Kuwait J. Sci. 46 (1) pp 33-42, 2019

A unified model to manage requirement engineering for global software development

Turki A. AlQarni[1], Rizwan J. Qureshi[2, *]
Faculty of Computing and Information Technology, King Abdulaziz University, 80221 Jeddah 21589, Saudi Arabia
[*]Corresponding author: rmuhammd@kau.edu.sa

Abstract

Change is an unavoidable activity during software development. Demands of customers,C organizational needs and scalability are some reasons requirements may change. Managing these changes on time is crucial in order to produce successful software. However, requirement change management (RCM) is not a straightforward activity, especially in global software development (GSD) due to distributed team structure and geographical challenges. Moreover, no collocated RCM model or organizational structure is currently available to do GSD. This paper proposes a novel unified model to manage requirement engineering for GSD. The research is validated using a survey. The results show that the proposed research will help software companies to solve the changing requirement problems in the GSD environment so that they can complete projects successfully.

Keywords: Global software development; requirment change management; requirement engineering; survey; unified model.

1. Introduction

Global software development (GSD) has gained popularity in the last several years in the software development community. The main reason for the increase in GSD is the monetary factor to reduce cost. Current and steady advancements in computer technology allow even complex projects and frameworks to be developed in different geographical regions (Iqbal *et al*., 2013). These changes lead the requirement change management (RCM) to be a difficult issue to handle. RCM is the procedure of controlling, analyzing, understanding, managing and following changes in requirements. It is important to satisfy the changing needs of a client in the GSD environment (Assawamekin, 2010). Traditional software development models depend upon the assumptions that requirements do not change (Mateen & Amir, 2016). However, the fact is totally the opposite as the customers regularly change requirements throughout the system's development life cycle (SDLC). In the same context, the alteration in requirements show up repeatedly in light of the adjustments in client requests, expanded comprehension of the partners, client association, extended vision, necessity details, and accessibility of innovative arrangements. The main issues with RCM are the cost attached to the process, such as time period, quality, and general cost, especially at the later stages of the SDLC. Therefore, it is evident that requirement change is a major cause for the failure of software projects (Mateen & Amir 2016; Hussain 2016; Ahmad *et al.* 2015). Due to this reason, RCM is not considered a simple procedure. In the case of a distributed environment such as GSD, the RCM process gets more complicated. This raises the need for a collocated model or organizational structure to manage requirements while a software product is developed in a distributed setting. This study first discusses related work. In Section 3, the problem is defined. Section 4 describes a novel solution, and Section 5 focuses on the evaluation of the proposed solution.

2. Literature review

Khan *et al.* (2012) developed a seven-core requirement change management model which could be used in collocated software development (CSD) organizations. It overcame the limitations that are found in the current requirement change management models. However, there was no requirement categorization function implemented in the model, and there was a need for a repository to store information related to changes. The repository can help development teams to keep track of change requests and expected changes throughout the SDLC.

Akhtar *et al.* (2014) introduced a framework that used a Twin Peak Model (TPM) to integrate the impact of the requirement change (RC) in software system. It was used to measure the impact of the RC in the system architecture. However, upcoming requirements in the system development can cause economic issues. Moreover, the use of TPM needs experts to review and evaluate the project from time to time, and the management of a project is very complex due to the amount of documentation required.

Mateen and Amir (2016) proposed a new model to improve the effectiveness of requirement change management process in GSD. The main advantage of their model is that requirement changes can be managed at any phase of SDLC. In addition, it helps to improve the understanding of roles. However, the model was applied in one organization. Hence, it needed more case analyses to get accurate results in GSD. This would have allowed for the development of large-scale software.

However, no unified model organizational structure is currently available for use in managing requirements while a software product is developed in a distributed setting.

Barrett (2011) discussed organization-consulting courses to enable a student to learn entrepreneurship thoroughly and be creative in choos ing a career. The e-Portfolio is a user-friendly and student-centered approach. It helps to personalize the consulting work easily. However, e-Portfolio is difficult to use for those who are not IT literate. Therefore, educators need to make particular changes in the curriculum to improve course offerings and incorporate teaching methods and techniques satisfying the requirements of the courses to encourage and motivate adult learners.

