



Linnæus University

School of Computer Science, Physics and Mathematics

Degree Project

Requirements Change Management in Global Software Development: A Case Study in Pakistan

Waqar Hussain
2010-06-11
Subject: Information System
Level: Master
Course code: 4IK00E

Abstract

Global software development has been a phenomenon of growing interest for almost past decade or so; and its adoption trend continues to gain momentum. Globally distributed work is taken up as an alternative to single-site mainly because of the economic and strategic benefits it offers. Software development at geographically distributed environment is not a straightforward task and entails numerous challenges which are unique to this form of development.

Requirements change management is considered challenging even in the best of conditions and it becomes even harder when performed at geographically distributed development locations. There is no existing model for managing requirements change in globally distributed software development context.

This study uses qualitative research method to explore requirements change management process and investigates the underlying causes of requirements change in geographically distributed software development. The research work proposes a model for requirements change management for global software development. This model tries to incorporate the roles, activities and artifacts identified in the change management models.

Acknowledgement and Dedication

All praise is to Allah the most passionate and the merciful. Peace and blessings be on his Prophet. I am thankful to Allah that he blessed me with the abilities to complete this work, without His help this huge task could not have been completed

I dedicate my work to my mother whose prayers and love never left me, while I was working on this task away from home. Her prayers and encouragement made my work simplified whenever I was faced with difficulties. Without the kind support, encouragement and dedication of my family it would not have been possible for me to pursue higher studies in Sweden. Special thanks to my brother *Muhammad Usman* for his encouragement and help in reviewing my work.

1

I am thankful to my supervisor, Osama Mansour, who kept us motivated during the whole work of thesis. His useful suggestions, advice and ideas were helpful in keeping the work on track.

Also, I thank Dr. Jan Aidemark for his useful suggestions, advice and ideas to bottleneck problems encountered during this thesis work were just immensurable. His expertise and constant encouragement were very helpful and made this effort an enjoyable one.

My special gratitude and respect to *Dr. Naveed Ikram* who has been a source of inspiration and knowledge. The enlightening discussions with him and the invaluable guidance and motivation played a major role in helping me complete my work. I am thankful for his kindness and patience and useful feedback whenever I needed. He is really the best. I would like to thank from the bottom of my heart to all my teachers who were light houses which did not let my ship get lost in the sea.

I am grateful to all the interviewees of GSD Inc. and especially to my friend Mr. Abdul Rahim & Mr. Amir Ikram for their timely cooperation and help by sparing his precious time and giving useful feedbacks during my thesis work.

I am thankful to all staff and friends at Linnaeus University, Vaxjo Sweden; they all were very supportive and kind. Special thanks to all my friends and loved ones back home, who prayed for me and also helped, guided and encouraged me whenever I required their support.

Table of Content

Chapter1: Introduction	1
1.1 Background: GSD	1
1.2 Problem Identification and Justification.....	1
1.3 Purpose and Aim of the Study	2
1.4 The Research Questions	2
1.5 Research Focus	3
1.6 Research Relevance and Contribution	3
1.6.1 Theoretical Contribution.....	3
1.6.2 Practical Contribution.....	3
1.7 Data Validation	4
1.8 Role of Researcher	4
1.9 Ethical Issues	4
1.10 Limitations	4
1.11 Disposition of the Thesis	5
Chapter 2 Literature Review	6
2.1 Definitions of Relevant Terms.....	6
2.2 Globalization and Global Software Development	7
2.3 Benefits of GSD	8
2.4 Challenges in GSD	8
2.4.1 Distance Problem in GSD Context.....	9
2.4.2 Challenges at Different Levels.....	11
2.5 Management of GSD.....	12
2.6 Globally Distributed Requirements Engineering.....	12
2.7 Why Requirements Change	14
2.7.1 Main Causes of Requirements Change in Literature.....	15
2.7.2 Reasons for Change.....	15
2.7.3 Origin or sources of Change	15
2.8 Taxonomy of Requirements Change.....	16
2.9 Requirements Change Management.....	17
2.10 Process of Requirements Change Management.....	17
2.11 Process and RCM Process Models.....	17
2.12 Need for Global Requirements Change Management Process Model.....	19
2.13 Tool and Technologies to Support GSD	19
2.13.1 ICT infrastructure	19
2.13.2 Collaborative technology	19
Chapter 3 Research Methodology	22
3.1 Qualitative Research	22
3.1.1 Case Study	22
3.2 Selection of Case.....	23
3.2.1 The Studied Projects	23
3.2.2 Project Sources of Change	24
3.3 Reason to Choose Case Study in Pakistan.....	24
3.4 Philosophical Assumptions of Research	24
3.5 Research Approach	25
3.5.1 Data Collection Procedure	25
3.5.2 Data Analysis	26
3.5.3 Analysis Method.....	26
3.5.4 Interview Criteria	27
Chapter 4 Empirical Findings.....	28

4.1 Case Description	28
4.1.1 Project Description: Project SDE	28
4.1.2 Project Description: DataDive 2.0.....	29
4.2 Data Collection	30
4.2.1 Interview	30
4.2.2 Data Collection from Artifacts.....	31
Chapter 5: Analysis and Discussion	36
5.1 Analysis of Requirements Change	36
5.1.1 Stage One: Understanding the Change	36
5.1.2 Stage Two: Change Analysis	38
5.1.3 Stage Three: Categorization of Requirements Change.....	38
5.2 Requirements Change in Light of Change Taxonomy by McGee & Greer	40
5.2.1 The Root Causes (Most Influential Triggers) of Requirements Change	40
5.2.2 Most Influential Uncertainties that Trigger Requirements Change	41
5.3 Discussion: Triggers and Uncertainty (modified after the original Authors).....	42
5.5 Influence of Uncertainty Constructs on GSD Projects.....	45
5.6 Modeling the Existing Process of Change Management.....	49
5.7 Summary of Change Management Process Analysis.....	50
5.7.1 Problems with Change Management Process	50
5.7.2 Recommendations	51
5.8 The Proposed Model for Requirements Change Management for GSD.....	51
5.9 Application of the Proposed Model	54
5.9.1 Proposed Model Score of Role, Artifacts and Activities identified in literature	54
5.9.2 Proposed Model and Identified Root Causes of Requirements Change.....	54
5.10 Limitations of the Proposed Model.....	56
Chapter 6 Summary and Conclusion	58
6.1 Summary.....	58
6.2 Conclusion	58
6.3 Future Work.....	59
10. References	60
Appendix A	65
Change Request Form.....	66

List of Figures

Figure 1..... Page 3

Figure 2Page 6

Figure 3Page 9

Figure 4Page 13

Figure 5Page 14

Figure 6Page 16

Figure 7Page 27

Figure 8Page 44

Figure 9Page 49

Figure 10.....Page 53

List of Tables

Table 1.....Page 10

Table 2.....Page 11

Table 3.....Page 18

Table 4.....Page 20

Table 5.....Page 35

Table 6.....Page 39

Table 7.....Page 39

Table 8.....Page 48

Table 9.....Page 56

Key Terms Used

CRF: Change Request Form

GSD: Global Software Development

RC: Requirements Change

RCM: Requirements Change Management

RE: Requirements Engineering

CMMI: Capability Maturity Model Integration

CM: Change Moderator

QAM: Quality Assurance Manager

TL: Team Lead

Chapter1: Introduction

1.1 Background: GSD

The development of Information systems and software are getting more and more globally distributed. The economic factor still remains the most influential driver for this phenomenon. The continuous growth, innovation and ongoing improvements in ICT's enable even complex projects and systems to get developed at places with geographical, temporal and cultural distance. Cost reduction, increased production, risk dilution and improvement in quality as well as flexibility in software development are the means to competitive advantage and are the motives common to the software industry worldwide. The inclination towards Global Software Development (GSD) is obviously because of its well identified and documented benefits (Conchuir et al 2009) 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 offshoring). GSD appears as a feasible alternative (Prikladnicki et al 2006) in such an environment.

More than a decade ago, the experimentation began with developing software using remote located facilities in order to lower cost of development and to access skilled resources available globally. It was the same time when the global market found itself having to deal with numerous crises not only through large number of project failures but also because of lack of competencies that were affecting the demand for new systems. However, during that time large investments paved the way towards the movement of globalization which then resulted in the creation of new forms of competition and collaborations (Prikladnicki et al 2006). The idea of globally distributed software development therefore continues to gain momentum. Sahay (2003) defines GSD as follows:

“Global software development is the software work undertaken at geographically separated locations across natural boundaries in a coordinated fashion involving real time (asynchronous) and synchronous interaction”

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. Our investigation work about root causes of continuous requirements change is aimed to better understand and support the process of requirements change management in GSD projects. For this purpose we propose a model for Requirements Change Management (RCM) to be utilized by global software development companies working on customized development of web and desktop application projects.

1.2 Problem Identification and Justification

This study is carried out owing to the need of investigation of RCM processes as pointed out by various other researchers. For example Lam et al (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). There remains a *gap* in this area to be filled up with more rigorous research on RCM process.

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). Other studies, e.g. Smite (2006) also hint towards having a whole new set of techniques and strategies is required to successfully carry out GSD projects. Similarly Zowghi (2007) states that requirements engineering phase is given relatively very little time and effort as compared to other software development activities. She suggests that elicitation, analysis, specification, validation and management of requirements remain one of the least explored and have the least satisfactory scientific foundations (*ibid*).

Considering the novelty of the GSD paradigm and the lack of research work in RCM process in GSD environment, this research work plans to explore RCM process in GSD to develop better understanding of the process and its possible future improvements. We believe, as pointed out by Lopez et al 2009, that overcoming the pitfalls posed by GSD on RE is the key issue to achieve success in GSD. The addition of empirical evidence, as a result of this work, in the area of RCM for distributed development also justifies this research. Damian & Moitra (2006) point out that the importance of GSD demands robust models, methods and processes that can efficiently and effectively execute GSD work. Therefore our proposed model for requirements change management in globally distributed development environment is a small contribution in this research area.

1.3 Purpose and Aim of the Study

The purpose of this research work is to investigate the process of Requirements Change management for Global Software Development 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. The objective is to provoke further understanding of the process of requirements change management in a global software development context.

1.4 The Research Questions

This study raises the following research questions:

- 1) What are the root causes of requirements change in GSD?
- 2) How requirements change is managed in GSD projects?

For the purpose of answering these questions this study explores Requirements Change Management (RCM) process in two distributed development projects carried out in a software development company situated in Pakistan.

We propose a RCM process model for managing requirements change for software development in a global environment. This model is based on our investigation of the

literature review, RCM process, the analysis of the empirically collected data and the findings from the case study conducted for this purpose.

1.5 Research Focus

The focus of this research is on the requirements change and the related change management process. We focus on the change of requirements that occur after the initial Software Requirements Specification (SRS) document is drafted but it has to undergo continuous changes, requested through Change Request Forms (CRF), throughout the development lifecycle. The objective is to identify the root causes of requirements change collected and analyzed through the CRF form in global software developments. Our thesis work focuses on commercial software development.

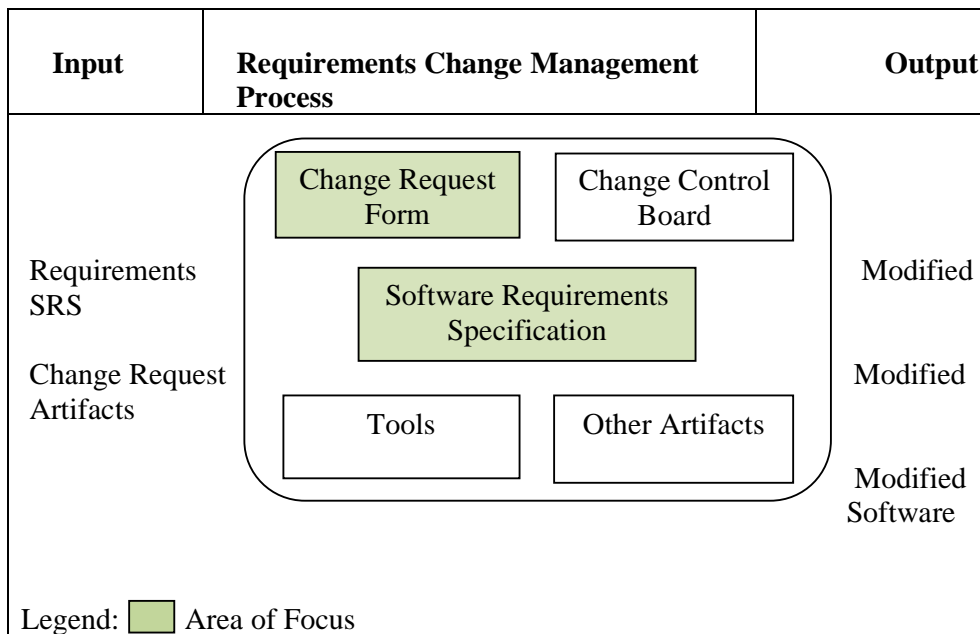


Figure 1: Focus of Research

1.6 Research Relevance and Contribution

This research work is relevant to information system research, management practice as well as process improvement area. It provides both academia and industry with theoretical as well as practical contributions respectively. Following are its contributions.

1.6.1 Theoretical Contribution

The research work conducted in this thesis has theoretical contributions in the form of identification of the root causes of requirements change that may act as an initial step to better understand, manage and measure requirements change. The improved knowledge and understanding gained through this work will contribute towards improving the process of requirements change management in GSD context.

1.6.2 Practical Contribution

This investigation will provide practical contribution to the body of knowledge by providing a process model for requirements change management specific to be used in global software development projects. Secondly it will provide more empirical data in this area of research and add to the scanty empirical evidence available in this field. Finally, the identified causes of requirements change has practical importance as it can be used by managers as a strategic tool to be used as impact analysis and also for effort estimates etc.

1.7 Data Validation

For qualitative validity of our results different reliability procedures will be adopted such as checking transcripts and notes to remove obvious mistakes, constantly comparing raw data, encoded data, conducting regular documented meetings and by sharing the analysis with peers the peer-review. Also triangulating different data sources of information and by examining evidence from different sources such as e-mails, minutes of meetings, spread sheets, bug reporting tools.

1.8 Role of Researcher

During the case study, we as researchers will have an exploratory role in which the researcher explores the process and does not participate decision making or does not interfere in the activities of work. Rather the role is merely to explore, investigate, record the activities related to requirements change process.

1.9 Ethical Issues

This study has two main components which have to be considered ethically. One of them is getting a permission to take part in the organization as a researcher role, who will observe employees and business processes, interview project manager and access critical data. This permission was obtained by signing of a contract of understanding of terms of research which defines the researcher's framework, parties' rights, responsibilities and damages in case of any compromise of data or information. Researcher's role poses challenges and encumbers a huge responsibility when it comes to the non disclosure or secrecy of information such as private, personal and company information used or reviewed for the purpose of research. This area is covered by data secrecy agreement which provides rules regarding using and publishing of secret data. Those agreements officially define roles and boundaries of both the parties collaborating for this research.

1.10 Limitations

Not all companies in Pakistan however are, as our case study will explore, do only web application project therefore our results may not be generalized or be applied to other types of software development such as desktop & mobile applications.

The research work is not generalizable to other countries because of the simple fact that the procedures and methodologies employed by the companies in different parts of the world are not the same and neither can their unique traits and style of the development be generalized on a large country-wide scale. The scope delimiting boundary of this work is the study of RCM process adopted by GSD Inc. Another limitation is the lack

of direct observation of activities because of where the company is situated i.e. Pakistan.

As a future work we would consider the application of the process model with the support of available tools of change management process. This would require a deep analysis of the technological capabilities. This part constitutes the technological aspect of the future study which is beyond the scope of this current research.

1.11 Disposition of the Thesis

The outline of this thesis is structures in the following manner. **Chapter 2, Literature Review**, describes the theoretical framework of the research work conducted. It defines most commonly used concepts in global software development. It discusses topics related to globalization, requirements engineering (RE), requirements change and management and the related change management process models and the need for such a model in GSD context. It also touches upon the use of tools and technologies prevalent in distributed development. **Chapter 3, Research Methodology** explains the choice of qualitative case study methodology adopted in this research, and the case selection criteria. It also describes the research process and specific techniques used for data collection and analysis. **Chapter 4, Empirical Findings** presents the empirical data collected from the case study performed for this research work. **Chapter 5, Analysis and Discussion**, is the actual analysis performed on the collected empirical data and presents our discussion based on it. Then a model for requirements change is proposed for managing requirements change in global software development. **Chapter 6, Summary and Conclusion** the chapter gives the concluding remarks and possible future work direction based on this research work.

Chapter 2 Literature Review

This chapter discusses the basic terms and definitions used in Global Software Development. It introduces the concepts of Globalization and Global Software Development and talks about the benefits it provides as well as the challenges it faces. Then the management of GSD in general is discussed leading to the area of our focus; i.e. globally distributed requirements engineering. In this chapter the causes and taxonomy of requirements change is also explained in light of literature. Finally the change management process, its process models and the need for a global software development model is highlighted. At the end of the chapter the collaborative technologies used for collaborating global software development work are discussed.

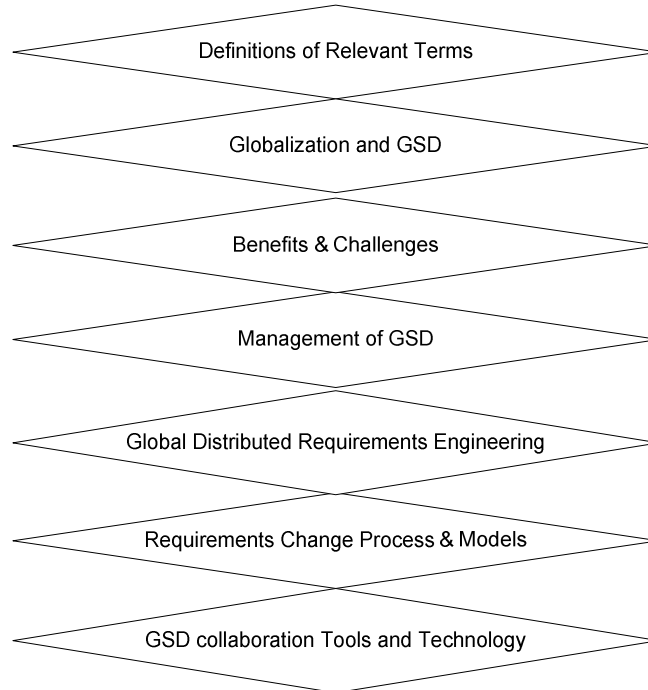


Figure 2 Literature Review Dispositions

2.1 Definitions of Relevant Terms

For the purpose of understanding of the reader we define the most commonly used terms used in requirements change management process when performed in a globally distributed environment.

