CONTENTS

Chief Patron

Dr. JOSE G. VARGAS-HERNANDEZ

Member of the National System of Researchers, Mexico

Research professor at University Center of Economic and Managerial Sciences,
University of Guadalajara
Director of Mass Media at Ayuntamiento de Cd. Guzman
Ex. director of Centro de Capacitacion y Adiestramiento

Patron

Dr. Mohammad Reza Noruzi

PhD: Public Administration, Public Sector Policy Making Management,
Tarbiat Modarres University, Tehran, Iran
Faculty of Economics and Management, Tarbiat Modarres University, Tehran, Iran
Young Researchers' Club Member, Islamic Azad University, Bonab, Iran

Chief Advisors

Dr. NAGENDRA. S.

Senior Asst. Professor,
Department of MBA, Mangalore Institute of Technology and Engineering, Moodabidri

Dr. SUNIL KUMAR MISHRA

Associate Professor,
Dronacharya College of Engineering, Gurgaon, INDIA

Mr. GARRY TAN WEI HAN

Lecturer and Chairperson (Centre for Business and Management),
Department of Marketing, University Tunku Abdul Rahman, MALAYSIA

MS. R. KAVITHA

Assistant Professor,
Aloysius Institute of Management and Information, Mangalore, INDIA

Dr. A. JUSTIN DIRAVIAM

Assistant Professor,
Dept. of Computer Science and Engineering, Sardar Raja College of Engineering,
Alangulam Tirunelveli, TAMIL NADU, INDIA

Editorial Board

Title

EMPIRICAL AND QUALITATIVE STUDIES BY ANALYZING REQUIREMENT ISSUES

IN GLOBAL SOFTWARE DEVELOPMENT (GSD)

Author(s)

Rabia Sultana

Department of Computer Science,

University of Agriculture, Faisalabad, Pakistan

Fahad Jan (Lecturer)

Department of Computer Science,

University of Agriculture, Faisalabad, Pakistan

Ahmad Mateen (Lecturer)

Department of Computer Science,

University of Agriculture, Faisalabad, Pakistan

Ahmad Adnan

Department of Computer Science,

University of Agriculture, Faisalabad, Pakistan

ABSTRACT:

Global software development has been a phenomenon of growing interest for almost past decade. Requirement engineering plays vital role in global software development projects. In this research work we focused on change management phase of requirement engineering. As we know change is the only thing which is constant. This result in continuously changing the requirements in any software development effort, so effective management of requirement is of vital important. In global software development it is not a straightforward task and it imposes numerous challenges which are unique to this form of development. We suggest a model in this paper for managing requirement change in globally distributed software development context. In this paper qualitative research method is used to explore requirement change management process. It investigated the underlying causes of requirement change in global software development. The research work proposed a model for requirement change management in global software development. Coverage of process elements has an effect on process model and its usage. So this model tries to incorporate the process element like role, activities and artifacts identified in the change management models. The proposed model will use to analyze an organization to see how well it performs requirement change management functions. The ultimate goal of this research work is to promote better understanding of the process of requirement change management in global software development.

Key Words: Global Software Development, Requirement Engineering, Change Management.

INTRODUCTION:

This paper is about the requirement change management in global software development. Global software development is most important because there are lots of advantages by using GSD. But global software development faces many problems such as language and time differences, cultural diversity and knowledge management bring additional challenges to communication. These problems have created specific difficulties in conducting requirement engineering activities. The inclination towards Global Software Development (GSD) is obviously because of its well identified and documented benefits that include cost savings,

access to large multi-skilled workforces, proximity and reduced time to customer market etc. (Conchuir *et.al.,* 2009). The quest for business excellence and competitive advantage compels organizations to look for solutions around the globe (offshore sourcing or off shoring). GSD appears as a feasible alternative in such an environment (Prikladnicki and Audy, 2006).

This study is carried out owing to the need of investigation of RCM processes as pointed out by various other researchers. For example Lam and Shankararaman, (1999) suggests that in software industry the collective guidance for managing requirements change is still weak and there is a need for developing systematic and methodical practices for managing requirements change. Many partial solutions have been offered for the implementation of RE in a GSD environment but they lack process level detail (Lopez *et.al.,* 2009).

