View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Universiti Teknikal Malaysia Melaka: UTeM Open Journal System

Current Challenges of Requirement Change Management

Hussin Ahmed, Azham Hussain, Fauziah Baharom
School of Computing, Universiti Utara Malaysia, 06010 UUM, Kedah, Malaysia.
azham.h@uum.edu.my

***Abstract*—Nowadays, responding to requirements change in software industry is essential for survival in the competitive market to achieve business objectives. However, it is clearly evident that changing requirements have many problems which causes software failure. This was a great motivation to analyse literature for identifying current challenges of Requirements Change Management (RCM); which in return can improve our ability to make better decisions and resolve changing requirements problems. Major challenges of RCM have been elucidated as reusability, change anticipation, change activity measurement, connectivity with software artifacts and change management automation. Identifying RCM challenges will help to draw a road map for researchers and practioners to find optimal solutions.**

***Index Terms*—Requirement Engineering; Requirement Change Management; Challenges.**

I. INTRODUCTION

Nowadays, business requirements and IT change and evolve from the initiation of the development and then during the whole life cycle of the system [1]. Generally, requirements change results from adding, removing or updating the following components; product, service, stakeholder role, business rule or any constraint which governs the previous elements. There are multiple reasons which in return drive software to accommodate requirements changes. For instance, in the early stages of software development, usually requirements are incomplete owing to an inconspicuous vision of required business objectives and goals [2] [3]. Additional factors drive business to develop new requirements like new government regulations; stock price change, etc., and internal changes, like business volatility, desire to remain competitive, etc. [4]. Furthermore, fixing errors and handling its impacts are considered as a trigger for requirement change. Accordingly, lacking to reflect changes on software hinders customers' satisfaction and blocks continuous progress of software functionality.

In the business world, being able to adapt information system rapidly to changing requirements is critical [5] [6]. This in return reflects on the need of making alignment between ongoing business requirements and software which has a pivotal role to achieve business objectives. It is noteworthy that overall success of the project is extremely affected by requirement changes [7] [8]. In fact, there were many cases where parts of the last product did not fulfill the customer's needs as the required change had not been implemented precisely [9]. In that respect, an improper RCM leads to the complete failure of the system and contributes to be a cause of business loss [10] [11]. Consequently, recognizing current problems and modern challenges of RCM has significant value to develop fully-featured software.

II. LITERATURE REVIEW

RCM is an important asset in requirement engineering and profound understanding of its process is a primary success factor to implement requirements change. RCM is defined as "*a procedure of managing changes in requirements throughout the requirement engineering process and system development*" [7]. Management means "*adding, deleting or updating requirements and fixing the errors*" [12]. The main steps of RCM process are identified as follow: initial change request, prepare change proposal note, evaluate impact of change, decision about change whether to accept or reject and implemet change to system [7]. There are many problems related to RCM which needs to be elucidated. As a result, these problems will be a trigger for depicting challenges to craft prospective solutions.

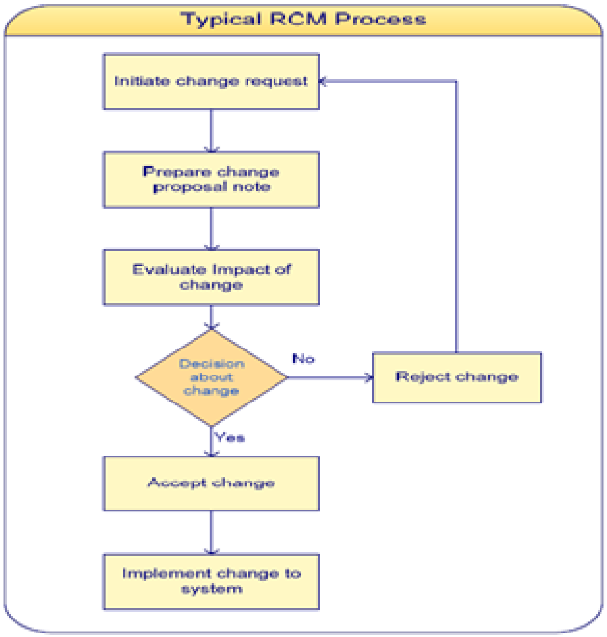

Figure 1: Typical RCM process flow [7]