Ziemba and Obłąk (2015) identified critical success factors in change management in information system (IS) projects. The study proposed that the relationship between change management and information system projects is one of the important factors to deliver a successful IS project. Then change management is optimizing the IS projects to be managed timely and effectively. Other than that, change is an unavoidable factor in the implementation of IS projects. The research was limited because it only focused on two case studies in the Polish public organizations. The study needs more data sets to generalize the results. Although limited, the critical success factors in the change management of IS projects and its practical implementation can still be identified (Ziemba and Obłąk, 2015).

Elezi *et al.* (2013) suggested an approach, which predicted the challenges in engineering change management (ECM) from the systems perspective. ECM processes, planning, and techniques were examined. However, the approach mainly focused on quantifiable and easily measurable items, and it lacked goal consistency of the system.

A successful project management and change management framework solution was proposed by Munassar *et.al.* (2013). The study aimed to identify the most critical factors from a change management perspective that would lead to the success of a project. The proposed approach enhanced the sustainability and the productivity of the organizations. However, the critical factors leading to successful completion of projects were not dealt with.

In Pressman (2015), lean methodology was proposed to achieve mass production in the manufacturing industry with high quality. The main principles of lean methodology are the elimination of waste, fast delivery, high quality through standardization, empowerment of the team, and flexibility in decision making. In addition, lean methodology includes the added value from customer perspective and optimizes the value stream through continuous improvements. Over the last few years, the software industry has shown great interest in the application of lean methodology. This is because it offers enormous benefits, including compatibility with agile methodologies, customer satisfaction and high quality products. It also saves time and is economically beneficial.

Xiong *et al.* (2016) proposed a change management approach based on Lean methodology. The aim was to help software industries start the process of change management. The researchers (Xiong *et al*., 2016) stated that the key process areas for enabling successful implementation of change are the readiness for change, leadership and direction, communication planning, organization resources, system controls and behavior of the workforce. The research concluded that it can be ensured only that a software company focused on change after applying all the key process areas. However, no proper measurements were defined that could determine the different grades for the success of management change.

Jeet and Dhir (2011) discussed implementing a software system based on a fuzzy interface approach. They sought to attain and manage maintainability. Many process models have implemented predicting the software quality, but most have failed to attain this goal. Current approaches that seek to model maintainability do not adequately explain disruptive factors and their repercussions. Fuzzy Interference Systems (FIS) use an open-source data set to motivate predictive models of software engineering. FIS can be verified, repeated, refuted and improved. Yet, the research solution is not relevant for all cases. An FIS-based approach needs extensive evaluation which includes hardware for verification and validation. In conclusion, defining exact fuzzy rules, membership functions, and optimization are difficult tasks to accomplish.

Wiboonrat and Kaewsiri (2015) proposed the engineering change process to improve the governance in data center project management (DCPM). DCPM structures the data center design process and governs engineering changes. Its limitations are inadequate analysis in the preliminary stage, improperly defined business objectives and requirements, and incomplete review of the requirements by the quality evaluators.

Minhas *et al.* (2014) suggested a framework for requirement change management in global software development (RCM_GSD. Their goal was to discover challenges faced during a change management requirement. The proposed framework met the necessary processes of change management requirements, and it minimized the impacts of global software development. In a GSD environment, communication can be problematic because of the distance between sites, time zone differences, and the language variations and culture, lowering communication rates. This study's framework solved most of the problems encountered during GSD. However, it required more effort and resources to

improve upon the decision-making phase of RCM. The decision-making depended on experts' feedback, and it could not ensure that decision-making was always valuable.

Andrade *et al.* (2016) explained a process to implement change management. It was used to manage systematic process control changes that were very compatible with the needs of the IT sector. Because the process required no financial investment, it was very feasible and highly recommended. The process was, however, hindered by a human factor like those employees showed unwillingness to readily adapt to the new system who were technologically illiterate.

Kumar and Kumar (2011) discussed a framework to control requirement management problems and challenges which manifest during GSD projects. This is because many GSD projects fail due to poor project management and substandard requirement management (RM) activities. Their framework used a knowledge management system based on an ontology with the goal of managing problems related to inadequate requirements. However, the authors did not clearly define a communication, and the method of dealing with the repository was also not thoroughly discussed.