Requirements change management process is one of the most collaboration intensive activities and it poses significant difficulties with distributed stakeholders (Sinha *et. al.,* 2006).

There remains a gap in this area to be filled up with more rigorous research on RCM process. GSD is facing more problems with changing requirements and their management as compared to single-site development. Software tends to continuously evolve throughout the development lifecycle which leads to the problem of continuous change about exactly what to build (referred as requirements). Managing and keeping track of this ever changing requirements pool is an arduous task especially in a multi-site development which has to additionally deal with temporal, geographical and cultural difference (Šmite *et.al.,* 2010).

A Process Model is the representation of a process. The model is useful and likely to reduce the impact of the challenges and causes of requirements change such as Culture, Communication, Knowledge Management, Time Difference, Trust etc. (Damian and Zowghi, 2003).

This research work is about the root causes of continuous requirements change and is aimed to better understand and support the process of requirements change management in GSD projects. For this purpose a model is developed for Requirements Change Management (RCM) to be utilized by global software development companies working on customized development of web and desktop application projects. It aims to explore and analyze the root causes of requirements change in distributed development projects. The research also aims to investigate the existing requirements change management process practiced in the case under investigation.

Without a Requirements Engineering (RE) process suitable for GSD especially designed for Requirements Change Management (RCM) it is difficult to avoid the challenges global software development is faced with (Sangwan *et.al.,* 2007).

The addition of empirical evidence as a result of this research work in the area of RCM for distributed development also justifies this research. The objective is to provoke further understanding of the process of requirements change management in a global software development context.

METHDOLOGY:

The research has done in this paper based on the published materials, journals and systematic literature review. Some materials also get from IEEE publications and websites. Our study opens the door of opportunity for the process related problem identification and also process improvement. Case study research is appropriate for investigation of a phenomenon within its real-life context, in order to answer how and why questions, when the investigator has little control over the events. We use a qualitative method to evaluate requirements change problems in software development process in a global software development context. The case under investigation is a real life company and it provides us with the opportunity to study or investigate contemporary event where relevant behaviors were not manipulated, this approach is advocated by (Yin, 2003).

RCM process is vital in the development of software for both collocated as well as distributed development. The existence of such process in an organization is likely to improve the outcome and predictability of the undertaken projects (Ramzan and Ikram 2006).

We develop model for a better understanding of the system to be developed. Models also are useful to support process improvement and process management. A process model itself is composed of various elements such as activities, roles and artifacts. The elements of the process models are the means for understanding what (activities) is to be performed, who is responsible (roles) for performing these activities and what are the outputs (artifacts) resulting from such activities. Our conducted a literature survey of RCM process models on the basis of the elements of a process model namely roles, activities and artifacts. A survey that was conducted suggests

RCM process area is not standardized even in the single site development and there is no consensus over the adoption and use of a single RCM process even in single-site development (Ramzan and Ikram, 2006).

Furthermore the models described in the literature are not detailed enough to be used in the real world by requirements engineers. The authors compared various available models for RCM they are not detailed enough to be used by software engineers. Out of the 34 activities mentioned in these authors (Ramzan and Ikram. 2006) only Leffingwell's model covers 13 activities which is the highest number of activities followed by a single model. This observation is surprising and shows the lack of detail as concluded by them. So that model reveals that there seems to be no consensus between the models used for RCM. In this study data was collected in multiple forms by means of Change Request Forms (CRF) which is the core document that would be used for analysis of the reasons or rationale of the requirements change. CRF has a central role in the process of RCM it is initiated by the person suggesting a requirements change which then was evaluated on the basis of its feasibility and then accepted or rejected by the Change Control Board (CCB).

A single case study design with a two units analysis (i.e. two project) in an industrial setting (GDS) was applied to investigate the causes of requirements change during global software development. This approach is appropriate for the researcher to conduct in-depth investigation the situation of a typical project in the real software industry environment. Data was collected by interview from the project members included was Change Moderator (CM), Quality Assurance Manager (QAM), Team Lead (TL) and Analyst. Change requests data were collected from two projects within GSD organization. A total of 24 change request forms were collected for project A and 12 collected for project B during the eight months of the projects' duration. The purpose of our analysis was to identify and understand the problems relating to changing requirements during the software development process and their underlying causes.

