Bulletin of Business and Economics, 13(3), 635-641
https://bbejournal.com
https://doi.org/10.61506/01.00547

Impact of Project Management on Requirements Engineering and Method for Managing Requirements Change for Global Software Development

Nadeem Rasool[1], Sidra Yousaf[2], Usama Haseeb[3], Muhammad Waseem Iqbal[4], Saleem Zubair Ahmad[5], Aqsa Afzaal[6]

Abstract
Change is an inherent process in software development. Customer demands, corporate needs, and scalability are just a few of the reasons why requirements may vary. Managing these changes on time is critical to developing effective software. However, due to distributed team structures and geographical obstacles, requirement change management (RCM) is not an easy task, particularly in global software development (GSD). Furthermore, no collocated RCM model or organizational structure is currently available for performing GSD. This study provides a revolutionary unified paradigm for managing requirement engineering in GSD. The study is validated by a survey. The findings indicate that the suggested research will assist software businesses in solving the shifting requirement difficulties in the GSD environment, allowing them to execute projects effectively.
Keywords: Requirements Engineering, Requirements Change Management, Global Software Development

1. Introduction

Global software development (GSD) is currently expanding quickly. It is not, however, a simple procedure. Due to the extensive communication requirements, “both requirements engineering and requirements change management are thought to be extremely challenging activities. This requirement makes RE and RCM” difficult because addressing geographic and cultural inequalities in GSD is crucial. “The significance of project management in RE and RCM procedures is examined in this paper. First, for RE and RCM, frameworks with the phenomenon of specialist project management are suggested. The suggested frameworks are then validated by conducting a survey and conducting blind interviews with the experts.” Finally, statistical analysis methods are used to examine the gathered data. Our findings, which make use of the examined data, demonstrate the significant effects of both.

The development of Software is a rapidly evolving field that requires continuous adaptation to stay competitive. The papers discussed in this analysis demonstrate the diversity of topics and approaches to addressing the challenges of software development. From project management to requirement engineering, these papers highlight the importance of coordination, communication, and collaboration among team members, stakeholders, and customers.

The first paper by Mishra (2023) emphasizes the significance of knowledge-based coordination in global software development. The author suggests that effective knowledge sharing and management can improve coordination among globally distributed teams. The second paper by Shameem et al. (2023) proposes a probabilistic model based on evolutionary algorithms for forecasting.

Leong et al. (2023) discuss the importance of for long-term sustainability; hybrid project management mixes the conventional development lifecycle with agile-based product development. The fourth paper by Khan et al. (2023) evaluates the analysis-based best practices for requirement engineering for secure software development in international software development.

The fifth paper by Dingsøyr et al. (2023) presents a Studying the effects of switching “from a first to a second-generation large-scale agile development method, this longitudinal case study on coordination in a very big development programmed”. International software development, the sixth paper by Lopes et al. (2023) provides a comprehensive method of expert detection that makes use of syntactic and semantic social network analysis.

The seventh work by Hernandez et al. (2023) uses a multivocal mapping study to examine requirements change management in DevOps contexts. The eighth paper by Sakka et al. (2023) proposes a Conceptual framework for IT projects that supports upstream decision-making and project management method selection.

The ninth paper by Ahmad et al. (2023) conducts a systematic mapping study of requirement engineering for artificial intelligence systems. Finally, the tenth paper by Quansah et al. (2023) examines the issues with requirement engineering affecting Sub-Saharan Africa's software quality.

The importance of effective coordination, communication, and collaboration among team members and stakeholders in software development. The papers highlight various challenges and propose solutions for software development, paving the way for further research and improvement in the field. Project management has a significant impact on requirements engineering and methods for managing requirements global software development has changed (GSD).

GSD projects require a large amount of coordination and communication between teams in different countries. This can make it difficult to manage requirement changes and ensure that all stakeholders are on the same page. Agitate: Without proper project management, GSD projects can quickly become chaotic and disorganized, leading to costly delays or even complete failure.

Solution: By implementing effective project management techniques such as requirements engineering and change management, GSD teams can ensure that their projects stay on track while meeting the needs of all stakeholders involved. With these methods in place, GSD teams will have the tools they need to successfully deliver good quality software solutions on time and within budget.