Hussain (2016) investigated challenges faced by practitioners and the role of collaborative technologies (CT) in carrying out RCM activities. Managing RCM is challenging when the stakeholers are globally distributed. A fit-for-purpose research framwork was used to analyze the challenges of RCM in the GSD and the associated role of CT. The limitations of the process were inadequate research funding, local culture, insufficient piloting of interview questions, a delay in starting interviews, and developing confidence and the right interviewing skills.

Ahmad *et al.* (2015) proposed a model to manage RC at any phase of SDLC. Two phases were presented, namely RM and the impact minimization technique. In the first phase, requirements are categorized. In the second phase, changes are managed in such a way that only the desired portion is affected. However, the proposed model was not implemented on a real software project nor was it validated. Also, further findings were not integrated into the model for clarification.

Chung and Pei (2009) presented a holistic approach that used attributes and linkage to characterize software contents. This approach analyzed the change impact in the software requirements, data and documents. Yet, their approach needed further development in regards to linkage construction and change patterns for it to be applicable in different software production environments. Sinha *et al.* (2006) conducted a research to highlight the importance of requirement management in the GSD environment. Several issues of GSD with respect to requirement management were discussed, such as geographical distances, cultural differences and inadequate tool support for distributed teams. An effective collaboration and a requirement management tool could help to reduce the issues of requirement management of distributed teams.

3. The Problem Definition

Most of the previous and current research on the customary development of software depends on the presumption of fixed requirements gathered during the requirement collection phase. The main problem is that with each requirement adjustment, quality, time period, and the general cost of the product is impacted. For this reason, software developers and researchers (Ahmad *et al.* 2015; Ziemba and Obłąk 2015; Minhas *et al*. 2014) evaluated change in requirements as one of the main causes of software project failure. In light of this fact, RCM is not a simple procedure in collocated software advancement. The issue gets even more complicated in the case of GSD. Mateen and Amir (2016) showed that there is no unified model or organizational structure currently available to manage requirements in distributed settings. This posits the following research questions in the field of RCM"

- How would a unified model for RCM positively impact software development?
- Would a novel unified model be appreciated and adopted by software development organizations at the individual and GSD levels as well?

4. The Proposed Unified Model

A model is proposed which can manage requirements for collocated and distributed teams as shown in figure 1.

Phase 1: Understanding the requirements

First, management teams must meet to ascertain whether or not there is a complete understanding of changes. The requirement change management procedure starts by setting up a comprehension of the asked for changes between various departments in caseof a collocated and distributed teams. The venture requirements taken from a database of requirements must be converted into a graphical structure. This process involves:

- Understanding the existing requirements.
- The impact of the change on any existing requirements.
- Estimation of the extent of change.

Phase 2: Analyzing the changes

At this stage, the required changes will be made in the requirement diagram. These changes might be the expansion, edition, or deletion of requirements. The extension and scope of a changed requirement would be recognized and understood by