Software requirement as given by Dorfman and Thayer (1990) cited by Leffingwell and Widrig (2003) is “a software capability needed by user to solve a problem to achieve an objective”. Alternatively; “a software capability that must be met or possessed by a system component to satisfy a contract, standard, specification, or other formally imposed documentation”. Similarly a requirement can also be seen as a documented, externally observable characteristic of a desired system. Finally we present a compact and workable definition by Oberg (2000) that “a requirement is the condition or capacity that a system that is being developed must satisfy”.

Requirements Change (RC) refers to “the emergence of new requirements or the modification or removal of existing requirements” (Lam & Shankararaman 1999).

Changing requirements are considered to be a cause of failure for new as well as legacy systems as both have to go through a number of requirement changes (Lam et al 1999). Requirements change encompasses:

Requirements volatility which is a term meaning ‘a measure of the number of requirements changes (addition, deletion and modification) (Davis 2005 Cited by Davis et al., 2008) divided by the number of requirements for a given period of time (Costello et al 1995 cited by Davis et al 2008).

Requirements Creep which refers to ‘frequent changes in requirements’ (Jones 1996 Cited by Davis et al 2008), and also described as changes that result “in extensions to and alterations of the software’s functionality and scope’ (Carter et al 2001 cited by Davis et al 2008)

Requirements Management (RM) means “the systematic process of organizing and storing relevant information about requirements, while ensuring requirements traceability, and managing changes to these requirements during the whole lifecycle of the information system” (Grehag 2001).

Requirements Change Management (RCM) is concerned with making rational decisions whether to implement a requested change or not. It is also concerned with supporting the identification of which information, e.g. documents and other requirements that are affected by the proposed change (Grehag 2001). Change management is not easy to perform even under the best of circumstances (Sangwan et al 2007) and it becomes more difficult when performed globally because of the nature of distributed development projects and the diversity of stakeholders (Damian & Zowghi, 2003).

2.2 Globalization and Global Software Development

Herbsleb and Moitra (2001) describe the phenomenon of globalization and its driving force as follows:

“The last several decades have witnessed a steady, irreversible trend toward the globalization of business, and of software-intensive high-technology businesses in particular. Economic forces are relentlessly turning national markets into global markets and spawning new forms of competition and cooperation that reach across national boundaries.”

Global software development (GSD), in combination with the advancement of new technology and advent of newly developed business models, is the prime enabler of globalization (SIIA 2006). This phenomenon has a huge impact on development as well as servicing of software. It has not only taken the customer service and technical support but also the development of ‘core product’, away (to geographically distant and different locations) from ‘home’ firm (SIIA 2006).

It is important therefore to understand the nature of globally distributed software development and the projects that are undertaken based on this concept. ***Globally distributed software projects*** consist of two or more teams working together to accomplish project goals from different geographical locations. In addition to geographical dispersion, globally distributed teams face time-zone and cultural differences that may include but are not limited to different language, national traditions, values and norms of behavior (Carmel 2009).

A subset of Global Software Development sprung up under the name of *Follow The Sun* (FTS) which is also termed as: 24-hour development. Yet another suitable name given to such development model is round-the-clock development (Carmel et al 1999). It shares with global software development the many issues and challenges of coordination, culture, and communication. What FTS aimed to achieve was the speed, in the development of the product and its time to the market in a reduced time period.

However what was promised did not get fully realized with FTS (Carmel et al 1999). This methodology of development did not have many success stories of FTS, and the main culprit for that is the *coordination*. The way of working in FTS is simple, you hand-over your work where the working shift is over to the other site where the work shift starts and this handing and taking over continues round the clock or 24 hours, hence the name 'round the clock development'. This reduced the development duration many fold (depending on the number of shifts used) for example by 50% if there are two sites and by 67% if there are three sites (Carmel et al 1999).

2.3 Benefits of GSD

Conchuir et al (2009) have collectively mentioned several trends that lead to the quick acceptance of the global software development trend. These factors have been discussed in other leading researchers of this field such as Herbsleb (2003); Damian & Moitra (2006); Damian (2006); Zowghi (2007) etc. These factors include

- Capitalize on global resource pool to successfully and cost-competitively use scarce resources, wherever located
- Gain the business advantages of proximity to the market, including knowledge of customers and local conditions, as well as the good will engendered by local investment;
- To quickly form virtual corporations and virtual teams to exploit market opportunities;
- To respond to severe pressure to improve time-to market by using time zone differences in "round-the-clock" development
- To respond to the need for flexibility to capitalize on merger and acquisition opportunities wherever they present themselves

2.4 Challenges in GSD

The organizations that have adopted a Global Software Development (GSD) strategy have realized Software development performed in geographically and culturally different locations is not a straight forward activity, it comes with its fair share of problems. This is because of the inherent complexity of software development its outsourcing to other organizations or offshoring to remote divisions becomes even more complex (Carmel 1999, Herbsleb & Moitra 2001). Coordination, lack of understanding requirements and testing of systems are but a very few of the difficulties faced by these types of projects (Carmel 1999; A°gerfalk and Fitzgerald 2006).

GSD faces additional challenges when taking up software development work, it not only has to deal with all the problems common to the single-site development but additionally it has to face problems that are unique to multiple-site model of development (Lanubile *et al* 2003).The problems of GSD are compounded due to culture, language geographic and temporal distance. Furthermore lack of communication, remote team members (virtual teams) and the customer, varied nature

of processes and their maturity levels, different development and testing tools, standards, technical skills and experience, exacerbate the problems faced in GSD. This is why globally distributed software development projects are recognized as difficult and complex endeavors (Lanubile *et al* 2003).

2.4.1 Distance Problem in GSD Context

GSD offers many benefits as described in section 2.3 which includes cheap labor or low salary cost, benefiting from a 24 hour model of development, having a larger pool of skilled workers to choose from and also the proximity to market and customers. All of this makes GSD a potential choice as a development model. However GSD also poses the challenges such as temporal, geographical and socio-technical distances. Figure 3 depicts these factors diagrammatically.

Distance has been identified as a key problem and by its very nature introduces barriers and complexity into the management of globally distributed projects (Rutkowski *et al* 2002). This can be directly ascribed to the fact that there are four key elements to distance in this context. The distance factor poses challenges to communication, coordination as well as control while performing various project activities. Due to distance communication and especially informal communication suffers a lot. The chat during drinks and coffee breaks that sometimes is very helpful is missed in this kind of development style. Coordination is another aspect which suffers because of the distance factor as most often reaching the right person at the right time is difficult because of the geographical distance and time difference. Yet another factor is the control over activities of the project, due to distance the visibility of the operations performed at a distant site is weak hence the control also gets compromised a little. The information flow from different sites regarding the operations and activities reaches late hence the managements control mechanism might sometimes get delayed.

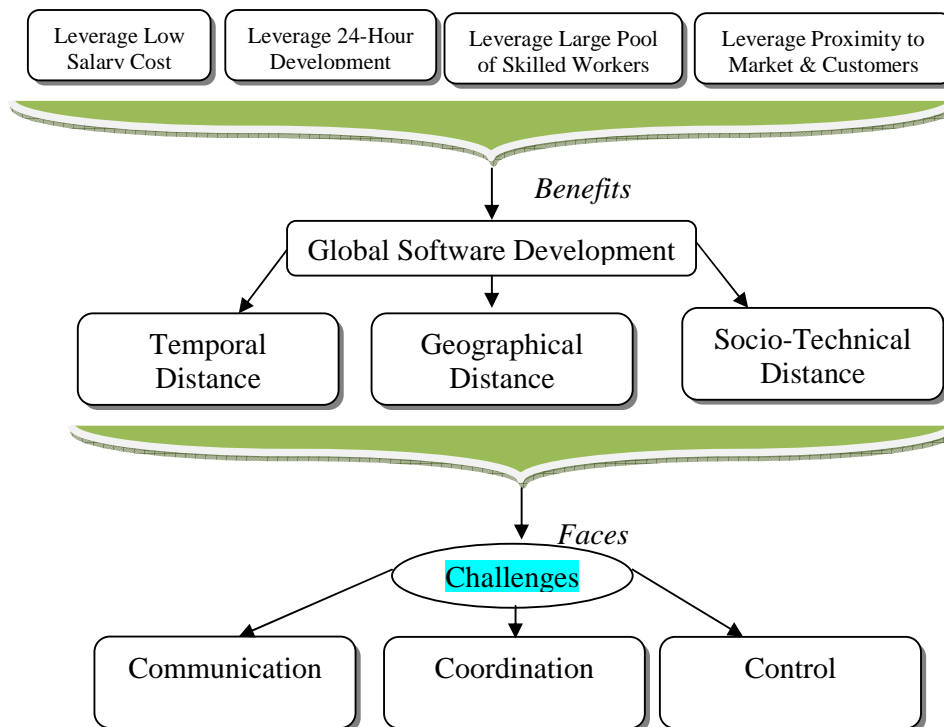


Figure 3: Effects of Temporal, Geographical and Socio-Technical Distance on GSD

Geographical distance introduces physical separation between team members and management (Casey and Richardson 2006). Temporal distance hinders and limits opportunities for direct contact and cooperation (Carmel 1999, Prikladnicki *et al* 2003). Linguistic distance limits the ability for coherent communication to take place (Jensen *et al* 2007). Socio-Cultural distance negatively impacts on the level of understanding and appreciation of the activities and efforts of remote colleagues and teams (Hayes 2002). The overall effect of the distance is a challenging factor for the global software development throughout software development lifecycle.

Table 1 describes different positives of global software development but at the same time highlights the effects of *distance* (temporal, geographical and socio-technical) on communication coordination and control.

	Temporal Distance	Geographical Distance	Socio-Cultural Distance
Communication	<ul style="list-style-type: none"> ⬇ Improved record of communications ⬇ Reduced opportunities for synchronous communication 	<ul style="list-style-type: none"> ⬇ Closer proximity to market ⬇ Access to remote skilled workforces ⬇ Face to face meetings difficult 	<ul style="list-style-type: none"> ⬇ Innovation and sharing best practice ⬇ Cultural misunderstandings
Coordination	<ul style="list-style-type: none"> ⬇ Coordination needs can be minimised ⬇ Typically increased coordination costs 	<ul style="list-style-type: none"> ⬇ More flexible coordination planning ⬇ Reduced informal contact can lead to lack of critical task awareness 	<ul style="list-style-type: none"> ⬇ Greater learning and richer skill set ⬇ Inconsistent work practices can impinge on effective coordination ⬇ Reduced cooperation arising from misunderstanding
Control	<ul style="list-style-type: none"> ⬇ Time zone effectiveness can be utilised for gaining efficient 24x7 working ⬇ Management of project artefacts may be subject to delays 	<ul style="list-style-type: none"> ⬇ Communication channels can leave an audit trail ⬇ Difficult to convey vision and strategy ⬇ Perceived threat from training low-cost "rivals" 	<ul style="list-style-type: none"> ⬇ Proactiveness inherent in certain cultures ⬇ Different perceptions of authority can undermine morale ⬇ Managers must adapt to local regulations

Table 1: Effect of Distance on Control, Coordination and Communication; adopted From: Conchúir *et al* 2009

The Temporal distance mentioned in the above table refers to the difference in time between various locations for example the time differential between a development team in USA (Virginia) and Pakistan (Islamabad) is +11 hours. Geographical distance means the distance between two or more development teams in terms of kilometers. And the socio-cultural difference refers to the difference of culture, language, customs and traditions among people living in different parts of the world.

As a result of distance, communication on GSD projects is normally electronic with limited opportunities for synchronous contact, depending on geographical location and time zone difference. Coordination, visibility, communication and cooperation are all negatively impacted by geographical, temporal, linguistic and cultural distance (Richardson & Casey 2004). If these elements of distance and their potential for negatively impacting on projects are not recognized and managed correctly, they can be responsible for creating serious barriers and complexity within GSD projects.

Offshoring and outsourcing is an uphill task because of the previously described circumstances and should be pondered over in great detail before embarking on this journey. It is a complex endeavor and the decision for its implementation has to be a product of a well thought-out and planned strategy. The common mistake that should be avoided is to consider GSD different from collocated development and avoiding the use of same strategies as employed in the single site development projects. This can result in a ‘short-sighted approach’ as pointed out (Carmel 1999, Prikladnicki *et al* 003). And it has led to serious problems and numerous failures. Owing to the importance of GSD and the need for a specific model in this area this study has put forward an initial working model for managing requirements change for global software development projects.

2.4.2 Challenges at Different Levels

GSD embodies the involvement of various people, processes (de Souza et al., 2007) and technology that provoke new issues. The main contributors of these problems are distance, culture and time differentials (de Souza et al., 2007). Compared to traditional systems development, GSD is complex (Geisberger et al. 2007) both technologically and organizationally. It poses numerous additional challenges (Herbsleb 2005; Hanisch & Corbitt 2004) that are problematic both at the organizational as well as team level (Richardson et al. 2005; Damian & Zowghi 2003). Moitra and Herbsleb (2001) have provided their categorization of GSD issues on various levels which they refer as the dimensions of the issues as shown in Table 2. Following is the list of these dimensions;

<i>Strategic Issues;</i>	How to divide tasks in a globally distributed environment among different sites? Which model to adopt while working on a particular project, should the sites be allowed to operate totally independently or not etc. How to align the management on different locations to the common goals and develop trust between the people involved in these sites
<i>Cultural issues</i>	GSD involves people from various cultures & backgrounds, this kind of diversity of cultural background & work ethics can create serious issues. Many cultures have different sense of time which often becomes a problem & is subject to many interpretations.
<i>Inadequate Communication</i>	Geographically distributed projects suffer lack of communication both formal and informal. Given the nature of software development which is communication and collaboration intensive suffers most due to this issue
<i>Knowledge Management Issues</i>	The resistance to document and update the artifacts at different locations causes vital information loss of knowledge and hampers the progress of the project and in worse case puts it in jeopardy.
<i>Technical Issue</i>	The issues related to technology use, standardized formats of data collection and transfer, use of compatible software and hardware between sites are some of the technical issues faced by GSD.
<i>Product and Process Management Issues</i>	In GSD projects alignment of processes among different sites often becomes difficult because of changing requirements, lack of synchronization of work & unavailability of tools to fully support collaborative work in different time zones and places.

Table 2: Summary of GSD Issues and Challenges at Different Levels, (Moitra & Herbsleb, 2001)

Another way of grouping the issues faced in GSD is considering the managerial, team and individual perspective. The issues can be grouped at different levels for example;

At *management level*; where the top executives have to confront with project control and management issues (Jiménez et al 2009), low visibility to the development process (Morisaki et al 2007; Sangwan et al 2007) and also having to manage larger and geographically distributed work groups (Bianchi et al 2008).

At the *team level* the members are faced with of issues like integration (Grinter 1999), collaboration (Herbsleb et al 2000), communication and coordination overhead (Herbsleb 2005; Herbsleb & Mockus 2003) (Bianchi et al 2008; Jiménez et al 2009; Lundell et al 2006).

At *individual level* other sets of problems have to be dealt with such as loss of rich, subtle interaction (Herbsleb et al. 2000) having to overcome language problems and understanding and coping with cultural issues.

2.5 Management of GSD

Research on the management of GSD started to emerge in the second half of the 1990s as a subset of research on management of globally distributed projects and, by the late 90s (e.g. Carmel 1999; Karolak 1999), established itself as a separate research area. Researchers have come up with various models and frameworks for managing GSD projects; however there still remains a universal model or framework which is accepted and applicable for all GSD projects.

More recently Agile methods and techniques such as SCRUM, are being proposed to manage GSD (Hossain et al 2009) projects however they are not so commonly practiced and have a limited acceptance. The main argument against the adoption of such methods and techniques is that fact that agile practices thrive on the interaction with the client and GSD inherently lacks this opportunity to involve the user frequently because the distance and time difference.

2.6 Globally Distributed Requirements Engineering

The literature identifies many factors that influence the globally distributed requirements engineering process. In this study focus on those factors which affect globally distributed requirements engineering. These factors have been highlighted in grey in Figure 4. To narrow down our research area we will focus exclusively on the Requirements Change Management Process and related activities in this research work.

Figure 4 shows numerous factors that affect globally distributed requirements engineering. The factors that this study focuses on are geographical, cultural, language difference, process management, strategic and challenges caused by time. The area of interest of this research study revolves around the requirements change management process hence other requirements engineering activities such as prioritization, elicitation and specification etc., and other processes are beyond the scope of this study.

Figure 4 shows these factors along with other factors that are common to collocated software development influence globally distributed requirements engineering; however the common factors were not focused in this study.

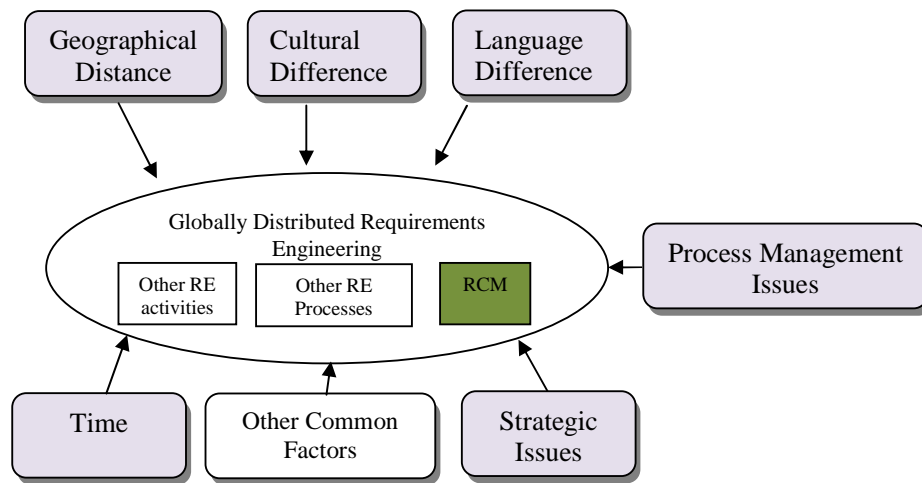


Figure 4: Factors Effecting Global Requirements Engineering Process

Furthermore these factors have influence over many requirements engineering activities and processes but our study is limited to their influence on the Requirements Change Management Process only.

Damian and Zowghi (2003) discuss the effects of GSD on Requirements engineering provides insight into the culture, conflict and affects of distance in globally distributed requirements negotiations. This research work mentions the known (identified in literature) problems that global software development is facing such as Cultural Diversity, Inadequate Communication, Knowledge Management and Time difference. Similarly Carmel & Agarwal (2001) talk about distance being one of the key factors that challenges organizations coordination and control without which the organizations cannot operate properly.

According to them distance has negative impact on all three crucial factors (communication, coordination and control) of success for a distributed development project. Gumm (2007) also talks about social distance and discusses cultural, language, process and view on a subject. The author argues that in a distributed project the involved organization hold different views about the subject under discussion which is causing problems. This complicates things and creates difficulty in reaching an agreement. He attributes cultural difference as the cause of such problems. Timea et al (2007) talk about challenges of distributed development caused by distance. They mention five barriers to the efficiency of distributed project these include, a) technical b) process c) communication d) domain specific and e) cultural barriers. The influence of these negatively impacts the resultant efficiency of a distributed project.