Project management and software development are two critical areas of business that have undergone significant changes in the years. The rapid benefits of technology and the increasing complexity of projects have led to the development of various methodologies, models, and frameworks aimed at improving project outcomes. In this research paper, we aim to provide an overview of some of the latest research in project management and software development. We have analyzed ten research papers covering

[1] Department of Software Engineering, Superior University, Lahore, Pakistan, msse-f21-007@superior.edu.pk
[2] Department of Software Engineering, Superior University, Lahore, Pakistan, msse-f21-015@superior.edu.pk
[3] Department of Software Engineering, Superior University, Lahore, Pakistan, usama.haseeb2210@gmail.com
[4] Associate Professor, Department of Software Engineering, Superior University, Lahore, Pakistan, waseem.iqbal@superior.edu.pk
[5] Professor, Department of Software Engineering, Superior University, Lahore, Pakistan, saleem.zubair@superior.edu.pk
[6] Faculty of CS/IT, Superior University, Lahore, Pakistan, aqsaafzaal489@gmail.com

different aspects of project management and software development. These papers cover a wide range of topics, including combining paradigms for knowledge management, software development, and project management, the possibility of creating information modeling for project management knowledge applications, and the impact of software quality models on healthcare software systems.

Global software development (GSD) has gained popularity in recent years among the software development community. The primary cause for the growth in GSD is the need to minimize costs. Advancements in computer technology enable remote development of complicated projects and frameworks (Iqbal et al., 2013). These developments make requirement change management (RCM) a challenging task to manage. RCM is the process of regulating, analyzing, comprehending, managing, and tracking changes in requirements. Assawamekin (2010) emphasizes the need to adapt to the changing demands of clients in the GSD environment. According to Mateen and Amir (2016), traditional software development models assume that needs remain constant throughout time. Customers' needs frequently change during the system's development life cycle (SDLC), which contradicts the conventional wisdom.

Changes in needs occur due to changes in customer demands, partner understanding, client association, extended vision, need details, and availability of new solutions.

RCM has significant economic implications, including time, quality, and overall costs, particularly in later phases of the SDLC. Software projects frequently fail due to requirement changes (Mateen & Amir 2016; Hussain 2016; Ahmad et al. 2015). Because of this, RCM is not a straightforward technique. In dispersed environments like GSD, the RCM process becomes more challenging. A collocated model or organizational structure is necessary to handle requirements during software development in a distributed environment.

2. Literature Review

Khan et al. (2012) proposed a seven-core requirement change management methodology for collocated software development firms. It addressed problems in conventional requirement change management approaches. The model lacked a requirement categorization function and required a repository to hold change-related data. The repository assists development teams in tracking change requests and expected changes throughout the SDLC process. Akhtar et al. (2014) proposed a Twin Peak Model (TPM) to assess the impact of requirement changes (RC) on software systems. It assessed the influence of the RC on the system architecture. However, potential system development requirements may generate cost concerns. TPM requires periodic inspection and evaluation by professionals, and project management is hard owing to extensive documentation requirements.

Mateen and Amir (2016) introduced a novel paradigm to enhance the efficacy of requirement change management in GSD. Their architecture allows for managing requirement changes throughout the SDLC process, which is a key benefit. Additionally, it improves knowledge of duties. However, the model only applied to one organization. To achieve reliable results in GSD, more case analyses were required. This would have enabled the creation of large-scale software. There is presently no unifying organizational framework for handling requirements in a dispersed software development environment.

Barrett (2011) emphasized the importance of organization consulting courses in preparing students for entrepreneurship and career exploration. The e-Portfolio provides a user-friendly and student-centered approach. This allows for easy personalization of consultancy services. E-Portfolios might be challenging for non-IT users. To inspire adult learners, educators should increase course options and use effective teaching approaches that meet course needs.

Ziemba and Obłąk (2015) highlighted key success elements for change management in information systems (IS) initiatives. The study suggests that successful information system initiatives rely heavily on effective change management. Change management optimizes IS project management to ensure timely and successful execution. Change is unavoidable while implementing IS initiatives. The research was confined to two case studies from Polish public organizations. To generalize the findings, the study requires additional data sets.

Ziemba and Obłąk (2015) identify essential success elements for change management in IS projects and their practical execution. Elezi et al. (2013) proposed a systems-based strategy for predicting engineering change management (ECM) issues. ECM procedures, planning, and methodologies were evaluated. However, the method prioritized quantitative items and lacked system consistency.

