Optimized Approach in Requirements Change Management in Geographically Dispersed Environment (GDE)

Shahid N. Bhatti[1], Mohammad A. Alqarni[3]
Department of Software Engineering
College of Computer Science and Engineering
University of Jeddah, KSA

Frnaz Akbar[2]
Dept. of Software Engineering
Bahria University Islamabad, Pakistan

Amr Mohsen Jadi[4]
Dept. of Software Engineering
College of Computer Science and Engineering
University of Hail, KSA

Abdulrahman A. Alshdadi[5]
Dept. Information System and Technology
College of Computer Science and Engineering
University of Jeddah
Saudi Arabia

Abdulah J. Alzahrani[6]
Dept. of Computer Engineering
College of Computer Science and Engineering
University of Hail
KSA

***Abstract*—Managing requirements is an essential trait in engineering development process as requirements change and emerge throughout the development process. In the following research work the primary prominence is to eke out requirements that are changing frequently in geographically dispersed setup. To efficiently and effectively cope up with the changing requirements are the key to fulfill customers' requirements in geographically dispersed environment (GDE) and thus, appropriate procedural modeling is presented in this work to covenant with changing requirements to cut overall cost of the project and increase profitability by gratify the customers and the stakeholders. In the following research we have proposed an approach to tackle changing requirements in software development that are geographically dispersed and we have validated the presented procedural model through case scenario. Comprehensive systematic literature review has been performed in this section (II) to propose the efficient methodology in GDE, traits and risk and further to effectively eke out the evolving project's requirements in geographically dispersed environment. Changing requirements in geologically dispersed environment can effectively be managed if the proposed MCR model followed and it will mitigate the risk and challenges which we have to face in global software development and as well as it will cut down the overall project's cost and profitability will expectedly increase.**

Keywords—CM (Change Moderator); GDE (Geographically Dispersed Environment); MCR (Managing Changing Requirements); MCR in the GDE framework; RM (Requirements Management)

I. Introduction

Managing requirements can be described as recording, examining, organizing and concurring on requirements, and after that controlling change and conveying to applicable partners. Geographically Dispersed Environment implies the development of software system from scatters territories [3]. Managing varying requirements indicate the way of dealing the changing prerequisites with a particular true objective to satisfy the need of customers additionally it plays a vital role in the successful accomplishment of the project. Project failure risks are high if requirements are mismanaged [8]. The way toward changing prerequisites is beneficial and healthier as well as challenging at the same time.

In literature, it is accounted that due to the requirements that are frequently changing in the geographically dispersed area leads to the project failure that is very high, in these suggestions are mentioned not to continue global development at all, due to poor planning of changing requirements [5].

The present study is designed to examine how we can effectively manage the requirements that are changing frequently specifically in GDE. In the geologically dispersed setup managing changing requirements are difficult to handle due to lack of common understanding and the persistent changing in requirements. Thus it is necessary to identify the ways and appropriate methods and techniques are required to handle the changing requirements in a geographically distributed environment [2]. The following research work examines the changing requirements in the geographically distributed environment and methodology will help to tackle the change in a way to avoid failure of the project.

II. Literature Review

We have contemplated, distinctive frameworks of handling changing requirements in the Geographically Distributed Environment (GDE) from various sources. Different approaches have been projected to handle varying requirements in geologically dispersed environment. The authors in [7], conducted a review, they stated that GDE supports the advancement of plans of action and innovation in the evolution but these may lead to the project failure of the project if functional, non-functional and other requirements not considered properly, the proper understanding of change request plays an important role but the ways to tackle the risks was not highlighted. In [3], explained distinctive advantages for the advancement of global software development, the authors have used UML (Unified Modeling Language) that tackle change in the geologically distributed area, their main focus was on impact analysis of project cost how the change affects the cost.

In the research work [6] [5], the most prominent and vital advantages for the globally distributed Software are the lower advancement cost of software, the availability of professional workforces at any time in any place, proximity to the market. In addition, organizations have the opportunity for the extension of Software activities by a large number of individuals situated at various geologically distributed locales, but there are many factors that lead to failure of whole projects, but how to handle that risk not identified appropriately [10].