In literature practitioners and researchers investigated to elucidate major problems regarding requirements change which will be highlighted as follow:

A. Requirement Inadequacy

One of the apparent problems of RCM is inadequacy in describing and identifying requirements change [9] [13]. Undoubtedly, adequacy is a strategic pillar in requirement development and requirement management as well to reasonably satisfy business goals. Clearly, many reasons lead to lack of adequacy concerning requirements changes. Firstly, there are some problems and challenges for software engineers to elicit, analyze, and understand requirements like business dynamicity, new and very different requirements may be needed because there are new stakeholders or new business processes that are previously not considered [14]. Secondly, some requirements cannot be defined easily and even the stakeholders may not aware about them [15]. As a result, this drives systems to be unsuccessful due to lack of important information required for software requirements [16]. Consequently, this raises a challenge to invent a rigorous process to guide all stakeholders for addressing the requirements change in a scientific and formal way.

B. Requirement Ambiguity

As new requirements emerge, it is possible that ambiguous and unclear requirements creep into the process which cause highly complex situation [17]. Ambiguity results from lacking to create a standard way to define requirements meaning explicitly. There is an emphasis emphasized on the importance of dealing with the difficulty in relaying the business requirements change down the IT development line, and handling misinterpretation of the requirements change and the business goals [9]. Resolving requirements ambiguity will promote developing plausible software to address stakeholders' needs precisely.

C. Requirement Traceability

Supplying RCM with a decision making model promotes change management in the system [9] [18] [19]. Decision making occupies a great interest in RCM which results in determining ideal decision about implementing a change. For the purpose of decision making, impact analysis is required to be aware about every consequence in the whole software. Impact analysis is a primary step for comprehensive understanding of change implications [3]. Traceability is the mechanism used to perform impact analysis via creating links between requirements and software artifacts. Artifacts include many objects like requirements, code modules, designs, test cases and any entity that construct behavior and characteristics of the system [20]. There are two apparent problems of traceability: scalability and inability to construct reliable automation [21]. Firstly, the main reason behind scalability is the huge number of trace links between requirements and affected artifacts. For instance, the number of links between requirements and code tends to be very high due to scattering (the implementation of a requirement is distributed over many classes) and tangling (one class contributes to the implementation of many requirements) [22]. Secondly, lack of reliable automation results from that requirements are typically captured informally and cannot easily be reasoned about [21].

D. Time and Cost

Researchers have identified that consuming time and high cost are critical problems of requirement change [8][11][18][23]. It is observable that a main contributing factor for time delay and huge cost is reliance on human factor which has many drawbacks. In practise, if we consider the high rework on a particular change for an application size of 50,000, we might see that the requirement for this change was 1500 lines of source code updates at a moderate estimated rework effort of 25 person-days [24]. It is also found that the costs of adding functionality to a system after it has been put into operation are usually much greater than providing similar functionality when software is originally developed [11]. It is noteworthy that time delaying to implement a change will reduce profits and hamper potential business opportunities in competitive world.

On the basis of this research, the following research question and research objective are formulated.

RQ: What are the current challenges of requirement change management?

RO: *The objective is to depict the current challenges of requirement change management from the existing literature.*

III. Research Methodology

In order to conduct this research, a literature review is analyzed to record potential challenges related to RCM. The main keyword to collect relevant publications was "requirement change" and focus on the last five years is taken into consideration. However, it was necessary to cite old publications which are coherent to current challenges.

IV. Challenges of Requirement Change Management

In literature key findings are identified as follow: reusability, change activity measurement, connectivity with software artifacts, change anticipation and change management automation. In the following sections every challenge will be addressed.

A. Reusability

Representation of the requirements objects and system functionalities must be reusable [25] [26]. Reusability has a major concern in requirement development and its importance is escalated when it comes to RCM. Nowadays rapid evolution of business requirements and necessity for flexibility urge the need of reusing functional and pretested business parts in current IS [25]. Indeed, reusability saves time and maintains a repository for repetitive tasks in dynamic environment for handling new requirements at an accelerated pace. Reusability has to be employed initially during requirement development and then reusable assets will be replicated to RCM. Thus, reusable objects have to support extensibility to modify them according to every change.