Munassar et al. (2013) provided an effective framework for managing projects and transformation. The study intended to uncover key change management elements that contribute to project success. The recommended technique improved the sustainability and productivity of organizations. However, crucial aspects that contribute to project success were not addressed. Pressman (2015) advocated lean methodology as a way to attain high-quality mass output in manufacturing. Lean technique focuses on waste removal, rapid delivery, quality standards, team empowerment, and flexibility in decision-making. The lean approach prioritizes customer value and continuously improves the value stream. In recent years, the software sector has demonstrated significant interest in using lean methodologies. It has several benefits, including compatibility with agile processes, client satisfaction, and high-quality goods. It saves time and is economically advantageous.

In 2016, Xiong et al. suggested a Lean-based change management strategy. The goal was to facilitate change management in the software industry. According to Xiong et al. (2016), effective change implementation requires preparedness, leadership, communication planning, organizational resources, system controls, and worker behavior. According to the report, a software business can only secure success by focusing on change and implementing critical process areas. However, no clear metrics exist to assess the performance of managerial changes. In 2011, Jeet and Dhir created a software system using a fuzzy interface method. They aimed to achieve and manage maintainability. Although many process models attempt to predict software quality, they often fall short. Current models of maintainability do not sufficiently address disruptive forces and their consequences. Fuzzy Interference Systems (FIS) leverage open-source data to develop predictive software engineering models. FIS can be validated, reproduced, disproven, and improved. However, the research answer does not apply to all circumstances. A FIS-based strategy requires

considerable assessment, including hardware for verification and validation. Defining precise fuzzy rules, membership functions, and optimization are challenging jobs to do.

Wiboonrat and Kaewsiri (2015) introduced an engineering change method to increase governance in data center project management (DCPM). DCPM organizes the data center design process and manages engineering revisions. The disadvantages of this approach include poor preparatory study, unclear corporate objectives and requirements, and insufficient assessment by quality assessors.

Minhas et al. (2014) proposed a methodology for managing requirement changes in global software development (RCM_GSD). The purpose was to identify issues encountered throughout a change management need. The suggested framework meets change management needs while minimizing the effect of global software development. Communication in a GSD context can be challenging because of distance, time zone differences, language discrepancies, and cultural differences, resulting in low rates of communication. The study's approach addressed most of the issues faced during GSD. However, improving the decision-making step of RCM needed additional time and money. The decision-making process relied on expert feedback, which did not always result in useful outcomes.

Andrade et al. (2016) described a method for implementing change management. It effectively managed process control modifications for the IT sector. The technique was highly recommended due to its affordability and lack of financial input. The procedure was hampered by staff who lacked technology literacy and were hesitant to adjust to the new system. Kumar and Kumar (2011) presented a framework to address requirement management issues and obstacles in GSD projects. Many GSD initiatives fail owing to poor project management and inadequate requirement management (RM) practices. The framework employed an ontology-based knowledge management system to address issues with insufficient requirements. The authors did not provide a clear definition of communication or clarify how to interact with the repository (AFZAL, M, et,al. 2023).

Hussain (2016) examined the issues faced by practitioners and the impact of collaborative technologies (CT) on RCM operations. Managing RCM is tough when stakeholders are located abroad. A suitable study framework was employed to examine the problems of RCM in GSD and the function of CT. The drawbacks of the procedure were limited research funding, local culture, insufficient piloting of interview questions, a delay in initiating interviews, and the need to improve confidence and interview skills. Ahmad et al. (2015) suggested a paradigm for managing RC throughout the SDLC. Two steps were presented: RM and impact mitigation approach. The first phase involves the categorization of needs. The second phase involves managing modifications to ensure just the targeted section is affected. However, the proposed approach was not tested on an actual software project or validated. Additional findings were not factored into the model for clarification.

Chung and Pei (2009) proposed a comprehensive method for software content characterization, utilizing qualities and connection. This technique examined the impact of changes on software requirements, data, and documentation. However, their technique requires additional research in connection creation and change patterns to be usable across various software production contexts.

Sinha et al. (2006) performed studies to emphasize the significance of demand management in GSD environments. GSD's demand management challenges include geographical distance, cultural gaps, and limited tool support for remote teams. Effective cooperation and a requirement management platform may alleviate the challenges of managing requirements for distant teams.

3. Methodology

Previous and current research on software development often assumes fixed needs acquired during the requirement-gathering phase. Changing requirements might affect product quality, timeliness, and cost. Software developers and researchers (Ahmad et al. 2015; Ziemba and Obłąk 2015; Minhas et al. 2014) identified change in requirements as a leading cause of software project failure. RCM is a complex process in advancing collocated software. GSD complicates the matter further.