Our analysis was based on descriptive and qualitative methods. Our data analysis process was a combination of both inductive and deductive inferences. Another important artifact that was analyzed that also related to change request was the cumulative CRF log spreadsheet. Every project has a specific Cumulative CRF log sheet which consists of the change request data for that project. This sheet is mainly consists of some basic information contained in the CRF form

and it simply shows the collection of all change requests for a project in one document. It includes information such as CRF number, reason for change, description for change, requested by and change type. The consolidated Change Request Log Sheet was available in the organization and was used for the analysis purposed in our study.

The model was drawn based on the interviews of the people involved and the related documents used in the process. The change management process is triggered by change request from any of the key stakeholders (Analyst, Development team, QA team and the client). The change request may be accepted or rejected based on the analysis and negotiation and may or may not therefore be implemented. Once the change gets implemented it is verified by the Quality Assurance team and if accepted it goes to the client's validation process. When it gets validated by the client the change is closed. In case of non acceptance from the Quality Assurance team of the client the change is reviewed and reported and goes back to the evaluation process. CRF form is the responsibility of a Change Moderator throughout the development life cycle. We outlined our findings below for the four main phases of the change request process.
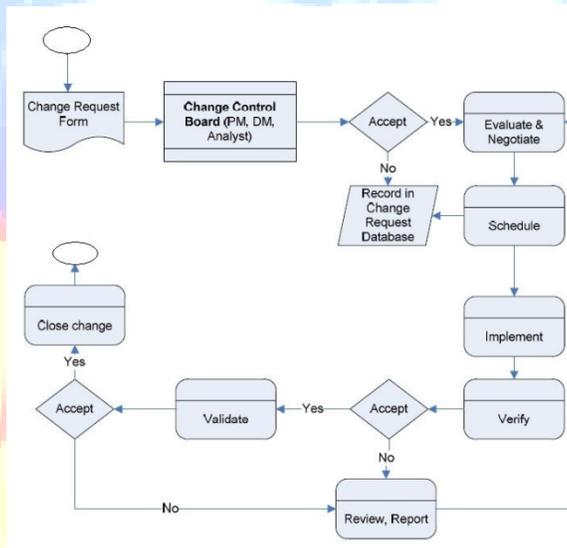
Figure 1: Model of existing Requirement Change Management Process

In change management process CRF is considered as the key document which contains most valuable information regarding the proposed change. Our analysis revealed that quite often the minor changes identified by the development teams are implemented without generating any CRF. When we explored further the Project Manager explained CRFs are meant for major changes which are most often sent by the client. This was an interesting finding, but there is no

A Monthly Double-Blind Peer Reviewed Refereed Open Access International e-Journal - Included in the International Serial Directories
Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A.
International Journal of Management, IT and Engineering
http://www.ijmra.us

11

record of how many 'small' changes were made. Also there is no record of the process of identification, approval, implementation and verification of these changes. We found a lack of integration of the collaboration tool with the RCM process. The documents filled out for requirements change do have specified area to cover this information. Our analysis also revealed that there is no record of how the rejected or deferred changes are handled and stored and at what stage during multiple iterations of the project development they are included back into the project work.

In this case study this did not have very big impact due to the fact that only two development teams on two development sites were involved and there were not many change request forms. Therefore at the end of our analysis we gave following recommendations to the company:

1. Fill all the required Information in the Change Request (CR) forms especially change rationale.

2. Perform detailed impact analysis.

3. Integrate online collaborative information with RCM artifacts.

4. Record technical changes made by development teams even on small scale.

5. Define the process of including deferred or rejected changes in the development process.

6. Develop central database for recording all changes as archived data for evaluation purpose.

This study has proposed RCM model that includes most of the roles, artifacts and activities mentioned and in their proposed model for requirements change management (Imtiaz *et. al.,* 2008).