Table 1. Summary of the literature review

Source title	Limitation
A process model for requirements change management in collocated software development (Khan *et al.*, 2012).	No requirement is defined, and there is a need for a repository to store expected changes.
Role of requirement change in software architecture using Twin Peaks Model (Akhtar *et al.*, 2014).	Must be expertized to review and evaluate the project from time to time, and the project management is very complex due to the amount of documentation required.
Enhancement in the effectiveness of requirement change management model for global software development (Mateen & Amir, 2016).	The proposed model is not evaluated deeply by more case analyses in order to get accurate results in the GSD and to develop large-scale software. It is difficult to meet user expectations when problems arise.
Creating change and innovation in human resource management courses: Developing a model organizational consulting project as a learning application (Barrett, 2011).	It is difficult to use for those who are not IT literate.
Change management in information systems projects for public organizations in Poland (Ziemba & Oblak, 2015).	Critical factor successes issues of change management in information system projects are not explored deeply.
Engineering change management challenges and management cybernetics (Elezi *et al.*, 2013).	Focuses mainly on quantifiable and easily measurable items. Goal consistency of the system is missing.
Change management framework and its benefit for an effective implementation of IT project (Munassar *et al.*, 2012).	Success factors are not dealt with or taken into consideration in order to prevent any unexpected outcomes.
Successful implementation of change management in projects (Xiong *et al.*, 2016).	It is required to apply the research setting in multiple organizations to check generalize the results.
A model for estimating efforts required to make changes in a software development project (Jeet & Dhir, 2011).	Fuzzy based approach is needed for an extensive evaluation with hardware for verification and validation. It is hard to define exact fuzzy rules, membership functions and optimize fuzzy systems.
Engineering changes to improve the governance in data center project management (Wiboonrat & Kaewsiri, 2015).	Gives an inadequate analysis of the preliminary stage and has poorly defined business objectives and requirements.
An improved framework for requirement change management in global software development (Minhas *et al.*, 2014).	Decision-making depends on expert feedback. Yet, this cannot ensure that decision-making always results in better solutions.
Change management: implementation and benefits of the change control in the information technology environment (Andrade *et al.*, 2016).	People who are not technical literate are unwilling to adapt to new systems.
Study the impact of requirements management characteristics in global software development projects: An ontology-based approach (Kumar & Kumar, 2011).	A method of communication is not defined nor is how to deal with the repository mentioned.
Reflections on requirements change management in global software development: A multiple case study (Hussain, 2016).	Study discusses local culture, delays in starting interviews, and a lack of proper and safe transportation.
Impact minimization of requirements change in software project through requirements classification (Ahmad *et al.*, 2015).	The proposed model is not implemented on a real software project nor is it validated. Any further findings are not integrated into the model for clarification.

individuals in different departments and at sites, in cases of an organization or GSD, respectively. There are two sub-processes involved in this phase:

- Analysis of the module expected to be affected by the new requirement or change in requirement.
- Involvement of experts from different departments or sites (for GSD) in the analysis process.

Phase 3: Settling the adjustments in the requirements

After the experts resolve the extension and scope of changes, change investigation will be performed for the advancement work which was completed in various departments and GSD sites. This could be influenced by potential changes of requirement requested by the client. With the results of progress investigation, a conclusion of the change can be made and recorded in the database of the requirement for future correspondence. Therefore, the steps involved in the third phase are:

- Analysis of the change effects to both the overall project and different departments or sites.
- Comparison of change effect before and after

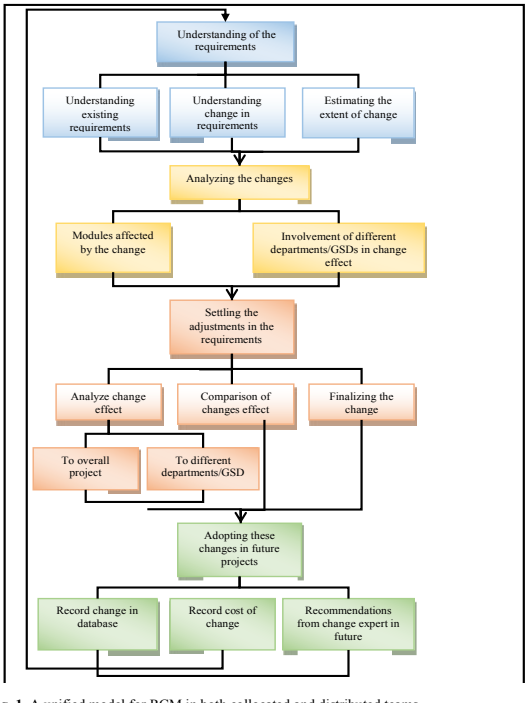

Fig. 1. A unified model for RCM in both collocated and distributed teams

the change.

- Finalization of the changes as desired by the client.

Phase 4: Adapting these changes in future projects

For the last step of the strategy, the recorded database change would be adapted to future projects, if the client requires a similar change. This process involves:

- Recording the change in the requirement database.
- The cost of the change in the requirement database.
- Recommendations from experts in the requirement database.

Software Development Organizations use different models for the process of Requirement Change Management. Earlier models described in the literature have their limitations. Some issues led to increased project costs or to project failure (Hussain 2016; Ahmad *et al.* 2015; Xiong *et al.* 2016). In order to overcome these limitations, we proposed a unified model for RCM both at the organizational and GSD level. This model would be helpful for software developers to manage the requirements while a software product in a distributed setting is developed. The following goals are narrated to check the validity of proposed unified model based on the guidelines of Mateen & Amir (2016).