- How may a single RCM model improve software development?
- Would a new unified paradigm be accepted and embraced by software development organizations at both individual and group levels?

3.1. Proposed Unified Architecture

Phase 1: Knowing the requirements

First, management teams must meet to ensure a thorough knowledge of developments. To manage requirement changes, it's important to first establish a shared understanding of the requested adjustments among departments, whether they're local or scattered. Convert venture needs from a database to a graphical form this fellow below process involves;

- Understanding the current needs.
- Evaluate the impact of the change on current requirements.
- Estimate the scope of the change.

Phase 2: Analyzing the changes

At this step, necessary adjustments will be made to the requirement diagram. Changes may involve adding, editing, or removing requirements. Khan et al. (2012) provide a process model for managing requirements change in collocated software development, which recognizes and understands the breadth of altered needs. A repository is necessary to hold anticipated modifications, but no specific requirements are established. Akhtar et al. (2014) used the Twin Peaks Model to analyze the role of demand change in software architecture. Expertise is necessary to assess and evaluate projects regularly. Project management is hard owing to extensive documentation requirements. Improved the efficacy of the requirement change management methodology for global software development (Mateen & Amir, 2016).

Ziemba and Oblak (2015) discuss change management in information systems initiatives for public entities in Poland. Elezi et al. (2013) discuss engineering issues in managing transformation and management cybernetics. Concentrates on quantitative and easily measurable elements. The system lacks consistency in its goals. Munassar et al. (2012) discuss the benefits of using a change management framework for successful IT project execution. Effective implementation of change management in projects (Xiong et al., 2016). Jeet and Dhir (2011) proposed a technique for predicting the work necessary to make modifications to a software development project.

Wiboonrat and Kaewsiri (2015) suggest engineering modifications for better governance in data center project management. Minhas et al. (2014) proposed an enhanced methodology for managing demand changes in global software development. Andrade et al. (2016) discuss the deployment and advantages of change control in the IT environment. Kumar & Kumar (2011) propose an ontology-based approach to studying the influence of requirements management characteristics on global software development projects. Hussain (2016) conducted a case study on managing needs changes in global software development. Ahmad et al. (2015) propose using needs classification to minimize the impact of changes in software projects. A repository is necessary to hold anticipated modifications, but no specific requirements are established.

Expertise is necessary to assess and evaluate projects regularly Project management is hard owing to extensive documentation requirements. To get accurate findings in GSD and construct large-scale software, the suggested methodology lacks thorough evaluation through case studies. It's challenging to match user expectations when difficulties develop. It is challenging to utilize for folks who are not IT savvy. Change management is often overlooked as a key success aspect in information technology development. Concentrates on quantitative and easily measurable elements. The system lacks consistency in its goals.

Failure to address success variables may result in unanticipated results. To ensure the generalizability of outcomes, research should be conducted across many companies. A fuzzy-based technique is required for comprehensive assessment with hardware verification and validation (Khan, S., Haseeb, et,al 2023). It is challenging to create precise fuzzy rules, and membership functions, and optimize fuzzy systems. Provides insufficient study of the preparatory stage and lacks clear business objectives and needs. Expert feedback is essential for decision-making. However, this does not guarantee optimal decision-making outcomes. Non-technical individuals struggle to adopt new systems.

Phase 3: Settling the changes in the requirements

Experts will conduct a change analysis to assess the extent and breadth of completed advancement work across departments and GSD locations. Client-requested modifications to requirements may impact this. After conducting a progress inquiry, the modification can be noted in the database for future communication purposes.

- Analyze the impact of changes on the project as well as specific departments or sites.
- Comparison of change effects before and after the client's intended modifications are finalized.