Description of Model:

The model uses the term R for role and S for the site to show the distribution of work at multiple sites. This model shows R1S1 which means any *Role1* at *Site1* played by a stakeholder role who can initiate change. Similarly *R2S2* means any other key stakeholder role (Role 2) at Site 2. The model is extendable to any number of sites involved in the development process

A Monthly Double-Blind Peer Reviewed Refereed Open Access International e-Journal - Included in the International Serial Directories
Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A.
International Journal of Management, IT and Engineering
http://www.ijmra.us
12

where *RnSn* shows any key stakeholder role *(Rn)* at any Site n. In the proposed model only one client is shown for simplicity. However if there are number of clients involved at multiple locations the terminology similar to multiple roles for different sites can be used; for example *C1L1* which would mean Client1 at Location1 and so on. The requirements change management process is initiated when the need for a change is identified by any of the key stakeholders at any development site which is involved in the project. This change identification activity shown in the model starts when any of the roles at any given site or the client communicates the need for a change. This activity leads to the second activity in which the identified change is discussed by the client and key roles from the relevant site(s) involved in the project. It is important to note that it is possible that only the client and a particular site may get involved it the change is only concerned with a particular area of development carried out at a particular site. However the model is flexible to accommodate the involvement of multiple roles participation. The discussion maybe conducted by means of an online video conference call where the identified change gets formulated defined and understood by all stakeholders involved. If the change need is commonly agreed upon and accepted a formal CRF document is generated in coordination with the Change Moderator (CM). If the change gets rejected in that conference no further action is taken. The Change Moderator is responsible for the process of requirements change management once the change enters the phase of development of the CRF. The Change Moderator finalizes the Change Request Form with the cooperation of the involved development personnel who provide the required detail of modules and artifact that are affected and the determination of effort involved in implementing the proposed change. In the next step CCB evaluates for the feasibility in terms of possibility of implementing the proposed change within the project time, budget and allocated resources. It is possible that the CCB may evaluate different alternatives based on the impact analysis performed for the proposed change and if such information is available. If the CRF gets accepted it is moved to the next phase which is negotiation with client. The negotiation is carried out between the client and CM who has the knowledge of the proposed change as well as the CRF generated and accepted for that change. If the CRF is not accepted, it gets recorded in Requirements Change Database and is also communicated to the Review and Re-evaluation Committee. The committee examines the rejection factors and prepares a report which is forwarded to the Change Moderator. From here this information is updated by the CM into the central repository that is accessible to all the sites involved in the project. When the proposed

change is in negotiation process and it gets accepted by the client it goes to implementation phase. But if the change gets rejected at this stage or it gets a deferred status it is forwarded to the Review and Re-evaluation Committee. The committee analyzes the rejection or deferral factors and prepares a report which is stored in the Requirements Change Database and which is also communicated to the CM. The CM updates the Central Repository with this information and also decides whether to develop a modified CRF and restart the whole process or not. Once the change gets implemented after the negotiation with the client is successful it moves to the stage of verification where QAD verifies the implemented change. If the implemented change gets accepted by QAD it moves to the validation stage. Otherwise it gets reported in the Bug Reporting Tool from where the status of the defect (bug) is visible to the development team, review committee and all development sites.
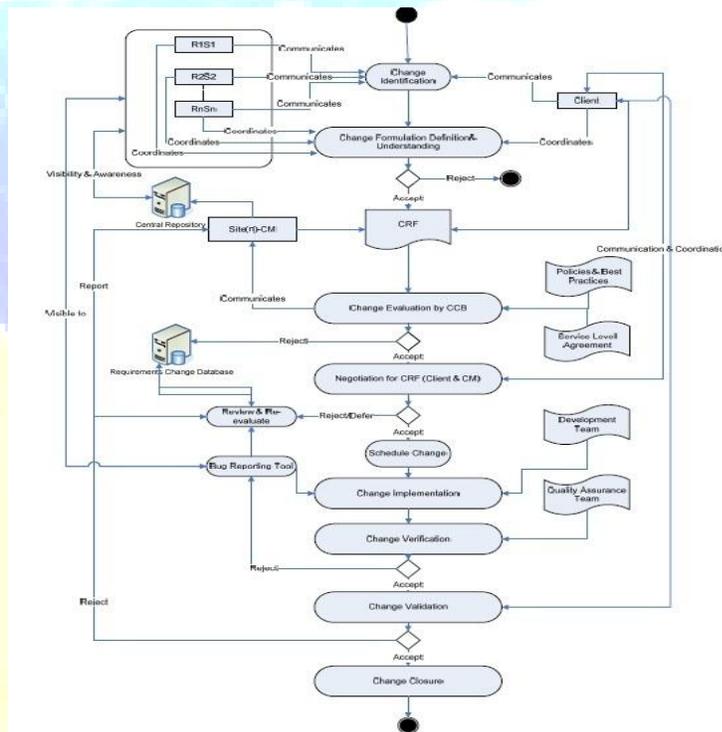
Figure 2: Proposed Model for Requirement Change Management in GSD