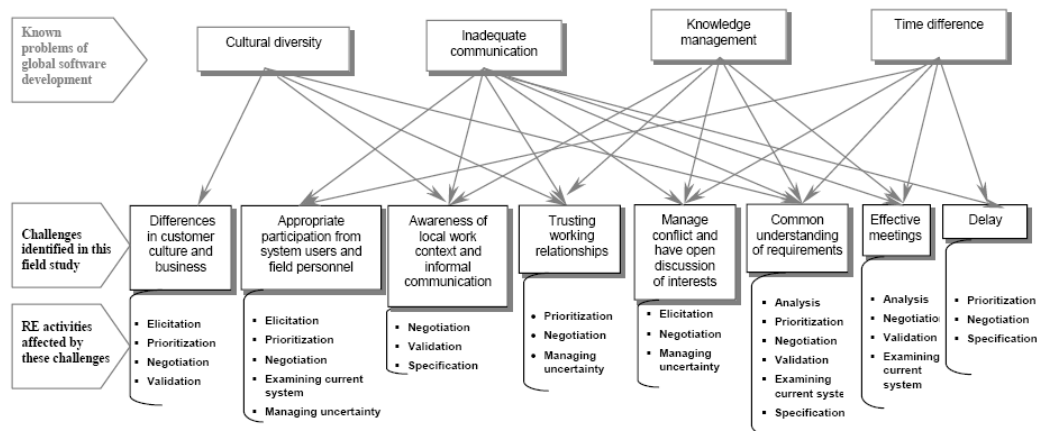


Figure 5: A model of impact of challenges and the affected RE activities due to problems of cultural diversity, inadequate communication, knowledge management and time differences in global software development *Adopted from Damian & Zowghi, (2003)*

Figure 5 shows the known problems of GSD and also the newly identified problems by Damian & Zowghi, (2003) through their field study. They mention that there are several other challenges faced by Global Software Development in terms of Requirements Engineering activities. The authors identified several other factors such as differences in culture, appropriate participation from system users and field personnel, awareness, trust, common understanding of requirements, effective meetings and delay and they claim that these newly identified factors were facing a combined influence by the previously identified and known GSD problems.

Furthermore the authors discuss the affect and challenges of these newly discovered factors on requirements engineering activities. The authors mentioned that these newly identified challenges affect most of the RE activities such as Requirements Elicitation, Prioritization, Negotiation, Validation, Specification, Management of Uncertainty.

2.7 Why Requirements Change

Much of the classical Software engineering literature is based on “the assumption of fixed requirements”; this misconception leads managers to believe that they should freeze the requirements before project starts (Lam et al 1999). However, contrary to this traditional belief most software tend to evolve (Lehman, 1996) and requirements change both in the beginning as well as later stages of software development lifecycle. Requirements continuously change because the "cultural matrix" of processes, users, laws, and technology keeps changing (Brooks 1987). Requirements volatility (change in requirements), poses threat to project schedule, cost (Zowghi & Nurmuliani 2002 cited by McGee & Greer 2009) and defect rates and constitutes one of the top ten risks to successful project development (Boehm 1987).

A recent report (2004) published by the Standish Group International which inquired 13522 software projects, indicated that out of the surveyed projects only 29 percent are successful ,18 percent are considered as “failed” and 53 percent are regarded as “suspected” and the main cause of the failed projects is the requirements change (Jiayi et al 2008). Lam et al (1999) suggest that, changing requirements are the main drivers for software maintenance and re engineering activities. Studies indicate that 80% of the software maintenance activities are attributed to adaptive and perfective maintenance (as opposed to corrective maintenance). For this reason organizations must learn to manage requirements evolution as part of the broader software evolution process

(Harker et al 1993 cited by Lam et al 1999). One estimate states that 40% of requirements need rework during the course of the software development project (Hutchings & Knox 1995 Cited by Zowghi 2007).

2. 7.1 Main Causes of Requirements Change in Literature

According to Nurmuliani et al 2003, based on a case study performed at a multi-site organization in USA, the main causes of requirements change are; 1) changes in customer needs (or market demands) 2) Developers' increased understanding of the products 3) Changes in the organizational policy. These identified main causes of requirements change may also hold in the GSD context but this research work explores the influence of other factors such as distance, culture and language on distributed development projects.

These main causes of requirements change can be viewed as the root causes which influence a change in the requirements. In the next section the reasons for change are given which can be seen as the 'what' needs to be changed. These reasons are the effects of something caused by some higher order reason, which we can view as the root causes.

2.7.2 Reasons for Change

The identified reasons for change according to Nurmuliani et al 2003 are; Defect Fixing, Missing Requirements, Functionality Enhancement, Product Strategy, Design Improvement, Scope Reduction, Redundant Functionality, Obsolete Functionality, Erroneous Requirements, Resolving Conflicts and Clarifying Requirements.

It is important to note that these identified reasons for change were mentioned on the basis of one study, and by no means are inclusive of all the possible reasons for change. Furthermore numerous other reasons (more than 60) have been reported in the literature by other authors such as McGee & Greer et al (2009), in their survey on requirements change. Our research work will explore which of the changes mentioned in the literature occur during our case study on the selected projects and then we would discuss these changes in the light of GSD context.

2. 7.3 Origin or sources of Change

Following sources of change were identified by Nurmuliani et al 2003, they show sources from which these changes are triggered. These sources are; Defect Reports, Engineer's call, Project Management Consideration, Marketing Group, Developer's Detailed Analysis, Design Review Feedback and Technical Team Discussion.

The connection between these sources of change and the reason for change is the fact that these sources require some change in area of 'what' needs to be changed which is covered by reasons for change which in turn are related to the root causes which were considered or which influenced this change.

It is important to note that these origins or sources of change were identified based on a particular case study and it is possible that other cases may reveal other sources of change.

2.8 Taxonomy of Requirements Change

Another recent literature survey was conducted by McGee and Greer (2009) on software requirements change source taxonomy. The authors have provided the complete picture of requirements change by describing the Change domain, triggers and uncertainties. The change domain can be viewed as the platform of possible classification or broad categorization of changes that occur in requirements. Triggers are the events that cause the requirements to change whereas the uncertainties are the cause of some event to happen that act as triggers for requirements change. The authors have cumulated a comprehensive list of causes and uncertainties that gives rise to requirements change. In this model they have not only used previously identified causes of change in the literature but also have added some new ones such as ;

First or re engagement of User representative
 Change to business case (ROI, Total Cost of Ownership)
 Customer Organization Strategic Change
 Change of Stakeholder Representative
 Quality of Requirements Specification
 Technical Complexity of Solution
 Technical Uncertainty of Solution
 Quality of Development Team Communication
 Age of Requirements
 Insufficient Sample of User Representatives
 Development Team Stability

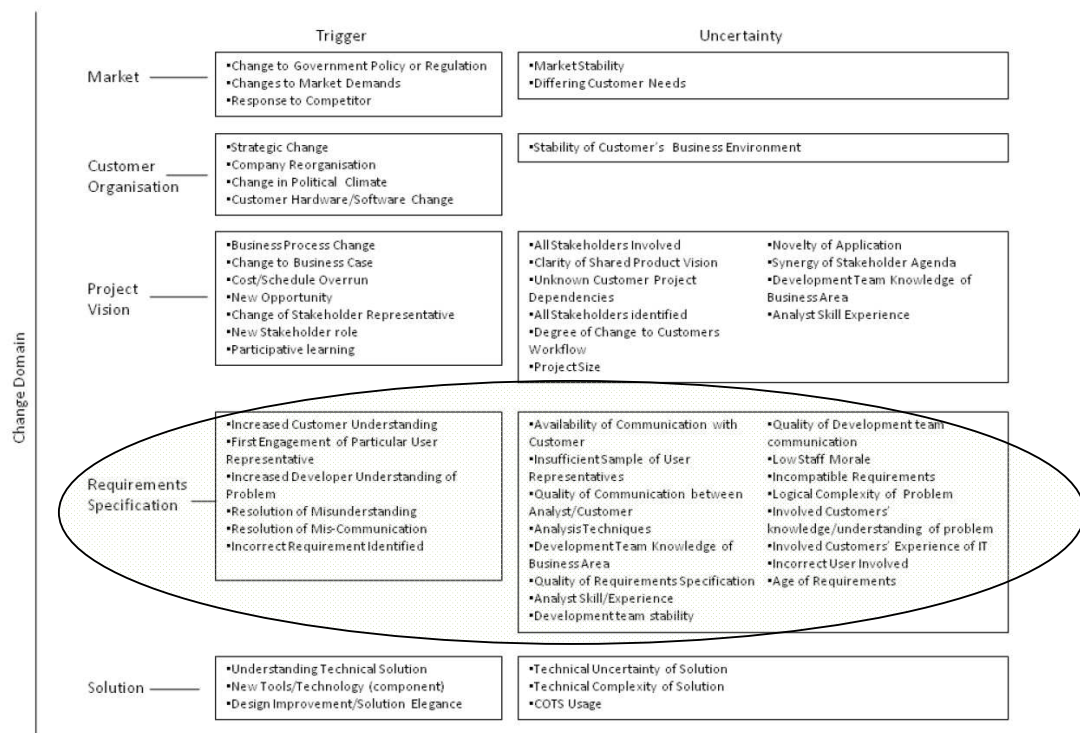


Figure 6: Requirements change Taxonomy by McGee and Greer 2009

The model presented above is used as a base for the development of a model that is more appropriate for use in GSD context. The Area of our main concern along the change domain is the *Requirements Specification* and the triggers as well as

uncertainties included in it. It is observed in this model that there is no mentioning of the fact that there are certain areas that are more likely to cause change in requirements in an *iterative manner* for example the Market, Customer Organization and project vision SRS. However the other areas on the change domain such as tools are relatively less likely to have an iterative effect on the requirements change. Moreover The Market situation , customer organization and project vision have a collective affect on the SRS and are more likely to cause requirements change during different stages of software development life cycle.

We also observed that the triad of geography, culture and language is missing from this taxonomy; these factors are the three main contributors of requirements change in GSD. The model above can be redefined by including the three influencing factors of GSD and their related triggers and uncertainties.

2.9 Requirements Change Management

Requirements change management process is one of the most collaboration-intensive activities and it poses significant difficulties with distributed stakeholders (Sinha et al 2006). Many of the requirements related issues such as specification, collaboration, communication, that hamper smooth progression of development in collocated environment become more potent hurdles to overcome in GSD. Sangwan et al (2006), have discussed some of these broad level factors such as: 1) Diversity of stakeholders 2) Temporal and cultural difference between stakeholders 3) Distance and ambiguity between teams/stakeholders 4) Lack of visibility in terms of details of activities performed across multiple sites.

2.10 Process of Requirements Change Management

Software quality is strongly dependent on the process used in its preparation (Prikladnicki et al 2006). The process of requirements change management is seen as a part of requirements engineering activity and not as an independent process (Kobayashi & Maekawa 2001). 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 & Ikram 2006). The lack of proper management of requirements change may lead to software failure or even loss of business as well however managing change is rewarding but challenging at the same time (Ramzan & Ikram 2005).

2.11 Process and RCM Process Models

A combination of related activities constitutes a process. Process can also be viewed as a set of partially ordered steps to achieve a goal. As defined by Pressman (2001), software process is defined by a set of activities, methods, practices and technologies that people and companies use to develop and to keep related software and products.

-The software quality is strongly dependent on the quality of the process used in its preparation;

- The software process can be defined, managed, measured and improved.

According to Prikladnicki et al 2006, Software quality is dependent on the quality of the process followed to develop the software.

A *model*, simplified defined according to Booch et al (1999), is “a simplification of reality”. A *Process Model* is the representation of a process (Feiler and Humphrey, 1993 Cited by Ramzan and Ikram 2006). But the question is why to use process models? The answer to this question is; process models are the means to facilitate human understating and communication. Also, according to Booch et al (1999), 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 (Ramzan and Ikram 2006). 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. Ramzan and Ikram (2006) 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 conducted by Ramzan & Ikram (2006) 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. Furthermore they concluded that the models described in the literature are not detailed enough to be used in the real world by requirements engineers as shown in Table 3. This renders the implementation of these models per say inadequate not only for single-site but also for multi-site development.

RCM Process Models Activities	Dean Leffingwell and Widrig Model [20]	Olsen's Model [25]	V- Like Model [22]	Ince's Model [22]	Spiral Model [22]	NRM [23]	S.A. Bohner Model [31]	CHAM [32]	S.A. Ajila Model [33]	Simon Lock Model [34]
Plan for change	√							√		
Baseline requirements.	√									
Establish channel to control change	√			√						
Use a change control system to capture changes	√									
Change Impact on functionality	√		√						√	√
Change Impact on cost	√							√		√
Change Impact on customers and other external stakeholders	√									
Potential of the change to destabilize the system	√									
Negotiation process	√							√		√
Budget reconciliation process	√									
Decision making	√									
Manage change hierarchically	√									
Update documents	√			√				√		
Change creation		√		√					√	√
Change implementation		√	√	√		√	√		√	
Verification		√				√		√		√
Problem understanding			√		√		√			
Solution analysis			√		√			√		√
Solution specification			√		√		√			
Submit solution			√							
Regression testing			√				√			
Acceptance testing (Validation)			√	√		√			√	√
Submission of modification report			√							
Change authorization note filled				√						
Current configuration record updated				√						
Examined from the non-technical viewpoint.					√					
Documenting the actions and observations					√					
Determine the type of change								√		
Estimate effort								√		
Release								√		
cost benefit analysis										√
Document impact, cost and decisions										√
System release planning					√			√		√
System release and integration										√

Table 3: Activities in Requirements Change Management Models by Ramzan & Ikram 2006

The authors compared various available models for RCM and using the table above make it clear that not only the models themselves lack the desired synergy between them but also they are not detailed enough to be used by software engineers. Out of the 34 activities mentioned above 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 Ramzan and Ikram (2006). Furthermore the table reveals that there seems to be no consensus between the models used for RCM.

2.12 Need for Global Requirements Change Management Process Model

The literature review reveals the absence of RCM model specific for distributed and global software development. Organizational processes including RCM differ greatly in distributed development teams hence the alignment of such processes is difficult. Distance requires more rigorous change management processes as ineffective propagation of change info across different locations leads to difficulties in coordination and increased development rework (Damian 2007). There is a need for a more detailed RCM model that can be used for managing requirements change in single site development and there is even stronger need to devise a RCM model for handling changing requirements in a globally distributed development environment.

2.13 Tool and Technologies to Support GSD

Tools and technologies suggested overcoming problems and breakdowns in GSD and enable collaboration in a distributed environment comprise (i) a powerful *ICT infrastructure* that allows the transfer of data at high speed, (ii) *generic collaborative technologies* enabling remote colleagues to connect and communicate, and (iii) *software engineering tools* that support software development activities conducted in parallel at remote locations.

2.13.1 ICT infrastructure

A reliable and high bandwidth ICT infrastructure is required to ensure connectivity between remote sites (Carmel 1999). A good infrastructure is important factor for the efficiency of communication and reliability. The establishment of a sound infrastructure and its optimal usage may result in mitigating certain issues that occur due to lack of proper communication and it support throughout the development process.

2.13.2 Collaborative technology

Collaborative technology can be used to improve collaboration in GSD teams. The most commonly suggested collaborative technologies as given in Table 4 by Huis et al 2002) are; email, chat (Instant Messaging), Phone/audio Conference, Videoconference, Internet/Intranet, Group Calendar, Discussion list, Electronic meeting System.

Typically, collaborative technologies recommended for GSD teams are classified according to the time/space dimension: the two-by-two same/different place and same/different time matrix and was widely supported in the research on GSD projects to classify collaborative technologies (e.g. Carmel 1999). This matrix contains four

categories and corresponding technologies: *same place/ same time* (collocated group decision support), *same place/different time* (workflow systems), *same time/different place* (telephone, chatting), *different place/different time* (bulletin board). A more advanced classification of collaborative technologies was suggested by Huis et al (2002): the authors distinguish between several types of collaborative technology that support different needs of globally distributed teams in different time/place settings,

	Setting		
	Different place/ different time (off-line), i.e. support between encounters	Different place/ same time (on- line), i.e. support for electronic encounters	Same place/ same time, i.e. support for face-to- face meetings
Communication Systems: <i>aim to make communications between remote people easy, cheap and fast</i>	•fax •email •voice-mail •video-mail	•telephone •mobile phone •desktop-video •video / audio-conferencing systems (multi-point) •chat system	
Information sharing systems: <i>aim to make the storage and retrieval of large amounts of information quick, easy, reliable and inexpensive</i>	•document sharing systems •computer conferencing	•tele-consultation systems •application for searching remote information sources	•presentation systems
Collaboration systems: <i>aim to improve teamwork by providing document sharing and co-authoring facilities</i>	•co-editing systems	•shared white-board, CAD, word-process or spread-sheet	•Group Decision Support Systems
Coordination systems: <i>aim to coordinate distributed teamwork by coordinating work processes</i>	Synchronizers: •group calendar •shared project planning •shared workflow system	•awareness / notification systems (e.g. 'active batch')	•command and control centre support systems
Social encounter systems: aim to facilitate unintended interactions		•media spaces •virtual spaces	

Table 4: Types of Collaboration Technology (adopted from Huis et al 2002)

The role of collaborative technology is of critical importance in global software development. This research work provides an insight into the collaborative mechanism and technology being used by the company under study.

In this chapter we introduced the concepts of GSD and discussed the motives for its adaption and the challenges faced by Global Software Development. The chapter discusses the concept of why requirements change and the necessity of managing requirements in a globally distributed environment. The chapter highlighted the need of a requirements change management model for managing changes in GSD. And finally the chapter discusses the importance of collaborative technology use in distributed development context. The literature review suggests that there is a need of understanding the nature of requirements change and its causes in a global environment where the management of these causes is more crucial to the success of the projects. The need of a requirements change management model suitable for GSD is also felt and also the use of technology to support the development work carried out in global software development.

Chapter 3 Research Methodology

This chapter discusses our research methodology and the means by which we gather our data for empirical analysis of the evidence collected. The research approach is also discussed in this chapter along with the case description, research settings, ethical issues and role of the researcher. This chapter is aimed at explaining the overall process of collection and analysis of data collected. In the following sections we describe our research plan to conduct this investigation.

3.1 Qualitative Research

This research adopted a *qualitative interpretive* approach and the empirical evidence collected for this research is qualitative. The selection of qualitative approach, which is based on words and not numbers (Miles and Huberman 1994) was used due to the needs of this research work. This investigation adopted qualitative approach because we wanted to study people, process as well as the process related technology in real life context. The need to obtain qualitative data from the people in real-life settings (Cresswell 2007) requires the adoption of qualitative approach which is suitable for such data collection, hence this strategy was chosen as our selected approach of investigation. Qualitative data has many forms including data collected from interview method, which was used in our study to get the empirical evidence; this also justifies our selection of qualitative method for our investigation. Miles & Huberman (1994) recommend that using a qualitative method can help in gaining a deeper understanding of data collected from people about their experiences and their local context.