In the following research work [11] [14], throughout the advancement of the structure MCR their most important recognition was on verbal exchange issues during the development of software in the dispersed area, to accommodate these challenges they have suggested removing redundancy is important that will reduce the overall development time and will be beneficial for developers and as well as users or stakeholders, and the alignment of the process is significant in a globally distributed environment to fulfill the project requirements.

An international Standish group [15], examined various software projects that are 13522 in number as a consequence, “Only projects that prove to be efficacious were 29 percent, the unsuccessful percentage of projects were 18 and 53 percentage of projects were tumbled and the foremost cause is prerequisite changes that are not manage properly due to which a high failure rate is face in globally distributed software development projects”.

In context of managing changing requirements in GDE comprehensive literature review [1] [4] [9] [5] [10] [12] [13] [16] has been performed and hence identifies that 85% failure of projects are due to poor management of changing requirements in the global software development environment [7] and also identifies a novel approach to overcome these problems an improved framework has been suggested. The methodologies, the significant findings and the limitation of the used approaches have been highlighted in the Table I.

TABLE. I. DEPICTING LITERATURE REVIEW IN GDE AND MCR

Approach	Methodology	Significance	Limitations
Method applied in Management of GDE projects[3]	Proposed an ontology based Requirement.	Managing the requirements change through an architectural repository.	Requirements Change activities are not specified appropriately.
Manage Dynamic Business Process in GDE [1]	Framework based on integration of UML & CPN.	The proposed framework enhances requirements change management in business.	Utilization of Petri Net abnormal state.
A Version Control Tool for Frame-work Based Application [7]	Based on incremental technique by using GREN tool.	Bolster re-designing and system requirements change effect abridged.	Problem in GREN Wizard Source Code
Requirements Management framework in GDE [17]	Framework proposed to manage communication issues.	Covers all RE activities like change initiation, evaluation and implementation.	GDE issues and impacts not addressed appropriately.
Requirements management using XP [10]	Framework formed for RM utilizing XP in Distributed Software Development.	Manipulate the requirements in conveyed setting based on agile method extreme programming.	Cooperation isn't adequately accomplished in a disseminated setting.
Requirements Measurement Framework. [8]	To deal with MCR, metrics and indicators are used.	Based on estimation the manager of the project made choices.	Errors may occur when indicators use metrics.
Requirements tracing approach in GDE [16]	The proposed model for requirements tracing.	Stakeholders are effectively associated with each progression.	It doesn’t address the Significant issues of GDE.

III. Proposed Model for MCR in GDE

After taking into account the different issues, concerns and features in requirements management in geographically dispersed environment via comprehensive literature review (in Section II). The consensus in this is of generic essence that when the system developers no longer satisfy the clients' prerequisites (obligation) then the whole venture windup aimless for the user and the organization may confront loss and failure of project, organization in the software industry [3] [5] [7] [15].

The proposed model (Fig. 1) enhances requirements change management in the geographically dispersed environment to accommodate organizations of all kinds to manage change effectively and efficiently in the distributed environment. The framework is split into eight fundamental levels and each of these levels has particular exercises that surface throughout the procedure of managing changing requirements in the geologically distributed environment. Fig. 1, is depicting the Management of Changing Requirements in the geographically distributed environment.

A. The Request for Change in Requirements

In the projected model the MCR initiate with the demand for a modification from any associate at any circulated site, the client can also demand for change. This first stage deals with the whole statistics regarding the amendment, for example, complete illumination of requested change, the main purposes behind the alteration are, the requestor who has requested for change and so forth.

B. Impact Analysis

In 2nd phase, the requested change will send for an appraisal. It comprehends the requested amendment, for example, understanding either the modification requests for adding upgraded features to the structure or eliminating error from the system or redesign's some parts of the structure. The effect of the change on spending plan, time and other framework segments is in like manner surveyed in this stage. During the 2nd phase, request for the amendment is assessed through numerous checks.