The identified bugs are worked upon by the development team till they get closed. If there is some problem in fixing a particular defect and it gets a deferred status this is evaluated by the Review and Re-evaluation committee and reported to CM for further action. The next stage is change validation which is performed once the change is successfully verified by the user. If the change is validated it gets closed and relevant closure procedure is adopted to

formally close the change. With the closure of the implemented, verified and validated change and the updating of related artifacts the change management process comes to an end.

RESULTS AND DISCUSSIONS:

During the analysis of this research work we found out that for the web application software development projects not all five were equally influential in triggering change. We found that the main contributor to the requirements change was Requirements Specification and associated triggers and uncertainties. Within the requirements specification the main triggers were:

1. Resolution of Misunderstanding

2. Resolution of miscommunication

3. Incorrect Requirements Identified

4. Increased client's Understanding of the problem and Solution

5. Increased development team understanding of the problem

6. Increase Domain Knowledge of Development team

The increased understanding of the client and development team appeared to be one of the most critical in triggering change 67% of the overall changes.

Developed Model Score of Role, Artifacts and Activities identified in literature:

Roles 8 out of 11

Artifacts 4 out of 5

Activities 13 out of 18

Due to the coverage of necessary activities mentioned in the literature for any requirements change management process. The model provides representation of the process which improves

visualization of how the activities can and should be performed that may improve the overall
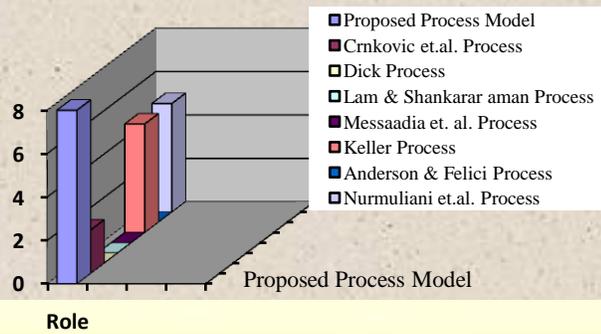
process.

Figure 3: Proposed Model Score of Role

Figure 4: Proposed Model Score of Artifacts

Figure 5: Proposed Model Score of Activities

As a result visualization of the process was improved and becomes clear that reduces misunderstanding and miscommunication that saves a lot of unnecessary work. Proper change evaluation and impact analysis reduces the likelihood of wrong estimates and improves resource delegation.

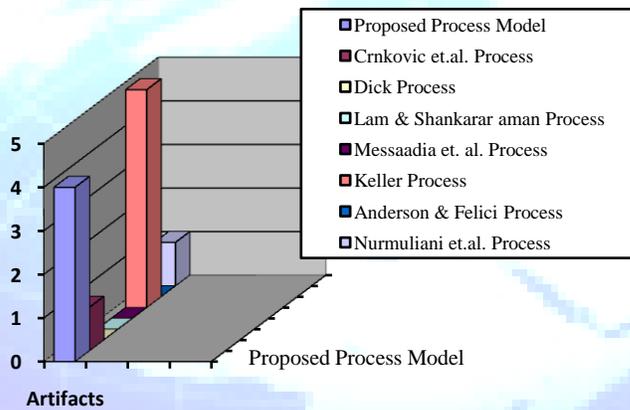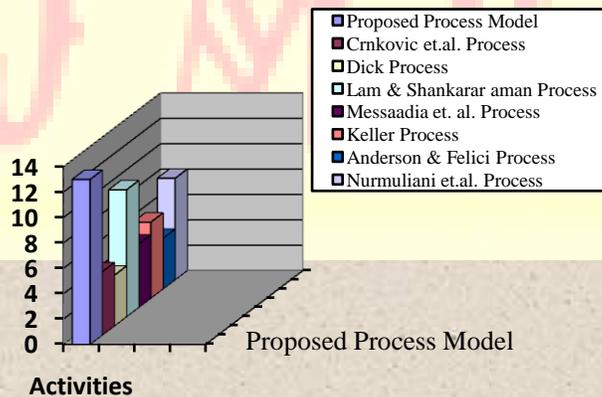This model would benefit requirements change process if some other factors or suggestions are taken into account when implementing the proposed model. The effect of distance can also be minimized by applying a 75- 25% development style where core activities of 25% are kept in proximity to the client and the rest 75% is given to remote site(s).