3.1.1 Case Study

According to Yin (1994), *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).

Interpretive research philosophy implies that ‘our knowledge of reality is gained only through social constructions such as language, consciousness, shared meaning, documents, tools and other artifacts’ (Klein and Myers 1999). Following this philosophy we acknowledged different experiences of individuals within the same context that allowed in-depth analysis of a unique situation about the object of our study in a real life context.

The RCM process is investigated using a case studies approach which is useful for exploration of an event, program, activity, process or one of more individuals, bounded by time and activity according to Creswell (2009, p.14). We used case study as our strategy of investigation in order to gradually make sense of a social phenomenon by contrasting, comparing, replicating, cataloguing and classifying the object of study as prescribed by Miles and Huberman (1984) cited by Creswell (2009, p.194).

The reason for selecting case study method as our favored method is not only because of its widespread use in IS research as per Palvia et al (2003), but also because it allows the researchers to use several methods and approaches to data collection and analysis (Yin, 1994). The conducted research based on interpretive perspective and utilizes data

collection and careful analysis of the evidence not only from the study object (the company under study) but also from the literature surveyed for the purpose of highlighting the problem, identifying literature gap and finding related literature to make the investigation more authentic.

Our Case study data collection involved gathering evidence from a variety of sources: documentation, archival records, questionnaires, interviews and observations as prescribe by Yin, 1994. Triangulation of data collected from multiple sources all by an in-depth study of a phenomenon from different angles and increased the validity of the research findings. In addition, data analysis methods did not involve any quantitative procedures (Strauss and Corbin 1998).

GSD projects are communication and collaboration intensive and are likely to face from issues rising from human interaction, communication, culture and collaborations. We used an interpretive approach to investigate and analyze the root causes of requirements change and the issues related to RCM during a GSD project, inductively which is suitable for investigating such phenomenon. The Case study is conducted for two projects and we performed in depth investigation on two web application development projects in the real life in Pakistan software industry.

3.2 Selection of Case

We have selected GSD Inc. (Names are not mentioned to protect privacy), a software development company situated in Pakistan. The company started its operations in 1999 in USA and expanded its operations to Pakistan in 2002 and made a branch office in Islamabad, and has a successful track record of a number of successful offshore development projects in this time period. GSD Inc. is a small to medium size company with almost 100 employees. The company is working on improving its processes and achieving Capability Maturity Model Integration (CMMI), level two for which they have started the training of their employees.

GSD Inc. has started following process improvement practices and hence it was easy to convince them to improve on their requirements change process for the distributed projects. The company is involved only in custom software development It has its head office in USA and branch office in Pakistan which is responsible for the major development work. The company performs software development, quality assurance and customer contact operations for both local and foreign clients. GSD Inc. is purposefully chosen because it is a typical case in the web based application development in Pakistan for GSD projects.

3.2.1 The Studied Projects

The projects studied for this research work are concerned with the development of customized commercial software for a particular client in publishing industry situated in US. The study is carried out at GSD Inc. at Islamabad site, which is one of the three sites used for carrying out the software development project. This site has approximately 100 employees involved in various software development and support activities. The team dedicated for the two projects under consideration consisted of one Development Manager, one Quality Assurance Manager and an Operations manager. The other members included two development teams with one Team Lead each, one Quality Assurance Engineer and software testers.

The other two sites in the development process are in US but at different locations namely Virginia and Pennsylvania. The US team in Virginia had one Systems Analyst and one Senior Developer. This team was mainly responsible and used for initial requirements analysis, specification and the initial software design as well as partial component and module development during these projects. The onsite senior developer and analyst were dedicated to this project and were also responsible for coordinating work with the team at Islamabad site as well as the clients.

Interestingly the client hired a third party to conduct the user testing of the customized software developed by the development company and this testing team was involved in the user as well as acceptance testing phase. This made the client site as an active member of development process and hence considered as part of the software development. Considering that both the projects under study in our research work were a success, the inclusion of a professional software testing team (a third party) gives a unique dimension to this project and an interesting perspective that might be explored as an option for the future studies.

3.2.2 Project Sources of Change

As the nature of software developed was customized application development, therefore there was no change recorded that was from market demands. Similarly no significant change was noticed from external government or policy making authority. We noticed that the major sources of change for these projects were: Client, Analyst and Development team.

3.3 Reason to Choose Case Study in Pakistan

The reason to perform this case study in Pakistan is because the software industry in Pakistan is growing and the number of projects outsourced to the country is growing rapidly. The country has a large pool of economical work force and skilled professionals equipped with the level of technical expertise required to handle technology related work. Further, the youth of Pakistan has shown a growing interest in education and training themselves in the fields of IT and software technology and the number of trained and skilled professionals who are relatively inexpensive to engage for the development work has increased even more. In addition to this the government has taken numerous steps to develop appropriate IT infrastructure and affordable connectivity rates which result in considerable saving of time and cost for entrepreneurs and at the same time result in faster time of delivery of the products. Therefore Pakistan software industry is an ideal place to conduct a case study for Global software development project as most of the work done for the international software market is done with GSD methodology.

3.4 Philosophical Assumptions of Research

Our philosophical assumptions of this research work are influenced by the qualitative research paradigm which has its roots in cultural anthropology and American sociology (Kirk & Miller 1986 cited in Creswell 2009, p.194). Through this form of scientific investigation researchers (Creswell 2009, p.12) make sense of a social phenomenon. The investigation of RCM process is exploratory research fittingly falls under qualitative research paradigm. Requirements change management in GSD is a set of highly collaborative activities involving diversity of stakeholders with different and sometimes conflicting aims. Studying the process of RCM to explore the activities of

interaction, communication and collaboration between the distributed teams is a social phenomenon.

The 'reasons' for requirements change provided by different stakeholders are subject to interpretation and will be evaluated and analyzed. The rationale given by people to justify or describe the need for requirements change is subjective and is based on human knowledge or is a result of negotiation, which makes it a qualitative data and thus requiring a qualitative investigation. The qualitative research method and a case study strategy is suited to explore and understand this type of problem. The adopted research design is influenced by social-constructivist and interpretative worldview (Creswell 2009) because we explore the phenomenon by learning the meaning that the stakeholders associate with the causes of requirements change. We rely on project stakeholders' opinion, rationale and available change related data for the projects in order to identify and classify causes of change. This enables us to provide answers for exploratory research questions posed in this study.

3.5 Research Approach

3.5.1 Data Collection Procedure

In this study, data was collected in multiple forms by means of Change Request Forms (CRF) which is the core document that will 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 is evaluated on the basis of its feasibility and then accepted or rejected by the Change Control Board (CCB). A template of a change request form used for this case study is attached in appendix A for the benefit of users. Along with CRF other artifacts, related to requirements change, such as emails, meeting notes, spreadsheets will also be used to collect data required for further analysis and interpretation. In addition to that bug reporting tool used identification and communication of defects will also be used to collect data from different development sites. The information contained in this tool is crucial for data collection because it holds change request data and the communication done between various sites for the purpose of collaboration. More qualitative data will be collected by means of interviews with stakeholders. However in our study the CRF and periodically generated consolidated CRF reports are used as the most important sources to find causes of requirements change. All these sources of data collection will provide time and place independent accessibility to study the issue at hand.

Also online video conference and email as well as online chat were used for unstructured interview with the project manager and other key stakeholders. The interviews were recorded when necessary and this structure has already been approved by the company. At for the video conference written notes were used to record important points as a precaution in case of malfunction of recorded material as suggested by Creswell (2009). Open-ended questions provided flexibility for the responders and better understanding about change reasons and their effects on the project time line, for the researchers.

In our case study, the data is collected is based on the views, communication, interaction, collaboration and negotiation of the participants of the RCM process. The data collected is relevant to requirements change by means of change request forms, emails, minutes of meeting, interviews etc. The data collection also included collaboration data while the participants worked on implementing these changes. Data

collection by and of the participants constitutes the social aspect of our study. We also considered the structure, procedures, communication culture and practices to improve our understanding to determine the characteristics of the organization. All these factors helped us explore the root causes of requirements change in this GSD organization and these factors comprise the organizational aspect of our study. Moreover, the tools used by the organization such as bug reporting tool, were examined for storing and documenting requirement change and its management operations. This constitutes the technology aspect. Therefore our study touches upon all the three important aspect of IS research.

3.5.2 Data Analysis

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.

Change request forms along with a consolidated excel sheet for CRF is used for analyzing the nature of change. To support this the interview data collected from the project members is used, these members include; Change Moderator (CM) , Quality Assurance Manager (QAM), Team Lead (TL) and Analyst. Change requests data were collected from two projects within GDS 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.

3.5.3 Analysis Method

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 is based on descriptive and qualitative methods.

Descriptive analysis provides rich information for understanding the requirements volatility problems as well as related aspects such as organizational policy, customer needs and software changes. Qualitative methods are employed to analyze the collected data and to evaluate the change process.

The data analysis framework we present in this paper is adapted from the general approach of Briand et al. 1994. This method is used to determine the causes of requirements change and its related aspects. Figure 7 illustrates our data analysis process, which is a combination of both inductive and deductive inferences. Our approach will be described in the next section.

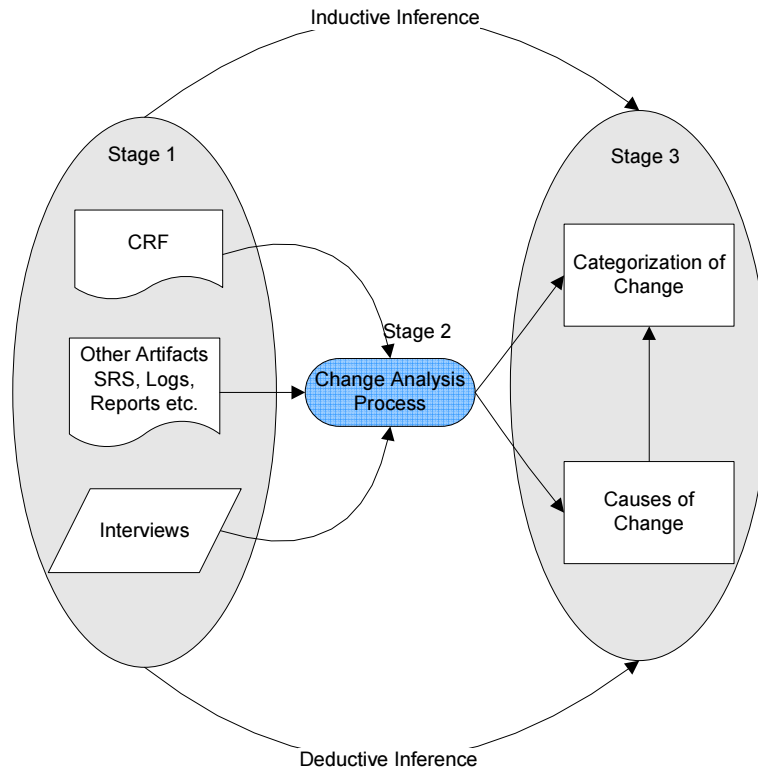


Figure 7: Data analysis framework

3.5.4 Interview Criteria

For this case study and the purpose of data collection team members from Pakistan site were interviewed. The members that were interviewed included

Project Manager (PM)
 Quality Assurance Engineer (QAM)
 Development Lead (DL)
 Change Moderator (CM)

The criteria for selection were; a work experience of more than four years in software development projects and at least two years of experience in distributed development projects.

The Development Project Manager interviewed has more than five years of experience in managing software development projects and has three years of experience in managing distributed software development projects. Similarly PM, QAM, DL and CM fulfill the criteria of selection. As these members are among the people who can initiate change, so they were asked about the reasons they associate with the change and how they view change rationale while understanding and implementing. Many of the changes are requested by the client a US based company in this case, and are written in English therefore culture and language are important factors that are to be taken into consideration when understanding a change request rationale. Their responsibilities and activities were identified and understood in change management process in order to gain understanding of the process of change.

Chapter 4 Empirical Findings

This chapter presents the empirical findings and the data collected during the case study from GSD Inc. The one of the two projects from which the data has been gathered are described. The chapter ends with inclusion of the Change Request Log used by this research work for the later on analysis (Discussed in Chapter 5) of the empirical data.

4.1 Case Description

4.1.1 Project Description: Project SDE

Project SDE was developed for a leading publishing organization working in USA. This project was about development and delivery of an end-to-end print-on-demand solution. SDE is a web application responsible for receiving publishable text from the clients (authors/organizations) as an order which is then processed, finally printed and sent to the client's desired location (book, article, journal or hard bounded copy). The order information, and all related transactions are handled by the SDE by a web application.

The major modules for this application were

- a) Electronic Warehouse (EW)

Electronic Warehouse is a central virtualized system for storage and management of print-ready customer content (products) for print production. GSD Inc. was to re-develop EW as a modular sub-system for the SDE using Microsoft .NET technologies. The EW stores product metadata in EW database and Product XML (PX) data structure along with product cover and inner work content PDF files in network storage. The EW utilizes Common Internet File System (CIFS) for the storage of content. GSD Inc. utilized an FTP interface layer for storage and retrieval of content metadata and PDF files from the CIFS file system.

- b) Third party Module Integration application

The development work included a major module for integration various third party applications that are required for printing and processing operations for the orders received.

- c) XML Data models

The XML data models were developed by GSD Inc. through Joint Application Development (JAD) process. These XML data models were developed that act as a data reference model for the exchange of information within SDE.

The project development work was carried out in 8 months and the data was collected for this project using the change request forms, change request logs, interviews with the project manager and other team members including, SQA Engineer, Development team leads etc. The total number of collected Change Request forms for *Project SDE* during the development cycle was 24, a sample CRF form is added in the appendix for the reference.

Although the change request log (excel sheet) which was maintained by the development teams was available it was not complete and was not updated regularly, moreover this change request log did not have the desired description columns which

were required to perform the analysis of the available empirical data. Furthermore the description in the column 'Approval Detail' was missing for a number of CRFs. To cover this limitation another change request log (in the form of an excel sheet) was created based on all the available CRF forms and with the desired columns. These columns were; Reason for change, Type of Change, Description of Change and Approval details. For the purpose of change classification and analysis an additional column was added into this sheet that was named Classification and Comments. The change request log is produced hereunder for the reference.

CRF No is changed and the type column is removed so that the columns more relevant for the analysis purposes could be presented within the space available. It is important to note that in the columns where the reason for change is not specified, shown as 'N/A' the type of the change, as mentioned in the CRF, was 'Business'. All the change types with this name had no detail in the 'Reason for Change' column. Also, important is to note that all the other changes which have some detail in the 'Reason for Change' column were having the type as 'Other'. The Change Request Log presented below contains only 19 of the total 24 change request, for keeping the data simple and due to the space limitations the number of CR presented is limited.

4.1.2 Project Description: DataDive 2.0

The second project for this case study in GSD Inc. is the development of DataDive 2.0. It is an online repository of data and audit reports for all services provided by the client's organization. DataDive 2.0 is a centralized web base application which provides tools for performing queries that will assist in analyzing and reporting on current and past client's evaluations. The database allows dynamic queries of multiple products and the ability to directly compare results to the specification requirements. The application allows the client to perform Testing Facilities and Auditors to enter real time data online. Also the industry partners and the client organization's administration may review product information and preliminary reports online for timely reporting of all evaluations.

DataDive 2.0 application provides facility to the manufacturer to submit applications online. This application is an effort to promote paper-less environment with a user friendly interface. The major modules of DataDive 2.0 are:

- Submission by Manufacturer Role
- Submit Evaluation Application
- Submitted Evaluation Application Status
- Role Functionality
- View Application
- Process Evaluation Application (Acceptance/Rejection)
- User Management
- Billing Worksheet
- System Administration Features
- User Management
- Manufacturer Product/Facility Management
- Module Management
- Registration Request Management

The developed project's duration was 5 months and after the requirements specification (in this case design specification document) it received twelve change requests.

4.2 Data Collection

4.2.1 Interview

Software Manager

In this case study the Software Manager of GSD Inc. was contacted numerous times for getting the desired information about the structure the organization, project and the teams involved. The SM was the key person in providing information regarding the description of the processes involved for development, the mechanisms how distributed development is achieved in the organization, number of people involved, description of development process, time schedule. To conduct these multiple interviews with the Software Manager different the author used multiple ways for example electronic mails, telephonic calls, online 'Skype' conferencing and last but not the least online text chat sessions were conducted a number of times to conduct the interviews and obtain the required data information. The SM was the key person through which company specific and private data, artifacts, information and resources such as online access to the collaboration tool was accessed, all of which was under 'non-disclosure' contract between the researcher and the company.

The author referred to Software Manager as in our study the Software Manager was also assigned the responsibility of Change Moderator which oversees the whole process. The queries regarding any clarifications on the change management processes and artifacts and project development related issues were discussed with the dual role of SM and CM.

Quality Assurance Manager

With the permission of the Software Manager the documentation available with the quality assurance department was accessed. The Quality Assurance Manager was contacted and interviewed on the documentation of the requirements change process and the involvement of the Quality Assurance Staff along with their responsibilities. She was interviewed to understand the overall quality assurance process in general and requirements change management process and its execution in particular. This included understanding of different activities such as documentation of change request, verification process, bug reporting and closure or deferral of requirements change. Based on the understanding gained from both Software Manager and Quality Assurance Manager, process activity diagram was developed, (shown in Figure 8) for Change Management Process carried out at GSD Inc.

Team Lead

The Team Lead at GSD Inc. was interviewed by the author for the purpose of understanding the team coordination and development related activities and issues. The working hours, overlapping time, communication related queries were put forward to the Team Lead during the interview. The team lead was very vital in understanding how the actual implementation of changes effected their work plan and schedule and how modifications were required in the code and module when changes were coming on a regular basis. During the interview it was realized that only one of the newly hired development team members were sometimes faced with the language problem when working with the design specification and other development related documentation.

The interview with the Team Lead was conducted over the telephone and notes were taken which were later on analyzed for the purpose of our study analysis.

4.2.2 Data Collection from Artifacts

For the purpose of our study we used information from the following data sources.

4.2.2 .1 Collection of Change Request Forms

The next step in the collection of empirical data was to gather the most crucial document in the requirements change management process i.e. Change Request Form. These forms were collected, screened and streamlined. These forms include all the valuable information regarding a particular change request. The following list shows the kind of information a Change Request Form held regarding a particular change.

- Type of Change: New Requirement, Requirement Change, Design Change, Other
- Reason for change: Legal, Business, Performance tuning and Other
- Description of change (with special attention to requirements change),
- Priority of Change (Low, Normal, High)
- Duration of Change
- Artifacts Impacted
- Approval Details
- Reason for Decision
- Change Closure details (successful, unsuccessful etc.)