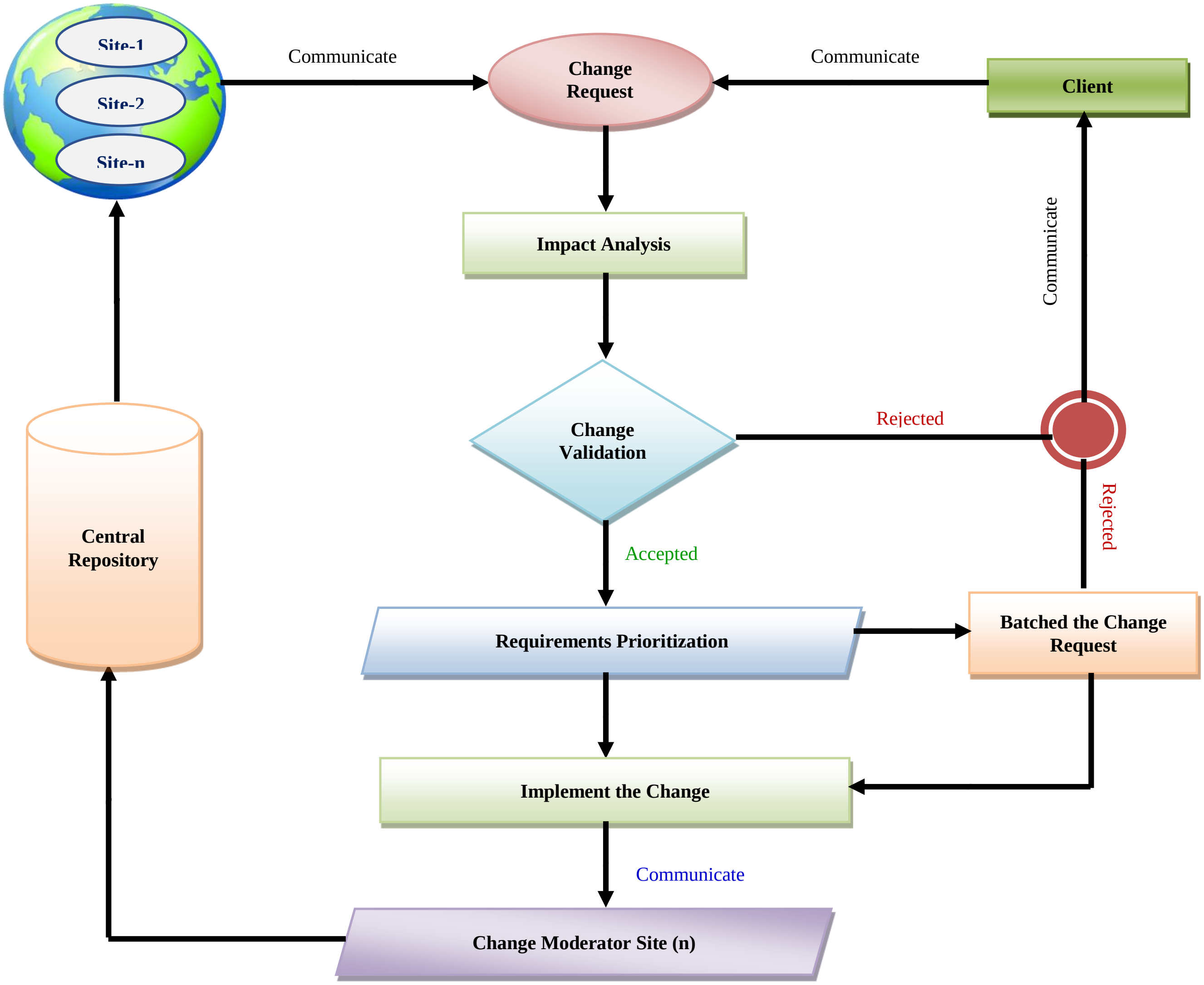

Fig. 1. Proposed Framework for Requirements Change Management in GDE.