4.1 Goal 1: Determine the benefits/suitability of a unified models over traditional models

According to the GSD implementation described in Casey & Richardson (2009), our first goal is to see whether the proposed model is better than traditional models adopted by software development organizations. Software development organizations are professional centers and follow some traditionally specified model for the development process. These models are adapted and executed for years. Therefore, our basic goal is to check the suitability of the proposed model in these organizations.

4.2 Goal 2: Introduction of the unified model in Software Development Organizations in GSD

Minhas *et al.* (2014) discovered that the requirement change management process can be improved in software development organizations and GSD if,

- The changes in requirement are understood well.
- The changes are analyzed thoroughly.
- The cost related to changes is estimated correctly.
- The changes are well communicated between departments and sites.
- All the modules are analyzed with respect to the change at an early stage.
- Changes are finalized and recommended by experts.
- Changes are recorded in a database for future use.
- Changes are analyzed with respect to a requirement database.
- Any new changes not presented in the database must be stored in the database with all specification, cost, analysis, and recommendations.

By adopting these guidelines for RCM, it would be helpful to create and introduce a unified model that would be accepted by the software development organizations and GSD.

4.3 Goal 3: Implementation of the unified model

Requirement Change Management is costly and time consuming. To be more specific, the change in the requirements cost more if they are introduced late in the software development process. Our last goal is to implement the proposed model in the existing setup of large-scale software industries and then to see its general impact over software development and specifically on RCM. However, this means implementing the proposed model in a setup where a traditional model has already been implemented. Therefore, it is a two-step process:

- To see the impact of the proposed model on overall software development.
- To see the impact of the proposed model on RCM.

5. Validation

We used a survey to validate our proposed solution. The

survey targeted three categories. The first category was software developers. The second category was system analysts. The third category was IT manager (Assawamekin 2010; Xiong *et al.* 2016; Jeet & Dhir 2011). These aims were considered equally relevant in the survey because the proposed model would be effected by the entire team dealing with RCM. A web-based survey was conducted instead of manual data collection. This method has been shown to reduce human error, save time, decrease cost, and provide a high response rate (Nardi, 2003). The record of software development companies was taken from the Pakistan software export board (PSEB, 2017). Fifteen out of one hundred software development companies voluntarily showed interest in participating in the study. The software companies have business in several countries and regions, including Australia, the United Kingdom, the United States, and Asia. Ten out of fifteen software companies were randomly selected to collect the data. Two hundred and fifty people were randomly contacted from the ten selected companies to participate in the study via email. The response rate was fifty percent (50%) of the population. Eighty percent of the respondents represented the development category, eight percent of the professionals represented the system analyst category, and twelve percent of the participants represent the IT manager category. Furthermore, each goal was evaluated with ten questions, and the questionnaire was evaluated on a five-point likert scale. (The questionnaire is shown in the Appendix). According to McLeod (2008), this scale is used to measure attitudes or opinions of participants by asking them fixed statements about the topic. Table 2 illustrates the likert scale that is used to evaluate the questionnaire.

Table 2. Likert scale used to evaluate the questionnaire

Very High	High	Nominal	Very Low	Low
1	2	3	4	5

Table 3. Cumulative statistical analysis of goal 1

Q. No.	Very High	High	Nominal	Low	Very Low
1	33.33	46.67	10	6.67	3.33
2	6.67	13.33	33.33	30	16.67
3	50	26.67	13.33	3.33	6.67
4	36.67	50	6.67	3.33	3.33
5	40	30	13.33	10	6.67
6	33.33	53.33	3.33	6.67	3.33
7	26.67	23.33	26.67	13.33	10
8	33.33	40	10	6.67	10
9	30	46.67	16.67	3.33	3.33
10	46.67	36.67	6.67	3.33	6.67
Avg.	33.6	36.6	14	8.6	7