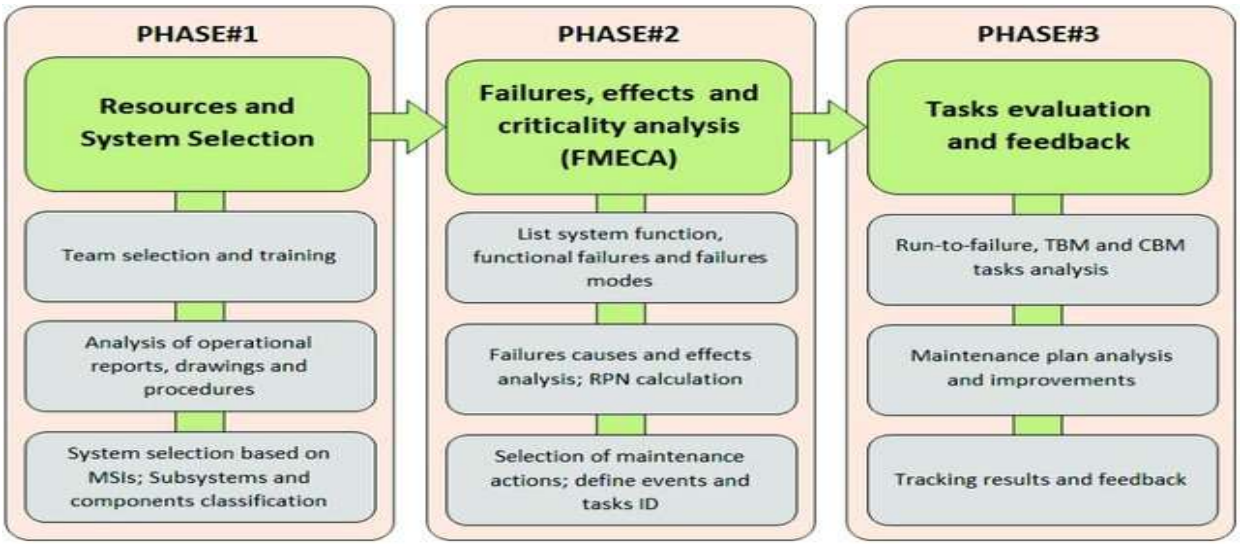


Figure 1: A unifying approach for RCM in both colocated and dispersed teams.

Phase 4: Implementing these modifications in future projects

The technique concludes with adapting the recorded database modification to future projects as needed by the customer.

- Recording changes in the requirements database.
- Determine the cost of changing the requirement database.
- Expert recommendations based on the requirement database.

Software development organizations use many approaches for managing requirement changes. Previous models reported in the literature had drawbacks. Certain concerns resulted in higher project expenses or failure (Hussain 2016; Ahmad et al. 2015; Xiong et al. 2016). To address these constraints, we suggested a unified paradigm for RCM at both the organizational and GSD levels. This strategy can assist software developers handle the needs for dispersed products. The following aims aim to validate the proposed unified model using Mateen & Amir's (2016) standards.

4. Results

Two strategies were used to carry out the survey: a web-based strategy and a manual strategy. The goal of the web-based survey was to gather general opinions from respondents about the suggested frameworks, while the goal of the manual survey was to gather actual opinions following the adoption of the suggested frameworks. The RCM framework is used in five GSD organizations, whereas the proposed RE framework is used in four GSD organizations. As a result, the questionnaire for the online survey was posted in October 2017 and made available for use for over two and a half months. From October 2017 to January 2018, a manual survey was carried out. We would like to point out that the web-based poll yielded 93 replies, of which only 24 were usable, with overwhelmingly consistent reliability. A total of 63 replies for RE were obtained through the manual survey, of which 29 were usable. Additionally, 26 of the 71 replies for RCM that were gathered might be used. We disregarded the comments provided by

academic researchers since we were unaware of their professional backgrounds in the software development sector. In Figure 2 for the web-based survey, Figures 3(a & b) for the physical-based survey, and Figure 4 for the interview participants who had firsthand knowledge of both frameworks, the responses were ordered geographically. To demonstrate that we have participants from all over the world, we gathered information on the participant's country.