C. Validate Changes

The reasonableness of the demanded change is assessed in the 3rd stage. The feasibility of the change is evaluated for further decision, as there are two conceivable options, whether the requested change is practical and feasible or not feasible to be executed. If the requested change isn't feasible to execute than the client informs about the impossibility of the alteration else the demand for alteration will be accepted and executed.

D. Prioritize Requirements

Organizing the requirements with the goal that the most astounding need prerequisite changes can be actualized first. Prioritization of requirements is figuring out the order of significance in the context of stakeholder as well as the developer. Prioritization of requirements may depend on individual inclination, business esteem, and the cost of usage or implementation order of the evolving prerequisite.

E. Batched the change request

A few alternate requests could be grouped together. The amendment request is assembled to execute later on according to the feasibility, because of few confinements. There are two potential outcomes with respect to clump change demands; if the requirements are easy going inside a given time, by then they will be executed else they will be rejected.

F. Change implementation

During the following stage all progressions are executed and actualize in the framework. While implementation all the Functional Requirements, Non-Functional Requirements and other Requirements considered by the implementer and by keeping the change impact in mind the implementer executes the change.

G. Inform Change Moderator

The second last step is to notify Change Moderator. The executed modification will be delivered to CM. Change Moderator will take the final decision. Change Moderator compares all previous requirements with the current change and evaluates the change.

H. Central Database

In the final phase, all the alterations are dispatched to the central database, and then all the stakeholders of the system are going to be enlightened concerning updates. Most of the system report will be revived with the change made. At long last, brand new refreshed changes will be applied to the system. Changes may reoccur at any phase at any time so the central repository of the change would be helpful for the future.

IV. Algorithm: PSEUDO

```
CRM (Change Request)
{
IF (impact analysis == accepted)
{
Prioritize Requirements;
Batch Change Request;
Implement Changes;
Change Moderator;
Update Central Repository;
}
ELSE
Inform the client & recall;
}
```

V. Validation via Case Study

We have selected xyz@ Software House Inc. (naming convention due to copy rights issue), a software program improvement corporation settled in Pakistan. The organization formed in 1990 in UK and in 2002 a branch office of the company inaugurated in Islamabad. The company develops offshore software development projects and successfully delivers a number of projects up till now. The company develops software of all types and many expert teams are available. Whenever there is a request for the change in software whether a functional change or non-functional change, the request is forward to the relevant expert team.

Let presume from the Brazil site; there is a change request in desktop app GUI; the stakeholder complaints that the GUI doesn’t meet the ease of use criteria; as they received many complaints from users that GUI is not a user friendly. So, they initiate change request the manager forward the project to the Graphic Designer team. The team assistant inquiries from the stakeholders what they actually want; in how many days they need the updated software; what were the previous flaws that they have to improve. The teams setup the meeting and decide what to change and assign the tasks to the workforces. Each team member performs their assigned tasks and informs the managerial head and the improved software forward to the stakeholders, but they face many issues during this change. As sometimes one single change affects many components and the overall cost, time for the development also increase. Sometimes it happens that the change request cost increased than the actual cost of software; another technical and non-technical risk also arises during the change; like difficult to synchronize, the stakeholder or the user dissatisfies with the updated change.

If the proposed framework is followed by the company, all above mentioned issues can easily be tackled. As if there is a change request first of all a meeting should be conduct to discuss the impact analysis of the requested change on the whole project, then validate that change if the change is feasible or not, if not then inform the requester. If the requested change is acceptable, then the project further handover to the relevant team, the team members have to conduct the meeting for prioritizing the requirements, and batch the change request that cannot be implemented right now due to some limitation and then tasks will be assigned to experts they will implement the change and inform the change moderator, the change moderator will review the change and update the central repository. Finally change requester will be informed about the change and reviews by the change requester should be kept safe.

The proposed model efficiently manages the changing requirements in the GDE environment and avoids the project failure percentage. Fig. 2 shows the graph of successful projects by applying the proposed model (Fig. 1), and Table II shows how effectively requirements change managed by following the proposed framework.