B. Change Activity Measurement

Constructing change activity measurement has great value to record change history of every requirement. Measuring change activity is a way to estimate the stability of the requirements and detect opportunities for process improvements that might lead to fewer changes in the future [3]. Furthermore, it helps to communicate the changes and maintain revisions histories [23]. Measuring change activity will help to gain a forward-looking insight for the purpose of optimal decision making and ideal implementation. Change activity measurement will take into consideration change type like add, update or delete, change date, requirement type like functional or non-functional and affected requirements.

C. Connectivity with Software Artefacts

Developing a clear connection between RCM process and other software artifacts is a pressing need; obviously, connection with requirement development and software code which have close tie with each other. Establishing this connection contributes to solve many problems, for instance, changes to requirements and then to code means that maintainers have to do change impact analysis twice: once at the requirements level where there is a need to determine the impact of change and then at the code level; this task is both expensive and error prone [22]. In essence, impact of time and cost is increased with the propagation of changes from one phase to the next in software development life cycle [8]. The major problem stems from that requirements represent high-level customer needs, while source code reflects many implementation and design decisions, hence, relating requirements to code is usually complex [22]. In reality, requirements are typically captured informally and cannot easily be reasoned about [21]. This sheds light of the need to make requirements objects communicate programmatically with software artifacts via formal specification. As formal specification gives clear and concise description about what the system must do [27].

D. Change Anticipation

Anticipating change in RCM is of high value in current competitive world. Some requirements appear in the future and they are unknowable at the time the information system is designed or built [11]. Anticipating market changes and customer requirements and reacting quickly receives a great deal of attention in new web applications framework [28]. It is practical and significant to identify potential changes in elicitation stage rather than allowing them to be delayed into late production stages [29]. There are many reasons trigger the value of business ontology to draw a road map for change anticipation. Firstly, ontology will capture and give a common understanding of the knowledge of the application domain through the definition concepts and concept relationships [30]. Secondly, ontology is one of the best practices to support the software development team throughout the software development life cycle [31]. As a result, business ontology will play a key role for change anticipation in advance.

E. Change Management Automation

One of the crucial problems in modern change management methodologies is too much dependence on human role which does not guarantee reproducibility of the result of a change [32] [33]. In fact, many IT departments and business units are consuming resources and spending most of time on maintaining the current system, and they do not have remaining strength to achieve the tasks of a new subject [34]. Thus, it is desirable to define a software process with sufficient precision so that many of the routine enactment tasks can be automated [35]. In practice, tooling change control process was one of the most successful process improvement initiatives [3]. It is evident that there is a crucial need for agent- oriented model to automate tasks in RCM in order to resolve the current problems for the purpose of saving time, reducing cost and most importantly implementing changes successfully. Software agents act autonomously on behalf of their users to solve increasing number of sophisticated problems [36]. Using agents in RCM will ease the process of handling massive amount of information with great level of accuracy and consistency. Typically, developing agent-oriented software requires agent-oriented software development processes [37]. In literature there is an emphasis on the importance of using agent-based process to improve software process productivity [36] [37] [38] [39] [40] [41]. Therefore, there is a definitive need for crafting a modern model to automate change management process via using agent-oriented approach.

V. DISCUSSION

Modern challenges have great effect on resolving current problems of RCM. Formal specification is a classical challenge which proves to be a corner stone for tackling requirements inadequacy, ambiguity and traceability. Moreover, it will sustain connecting requirements with software artifacts and laying the foundation for change management automation with using agent-oriented approach. A key point of formal specification is to construct innovative business ontology which has many values; drawing a roadmap for all functional requirements, considering potential changes and specifying expected relationships between requirements. Additionally, approaching formal specification in return will enable reusability of requirements assets. At last, automating change management will result in monitoring changes and crafting change activity measurement. Achieving these challenges will sustain to decrease time, reduce cost and leverage productivity in RCM to a higher level.

VI. CONCLUSION

In this work, current problems and challenges of RCM have been identified. A hierarchy for achieving challenges has been determined as follow: creating business ontology, developing formal specification, change management automation via using agent –oriented approach and change activity measurement. Reusability and connectivity of requirements with software artifacts will result from achieving formal specification and change management automation. In future work, systematic literature review will be employed to validate and extend the current findings. Additionally, an agent-oriented approach will be used for change management automation to address the current challenges and fulfill ongoing needs in requirements change.