It is important to note that some of the Change Request forms that did not included the approval details in some cases which were not available till the time of collection of this data. A sample CRF is attached in Appendix 1.

4.2.2 .2 Consolidated Change Request Log Sheet

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, shown in Table 5, was available in the organization and was used for the analysis purposes in our study as shown in Table.

4.2.2 .3 Requirements Management Workbook

Another important artifact that was used for the collection of information regarding the requirements change was the Requirements Management Workbook. This book contains all the requirements that are initially proposed to be implemented into the system. The artifact gets updated with each new addition or modification of the requirement that is proposed during the project development. The workbook is also used for tracing and linking various requirements with other requirements and is an important document to verify about redundant or conflicting requirements. For validation purposes the Change Request Forms that were issued during the projects were matched in

Requirements Management Workbook as crosscheck to ensure each item mentioned in the workbook corresponds to the right CRF and the CRF log sheet.

4.2.2 .4 Other Reference Materials

Some other reference material was also studied and reviewed to establish understanding of the development and management of change procedure in GSD Inc. these item were Configuration plan, project plan, SQA plan, Bidirectional Traceability Matrix etc.

In some cases the minutes of meeting and emails were looked up regarding a particular change request which was missing certain information. We also found the available development status reports and bug fixing reports to be very helpful in understanding the procedure of implementing change, correcting defects and bugs (corrective actions) with the help of the online tool. This tool also served the purpose of asynchronous collaboration among the stakeholders from management, quality assurance and the two development team in Pakistan and USA.

CRF No.	Reason for Change	Description of Change	Approval Details	Classification & Comments
1	Misleading / dual meaning field name. Change required for clarification on the interface for the user.	Change 'Number' Postfix to 'Code' in all the labels of the module.	The required text changes were specified by the client. It is important that terminology familiar to the client is used in an application.	MODIFY Increased understanding of application by the client after usage
2	N/A	Change UOMs of Paper Type	Technical Specification. Business logic changed the client wants new Unit of measurement for paper type	MODIFY Increased understanding of application by the client after usage
3	Attribute name change required as the new one is more description, related to technical jargon(Jargon standardization) used by client	Change field from 'Paper Quality' to 'Paper Finish'	Attribute name change required as the new one is more description, related to technical jargon(Jargon standardization) used by client	MODIFY Increased understanding of application by the client after usage
4	N/A	Add PPI(Paper per Inch) field in Paper Type	Technical Specification. Business logic changed the client wants new Unit of measurement for paper type	ADD Increased understanding of application by the client after usage
5	Additional functionality developed by development team. Not specified in the requirements but agreed upon later by the client and modified when developed and used by the client.	Separate Journal Preference and Publisher Preference. Provide ability to copy the data from the publisher preferences to the journal preferences.	Additional functionality developed by development team. Not specified in the requirements but agreed upon later by the client and modified when developed and used by the client	ADD Increased understanding of application by the client after usage
6	Delete attribute as it is not required	Remove Spine Type.	Delete attribute as it is not required	DELETE Increased understanding of application by the client after usage
7	Add new attribute. Missed in initial Analysis added by user upon review	Add Treatment Type. Add Spine Type. Add Barcode Position in Preferences.	Add new attribute. Missed in initial Analysis added by user upon review.	ADD Increased understanding of application by the client and development team after usage. Missed Requirement

8	N/A	Add E-Mail Notification Template. Add XML Notification Template	Initially not realized that email needs to be sent in a template otherwise it would be difficult to send thousands of mails without one standardized format. Missed in initial Analysis	ADD Increased understanding of application by the client and development team after usage
9	Not Initially communicated by the client identified at review.	Order Types have been added.	Not Initially communicated by the client identified at review	ADD Increased understanding of application by the client and development team after usage. Missed Requirement
10	Requirement Added at review. Gate review documents were sent but client omission of communicating the right requirements.	Mailing Sub system has been included which depends on the type of Distribution job for Springer.	Requirement Added at review. Gate review documents were sent but clients omission of communicating the right requirements	ADD at review meeting. Missed requirement
11	Functionality not required and therefore removed by client.	DDS Fetcher has been removed	Functionality not required and therefore removed by client	DELETE Increased understanding of application by the client after usage
12	GSD Problem	Data elements have been changed /added.	Change in the third party sub-system. For successful and complete integration	ADD/Modify. Miscommunication
13	GSD problem. One party agreed to develop (a functionality which was considered to be present when the part of a sub system being developed by US team) later realized that Team US will not develop it. Therefore Team Pak had to develop this additional functionality	Intent vs Content Comparison.		ADD. Addition of requirement. Missed due to misunderstanding
14	Missed Use Case by Team Pak.	Versioning of products		ADD. Addition , missed requirement

15	Functionality required by Dev Company for verification purposes and business needs	Save SPX.	ADD, additional requirement. Increased understanding of the development team
16	Functionality required by Dev Company. For verification purposes and business needs	Save SOX. Signal Processing	ADD, additional requirement. Increased understanding of the development team
17	Functionality required by Dev Company for verification purposes and business needs. Additional functionality highlighted by client, requested change. Logic Chagne	Save SOX.	ADD, Functionality required by Dev Company for verification purposes and business needs. Additional functionality highlighted by client, requested change. Logic Change
18	N/A	The functionality to calculate Total Inner Work Page Count needs to be changed.	MODIFY. The required change is specified by the client.
19	N/A	SEW Logging and Retry Enhancement (Ticket# 0002622).	MODIFY. Increased understanding of the application after usage by client and development team

Table 5: Consolidated CRF Sheet

Chapter 4 provided the empirical evidence for further evaluation and analysis in the coming chapters. Now chapter 5, Analysis and Discussion, which follows, uses the collected data (e.g. consolidated sheet of change request form) for our analysis and discussion regarding causes of requirements change and their classification.

Chapter 5: Analysis and Discussion

In this chapter we perform analysis on the change data and provide classification of requirements change to understand the nature and areas of impact due to change. Then using a change taxonomy model we discuss the root causes and most influential uncertainties that trigger requirements change. Then in next section the taxonomy is modified into a model to show the interaction and interdependence of factors that influence change. After that the change management process studied in this research work is modeled and the problem areas are highlighted with some recommendations on how to improve on these problems. Finally this study proposes a model for requirements change management in Global Software Development. The application of the model in the GSD context and its limitations are discussed at the end of this chapter.

5.1 Analysis of Requirements Change

Requirements change analysis was performed on the data and information collected through interviews and artifacts such as CRF, log sheet, requirements management workbook, SRS and other reference material as mentioned in section 4.2.2. Special emphasis was given on the reason for change in the change request forms which was the basis for our qualitative analysis for causes of requirements change in GSD. The data analysis framework shown in Figure 7 was used as the basis of our analysis. This framework consists of three stages which as described in detail here

5.1.1 Stage One: Understanding the Change

Artifacts Used to Understand Change

Our analysis process started by the collection of change request data by means of change request forms (CRF). These forms were collected, screened, and analyzed. Our analysis focused on those change requests that related to requirements change;

We evaluated the reasons for change and the description of change along with the reasons for decision to better understand the root causes of requirements change for this globally distributed project. Also by the reasons given by various interviewees in textual form (through chat or email), various stakeholders are in the form of descriptive text, the researchers analyzed the text to identify the root causes of requirements change and explore whether the changes have been influenced by the distance, culture or time difference. The researchers then categorize similar type of changes into categories which are representative of the change. For triangulating our findings, we often crosschecked other related documents during our inspection of the CRF. The artifacts included, design specification, minutes of meeting for the weekly status review, consolidated requirements change sheets, online tool for managing change requests etc.

Another artifact that was used for understanding the changes was *Consolidated CRF Log Sheet*. This CRF log excel sheet was available in the organization however it was found incomplete (missing columns and details in some cases). With the help of screening and streamlining the CRF data and with the help of interview data we were able to fill in the missing data in the consolidated CRF log sheet. Therefore another excel sheet for the summary of change request

forms was made and some of the missing information was completed and filled out with the help of discussions with the project manager regarding the missing data. The newly made excel sheet has two additional columns which were not present in the original summary sheet namely, *Approval Details* and *Comments & Classifications* as shown in Table 5.

The reason to included Approval Details from the CRF was to understand why a particular change was approved and why it was a necessary change. The inclusion of this information was thought would be helpful in understanding the value of change as viewed by the members of CCB. Later on this information was qualitatively analyzed. The second column which was added (*Comments & Classifications*); for the analysis purpose after the collection of empirical evidence. This column helped in classifying the changes in requirements for the project.

Other Artifacts such as Software Requirements Specification, Email, Minutes of Meeting record, status and bug reports, online system for communicating and tracking requirements and other changes. These sources were used when there was some clarification required about a specific change which was unclear from CRF and the log sheet. The clarification was sought through other artifacts or consulting the team members involved in the project. The emails, minutes of meeting, chat and interview notes are also be used for analyzing the causes of requirements change. These artifacts contain valuable information regarding the causes of requirements change, negotiation among stakeholders as they interpret a particular change and later on how the change control board evaluates a given change for acceptance or rejection. All of the collected data from the various sources mentioned above are evaluated and analyzed to understand the process of requirements change. At the end of data collection and interpretation procedure, extracted root causes were categorized and compared with existing literature about causes and categories of requirements change for comparison of the causes found in literature.

Interview Data to Understand Change

Another source used for analysis was the interviews conducted with the key personnel of the project. Their input was vital in understanding and in a way crosschecking the activities performed in the change management process. Our aim was to understand the change management process and the role of these people therein and also to capture information that was not available in the change request form. Interviews were conducted with the key persons in the project that were Project Manager, Quality Assurance Engineer, Team Lead and Change Implementers.

Our analysis of interview data revealed that the company was following most of the procedures prescribed by Capability Maturity Model Integration (CMMI) at level 2. As they were trying to achieve that level and this level requires proper change management and configuration management plans therefore the company had established the process and was recording process activities using required documents and artifacts.

The interview data was vital for understanding the working environment and relationship between the two teams and also with the client. The analysis revealed that the teams were informally structured as higher and lower ranked. The reason for this was that the team members at the US site were of higher position/designation in the company as compared to the ones in

Pakistan and also because the key position of Analyst was present at that site. Team US was enjoying better communication and trust (mainly due to proximity to the client) and frequent communication. The working hours were flexible and both team had at least 2 to 3 hours which overlapped every day. This gave them the necessary handing over or discussion time for the work done or issues that needed attention. This was important for sorting out misunderstanding and miscommunication issues that might take longer if there were no overlapping hours and no communication on a daily bases.

5.1.2 Stage Two: Change Analysis

Due to the importance of CRF for understanding the causes of change special care was devoted in carefully examining the form. The CRF form contained valuable information regarding change such as nature, type, causes/rationale for change, and approval details. The type of change were not strictly categorized in the CRF form as (addition, modification or deletion) but the text written in description of change contained similar words which were picked up to categorize the type of change.

Causes for Requirements Change

This was the crucial stage in the analysis as the data collected and information gathered was vital for performing the causal analysis of requirements change and characterizing the causes of requirements change. Change request data was analyzed in terms of the reason or rationales behind the proposed changes. The findings of the change request data revealed the following causes for change;

- Lack of Domain Knowledge
- Increased Understanding of the client
- Increased Understanding of the development team
- Defect Fixing: - corrective actions to remove defects identified during testing or evaluation
- Missing requirements: requirements not captured or discovered later.
- Unnecessary Functionality: - Redundant or unnecessary functionality
- Erroneous Requirements: - Incorrect requirements due to lack of understanding, miscommunication etc.
- Clarifying Requirements: - rewording requirements text for clarification

Origin of Change

Origin of change corresponds to the source of the proposed change, that is, where it was originated. The sources or requirements change were found to be the:

Client, Project Management Consideration, Status Meetings, Technical Team Discussion, Functional Review Meetings Development Team, Defect Reports, Developer's Detailed Analysis, Design Review Feedback.

5.1.3 Stage Three: Categorization of Requirements Change

Based on the collected information in Stage 2, changes in requirement were divided into three categories namely; division of change, category of change, and classification of change. This categorization helped us understand the causes of requirements change and the purpose was to

improve our understanding of the nature of requirements change. Change data from the two projects was analyzed and presented in Table 6 and 7.

Project A: SDE Project

Data Collection		
<i>Project A: SDE</i>		
Division of Change		
Change Request: Client	17	71%
Change Request: Dev Team	7	29%
Total Change Request	24	
Category of Change		
GUI Change Request	2	8%
Functionality Change Request	15	63 %
Business Logic Change Request	7	29%
Classification of Changes		
Add Functionality / Requirement	14	33%
Modify Functionality / Requirement	8	59%
Delete Functionality / Requirement	2	8%

Table 6: Taxonomy of Change: SDE

Project B: DataDive 2.0 Project

Data Collection		
Division of Change		
Change Request: Client	12	100%
Change Request: Dev Team	0	0%
Total Change Request	12	
Category of Change		
GUI Change Request	8	67%
Functionality Change Request	2	16.67%
Business Logic Change Request	2	16.67%
Classification of change		
Add Functionality	3	25%
Modify Functionality	4	33.3%
Delete Functionality	3	25%
Other (Not related to requirements)	2	16.67%

Table 7: Taxonomy of Change: DataDive 2.0

The purpose for this categorization is to show which areas are most affected by in this case study due to the changing nature of the requirements. This categorization is also helpful in understanding the major areas of concern for management when it comes to global software development and requirements change and highlight the need for special attention in GSD projects. As a result of our analysis our understanding improved regarding the nature, attributes and the driving causes for the change. Our process of analysis was iterative and we have derived the classification of our changes inductively and also in light of the change classification presented by Nurmuliani et al (2003).

5.2 Requirements Change in Light of Change Taxonomy by McGee & Greer

This research work utilized the taxonomy by McGee and Greer, 2009 as a model for the discussion on root causes of requirements change in GSD. Although the taxonomy is not exclusive to GSD projects but it provides a comprehensive collection and categorization of causes of requirements change. The causes of requirements change were mapped to the taxonomy for verification of their presence in globally distributed environment.

We analyzed the collected empirical evidence on the bases of taxonomies of change by McGee and Greer, 2009. The taxonomy proposed by McGee & Greer (2009) uses change trigger constructs to describe the origin or source of change. These constructs (although not specific to GSD) have previously not been validated (McGee & Greer 2009), we used these constructs as guides to highlights their applicability in a GSD context. The sources of change (Market, Customer Organization, Project Vision, Requirements Specification and Solution) together make the Change Domain. The reason to try and map these triggers in a GSD environment is because they provide a comprehensive set of constructs for requirements change in single site development and are also found useful in the GSD context. However the degree of influence in GSD context is different than in single site development.

5.2.1 The Root Causes (Most Influential Triggers) of Requirements Change

There is scanty empirical evidence available that provides causes of requirements change which is very surprising as requirements changes is quoted as one of the major causes of project failure. Owing to this need our research work conducted the case study and collected data for the causes of requirement change management in two global software development projects.

These causes were categorized and mapped on to a taxonomy based by McGee and Greer 2009. It was used as a base model to map the causes identified as being most influential in our case study shown in Figure 10.

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:

- Resolution of Misunderstanding
- Resolution of miscommunication
- Incorrect Requirements Identified
- Increased client's Understanding of the problem and Solution
- Increased development team understanding of the problem
- 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. One reason that can be associated with this can be the process of work evaluation, which composed of modular development and validation through the third party testing team hired by the client. During these sessions client and its testing team identified the required changes based on their improved understanding of the application. These were notified in the CRF and later on implemented after approval from CCB.

The projects studied used emails, video conferencing, requirements negotiations for coping with these influential triggers. Some of the triggers during these projects were influential as mentioned above however some others such as First Engagement of Particular User Representative, insufficient sample of user representatives, development team stability, low staff morale, logical complexity of the problem, incorrect user involved and age of the requirements, did not play a very significant role for introducing requirements change and were not as influential.

5.2.2 Most Influential Uncertainties that Trigger Requirements Change

In the category of Requirements Specification the uncertainties mentioned by McGee & Greer, 2009 were identified as the main contributors towards triggering a change in GSD environment. In the following Section 5.3 the description is given about how these uncertainties were managed to have minimal impact of the challenges imposed by GSD on the projects undertaken. Hereunder we present a list of those most influential uncertainties as under;

- Availability of communication with customer
- Quality of communication between Analyst and Customer
- Analysis Techniques
- Quality of requirement specification
- Analyst Skill / Experience
- Quality of Development team communication
- Involved customers knowledge/understanding of problem

The project being a globally distributed development work was heavily dependent upon quality communication and coordination and that is why these uncertainty factors were classified as most influential. The SDE Project was a small sized project and was completed on time, within budget and was acceptable to the user so it can be classified as a successful project.

The communication channels used, email, chat (between Manager, developers and analyst), video conferencing and group meetings paved the way for a successful communication and as a

result lead to successful project. Inappropriate or deficient communication could have triggered many changes however this did not happen and as a result only two (8%) of the total changes were caused by miscommunication or misunderstanding.

Some of the technical changes that were required were mainly caused by the lack of domain knowledge of the development team which increased with the usage of application as it was being developed and visualized. The client also made various change requests once the system module or iteration was ready for testing and the functionality could be visualized. Both the client and development team's increased understanding of the application appeared to be the most influential factor (80%) for requirements change.

Both projects studied used an online tool as bug reporting tool. This helped development teams in USA, Pakistan and the client's testing team with clear visibility of the work carried out at different locations. It also acted as means for collaboration tool for managing, traceability and progress monitoring for the stakeholders involved.

5.3 Discussion: Triggers and Uncertainty (modified after the original Authors)

We have modified the taxonomy used by McGee and Greer, (2009) to show that there is a link between the factors mentioned in the Change Domain column. The analysis also revealed that these factors influence other factors individually and in combination with each other in order to trigger change. We highlighted the most influential factors as found out in our study but at the same time these factors are so intertwined that it is difficult to specify them individually as a root cause. Most of the time presence of one factor of uncertainty or trigger influences another factor and the degree of which is not measureable.

Some of the factors have been pointed out by this study in 5.2.3 as most influential triggers that may be classified as root causes however many other factors can also be candidates of root causes of requirements change such as the following

- Change in Customer Needs (based on market situation, stability or competition)
- Strategic or Organizational, Governmental policy Change
- New Technology Adoption

We have not put them as the root causes in this case study as their influence in the projects studied was not present. This leads us to the understanding that giving an exhaustive list of causes of requirements change applicable to all projects is not only difficult but rather impractical as each project carried out in global context is different from the other.

The last factor 'Incorrect Requirement' mentioned in the list above is given not as a root cause but as the effect of a combination of factors. For example, improved understanding of the problem or system after usage may cause identification of an incorrect requirement. This may also result from incorrect or inadequate specification which gets realized once better understanding is developed both by the client and the development team about the system.