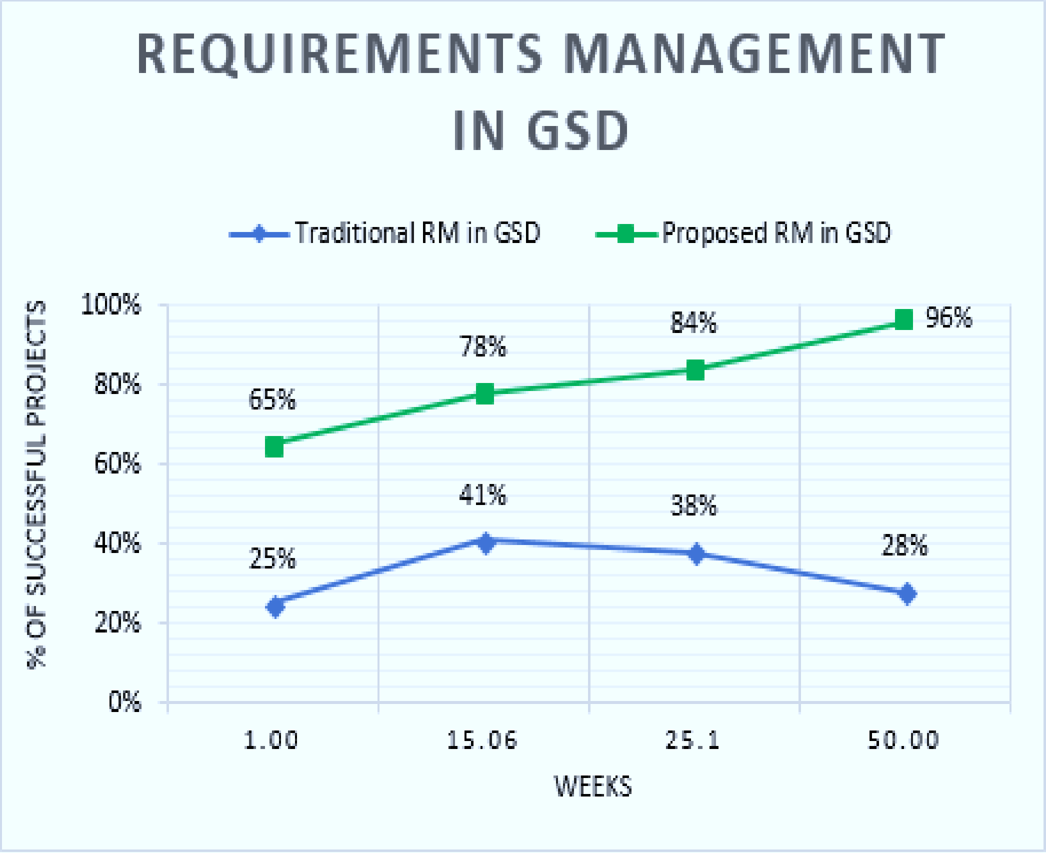

Fig. 2. The Percentage of Projects that Met Objectives.

TABLE. II. MCR IN GDE TRADITIONAL VS. PROPOSED MODEL

Requirements Management in GDE % of Successful Projects		
Change Management	**Traditional RM in GDE**	**Proposed RM in GDE**
Poor	25%	0%
Fair	70%	1%
Good	5%	7%
Excellent	0%	92%

VI. Conclusion

The requirements emerge, change throughout the software development process and requirements are needed to be prioritized and hence managed with utmost priority, especially when the scenario is that of Global Software Development (GSD) and especially geographically distributed environment (GDE). The essential intention of the following research work is to formalize a framework that efficiently oversees managing change requirements (MCR) in the GDE environment. As efficiently handling MCR in the GDE environment save cost, time and as well as ensure the availability of resources, highlighted in detail in literature review of this paper. A complete set of steps are presented, each step if carefully process the failure risk can be lessened. The proposed framework will lead to efficiently and effectively manage the change in requirements in the GDE environment.