CONCLUSION:

The model for requirements change management proposed in this case study is not specific to any organization or type of software development project. The model is present here for the adoption by all kinds of GSD projects. The reason is that it covers major activities, artifacts and roles identified from the literature of change management processes. The proposed model highlights multiple roles and locations in a GSD project. It is an effort to depict the way of working in a distributed development environment. It attempts to highlights the communication and coordination requirements in a GSD project. The limitation of this work is that it does not describe all the roles that can be involved in the process. However it is challenging to give the name of all the involved roles if multiple sites are involved and there is organizational difference among different sites and the roles nominated to handle the change management process. The major limitation of this model is that the model has not been tested in a real life project. The only factor which can be said in its favor is that the model consists of mainly those activities which are present in some of the available change management models used in single site development hence in a way it is not absolutely novel in its nature.

REFERENCES:

- Conchuir, E., O. Holmstrom, H. Olsson, P. Agerfalk, B. Fitzgerald. 2009. "Benefits of Global Software Development: Exploring the Unexplored". Software Process: Improvement and Practice, 14(2): 201-212.

- Damian, D. and Zowghi, D., 2003. "Requirements Engineering challenges in multi-site software development Organizations". Requirements Engineering Journal, Proceedings of the IEEE Int'l Conference on Requirements Engineering, 8(2003):149-160.

- Imtiaz, S., N. Ikram and S. Imtaiz. 2008. "A Process Model for Managing Requirements Change". 4th IASTED International Conference on Advances in Computer Science and Technology, 3(25): 152 – 161.

- Lam, W., V. Shankararaman. 1999. "Requirements Change: A Dissection of Management Issues". EUROMICRO Conference, 1999. Proceedings 25th, 18(2): 54-60.

- Lopez, A., J. Nicolas and A. Toval. 2009. "Risks and Safeguards for the Requirements Engineering Process in Global Software Development". Fourth IEEE International Conference on Global Software Engineering. 19(4): 394-399.

- Prikladnicki, R., N. Audy. 2006. "A Reference Model for Global Software Development: Findings from a Case Study". IEEE International Conference on Global Software Engineering (ICGSE'06), 24(2006): 18-28.

- Ramzan, S. and N. Ikram. 2006. "Requirement Change Management Process Models: Activities, Artifacts and Roles". Multitopic Conference, 2006 INMIC '06 IEEE, 219-223.

- Sangwan, R., M. Bass, N. Mullick, N. Daniel, J. D. Paulish and J. Kazmeier. 2007. "Global Software Development" Handbook. New York: Auerbach Publications, 10(6): 72-85.

- Sinha, V., B. Sengupta and S. Chandra. 2006. "Enabling Collaboration in Distributed Requirements Management Software". IEEE, 23(5): 52 – 61.

- Šmite, D., C. Wohlin, T. Gorschek and R. Feldt. 2010. "Understanding of Trust in Global Software Teams: A Multi-Case Study". In proceedings of the Int. Conf. on Product Focused Software Process Improvement (PROFES) in Lecture Notes in Computer Science (LNCS) by Springer Verlag, 4(12): 20-34.

- Yin, R. K. 2003. "Case Study Research, Design and Methods" (3rd edition) London: Sage, 5(1):231-239

A Monthly Double-Blind Peer Reviewed Refereed Open Access International e-Journal - Included in the International Serial Directories
Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A.

International Journal of Management, IT and Engineering
http://www.ijmra.us

18