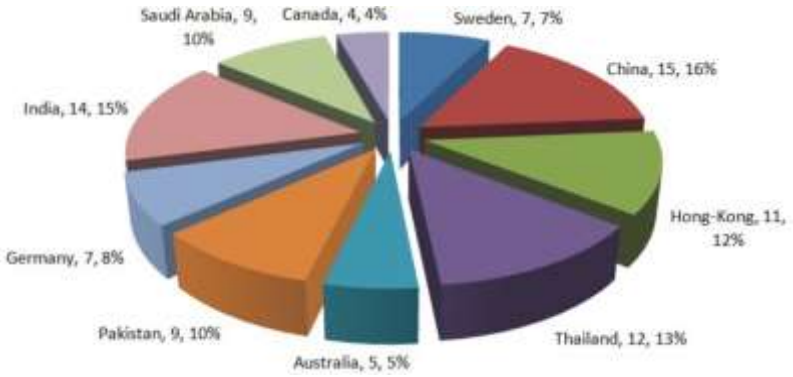
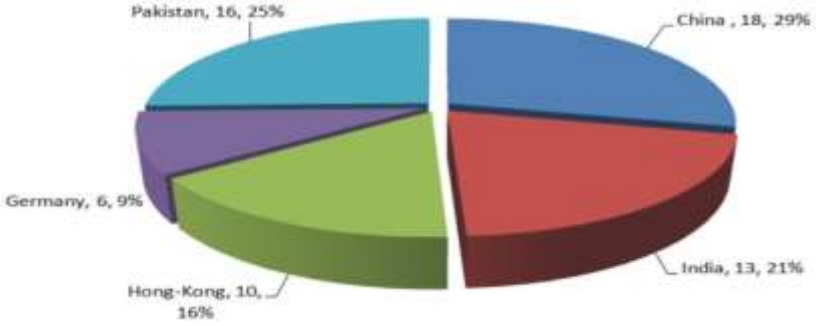


Figure 2: Continent-wise web survey responses from the RE and RCM frameworks.

(a)



(b)

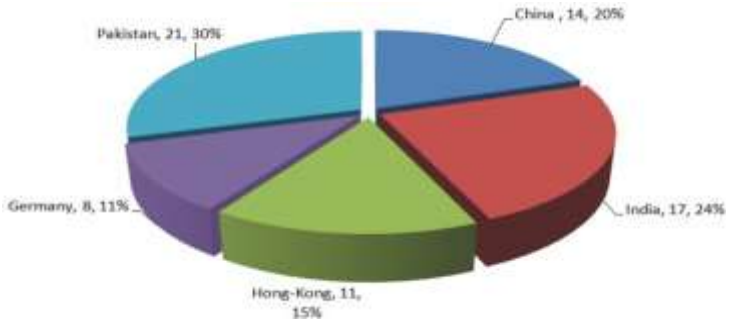


Figure 3: (a) Continent-wise manual survey responses for RE
(b) Continent-wise manual survey responses for RCM.

Figure 4: Continent-wise interview participants for RE and RCM.