Although it is suggested here that further work is required to handle communication risks and quality maintenance of requirements in the GDE environment for more effective results. Communication issues in the GDE environment also result in project failure so for better understanding of change there should be check and balance. Quality is the first priority of every stakeholder, so by centering quality maintenance in the GDE environment, the project reliability can be increased.

References

[1] Minhas, Nasir., & Qurat.,& Zafar.,& Atika, Zulfiqar. (2014).“An Improved Framework for Requirement Change Management in Global Software Development”. Journal of Software Engineering and Applications.7, 779-790.

[2] Bhatti Shahid., & Usman Maria., & Jadi Amir. (2015). “Validation to the Requirement Elicitation Framework via Metrics”. ACM SIGSOFT, USA, 40, 1-7.

[3] Hussain, Waqar. (2010). “Requirements Change Management in Global Software Development: A Case Study in Pakistan.”Retrieved from https://pdfs.semanticscholar.org/41cb/6b1d7b2b03f1214282c812c74ad1e65860d6.pdf

[4] Aneesa R. Asghar, Shahid N. Bhatti, (2017). “The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study” International Journal of Advanced Computer Science and Applications (IJACSA), 8(2), 2017.

[5] Holmstrom, Helenna., & Conchr, Eoen., &Agerfalkh, Paar., & Fitzgerald, Brian. (2006). “Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance”, ICGSE, Brazil, 2006. Brazil: Costão do Santinho, Florianópolis.

[6] Rida, Shahid Bhatti. (2017). “Impact and Challenges of Requirements Elicitation & Prioritization in Quality to Agile Process: Scrum as a Case Scenario, Conference: IEEE, International Conference on Communication Technologies (ComTech-2017), Rawalpindi, Pakistan.

[7] Mighetti, Juan., & Hadad, Graciela. (2016). “A Requirements Engineering Process Adapted to Global Software Development, Argentina”, 2016. Argentina: CLEI Electronic Journal.

[8] Sehrish Alam, Amr Mohsen Jadi. (2017). “Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review” International Journal of Advanced Computer Science and Applications(IJACSA), 8(4), 2017.

[9] Jayatilleke, Shalinka., & Lai, Richard. (2017). “A systematic review of requirements change management”, SIIA, Bundoora, 2017. Australia: INFSOF.

[10] Aneesa R., Atika. (2016). “Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development” International Journal of Advanced Computer Science and Applications (ijacsa), 7(12), 2016.

[11] Ahmad, Zahoor., & Hussain, Mussarat., & Rehman., & Qamar, Usman., & Afzal, Muhammad. (2015). “Impact minimization of requirements change in software project through requirements classification, International Conference on Ubiquitous Information Management and Communication”, Bali, 2015. Indonesia: ACM.

[12] Amna Sadiq, Makkia Abbasi. (2017). “Requirements Prioritization, management Techniques: An Empirical Study”. International Conference on Computing and Mathematical Sciences (ICCMS 2017), Invent, Innovate and Integrate for Socioeconomic Development. 25th and 26th February, 2017, ICCMS 2017.

[13] Raffo, David., & Wakeland, Wayne., & Setamint, Siri. (2006). “Planning and improving global software development process using simulation. Proceeding”, 18, 8-14.

[14] Sabahat, Nosheen., &Iqbal,Faiza., &Azam, Farouqee.,& Javed, Younas. (2010). “An Iterative Approach for Global Requirements Elicitation: A Case Study Analysis”, 2010.“International Conference on Electronics and Information Engineering (ICEIE)”.

[15] Prikladnicki, Rafaeel., & Audy, Jorge., & Evaristo, Roberto. (2014). “Global Software Development in Practice Lessons Learned”,8,267-281.

[16] Ali, Shahid, Tayyab, “Impact of Story Point Estimation on Product using Metrics in Scrum Development Process” International Journal of Advanced Computer Science and Applications(IJACSA), 8(4), 2017.

[17] Khan, Arif., &Basri, Shuib. "A propose framework for requirement Changes Management in Global Software Development", 2012.“International Conference on Computer & Information Science (ICCIS).