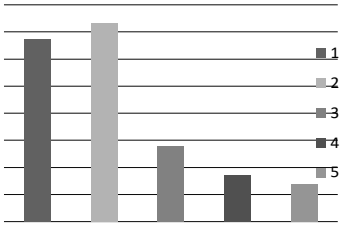

Fig. 2. Cumulative statistical analysis of goal 1

5.1 Cumulative analysis of Goal 1

Table 3 shows that eighty percent (80%) of the participants faced problems with RCM in the GSD environment. Only twenty percent (20%) of professionals are comfortable with RCM in the GSD environment. In addition, twenty percent (20%) of the participants think that existing traditional models of RCM are sufficient enough to manage change in the GSD environment. More than three-quarters of participants (76%) believe that there is some effort being made to design a unified model in the GSD environment. An overwhelming majority of participants (86%) support the suitability of a unified model. The survey also shows that

Table 4. Cumulative statistical analysis of goal 2

Q. No.	Very High	High	Nominal	Low	Very Low
1	33.33	40	16.67	6.67	3.33
2	23.33	56.67	6.67	3.33	10
3	56.67	30	6.67	3.33	3.33
4	33.33	53.33	3.33	3.33	6.67
5	6.67	3.33	3.33	36.67	50
6	23.33	66.67	3.33	3.33	3.33
7	33.33	30	20	10	6.67
8	43.33	36.67	13.33	3.33	3.33
9	16.67	43.33	16.67	13.33	10
10	26.67	56.67	6.67	6.67	3.33
Avg.	29.6	42.6	9.6	9	10

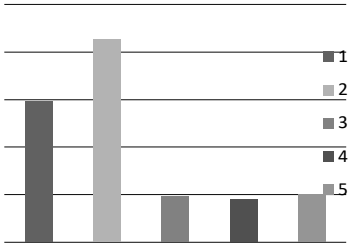

Fig. 3. Cumulative statistical analysis of goal 2

seventy percent (70%) of the participants believe that a unified model is acceptable to software development organizations in comparison to traditional techniques. Eighty six percent (86%) of respondents agreed when it is inquired that the unified model in the organization would provide better feedback and experience related to RCM processes. Half of the participants (50%) reported that software development organizations should establish the efforts to unified model development. Most of the participants (73%) agreed that software decision makers should learn about the importance of implementing unified model for RCM in the GSD. As for as user expectations are concerned, seventy six percent (76%) of the respondents believe that the proposed model is applicable which would overcome problems arising when meeting user expectations. Eighty three percent (83%) participants

think that by implementing the unified model in the GSD, the level of resource usage will improve as shown in figure 2.

5.2 Cumulative analysis of Goal 2

Table 4 and figure 3 show the survey results regarding goal 2.

First, seventy three percent (73%) of respondents find the "understanding requirements" phase suitable for RCM in both organizational and GSD. Second, eighty percent (80%) of the participants believe that "settling adjustments in requirements" phase important for the proposed unified model. Third, eighty six percent (86%) people think that "analyzing the changes" phase is suitable for the unified model for RCM in the GSD. Fourth, eighty six percent (86%) participants are in the favor of that a repository has a beneficial effect when adopting these changes in future projects in the GSD. Fifth, only ten percent (10%) of the respondents believe that when introducing the unified model to the software development organizations, issues or challenges arise. In the case of unified model implementation failure, ninety percent (90%) of the participants think that it results from a lack of skills or workforce resources. Sixth, sixty three percent (63%) of respondents think that the introduction of the proposed model must be undertaken phase by phase. Furthermore, eighty percent (80%) of participants find that software development organizations react when moving from a distributed setting to unified model development. In addition, sixty percent (60%) of respondents think that a software company will be able to introduce the proposed model successfully. Finally, eighty three percent (84%) of participants believe that the software industry has successfully mitigated the effects of existing model implementation issues of the new model.

Table 5. Cumulative statistical analysis of goal 3

Q. No.	Very High	High	Nominal	Low	Very Low
1	26.67	26.67	23.33	13.33	10
2	36.67	56.67	3.33	3.33	0
3	46.67	23.33	20	6.67	3.33
4	26.67	50	13.33	3.33	6.67
5	16.67	63.33	20	0	0
6	50	33.33	3.33	10	3.33
7	40	46.67	3.33	3.33	6.67
8	0	16.67	60	16.67	6.67
9	16.67	50	20	3.33	10
10	53.33	26.67	13.33	3.33	3.33
Avg.	31.3	39.3	18	6.3	5

