0794-X/06/$20.00 ©2006 IEEE

[10] Lorena Delgadillo and Orlena Gotel, "Story-Wall: A Concept for Lightweight Requirements Management", 1090-705X/07 $25.00 © 2007 IEEE, DOI 10.1109/RE.2007.41

[11] Xiang Luo, Koushik Kar, Sambit Sahu, Prashant Pradhan and Anees Shaikh, "On Improving Change Management Process for Enterprise IT Services", 2008 IEEE International Conference on Services Computing, 978-0-7695-3283-7/08 $25.00 © 2008 IEEE DOI 10.1109/SCC.2008.136

[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 (2008) 922–936, 2008 Elsevier B.V.

[13] M. W. BHATTI and I.A. MANARVI, " MEASUREMENT OF THE MATURITY OF THE SOFTWARE REQUIREMENTS CHANGE MANAGEMENT PROCESS", The Nucleus 48, No. 2 (2011) 107-119

ISBN: 978-981-07-5939-1

[14] Qing Wang, Xufang Lai," Requirements Management for the Incremental Development Model", 0-7695-1287-9101 $17.00 *0* 2001 IEEE

[15] S. M. Ghosh, H. R. Sharma and V. Mohabay, "A Study of Software Change Management Problem", International Journal of Database Theory and Application Vol. 4, No. 3, September, 2011

[16] S. M. Ghosh1, H. R. Sharma1, V. Mohabay, "Software change management – Technological dimension", International Journal of Smart Home Vol. 5, No. 2, April, 2011

[17] Nazri Kama, Tim French and Mark Reynolds, "Design Patterns Consideration in Class Interactions Prediction Development", International Journal of Advanced Science and Technology Vol. 28, March, 2011

[18] Marie-Agnès Peraldi-Frati and Arnaud Albinet, "Requirement traceability in safety critical systems", CARS '2010, April 27, Valencia, Spain, Copyright © 2010 ACM 978-1-60558-915-2/10/04