5.4 Modification of the Taxonomy into a Model

The original change source taxonomy discussed by McGee and Greer, 2009 shows no link or influence of various categories on the change domain and they appear as entities independent

from each other as shown in Figure 8. However our experience of the studied case revealed that the categories are not independent from each other, rather they have a combined effect which ripples through to Requirements Specification and may trigger changes in other entities on the change domain. For example changes in the *Market* situation affects the *Customer Organization* that can be structural, strategic etc. which in turn cause the *Project Vision* and as a result of the combination of these factors the *Requirements Specification* has to be changed. A significant change in the Requirements Specification may result in the tools and technology adopted, as a *Solution for* the project. Therefore this research work does not see these entities as independent from each other. The most influential factors of change for this case study and the interdependency of entities on the change domain can be described in the figure below;

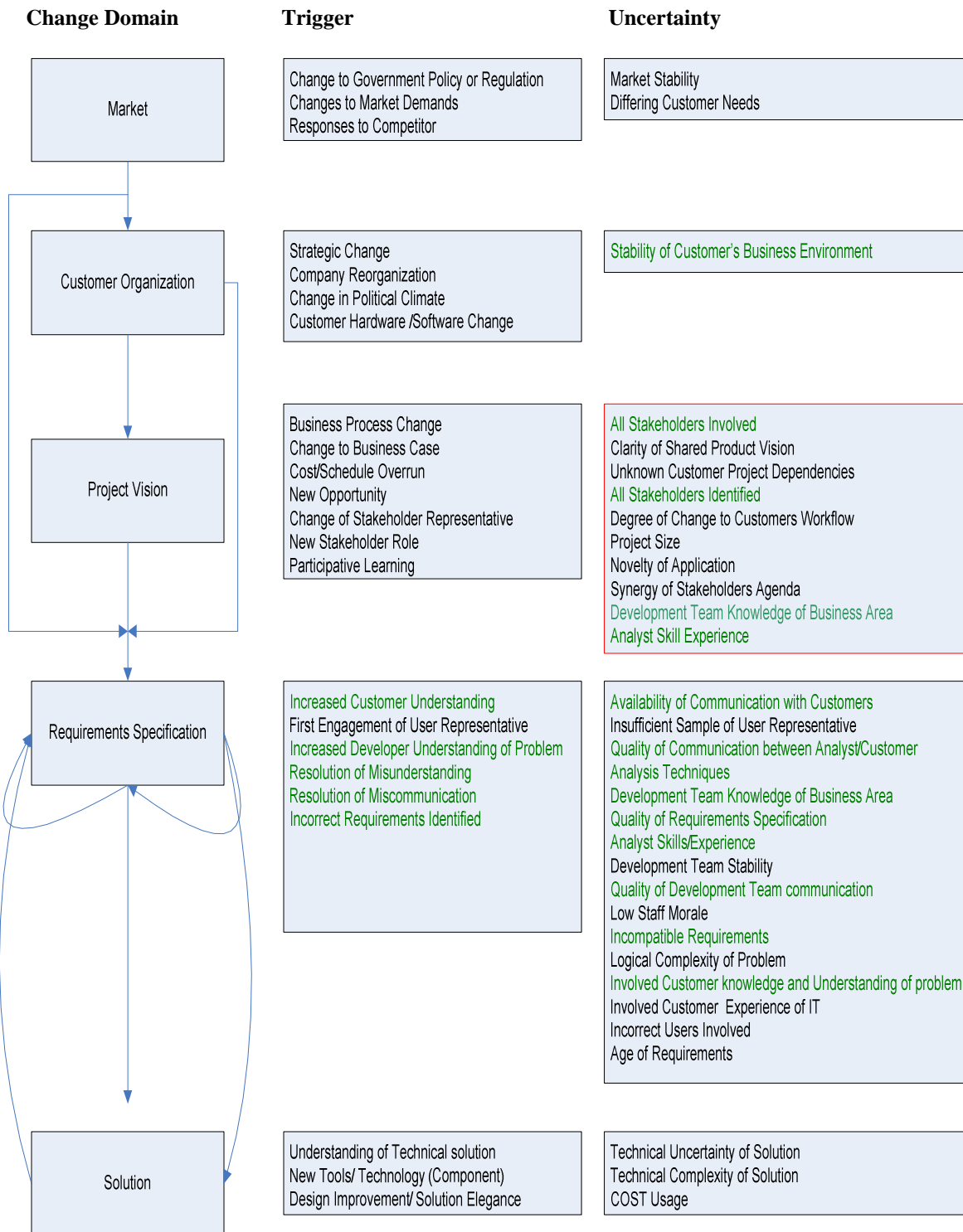


Figure 8: Modified Software Requirements Change Source Taxonomy, Adopted from McGee & Greer 2009.

The above figure shows the role of different entities in the change domain and how some of them affect as a whole the Requirements Specification. The figure illustrates that the Market, Customer Organization, Project vision may individually or in a combination can effect or trigger change in the Requirements specification. During our case study the uncertainty factors and

triggers (highlighted in green) which were found significant in driving change for the Requirements Specification entity *after* the initial requirements specification document was drafted are highlighted. It is important to note that the taxonomy of McGee and Greer, 2009 is used as a guide and these factors are matched with the factors of change identified during the case study with the help of CRF log, interviews and other related document. While there may appear similarity in the factors contributing to change in other similar web application development projects carried out in the GSD context the factor highlighted here are specific to this case study and may not be generalized to all other GSD projects.

We have used the following collection of uncertainty constructs list, which is quite exhaustive set of the major causes of requirements change found throughout the literature and also some additional causes added by the authors McGee and Greer. We have used this comprehensive list to verify our findings regarding the causes of requirements change in our case study and compare them with the ones given in the following list. This gives a very good idea, although a number of the causes are similar but not all causes described for requirements change in the literature are necessary to be present in the global software development project. Another important factor to note is that the causes given in the original list are stated without any reference to their influence in a particular case. We have commented on the most important causes and how much and how influential they have been in this particular globally distributed development project. The aim is to highlight the fact that there are certain causes which are more influential in the GSD projects which are not as important or influential in other forms of software development (collocated or single site software development).

5.5 Influence of Uncertainty Constructs on GSD Projects

This research work has modified the list of uncertainty constructs given by McGee and Greer, (2009) by providing two additional columns, 'Influence' and 'Comments'. The original list, although very comprehensive, does not specifically deal with uncertainty constructs (causes that may lead to requirements change) in GSD context. This research work modifies it to map these causes or uncertainty constructs in distributed development environment and show, based on our studied case, which causes we found to be more influential than other in our research as shown in Table 8. The compiled list by McGee and Greer is a comprehensive set of uncertainty construct found in the literature and serves a good purpose to base our observations regarding causes of requirements change in GSD.

Uncertainty Construct	Influence	Construct Mentioned by author(s)	Comments
Use of Prototype	√	Boehm, (2000); Harker, (1993)	Positive effect
New Stakeholder (role)		Boehm, (2000)	Clients Composite role; but with no negative impact
Customer Company Reorganization		Lamsweerde, 2009; Nurmuliani et al. 2004, Harker, (1993); Boehm, (2000); Pressman 2005	N/A
New solution Tools/technology		(1993); Boehm, Lamsweerde, 2009, Curtis, 1988; Harker, (1993); Wiegers 2003	N/A
Change to government policy or regulation		Lamsweerde, 2009, Harker, (1993) Wiegers 2003	N/A
Participatory Learning	√	Harker, (1993)	Positive effect and participatory learning due to user involvement during design and development
Local Customization		Harker, (1993)	N/A
Customer migration to new solution		Harker, (1993)	N/A
Customer needs change	√	Christel and K. Kang, 1992; Nurmuliani et al. 2004, Pressman 2005	Present but due to more accurate SRS and stable business environment it did not change drastically
Developers Increased Understanding of problem	√	Curtis, 1988; Nurmuliani et al. 2004; Mathiassen et al. 2004	One of the major causes of change in this research
Scope Reduction		Nurmuliani et al. 2004	N/A
Changes to packaging/licensing/branding		Nurmuliani et al. 2004	N/A
Solution Elegance (Design Improvement)	√	Curtis, 1998; Nurmuliani et al. 2004	Present but minimal impact
Resolution of miscommunication	√	Nurmuliani et al. 2004	Present and resolved but with communication overhead
Testability	√	Nurmuliani et al. 2004	Performed but no negative impact
Business Process change (continuous improvement)		Sommerville., 2007; Wiegers 2003	N/A
Response to competitor		Curtis 1988,	N/A
Functionality Enhancement	√	Nurmuliani et al. 2004	Present, mostly due to increased understanding and managed within schedule
Defect Fixing	√	Nurmuliani et al. 2004	Present but managed within schedule
Redundant Functionality		Nurmuliani et al. 2004	Present , minimal
Missing Requirement Identified (Not a reason/source)	√	Nurmuliani et al. 2004	Present, minimal
Clarification of Requirement	√	Nurmuliani et al. 2004	Clarification sought through status meetings, discussions and sometimes causing requirements change
Increased customer understanding	√	Lamsweerde, 2009; Sommerville, 2007; Wiegers 2003; Mathiassen et al. 2004	One of the major causes of change in this research
New Class of User		Lamsweerde, 2009	N/A
New Usage Condition		Lamsweerde, 2009	N/A
New way of doing things		Lamsweerde, 2009	N/A
Correction to Requirements specification		Lamsweerde, 2009	Present, minimal
New Opportunity		Lamsweerde, 2009	N/A
Change in the use of the information		Mathiassen et al. 2004	Unclear construct
Cost or schedule overrun		Lamsweerde, 2009; Pressman 2005	Project did not suffer cost of schedule overrun. The changes that were to threat existing schedule and cost were analyzed, evaluated negotiated and were either

			deferred or left.
hange to Customer's hardware/software		Nurmuliani et al. 2004(b)	Present and did drive numerous changes due to integration
System Usage (after installation, not prototype)	√	Boehm, (2000); Sommerville, 2007; Harker, (1993)	Present , minimal
Changes to Market Demands		Wiegiers 2003; Harker, (1993); Nurmuliani et al. 2004 (a); Pressman 2005	N/A
Resolution of Conflicting Requirement	√	Nurmuliani et al. 2004 (a)	Present, not very significant
New Functional Feature New Requirement	√	Lamsweerde, 2009	Performed till late in development cycle.
Improved Quality Feature	√	Lamsweerde, 2009	Performed till late in development cycle.
Result of Change in political climate (needs of particular group emphasized)		Christel and Kang, 1992; Wiegiers 2003; Curtis, 1988;	N/A
Change to customer's environment		Sommerville, 2007	N/A
Changes in Underlying technologies		Curtis, 1988	N/A
Incorrect Requirement Identified	√	Nurmuliani et al. 2004 (a);	Present but not very significant
Resolution of Misunderstanding	√	Curtis, 1988	Found and eliminated but with communication overhead and performed through both synchronous and asynchronous communication
First or re-engagement of user representative Added		McGee & Greer 2009	N/A
Change to business Case (Return on Investment, Total cost of Ownership)		McGee & Greer 2009	N/A
Customer Organization Strategic Change (New Marketing/Sales direction, change to organization goals)		McGee & Greer 2009	N/A
Change of Stakeholder Representative		McGee & Greer 2009	N/A
Analyst skill or experience	√	Ferreira et al. 2003; Christel and Kang, 1992; Moynihan, 2000; Wiegiers 2003	Positive influence of well experienced analyst resulting in less misunderstanding between client, analyst and the development team
Development team knowledge of business area	√	Christel and Kang, 1992; Moynihan, 2000;	Lack business domain knowledge resulted in lack of understanding of user needs. However managed in later stages of the project by increased user involvement, analyst interaction and participatory learning.
Analysis techniques employed (workshops, interviews, modeling etc)	√	Ferreira et al. 2003; Wiegiers 2003; Sommerville, 2007; Pressman 2005; Lamsweerde, 2009	Present, positive impact
Novelty of product (business novelty)		Christel and Kang, 1992; Mathiassen et al. 2004; Sommerville, 2007	N/A
Logical complexity of problem	√	Mathiassen et al. 2004; Moynihan, 2000; Christel and Kang, 1992	Present but managed within available resources
Availability of communication with customer	√	Mathiassen et al. 2004; Wiegiers 2003	Good communication channels resulting in positive impact on the building trust and project's successful outcome.
Involved customer's knowledge/understanding/clarity of requirements	√	Christel and Kang, 1992; Davis and Nori, 2007; Moynihan, 2000; Wiegiers 2003	Good understanding and clarity hence positive effect
Quality of Communication between analyst and	√	Christel and Kang, 1992; Mathiassen et al. 2004; Davis	Quality communication resulting in positive impact on

customer		and Nori, 2007; Wiegers 2003	the building trust and project's successful outcome.
Involved customers experience with working alongside IT to produce solutions	√	Moynihan, 2000	Experienced customer, positive effect
Diverse User Community		Moynihan, 2000; Sommerville, 2007; Wiegers 2003	N/A
Incompatibility between requirements		Lamsweerde, 2009	N/A
Lack of well understood model of utilizing system		Mathiassen et al. 2004	N/A
Lack of well-understood model of the utilizing system		Mathiassen et al. 2004	N/A
Lack of structure for activity or decision being supported		Mathiassen et al. 2004	N/A
Stability of Customers Business Environment	√	Christel and Kang, 1992	Present, positive influence
COTS usage		Lamsweerde, 2009; Boehm, (2000);	N/A
All stakeholders identified	√	Ebert and Man, 2005, Boehm, (2000)	Present, positive influence
All Stakeholders involved	√	Ebert and Man, 2005; Boehm, (2000); Christel and Kang, 1992; Wiegers 2003	Present, positive influence
Clarity/unity of shared product vision	√	Moynihan, 2000; Harker, (1993); Ebert and Man, 2005; Wiegers 2003	Present, positive influence
Synergy of stakeholder agenda		[28], [31], [14] Lamsweerde, 2009; Moynihan, 1997; Harker, (1993)	Present, positive influence
Unknown Customer Project Dependencies		Ebert and Man, 2005	N/A
Market Stability	√	Curtis et al. 1988; Davis and Nori, 2007; Harker, (1993)	Present, positive influence
Differing Customer Needs		Curtis et al. 1988	N/A
Type of user doing specification (incorrect user involved)		Wiegers 2003; Moynihan, 2000; Mathiassen et al. 2004	N/A
Change in the utilizing system		Mathiassen et al. 2004	N/A
Low Staff morale		Ferreira, 2003	Not present, positive influence
Large number of users		Mathiassen et al. 2004	N/A
Level of participation of users in specification	√	Mathiassen et al. 2004	Active participation by users in specification
Lack of user experience of utilizing system		Mathiassen et al. 2004	N/A
Degree of Change to customers' workflow		Moynihan, 1997; Wiegers 2003	Negligible, positive influence
Quality of Requirements specification	√	McGee & Greer 2009	Well defined requirements
Technical Uncertainty of Solution	√	McGee & Greer 2009	Minimal, positive influence
Technical Complexity of Solution	√	McGee & Greer 2009	Present, handled well within resources, positive influence
Quality of Development team Communication	√	McGee & Greer 2009	Good quality communication using collaborative technologies (chat , email, video conferencing, phone calls etc)
Age of Requirements (elapsed time since completion of requirements documentation)		McGee & Greer 2009	Age of requirements was not an issue as the project was not very long.
Insufficient Sample of User Representatives		McGee & Greer 2009	N/A
Development team (PM and analyst) stability	√	McGee & Greer 2009	Staff remained stable during project

Table 8 Modified Table of Uncertainty Constructs Used by McGee and Greer 2009

5.6 Modeling the Existing Process of Change Management

The change management process was studied that was in use by this organization. The model was drawn (Figure 9) based on the interviews of the people involved and the related documents used in the process. This process model is used to communicate and manage changes during software product development. Our study opens the door of opportunity for the process related problem identification and also process improvement.

The change management process is triggered by change request from any of the key stakeholders (Analyst, Development team, QA team and the client). This change request is formalized into a Change Request Form and sent to Change Control Board (CCB) for analysis and negotiation with the stakeholders. 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.

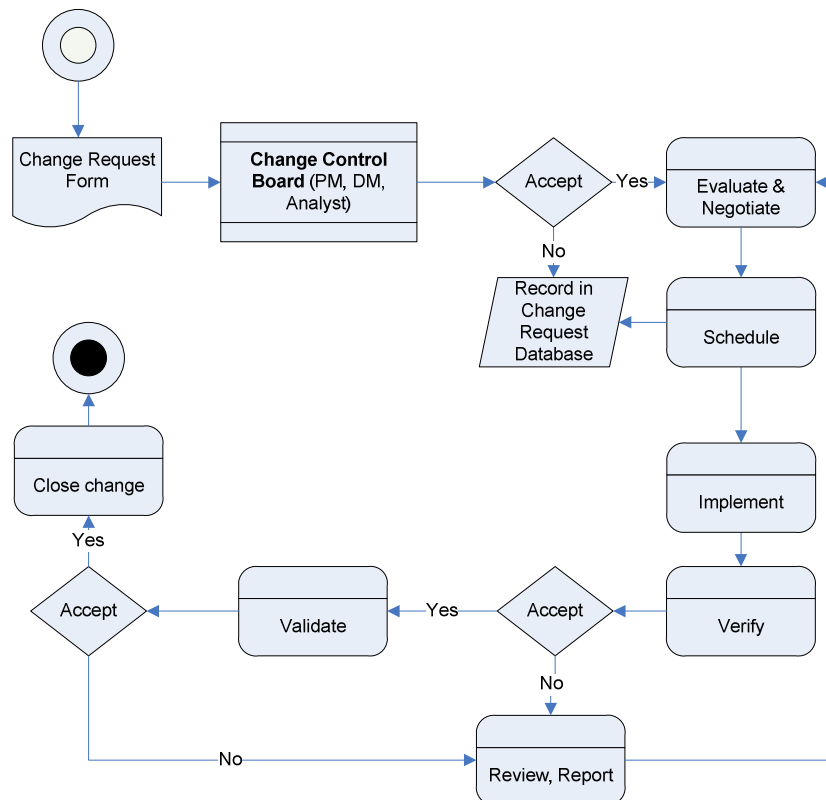


Figure 9: Model of existing Requirements Change Management Process

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 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. Certain change requests initiate from the bug identified and noted by the Quality Assurance (QA) team and in this case noted as 'tickets' in the online bug reporting and tracking system used by GSD Inc. These bugs may contain change requests for addition, modification and (or) deletion of certain functionality.

5.7 Summary of Change Management Process Analysis

In summary, our analysis identified the following problems were identified in the change management process at GSD Inc.;

5.7.1 Problems with Change Management Process

The analysis also revealed that there is limited information about the rationales of the proposed change.

The consolidated sheet which is used to record the important information was found to be missing some necessary information and for this reason was completed with the help of interview data where possible.

Another deficiency found in this consolidated sheet was that there is no record of the tickets submitted online (for bugs identified and changes that followed) and these had to be studied independently from the data contained in the consolidated sheets.

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 but not enough attention is devoted to putting down the complete rationale of change requested which makes the analysis and evaluation difficult.