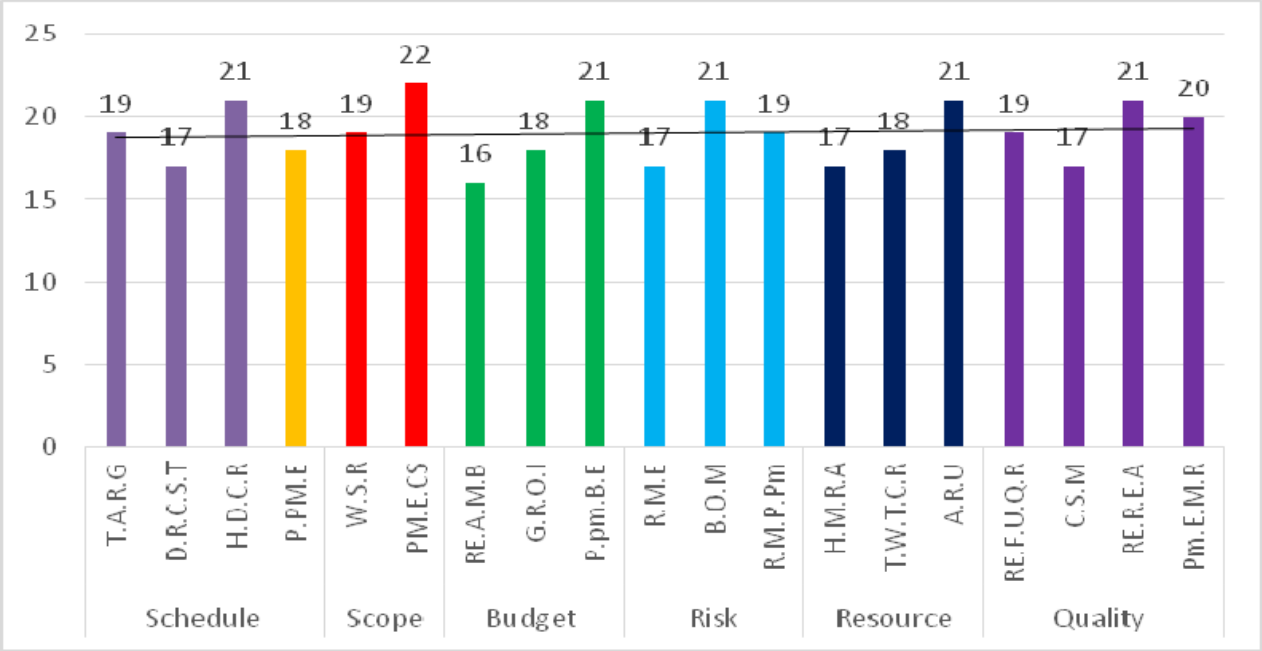


Figure 5: Frequency analysis of the web survey of the RE framework.

4. Statistical Analyses

Factor-wise data were gathered by the survey's second portion of the questionnaire. Final findings were obtained by using statistical tools and procedures. As a result, the Likert scale is divided into three stages: Strongly agree + agree, neutral, and strongly disagree + disagree are examples of responses. Only component a) of the data, which displays the respondents' perceptions of each framework factor, will be utilized to illustrate the results.

4.1. Results of the Re-Framework

Results of a web survey about the RE Framework. Figure 8 displays the online survey results for the suggested RE framework. The findings of the analysis show that the RE framework combined with specialized project management is particularly efficient at gathering precise and comprehensive requirements. The suggested framework has a very substantial impact on every aspect of the six-pointed star project management paradigm. The variables are placed in the following decreasing sequence based on the factor-wise significance level of the proposed framework: scope factors, quality factors, risk factors, budget factors, resource factors, and scheduling considerations. The trend line demonstrates that all of the criteria are therefore very important for the suggested RE framework.

5. Conclusion

RCM is an inevitable process in all stages of software development. A customer may demand a modification at any moment owing to changes in company structure, market conditions, revised project objectives, or the introduction of new technology. The impact of intended modifications benefits the customer if they are addressed appropriately and at the appropriate time. Otherwise, adjustments may result in delays, additional expenditures, and implementation failures.

This study proposes a four-phase model for RCM in organizational and global software development. This method may enhance the change management required to execute a project effectively. To back up our assertion, we conducted a poll. The survey findings suggest that the recommended technique significantly improves the RCM process. Eighty percent of respondents expressed trust in the proposed unified model. Eighty-four percent (84%) of respondents believe that the suggested unified model may fail to be implemented in software organizations owing to a lack of developer skills and expertise.

Future studies should include adopting the suggested model in industry case studies to test its efficacy with the required improvements. Furthermore, interviews with respondents would be an important data collection strategy since they would provide us with more information about their perspectives on the applicability of the suggested model.

References

Afzal, M., Hameed, U., Ahmed, S. Z., IqbaL, M. W., Arif, S., & Haseeb, U. (2023). Adoption of continuous delivery in DevOps: future challenges. *J. Jilin Univ.*, *42*, 20.

Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., & Grundy, J. (2023). Requirements engineering for artificial intelligence systems: A systematic mapping study. *Information and Software Technology*, *158*, 107176.

Ahmad, Z., Hussain, M., Rehman, A., Qamar, U., & Afzal, M. (2015, January). Impact minimization of requirements change in software project through requirements classification. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication* (pp. 1-5).

Akhtar, A., Motla, Y. H., Aslam, H., & Jamal, M. (2014, June). Role of requirement change in software architecture using Twin Peaks Model. In *2014 IEEE 5th International Conference on Software Engineering and Service Science* (pp. 174-177). IEEE.

Assawamekin, N. (2010, August). An ontology-based approach for multiperspective requirements traceability between analysis models. In *2010 IEEE/ACIS 9th International Conference on Computer and Information Science* (pp. 673-678). IEEE.

Barrett, B. G. (2011, May). Creating change and innovation in Human Resource management courses: Developing a model organizational consulting project as a learning application. In *2011 International Conference on Business Management and Electronic Information* (Vol. 3, pp. 267-270). IEEE.

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

Chen, C. Y., & Chen, P. C. (2009). A holistic approach to managing software change impact. *Journal of Systems and Software*, *82*(12), 2051-2067.

de Andrade, P. R. M., Albuquerque, A. B., Teófilo, W. D., & da Silva, F. A. (2016). Change management: Implementation and benefits of the change control in the information technology enviroment. *arXiv preprint arXiv:1603.03110*.