REFERENCES

[1] M. Asif Khan and H. Zedan, “Alignment strategies and frameworks in co-evolution of business and information technology,” *ICINA 2010 - 2010 Int. Conf. Information, Netw. Autom. Proc.*, vol. 1, 2010.

[2] N. Ali and R. Lai, “Managing Requirements Change in Global Software Development,” no. ii, 2014.

[3] K. Wiegers and J. Beatty, *Software Requirements*, 3rd Edition. Microsoft Press, 2013.

[4] T. Rusinait and D. Kalibatien, “Requirements of Dynamic Business Processes – a Survey,” pp. 0–3, 2015.

[5] S. Schiffner, T. Rothsch, M. Ag, and N. Meyer, “Towards a Subject-Oriented Evolutionary Business Information System,” 2014.

[6] A.-L. A.-L. . Lamprecht, S. . Naujokat, and I. . Schaefer, “Variability Management beyond Feature Models,” *Computer (Long. Beach. Calif).*, vol. 46, no. 11, pp. 48–54, 2013.

[7] H. Naz, Y. H. Motla, S. Asghar, M. A. Abbas, and A. Khatoon, “Effective usage of AI technique for requirement change management practices,” *2013 5th Int. Conf. Comput. Sci. Inf. Technol. CSIT 2013 - Proc.*, pp. 121–125, 2013.

[8] M. Bano, S. Imtiaz, N. Ikram, M. Niazi, and M. Usman, “Causes of requirement change - a systematic literature review,” *16th Int. Conf. Eval. Assess. Softw. Eng. (EASE 2012)*, pp. 22–31, 2012.

[9] S. Jayatilleke and R. Lai, “A method of specifying and classifying requirements change,” *Proc. Aust. Softw. Eng. Conf. ASWEC*, pp. 175–180, 2013.

[10] A. A. Khan, S. Basri, P. D. D. Dominic, and Fazal-E-Amin, “A process model for requirements change management in collocated software development,” *2012 IEEE Symp. E-Learning, E-Management E-Services, IS3e 2012*, pp. 77–82, 2012.

[11] A. Al Kalbani and K. Nguyen, “Designing flexible business information system for modern-day business requirement changes,” *ICSTE 2010 - 2010 2nd Int. Conf. Softw. Technol. Eng. Proc.*, vol. 2, pp. 112–118, 2010.

[12] S. Hassan, U. Qamar, and M. A. Idris, “Purification of Requirement Engineering Model for Rapid Application Development,” 2015.

[13] M. Hanafiah and R. Abdullah, “An evaluation on components of experience based factory model in requirement engineering process: A preliminary study,” *Conf. Proc. - 6th Int. Conf. Inf. Technol. Multimed. UNITEN Cultiv. Creat. Enabling Technol. Through Internet Things, ICIMU 2014*, pp. 308–313, 2015.

[14] B. Suranto, “Software prototypes: Enhancing the quality of requirements engineering process,” *2015 Int. Symp. Technol. Manag. Emerg. Technol.*, pp. 148–153, 2015.

[15] S. K. Saha, M. Selvi, G. Büyükcan, and M. Mohymen, “A systematic review on creativity techniques for requirements engineering,” *2012 Int. Conf. Informatics, Electron. Vision, ICIEV 2012*, pp. 34–39, 2012.

[16] M. W. Grenn, S. Sarkani, and T. Mazzuchi, “A Theory of Information Quality and its Implementation in Systems Engineering,” *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–10, 2014.

[17] C. K. Chang, “Situation Analytics: A Foundation for a New Software Engineering Paradigm,” *Computer (Long. Beach. Calif).*, vol. 49, no. 1, pp. 24–33, 2016.

[18] A. Akhtar, Y. H. Motla, H. Aslam, and M. Jamal, “Role of requirement change in software architecture using Twin Peaks Model,” *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, pp. 174–177, 2014.

[19] S. Ramzan and N. Ikram, “Making decision in requirement change management,” *Proc. 1st Int. Conf. Inf. Commun. Technol. ICICT 2005*, vol. 2005, pp. 309–312, 2005.

[20] J. Cleland-huang, C. Carl K., and C. Mark, “Event-Based Traceability for Managing Evolutionary Change,” vol. 29, no. 9, pp. 796–810, 2003.

[21] B. Burgstaller and A. Egyed, “Understanding where requirements are implemented,” *2010 IEEE Int. Conf. Softw. Maint.*, pp. 1–5, 2010.