The analysis highlighted that not enough impact analysis of the proposed change is performed which sometimes results in wrong conclusions about the impact of the change (on artifacts, code, schedule etc.) might have and as a consequence a mismatch occurs between the resources and time allocated.

Communication of and recording of CRF form is bound and limited to the implementing site and not electronically distributed to other site.

We also found certain problems with the existing process that nearly all the technical modifications and changes which are considered 'small' by the development teams by both Team Pak and Team US are not recorded. This is a deficiency in the process and because of which the time consumed, the code written, the artifacts and modules affected cannot be traced. Also the absence of such data makes the analysis less effective.

Our analysis also revealed that there is no record of how the rejected or deferred changes are handled and stored in the company and at what stage (during multiple

iterations of the project development) they are included back into the project work. There is no mentioning of how these changes are included into the change request process again.

The analysis also highlights the problem of synchronization of available data online regarding the changes due to bugs identified with the CRF.

The analysis also indicated that there was no central database available for all the teams to view and keep themselves updated regarding the archived change request form. 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. However if the sites were to be added in a more globally distributed project and the number of change request forms and data is included for a longer period there will be an issue of lack of visibility into the project operations and past progress.

5.7.2 Recommendations

Therefore at the end of our analysis we gave following recommendations to the company;

- Fill all the required Information in the Change Request (CR) forms especially change rationale
- Perform detailed impact analysis
- Integrate online collaborative tool (bug reporting tool) information with RCM artifacts
- Record technical changes (even on small scale) made by development teams
- Define the process of including deferred or rejected changes in the development process.
- Develop central database for recording all changes as archived data for evaluation purposes

It is hoped that the process activities performed for requirements change management at GSD Inc. in conjunction with the proposed recommendation can improve the existing process. The resulting process after the inclusion of these recommendations can be adopted by other similar organizations working on distributed environment

5.8 The Proposed Model for Requirements Change Management for GSD

The proposed model depicts the complete requirements change management process in Global Software Development environment. This study has proposed RCM model that includes most of the roles, artifacts and activities mentioned by Imtiaz et al 2008, p.123 and in their proposed model for requirements change management. Their model is not developed for GSD environment, whereas the proposed model in this research work is developed specifically for GSD projects.

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. In the Figure 10, 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 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 $CLL1$, 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 if 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, 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.

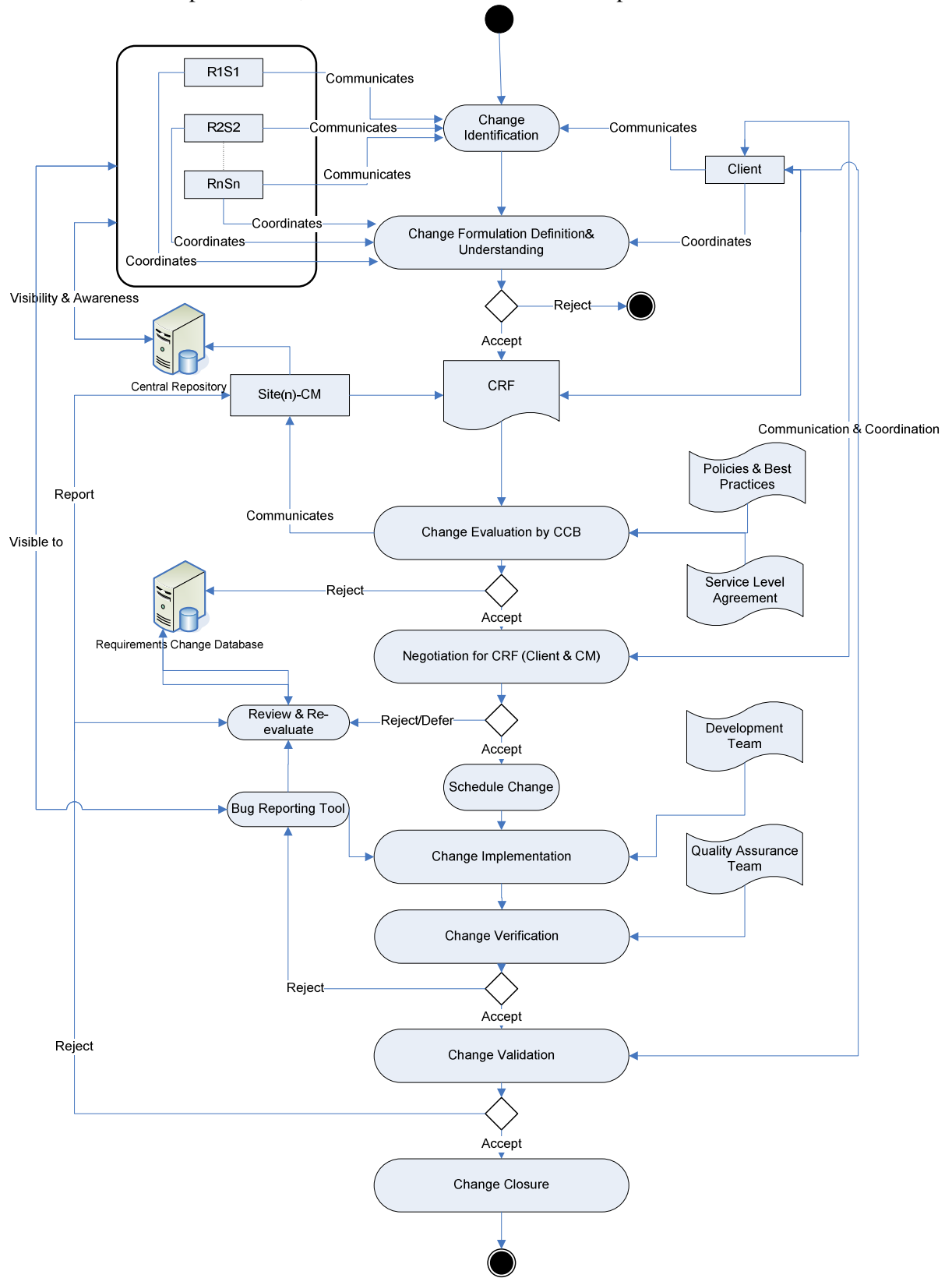


Figure 10: Proposed Model for Requirements Change Management in GSD

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. However if it is not validate it reported in the Bug Reporting Tool for review and further work by the development team till it gets resolved. With the closure of the implemented, verified and validated change and the updation of related artifacts the change management process comes to an end.

5.9 Application of the Proposed Model

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 as shown in Table 9.

5.9.1 Proposed 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

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 & Zowghi 2003) 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.

5.9.2 Proposed Model and Identified Root Causes of Requirements Change

In section 5.2.1 we discussed the identified root causes based on our case study at GSD Inc. Hereunder we discuss what role the proposed model plays in minimizing the effects of root causes of requirements change.

Resolution of Misunderstanding

Common understanding, common definition and common goal of the required changes which are discussed in '*Change Formulation Definition and Understanding*' activity reduce the likelihood of misunderstanding. Once the change is formulated and evaluated by CCB, the change with its full impact is negotiated with the client further eliminating the chances of misunderstanding and possible resolution of misunderstanding which took place in the earlier activity. The proposed model shows the link between stakeholders from all the related sites, their collaboration and communication with each other which also allows better means to resolve possible misunderstanding regarding requirements. The good thing about this activity is that it is kept at the top of other

activities after identification of change and done at very early state of the process. This makes it less costly and easier to rollback if something goes wrong.

Resolution of miscommunication

Global software development is collaboration and communication intensive mode of software development. The chances of miscommunication increase in this form of development. However the model shows that all related key stakeholders from all development sites are in contact with each other with the help of different activities. They come into frequent contact with each other many times during the change management process such as 'Change Identification', 'Change Formulation Definition and Understanding', 'Negotiation', 'Verification' etc. This frequent contact (done through meetings enabled by Technology such as video conferencing etc.) is extremely effective in terms of reducing and resolving any miscommunication.

Incorrect Requirements Identified

This type of root cause for requirements change is not directly related to change management process rather it is more related to requirements specification document. The proposed model does not solve this problem directly and has no control over eliminating this cause of change. However because of the frequent meetings through initiation and formulation of change and also because of frequent verification by the QA team and validation by the client such incorrect requirements are identified earlier. It minimizes the huge cost involved in identifying and correcting such incorrect requirement at later stages of development.

Increased client Understanding of the problem and Solution

The client understanding increases with the use of prototypes or validation of application versions or modules. This increased understanding needs requirements change and the indirect benefit of this model is that it is meant for managing such kind of requirements changes in a better and more efficient manner. The same argument covers the other two identified root causes of requirements change i.e. 'Increased development team understanding of the problem' and 'Increased Domain Knowledge of Development team' and this model does not discourage changes rather it minimizes the adverse affects of change as much as possible.

The model improves the challenging affects of by the root causes of requirements change such as cultural and communication by activities such as frequent reviews, client involvement, negotiation and verification. This improves and clarifies misunderstandings early which generally get identified late in a typical GSD project and hence drastically affect the schedule and cost. The model provides the activities, roles and artifacts involved to collaborate by means of technology and communication (formal, informal, synchronous, asynchronous etc.) As a result visualization of the process is 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.

We believe that the model will benefit requirements change process if some other factors or suggestions are taken into account when implementing the proposed model. For example, if well trained, experienced and culturally aware of clients business and work activities are assigned as key people that interact with remote sites the quality of communication, collaboration and trust will improve. On the other hand this would result in less misunderstanding and miscommunication. Also if the number of sites of

remote sites is kept minimal and some of the working hours are kept flexible with some overlapping hour present between the sites this will also provide more control and visibility over the activities of the project development. 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).

Socio cultural challenges can be solved by introducing, language, culture Email (formalizing communication) Introducing fast channels for response for immediate feedbacks incase of misunderstanding and also breaking the formal red tape and barriers to communication and trust with informal and frequent communication via (such as text or voice chat) Holmstrom et al 2006.

	Nurmuliani et al Process	Crnkovic et al Process	Dick Process	Lam & Shankararaman Process	Messaadia et al Proceses	Keller Process	Anderson & Felici Process	Proposed Process Model
Roles								
Change Initiator	x	√	x	x	x	x	x	√
Engineering Manager	√	x	x	x	x	x	x	x
Project Manager	√	x	x	x	x	x	x	√
Development Team	√	√	x	x	x	x	x	√
Quality Assurance Team	√	x	x	x	x	x	x	√
Reviewers and Approvers	√	x	x	x	x	x	x	√
Customer/Client	x	x	x	x	x	√	x	√
Change Manager	x	x	x	x	x	√	x	√
Change Advisory Board / Change Control Board	x	x	x	x	x	√	x	√
Change Builder	x	x	x	x	x	√	x	x
Release Manager	x	x	x	x	x	√	x	x
Artifacts								
Change Request Form	√	√	x	x	x	√	x	√
Service Level Agreement	x	x	x	x	x	√	x	√
Policies and Best Practices	x	x	x	x	x	√	x	√
Forward Schedule of Change	x	x	x	x	x	√	x	√
Configuration and Management Plan	x	x	x	x	x	√	x	x
Activities								
Anticipate/Predict Change	x	x	x	√	x	x	x	x
Change Initiation	√	√	x	x	√	√	√	√
Change Classification	x	x	x	√	x	√	x	x
Change Planning	√	x	x	√	x	x	x	x
Change Evaluation Detailed Description of Change	√	x	x	x	x	x	x	√
Impact Analysis (IA)	√	√	√	√	√	√	√	√
Prune and Elaborate IA	x	x	√	x	x	x	x	x
Change Estimation	x	x	x	√	x	x	x	√
Change Review and Approval	√	√	x	√	√	√	x	√
Negotiation Process	x	x	x	√	x	x	√	√
Schedule Change	x	x	x	x	x	√	x	√
Change Definition	√		x		x	x	x	√
Change Implementation	√	√	√	√	√	√	√	√
Configuration Management or change Documentation	x	√	√	√	x	x	x	√
Integrity Assurance	x	x	x	√	x	x	x	x
Change Verification/Validation	√	x	x	x	√	x	x	√

Table 9: Comparison of Proposed Model with Roles, Artifacts, Activities identified by Imtiaz et al 2008, p.123.

5.10 Limitations of the Proposed Model

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 same is applicable to the identification and description of the artifacts involved in the process. The reason why it is considered difficult to name all the involved artifacts is the same as was given for the roles; different companies may use different artifacts to use for the change management process. If on the other hand all the sites use standardized roles and artifacts for the change management process this limitation may be eliminated.

Another limitation of this model is lack of description of the mediating technology that maybe employed for communication and coordination purposes. Although many of these technologies are in use for GSD projects but because this model needs to be applied in practice first and experimented with the available technologies it is difficult to prescribe the technology to be used.

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.

Chapter 6 Summary and Conclusion

6.1 Summary

This thesis work reports the findings from a case study that explored the root causes of requirements change management in Global Software Development. It discusses important challenges that a geographically distributed development project faces due to requirements change. The research work highlights the most influential causes of requirements change and their impact on change management. The result of analysis showed that it is difficult to come up with a list of root causes that are applicable to all globally distributed projects. The reason for this is because there is a complex relationship between the factors that trigger requirements change. These factors are so intertwined that isolating and specifying them is very challenging. The elements on the change domain also work together to instigate requirement change. The analysis revealed that defined and established change management processes, proper communication channels, lesser number of development sites, active stakeholders and stable business environment played vital role in the success of these projects.

The two main contributions of this research work have been the proposed model of requirements change management in GSD context and the process improvement guidelines given to the organization studied. The guidelines resulted from the detailed investigation of the change management process employed by the organization and the subsequent modeling of the process. This leads to the identification of problems with the existing process. Some guidelines were proposed for process improvement.

The proposed model is flexible to accommodate multiple development sites for managing requirements change. The model stresses the need of recognition for the fact that extra communication and coordination overhead has to be planned while taking a decision on going global with the development of software projects. The model is aimed at describing the process of change management for both the organizations performing GSD projects as well as the ones thinking about embarking on this journey. The model remains to be tested in the real time software project that would validate its effectiveness rather than its accuracy and efficiency.

6.2 Conclusion

Change in software requirements is necessary and unavoidable. Managing frequent requirements change is an uphill task and is even harder when performed at geographically distributed locations. However still there are numerous system errors that can be traced back to requirements misunderstandings, miscommunications or mismanagement. Difficult as it maybe but proper management of requirements change often leads to completion of the development work within the allocated time and possibly within budget. Failure to do so may severely curtail the chances of project success and may rather result in failure.

Finding the root causes of requirements change in a globally distributed project leads to better understanding of how to cope with the challenges of software development likely to occur due to frequent requirements change. It is more important to manage frequent change that might result in extra communication and collaboration overhead in GSD which is already a communication intensive endeavor. This may also result in managing

change in a significantly better and well informed manner and as a result increases accuracy in planning, scheduling and resource delegation.

Due to the importance of proper management of changing requirements many models have been proposed in the literature however none of these models is specifically developed for GSD environment. Our proposed model is a modest initiative in that direction. It is proposed for the purpose of model verification for its effectiveness rather than efficiency in all types of GSD projects. The model is supposed to improve on the understanding of roles, artifacts and activities involved in GSD projects. This would result in improved visualization of the change management process specific to the global software development context.

User involvement and use of good communication and collaboration greatly improves the chances of project success. This is why the proposed model stresses the importance of intense communication and involvement of the client and collaboration among different teams through collaborative technology. It would increase the chances of early identification of potential changes early in the development process when it is less costly and cheap to fix them rather than later in the development process. The outcome should be a substantial contribution to the realization of the goal to minimize the adverse impact of changing requirements on the project objectives.

This research work was set to answer the question, “what are the root causes of requirements change in GSD project? We have answered this question and given a list of the root causes found in the case study performed at GSD. Inc. The other question which was meant to be answered was, how requirements change is managed in GSD projects?, This research work has evaluated the change management process in two GSD projects, and has modeled existing process. The author has highlighted the problems in the existing process as well as proposed a model of requirements change management specific to GSD environment.

The results of this study provide some root causes of requirements change, the mentioned causes may not be same for all projects in GSD as there are factors on the change domain, such as market, technology solution, and SRS have different type of uncertainty constructs and triggers that instigate change. However we believe that in a stable business environment and with not much technological and competitor pressure a GSD project will have similar root causes of requirements change.

6.3 Future Work

In the future work it is proposed that the model should be taken up and implemented on a global software development project for its testing and validity. The limitations mentioned regarding the proposed model need to be improved on as a future direction of this work. Also it is encouraged that a similar analysis be performed on a project of a longer duration with more requirements change request to verify our findings.

References

- Bianchi, A. Caivano, D. Lanubile, F. Rago, F. Visaggio, G., 2008. Towards Distributed GQM, [Online], Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.9136> [Accessed 24 October 2009]
- Boehm, B., 2000. Requirements that Handle IKIWISI, COST and Rapid Change. *Computer*, 33(7).
- Boehm, B., 1987. Industrial Software Metrics: Top 10 List? *IEEE Software*, 4(5), 1987.
- Brooks, F.P., 1987. No Silver Bullet, Essence and Accidents of Software Engineering, *IEEE Computer*, vol.20 no.4, 1987, pp.10-19.
- Carmel, E., 1999. *Global Software Teams: Collaborating Across Borders and Time Zones*, Prentice Hall.
- Carmel, E. Dubinsky, Y. Espinosa, A., 2009. Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory Field Study, *Proceedings of the 42nd Hawaii International Conference on System Sciences*.
- Casey, V. Richardson, I., 2009. Implementation of Global Software Development: A Structured Approach. *Software Process Improvement and Practice Process Improve. Pract.* 2009; 14: 247–262 Published online 23 June 2009 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/spip.422.
- Cataldo, M. Herbsleb, D., 2009. End-to-End Features as Meta-Entities for Enabling Coordination in Geographically Distributed Software development. *International Conference on Software Engineering Proceedings of the 2009. ICSE Workshop on Software Development Governance* Pages: 21-26 Year of Publication: 2009 ISBN:978-1-4244-3736-7.
- Christel, M. Kang, K., 1992. Issues in Requirements Elicitation. *Software Engineering Institute*, Technical Report No. CMU/SEI-92-TR-012.
- Conchuir, E. Holmstrom, O. Olsson, H. Agerfalk, P. Fitzgerald, B., 2009. Benefits of Global Software Development: Exploring the Unexplored. *Software Process: Improvement and Practice*, 14(5), Date: September/October 2009, p. 301. ,[Online], Available at <http://www3.interscience.wiley.com/cgi-bin/fulltext/122579539/PDFSTART> [Accessed 21 October 2009]
- Cresswell, J.W., 2009. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 3rd ed. Thousands Oaks, CA. Sage Publications.
- Curtis, B. Krasner, H. Iscoe, N., 1988. A field study of the software design process for large systems. *Communications of the ACM*, 31(11).
- Damian, D., 2001. An Empirical Study of Requirements Engineering in Distributed Software Projects: Is Distance Negotiation More Effective. In: *APSEC, 8th Asia-Pacific Software Engineering Conference*, Macau, China. 4-7 December 2001.
- Damian, D., 2006. Requirements Engineering in Distributed Projects. In: *ICGSE, International Conference on Global Software Engineering*. Costão do Santinho, Florianópolis, Brazil 16-19 October 2006.