Dingsøyr, T., Bjørnson, F. O., Schrof, J., & Sporsem, T. (2023). A longitudinal explanatory case study of coordination in a very large development programme: the impact of transitioning from a first-to a second-generation large-scale agile development method. *Empirical Software Engineering*, *28*(1), 1.

dos Santos, J. L., & Sampaio, R. R. (2023). Integrating project management, software development, and knowledge management models: a case study in a public ICT services organization. *Social Networking*, *12*(1), 1-27.

Elezi, F., Maier, T. G., & Lindemann, U. (2013, April). Engineering change management challenges and management cybernetics. In *2013 IEEE International Systems Conference (SysCon)* (pp. 567-573). IEEE.

Hernández, R., Moros, B., & Nicolás, J. (2023). Requirements management in DevOps environments: a multivocal mapping study. *Requirements Engineering*, *28*(3), 317-346.

Hussain, W. (2016, August). Reflections on requirements change management in global software development: a multiple case study. In *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)* (pp. 77-79). IEEE.

Iqbal, J., Ahmad, R., & Noor, M. A. (2013, June). A framework to improve the requirements engineering process for software development outsourcing. In *22nd Australian Software Engineering Conference, Melbourne* (pp. 4-7).

Jeet, K., & Dhir, R. (2011, July). A model for estimating efforts required to make changes in a software development project. In *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence* (pp. 175-178).

Khan, A. A., Basri, S., & Dominic, P. D. D. (2012, October). A process model for requirements change management in collocated software development. In *2012 IEEE Symposium on E-Learning, E-Management and E-Services* (pp. 1-6). IEEE.

Khan, R. A., Akbar, M. A., Rafi, S., Almagrabi, A. O., & Alzahrani, M. (2024). Evaluation of requirement engineering best practices for secure software development in GSD: an ISM analysis. *Journal of Software: Evolution and Process*, *36*(5), e2594.

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

Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L., & Ramasamy, R. K. (2023). Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. *Sustainability*, *15*(2), 1121.

Lopes, T., Ströele, V., Braga, R., David, J. M. N., & Bauer, M. (2023). A broad approach to expert detection using syntactic and semantic social networks analysis in the context of Global Software Development. *Journal of Computational Science*, *66*, 101928.

Mateen, A., & Amir, H. (2016). Enhancement in the effectiveness of requirement change management model for global software development. *arXiv preprint arXiv:1605.00770*.

Minhas, N. M., & Zulfiqar, A. (2014). An improved framework for requirement change management in global software development. *Journal of Software Engineering and Applications*, *2014*.

Mishra, D. (2023). Developing a knowledge-based perspective of coordination in global software development. *VINE Journal of Information and Knowledge Management Systems*.

Piechnicki, F., Loures, E., & Santos, E. (2017). A conceptual framework of knowledge conciliation to decision making support in RCM deployment. *Procedia Manufacturing*, *11*, 1135-1144.

Quansah, A., Emmanuel, A., Kyeremanteng, B., & Kesse, E. N. (2023). Requirement engineering problems impacting the quality of software in Sub-Saharan Africa. *Indonesian Journal of Electrical Engineering and Computer Science*, *30*(1), 350-355.

Rasool, N. A. D. E. E. M., Khan, S., Haseeb, U. S. A. M. A., Zubair, S., Iqbal, M. W., & Hamid, K. H. A. L. I. D. (2023). Scrum And The Agile Procedure's Impact On Software Project Management. *Jilin Daxue Xuebao GongxuebanJournal Jilin Univ. Eng. Technol. Ed*, *42*, 380-392.

Sakka, A., Kourjieh, M., & Kraiem, I. B. (2023). An IT projects' conceptual model to facilitate upstream decision-making: project management method selection. *International Transactions in Operational Research*, *30*(6), 3687-3718.

Shafiq, M., Zhang, Q., Akbar, M. A., Khan, A. A., Hussain, S., Amin, F. E., ... & Soofi, A. A. (2018). Effect of project management in requirements engineering and requirements change management processes for global software development. *IEEE Access*, *6*, 25747-25763.

Shameem, M., Nadeem, M., & Zamani, A. T. (2023). Genetic algorithm based probabilistic model for agile project success in global software development. *Applied Soft Computing*, *135*, 109998.

Sinha, V., Sengupta, B., & Chandra, S. (2006). Enabling collaboration in distributed requirements management. *IEEE software*, *23*(5), 52-61.