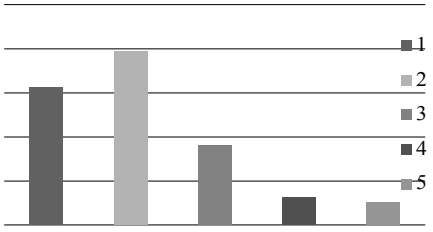

Fig. 4. Cumulative statistical analysis of goal 3

5.3 Cumulative analysis of Goal 3

The results of goal 3 are shown in Table 5 and figure 4. First, fifty three percent (53%) participants believe that software companies have enough time to implement the unified model. The respondents also believe that software development organizations also have sufficient funds to implement a unified model (93.34%). Second, seventy percent (70%) think that a unified model can be integrated into an existing setup on software industries. Third, seventy six percent (76%) believe that the software companies could minimize the risk of failure by integrating the proposed unified model with traditional techniques. Fourth, eighty percent (80%) of respondents think that integrating the unified model within an existing setup will improve the RCM process in the GSD. In the same context, eighty three percent (83%) people find that the proposed model will influence overall software development. Fifth, eighty six percent (86%) participants believe that the proposed model will have a significant effect on RCM. However, only sixteen percent (16%) believe that software developers have the necessary skills and experience to implement the proposed unified model. More than half of the survey respondents (66.67%) believe that the proposed model could be adopted within the requirements and changing technology. Finally, eighty percent (80%) of the participants are satisfied with the implementation of the unified model in the GSD and organizational development.

6. Conclusion

RCM is an unavoidable action in all phases of software development. A client may require a change at any time due to changes in corporate structure, the market, altered project goals, or due to the introduction of new technology. The impact of desired changes is advantageous for the client if changes are tended to properly and at the optimal time. Otherwise, changes may cause delays, extra costs, and implementation failure.

In this paper, a four-phase strategy for RCM in both organizational and global software development was proposed. This strategy could improve the change management needed to complete a project successfully. To validate our claim, we performed a survey. The results of the survey show a strong improvement in the process of RCM due to the proposed strategy. Eighty percent of the respondents show their confidence on the proposal of unified model. Eighty four percent (84%) of the respondents are of the opinion that the proposed unified model could be failed to implement in software companies due to lack of developers' skills and experience.

Future research should involve implementing the proposed model in industrial case studies in order to check its effectiveness with the requirement changes involved. In addition, interviews with respondents would be a valuable data gathering tool because it would increase our knowledge on their opinions about the suitability of

proposed model.

References

Ahmad, Z., Hussain, M., Rehman, A., Qamar, U. & Afzal, M. (2015). Impact minimization of requirements change in software project through requirements classification. Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (IMCOM 2015). Ottawa, Canada.

Akhtar, A., Motla, Y.H., Aslam, H. & Jamal, M., (2014). Role of requirement change in software architecture using Twin Peaks Model. Proceedings of the 5th International Conference on Software Engineering and Service Science. Beijing, China.

Andrade, P.R.M.D., Albuquerque, A.B. & Teófilo, W. (2016). Change management: implementation and benefits of the change control in the information technology environment. International Journal of Advanced Information Technology (IJAIT), 6(1): 23-33.

Assawamekin, N. (2010). An ontology-based approach for multi-perspective requirements traceability between analysis models. Proceedings of the 9th International Conference on Computer and Information Science (ICIS). Kaminoyama, Japan.

Barrett, G. (2011). creating change and innovation in human resource management courses: developing a model organizational consulting project as a learning application. Proceedings of the International Conference on Business Management and Electronic Information. Guangzhou, China.

Casey, V. & Richardson, I. (2009). Implementation of global software development: A structured approach. Software Process Improvement and Practice, 14(5): 247-262.

Chung Y. & Pei C. (2009). A holistic approach to managing software change impact. The Journal of Systems and Software 82(12): 2051–2067.

Elezi, F., Maier, T.G. & Lindemann, U. (2013). Engineering change management challenges and management cybernetics. Proceedings of the 7th Annual IEEE International Systems Conference (SysCon). FL, USA.

Hussain, W. (2016). Reflections on requirements change management in global software development: A multiple case study. Proceedings of the 11th International Conference on Global Software Engineering Workshops (ICGSEW). California, USA.