- Damian, D., 2007. Stakeholders in Global Requirements Engineering: Lessons Learned from Practice. *IEEE Software*, p.21-27.
- Damian, D. Moitra, D., 2006. Global Software Development: How Far Have We Come? *IEEE Software*, 23(5), pp. 17-19.[Online], Available at: <http://www.computer.org/portal/web/csdl/doi/10.1109/MS.2006.126> [Accessed 18 October 2009]
- Damian, D. 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), pp. 149-160.
- Davis, M. A. Nurmuliani, N. Park, S. Zowghi D., 2008. Requirements Change what's the Alternative? *Annual IEEE International Computer Software and Applications Conference*.
- Davis, A., 2005. Just Enough Requirements Management, New York: Dorest House.
- de Souza, C.R.B., Hildenbrand, T., Redmiles, D., 2007. Toward Visualization and Analysis of Traceability Relationships in Distributed and Offshore Software Development Projects. In: *Proceedings of the First International Conference on Software Engineering Approaches For Offshore and Outsourced Development (SEAFOOD)*, Zurich, Switzerland, 2007.
- Davis, A., Nori, K., 2007. Requirements, Plato's Cave, and Perceptions of Reality. *Proc. International Conference on Computer Software and Applications*, Beijing, China, 2007.
- Ebert, C. De Man, J., 2005. Requirements uncertainty: influencing factors and concrete improvements. *Proc. 27th International Conference on Software Engineering, ST Louis, Missouri, USA*.
- Ferreira, S. Collofello, F. Shunk, D. Mackulac, G. Wolfe, P. 2003. Utilization of Process Modeling and Simulation in Understanding the Effects of Requirements Volatility in Software Development. *International Workshop on Software Process Simulation and Modeling*, Portland, Oregon, 2003.
- Geisberger, E. Keil, P. Kuhrmann, M., 2007. International Workshop on Tool Support and Requirements Management in Distributed Projects. In: ICGSE, *International Conference on Global Software Engineering*, Minneapolis, Minnesota, USA 20-26 May 2007.
- Grinter, J. D., 1999. Splitting the Organization and Integrating the Code: Conway's Law Revisited. In: ICSE, *International Conference on Software Engineering*, Los Angeles, California, USA, 16-22 May 1999. ACM : Los Angeles, CA.
- Grehag, Å., 2001. Requirements Management in a Life-Cycle Perspective - A Position Paper, In Ben Achour-Salinesi, C., Opdahl, A.L., Pohl, K. and Rossi, M. (Eds) *Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ'01*, Interlaken, Switzerland. Essenere Informatik Beiträge, pp. 183-188.
- Gumm, D., 2007. A Model of Requirements Engineering at Organizational Interfaces: An Empirical Study on Distributed Requirements Engineering. *GREW'07* – pp. 36 to 44.
- Hanisch, J. Corbitt, B., 2004. Requirements engineering during global software development: some impediments to the requirements engineering process-A case study, *Proceedings of the 13th European Conference on Information Systems, ECIS* (ed.), 14-16 June 2004, Turku, Finland.[Online] Available at: <http://is2.lse.ac.uk/asp/aspecis/20040057.pdf> [Accessed 21 October 2009].

Harker, S. D. P. Eason, K. D. Dobson, J. E., 1993. The change and evolution of requirements as a challenge to the practice of software engineering. *Proc. IEEE International symposium on Requirements Engineering, San Diego, CA, USA*.

Herbsleb, D. J., 2005. Global software development at Siemens: experience from nine projects. In: *ACM, Proceedings of the 27th international conference on Software engineering*, 15-21 May 2005, St. Louis, Missouri, USA.
MO: ACM New York, NY, USA.

Herbsleb, D.J. Mockus, A., 2003. An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Trans. Softw. Eng.*, 29(6), pp. 481-494.

Herbsleb, D.J. Mockus, A. Finholt, T. A. Grinter R. E., 2000. Distance, dependencies, and delay in a global collaboration. In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2-7 Dec., Philadelphia, PA.

Hervner, A.R. March, S.T. Park, J. and Ram, S., 2004. Design science in information systems research. *MIS Quarterly*, 28(1), 75-105.

Holmstrom, H. Conchuir, E. Agerfalk, P. Fitzgerald, B., 2006. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. *ICGSE 2006*.

Hossain, E. Muhammad Ali Babar, M. A. Paik, H., 2009. Using Scrum in Global Software Development: A Systematic Literature Review 2009 *Fourth IEEE International Conference on Global Software Engineering* 978-0-7695-3710-8/09 DOI 10.1109/ICGSE.2009.25

Huis, M. A. Soekijad, M., 2002. ICT Facilitation of Distributed Groups and Communities. *Building Blocks for Effective Telematics Application Development and Evaluation*, Delft University of Technology: pp.39-46.

Imtiaz, S. Ikram, N. Imtiaz, S., 2008. A Process Model for Managing Requirements Change. *4th IASTED International Conference on Advances in Computer Science and Technology*

Jiayi, Z. Yunjuan, L. Yuesheng, G., 2008. The Requirements Change Analysis for Different level Users. IEEE Computer Society. *International Symposium on Intelligent Information Technology Application Workshops*.

Jiménez, M. Piattini, M. Vizcaíno, A., 2009. Challenges and Improvements in Distributed Software Development: A Systematic Review Advances in Software Engineering, *University of Castilla-La Mancha*. [Online] Available at: <http://www.hindawi.com/journals/ase/2009/710971.html> [Accessed 17 October 2009].

Kobayashi, A. Maekawa, M., 2001. Need-Based Requirements Change Management. *Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '01)*.

Lamsweerde, A., 2009. *Requirements Engineering: From System goals to UML models to software specifications*, John Wiley & Sons Ltd.

Lam, W. Shankararaman, V., 1999. Requirements Change: A Dissection of Management Issues. *EUROMICRO Conference, 1999. Proceedings. 25th*

Lehman, M., 1998. Software's Future: managing evolution, *IEEE Software*, 15(1):41-44, 1998.

- Lescher, C., 2009. Global Requirements Engineering: Decision Support for Globally Distributed Projects. *Fourth IEEE International Conference on Global Software Engineering*. ISBN: 978-0-7695-3710-8
- Lopez, A. Nicolas, J. Toval, A., 2009. Risks and Safeguards for the Requirements Engineering Process in Global Software Development. *Fourth IEEE International Conference on Global Software Engineering*.
- Lundell, B. Lings, B. Ågerfalk, P.J. Fitzgerald, B., 2006. The Distributed Open Source Software Development Model: Observations on Communication, Coordination and Control. *Supported by the European Commission via FP6 Coordinated Action Project*.
- Mathiassen, L. Saarinen, T. Tuunanen, T. Rossi, M., 2004. Managing Requirements Engineering Risks: Analysis and Synthesis of the Literature. *Helsinki School of Economics Working Papers W-379*, 2004.
- McGee, S. Greer, D., 2009. A Software Requirements Change Source Taxonomy. *Fourth International Conference on Software Engineering Advances 2009*.
- Miles, M. B. Huberman, M., 1994. *Qualitative Data Analysis: an expanded sourcebook*, SAGE Publication.
- Moitra, D., 2008. Keynote: Meeting Complexities Head-On: Leveraging Global Software Development for Competitiveness. In: ICGSE, *International Conference on Global Software Engineering*, 17-20 August 2008, Bangalore, India.
- Morisaki, S. Matsumura, T. Ohkura, K. Fushida, K. Kawaguchi, S. Iida, H., 2007. Fine-grained Analysis of Global Software Development Process. In: SOFTPIT, *Proc. Of 1st Workshop on Measurement-based Cockpits for Distributed Software and Systems Engineering Projects (SOFTPIT)*, August 2007 Munich, Germany.
- Moynihan, T., 1997. How experienced project managers assess risk. *IEEE Software*, 14(3).
- Moynihan, T., 2000. Requirements-uncertainty: should it be a latent, aggregate or profile construct? *Proc. Australian Software Engineering Conference, Canberra, ACT, Australia*.
- Nurmuliani, N. Zowghi, D. Powell, S., 2004. Analysis of requirements volatility during software development lifecycle. *Proc. Australian Software Engineering Conference, Melbourne*.
- Nurmuliani, N. Zowghi, D. Williams, S. P., 2004. Using card sorting technique to classify requirements change, *Proc. 12th IEEE International Conference on Requirements Engineering, Kyoto, Japan*.
- Palvia, P. Mao, E. Salam, A.F. Soliman, K.S., 2003. Management information systems research: what's there in a methodology? *Communications of the Association for Information Systems*, Vol. 11 pp.289-309.
- Pressman, R., 2005. *Software Engineering. A Practitioner's Approach*, McGraw Hill.
- Prikladnicki, R. Audy, J. Evaristo, R., 2003. Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context. In: ICSE, *Second International Workshop on Global Software Development*, 2003, Portland, Oregon. Available at: <http://gsd2003.cs.uvic.ca/upload/Prikladnicki.pdf> [Accessed 16 October 2009].

- Prikladnicki, R., Audy N., 2006. A Reference Model for Global Software Development: Findings from a Case Study. *IEEE International Conference on Global Software Engineering (ICGSE'06)*.
- PSEB, 2009. Pakistan Software Export Board. *Government of Pakistan official Site*. Available at: http://www.pseb.org.pk/item/why_pakistan Retrieved on: 26th March 2010
- Richardson, I. Casey, V. Zage, D. Zage, W., 2005. Global Software Development - the Challenges, *SERC Technical Report 278*, University of Limerick, Ball State University, September 2005.
- Ramzan, S. Ikram, N., 2005. Making Decisions in Requirements Change Management. *Information and Communication Technologies. ICIT First International Conference. 2005, 27-28: 309-*
- Ramzan, S., Ikram, N., 2006. Requirement Change Management Process Models: Activities, Artifacts and Roles. *Multitopic Conference, 2006 INMIC '06 IEEE*.
- Sangwan, R. Bass, M. Mullick, N. Daniel, Paulish, J.D., Kazmeier, J., 2007. *Global Software Development Handbook*. New York: Auerbach Publications.
- Sahay, S., 2003. Global software alliances: the challenge of 'standardization'. *Scandinavian Journal of Information Systems. Vol. 15, 2003, pp.3-21*.
- Sengupta, B. Chandra, S. Sinha, V., 2006. A Research Agenda for Distributed Software Development. *Proceedings of 28th International Conference on Software Engineering, Shanghai, China*.
- Sinha, V. Sengupta, B. Chandra, S., 2006. Enabling Collaboration in Distributed Requirements Management Software. *IEEE*, 23(5), Sept.-Oct. 2006 pp.52 – 61.
- SIIA, 2006, SIIA Global Software Development Survey Report. *SOFTWARE & INFORMATION INDUSTRY ASSOCIATION (SIIA) AND SYMPHONY SERVICES*
- Šmite, D., 2006. Requirements Management in Distributed Projects. *Journal of Universal Knowledge Management*, 1(2), pp.69-76. The Standish Group, 1995. *CHAOS*, The Standish Group Report. Available at: <http://www.cs.nmt.edu/~cs328/reading/Standish.pdf> [Accessed 18 October 2009].
- Sommerville, I., 2007. *Software Engineering*, Personal Education Ltd.
- Strauss, A. and Corbin, J., (1998). *Basics of Qualitative Research*. Thousand Oaks, CA, Sage Publications.
- Timea, I. Herrmann A. Geisser M. Hildenbrand T., 2007. The Challenges of Distributed Software Engineering and Requirements Engineering: *Results of an Online Survey*.
- Yin, R., 1994. *Case Study Research: Design and Methods*. Newbury Park, CA Sage Publications, Inc.
- Yin, R., 2003. *Case Study Research: Design and Methods*. Publisher: Sage Publications, Inc; 3rd edition (December 24, 2002).
- Zowghi, D., 2007. Does Global Software Development Need a Different Requirements Engineering Process? Available at: <http://www.requirementsnetwork.com/sites/requirementsnetwork.com/files/Does%20Global%20Software%20Development%20Need%20a%20Different%20Requirements%20Engineering%20Process.pdf>

0Software%20Development%20Need%20a%20Different%20Requirements%20Engineering%20Process.pdf. Retrieved on 19 Feb 2010

Wiegers, K., 2003. *Software Requirements*, Microsoft Press.

Appendix A

Sample Change Request Form

CHANGE REQUEST FORM

CHANGE REQUEST #: (To BE ASSIGNED BY CM MODERATOR)	ABCDEF-CRF-0001
--	-----------------

PROJECT DETAILS			
PROJECT NAME:	XYZ	VERSION/BUILD:	N/P
MODULE/AREA:	CDF	SUB-MODULE:	GHI

CHANGE REQUEST DETAILS		
CHANGE REQUESTED BY:	ASDFD	
REQUESTOR PROFILE:	ASDFA	
DATE OF REQUEST:	ASDFASDF	
CHANGE TYPE:	<input type="checkbox"/> New Requirement <input checked="" type="checkbox"/> Requirement Change <input type="checkbox"/> Design Change <input type="checkbox"/> Other	
	SPECIFY DETAILS (IF OTHER IS SELECTED):	
REASON FOR CHANGE:	<input type="checkbox"/> Legal <input type="checkbox"/> Business <input type="checkbox"/> Performance Tuning <input checked="" type="checkbox"/> Other	
	SPECIFY DETAILS (IF OTHER IS SELECTED):	Functionality required by USER for verification purposes and business needs.
PRIORITY OF CHANGE IMPLEMENTATION:	<input type="checkbox"/> Urgent <input checked="" type="checkbox"/> Normal <input type="checkbox"/> Low	
REQUIRED RESOURCES:	No additional resources are required.	
DESCRIPTION OF CHANGE:	Change UI for groups. There should be added a field in group(s) as a 'Group Acronym'.	

CCB DETAILS (TO BE FILLED BY CCB)	
CCB MODERATOR:	CM Officer

EVALUATION DETAILS (TO BE FILLED BY CCB MODERATOR)		
	NAME	IMPACT DETAILS (WHAT WAS THE IMPACT?)
MODULES AFFECTED:	<ul style="list-style-type: none"> Group Management 	<ul style="list-style-type: none"> To store the value of 'Group Acronym' the module to Add Group is updated. To View the 'Group Acronym' the module to View Group is updated.

PAGE UI AFFECTED:	<ul style="list-style-type: none"> Group Management 	<ul style="list-style-type: none"> Insert column for the field 'Group acronym' in the GridView of the GroupView.aspx Introduce new textbox to insert 'Group Acronym' in the AddGroup.aspx
DATABASE TABLES AFFECTED:	<ul style="list-style-type: none"> GROUP USER_GROUP BALLOT_GROUP BALLOT_SECTION_GROUP 	'GROUP_ACRONYM' field added in the mentioned tables.
USE CASE AFFECTED:	<ul style="list-style-type: none"> AEB-PU-UC-001 	Updated w.r.t. add group acronym of Groups.
WORK PRODUCTS AFFECTED:	N/A	N/A
OTHER ITEMS AFFECTED:	N/A	N/A

IMPACT ANALYSIS (TO BE FILLED BY CCB MODERATOR)	
IMPACT LEVEL:	<input type="checkbox"/> Significant <input type="checkbox"/> Negligible <input type="checkbox"/> No Impact <input type="checkbox"/> NA
ALL PROJECT LIFECYCLE STAGES (AS SPECIFIED IN THE WBS):	<ul style="list-style-type: none"> eBallot System <ul style="list-style-type: none"> Joint Application Development Specifications/Database Model Software Development QC/QA Deployment and Rollout
LIFECYCLE STAGE(S) AFFECTED:	<ul style="list-style-type: none"> eBallot System <ul style="list-style-type: none"> Joint Application Development Specifications/Database Model Software Development QC/QA
IMPACT IN MAN HOURS:	8 hours

CHANGE REQUEST APPROVAL DETAILS (TO BE FILLED BY CCB MODERATOR)	
CHANGE REQUEST APPROVAL:	<input checked="" type="checkbox"/> Approved <input type="checkbox"/> Rejected <input type="checkbox"/> Postponed
REASON FOR DECISION:	Functionality required by CLIENT for change UI of Group Management.
DATE OF DECISION:	September 02, 2009

RESOURCE ALLOTMENT (TO BE FILLED BY CCB MODERATOR)
--

RESOURCE ALLOTMENT APPROVAL:	<input checked="" type="checkbox"/> Fully Approved <input type="checkbox"/> Partially Approved <input type="checkbox"/> Rejected
REASON FOR DECISION:	Functionality required by CLIENT for change UI of Group Management.
ALLOTTED RESOURCES:	XTSFSDFD

TASK AND TARGET DATE ASSIGNMENT (TO BE FILLED BY SOFTWARE MANAGER IN CASE OF APPROVAL)				
MODULES				
No.	MODULE NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
1.	eBallot System	ASTEWR	September 04, 2009	Completed
2.	Group Management	ASREDFG	September 04, 2009	Completed
SCREENS				
No.	SCREEN NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
DATABASE TABLES				
No.	DATABASE TABLE NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
1.	BALLOT_GROUP BALLOT_SECTION_GROUP	DFDFSDF	September 04, 2009	Completed
2.	<ul style="list-style-type: none"> GROUP USER_GROUP 	FGGFHFGH	September 04, 2009	Completed
USE CASES				
No.	USE CASE NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
1.	AEB-PU-UC-004 AEB-PU-UC-005		September 02, 2009	Completed
WORK PRODUCTS				
No.	WORK PRODUCT NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
1.	eBallot System		September 02, 2009	Completed
OTHER ITEMS				
No.	ITEM NAME	ASSIGNED TO	TARGET DATE	COMPLETION STATUS
	N/A	N/A	N/A	N/A
	N/A	N/A	N/A	N/A

CLOSING DETAILS (TO BE FILLED BY CCB MODERATOR)	
CLOSED BY:	DFSFGSF
DATE OF CLOSING:	September 04, 2009
STATUS:	<input checked="" type="checkbox"/> Successfully Completed <input type="checkbox"/> NCR(s) Generated <input type="checkbox"/> Terminated <input type="checkbox"/> Postponed

NOTES:

1. It is noted that all fields must be filled. In other case one of following can be written in that field of form:-
 - Not Applicable
 - Not Available
 - Not Provided
2. Priority of Change Implementation is defined as follow:-
 - Urgent Change - Implemented Time 24 Hours
 - Normal Change - Implemented Time 3 Days
 - Low Level Change - Implemented Time One Week



Linnæus University

School of Computer Science, Physics and Mathematics

SE-351 95 Växjö / SE-391 82 Kalmar

Tel +46-772-28 80 00

dfm@lnu.se

Lnu.se