[22] E. Ben Charrada, “Updating Requirements from Tests During Maintenance and Evolution,” *Proc. Eighteenth ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, pp. 337–340, 2010.

[23] S. Hussain, N. Ehsan, and S. Nauman, “A Strategic Framework for Requirements Change in Technical Projects : Case Study of a R & D,” *Int. Conf. Comput. Sci. Inf. Technol.*, pp. 354–358, 2010.

[24] B. B. Chua, “Rework requirement changes in software maintenance,” *Proc. - 5th Int. Conf. Softw. Eng. Adv. ICSEA 2010*, pp. 252–258, 2010.

[25] M. Radgui, R. Saidi, and S. Mouline, “Business models alignment with reuse approach,” *Res. Challenges …*, no. URAC 29, 2014.

[26] S. . Banerjee, A. . Sarkar, and N. C. . Debnath, “Quality evaluation of requirement engineering framework: Business object based approach,” *2013 Int. Conf. Comput. Manag. Telecommun. ComManTel 2013*, pp. 374–379, 2013.

[27] M. Ali and T. Saha, “A proposed framework for full automation of software testing process,” *Informatics, Electron. Vis. (ICIEV), …*, pp. 436–440, 2012.

[28] R. R. Harmon, “The FuTure oF Web Apps A Service-Oriented Web Application Framework,” pp. 15–21, 2011.

[29] L. Shi, Q. Wang, and M. Li, “Learning from evolution history to predict future requirement changes,” *2013 21st IEEE Int. Requir. Eng. Conf. RE 2013 - Proc.*, pp. 135–144, 2013.

[30] B. Xu, J. Wu, and H. Cai, “Business process driven ontology discovery method from distributed data environment,” *Proc. - 2011 8th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2011*, vol. 2, pp. 1246–1251, 2011.

[31] M. U. Bokhari and S. T. Siddiqui, “A Comparative Study of Software Requirements Tools for Secure Software Development,” vol. 2, no. 2, 2010.

[32] N. Kushwaha, S. Sahu, and R. K. Tyagi, “Evolving intelligent agents for hospital management system,” *Proc. 2013 3rd IEEE Int. Adv. Comput. Conf. IACC 2013*, pp. 899–907, 2013.

[33] A. Shaban-Nejad and V. Haarslev, “AUTOEXC towards a framework for requirement change management in HEALTHcare software applications,” *Companion to 22Nd ACM SIGPLAN Conf. Object-oriented Program. Syst. Appl. Companion*, pp. 807–808, 2007.

[34] S. Konno, “Service oriented organizational management system for an information systems business,” *Icsssm12*, pp. 738–743, 2012.

[35] P. H. Feiler and W. S. Humphrey, “Software process development and enactment: concepts and definitions,” *Entwicklung und Durchführung von Software-Prozessen Konzepte und Defin.*, no. September, p. 1, 1994.

[36] V. Gaur, A. Soni, and P. Bedi, “An application of multi-person decision-making model for negotiating and prioritizing requirements in agent-oriented paradigm,” *DSDE 2010 - Int. Conf. Data Storage Data Eng.*, pp. 164–168, 2010.

[37] F. Golpayegani, K. Azadbakht, and R. Ramsin, “Towards process lines for agent-oriented requirements engineering,” *IEEE EuroCon 2013*, no. July, pp. 550–557, 2013.

[38] A. Fleischmann, U. Kannengiesser, W. Schmidt, and C. Stary, “Subject-oriented modeling and execution of multi-agent business processes,” *Proc. - 2013 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. IAT 2013*, vol. 2, pp. 138–145, 2013.

[39] C. Lin, K. M. Kavi, F. T. Sheldon, K. M. Daley, and R. K. Abercrombie, “A Methodology to Evaluate Agent Oriented Software Engineering Techniques University of North Texas,” pp. 1–10, 2007.

[40] A. A. Framework, S. Ali, B. Soh, S. Member, T. Torabi, and A. B. Rules, “of Rules Into Business Processes Using,” vol. 2, no. 3, pp. 145–154, 2006.

[41] Z. Hua, Z.-C. Wang, J. Hua, G.-X. Yang, and Y.-W. Liu, “the Framework of Agent-Oriented Programming,” no. August, pp. 18–21, 2005.