Iqbal, J., Ahmad, R. & Noor, A. (2013). Framework to improve the requirements engineering process for software development outsourcing. Proceedings of the 22nd Australian Software Engineering Conference. Melbourne, Australia.

Jeet, K. & Dhir, R. (2011). A model for estimating efforts required to make changes in a software development project. Proceedings of the International Conference on Advances in Computing and Artificial Intelligence–ACAI '11. Rajpura/Punjab, India.

Khan, A., Basri, S., Dominic, P.D.D. & Amin, F. (2012). A process model for requirements change management in collocated software development. Proceedings of the IEEE Symposium on E-Learning, E-Management, and E-Services. Kuala Lumpur, Malaysia.

Kumar S.A. & Kumar, T.A. (2011). Study the impact of requirements management characteristics in global software development projects: An ontology-based approach. International Journal of Software Engineering & Applications, 2(4): 107-125.

Mateen, A. & Amir, H. (2016). Enhancement in the effectiveness of requirement change management model for global software development. Journal of Science International Lahore, 28(2): 1161-1164.

McLeod, S.A. (2008). Likert scale. Retrieved from www.simplypsychology.org /likert-scale.html.

Minhas, N.M., Qurat-ul-Ain, Zafar-ul-Islam & Zulfiqar, A. (2014). An improved framework for requirement change management in global software development. Journal of Software Engineering and Applications, 7(9): 779-790.

Munassar, F., Ghanim, A. & Dahlan, A. R.A. (2013). Change management and its contribution to the success of IT project implementation. International Journal of Information and Communication Technology Research, 3(4): 134-140.

Nardi P. (2003). Doing survey research: A guide to quantitative methods. Pearson Education, New York. \Pp. 98. Pakistan Software Export Board. (2017). Retrieved from http://www.pseb.org.pk/.

Pressman, R. S. (2015). Software engineering. McGraw Hill, New York. Pp. 99. Sinha, V., Sengupta, B., and Chandra, S. (2006). Enabling collaboration in distributed requirements management. IEEE Software, 23(5): 52-61.

Wiboonrat, M. & Kaewsiri, U. (2015). Engineering changes to improve the governance in data center project management. Proceedings of the 9th Annual IEEE Systems Conference (SysCon). Vancouver, Canada.

Xiong, G., Nyberg, T.R., Zhao, A. & Xiong, G. (2016). Change management on an improvement project for success. Proceedings of the International Conference on Service Operations and Logistics, and Informatics (SOLI). Beijing, China.

Ziemba, E. & Obłąk, I. (2015). Change management in information systems projects for public organizations in Poland. Interdisciplinary Journal of Information, Knowledge, and Management, 10: 47-62.

Submitted: 01-03-2017
Revised: 10-12-2017
Accepted: 25-04-2018

نموذج موحد لإدارة هندسة المتطلبات الخاصة بتطوير البرمجيات العالمية

تركي القرني، * رضوان القرشي

كلية الحاسبات وتقنية المعلومات، جامعة الملك عبد العزيز
80221 جدة 21589، المملكة العربية السعودية

* rmuhammd@kau.edu.sa

الملخص

أثناء تطوير البرمجيات لابد من حدوث تغييرات على المتطلبات اللازمة وذلك للعديد من الأسباب مثل متطلبات العملاء والاحتياجات التنظيمية وقابلية التوسع. فمن الضروري معالجة هذه التغييرات في الوقت المحدد للحصول على برنامج ناجح. وبالرغم من ذلك، فإن إدارة التغييرات لتلك المتطلبات ليس بالأمر الهين خصوصاً في حالة تطوير البرمجيات العالمية بسبب توزع فرق العمل والتحديات الجغرافية. علاوة على ذلك، لا يوجد أي نموذج لمعالجة متطلبات التغيير أو الهيكل التنظيمي حالياً من أجل تطوير تلك البرمجيات. يقترح هذا البحث نموذجاً موحداً حديثاً لذلك، وتم اثبات صحتة بواسطة اجراء استقصاء، وكانت النتائج مشجعة. ومن المتوقع أن تساعد الأبحاث المُقترحة شركات البرمجيات على حل مشاكل المتطلبات المتغيرة في مجال تطوير البرمجيات عالمياً لإنجاز المشاريع بنجاح.