Agile Requirements Change Management Model

For

Global Software Development

by

Neha Koulecar
B.E., Goa University, 2013

A Project Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Agile Requirements Change Management Model

For

Global Software Development

by

Neha Koulecar
B.E., Goa University, 2013

Supervisory Committee

Dr. Daniela Damian, Supervisor
(Department of Computer Science)

Dr. Sajin Koroth, Departmental Member
(Department of Computer Science)

ABSTRACT

We propose a comprehensive and robust agile requirements change management (ARCM-GSD) model that addresses the limitations of existing models and is tailored for agile software development in the global software development paradigm. To achieve this goal, we conducted an exhaustive literature review and an empirical study with RCM industry experts. Our study evaluated the effectiveness of the proposed RCM model in a real-world setting and identifies any limitations or areas for improvement. The results of our study provide valuable insights into how the proposed ARCM-GSD model can be applied in agile global software development environments to improve software development practices and optimize project success rates.

Contents

List of Tables

List of Figures

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Daniela Damian, whose invaluable support, guidance, and expertise were critical to the success of this project.

DEDICATION

I dedicate this project to my colleagues in the Department of Computer Science, who have supported and encouraged me throughout this journey.

Chapter One

Introduction

Global Software Development is a software development style that facilitates skilled software professionals to collaborate on developing high-quality software irrespective of their geographical location [1]. The adoption of Global Software Development (GSD) has led to the transition of software companies into global software development enterprises. The primary reason for this transition is the strategic and economic benefits gained from GSD [2]. The most attractive GSD benefits include lower development cost and time, round-the-clock development, skilled and cost-effective global workforce and access to an international customer base [3]. However, transforming into global software development introduces multiple unique challenges not faced in single-site development environments [4]. GSD teams are spread across different geographical locations, creating a physical divide among the members, which disrupts effective communication [5]. Additionally, cultural variations, language barriers, and time zone differences substantially aggravate communication and coordination challenges [2].

Software requirements change is a significant barrier to the software development cycle, negatively impacting project duration and budget [2]. Requirement changes have become dynamic and frequent for a variety of reasons. The most common causes include changing business priorities, availability of the latest technologies, market competition and opportunities [2]. Additionally, with time, clients/stakeholders acquire a better knowledge of the

software and request system improvements. Continuously changing requirements have a detrimental impact on the project deadline, cost and quality [2]. However, changes cannot be avoided; therefore, the success of software projects depends on effectively managing these evolving requirements [6].

In traditional development approaches, project requirements are collected from the client at the start of the software development life cycle process. The actual project implementation starts after finalizing the requirements. The development process is initiated after thorough system planning and designing. But this requirements management approach fails when the client demands change to requirements at a later phase of development as it incurs rework and adversely impacts project deadlines and cost [2].

Newer lightweight development techniques like agile development are more receptive to the dynamic nature of requirements [2]. Consequently, software organizations are approaching requirement changes through agile methodologies as agile accommodates software changes at any stage due to the iterative development style. Additionally, both agile software development (ASD) and requirement change management (RCM) processes are communication/coordination intensive and rely on teamwork [4]. Due to these commonalities, ASD and RCM are an excellent match. The RCM activities can readily integrate with agile ceremonies. Thus, there is a demand to study the requirement change management process in the agile paradigm. However, the literature highlights a research gap which suggests that the RCM process in agile software development has not received enough attention from researchers, especially in the GSD context [4]. Kamal et al. [4] mention that the RCM models/frameworks in the literature lack the roadmap for agile development. Hence there is a strong need for a robust, agile requirement change management process for distributed software development.

1.1 Motivation

According to a survey conducted by SGoI (Standish Group of International), only 29% of software products are considered successful, while 53% are doubtful, and 18% are unsuccessful [2]. The main reason for this high failure rate is attributed to the improper management of requirements [7]. In addition, McGee & Greer's study [8] revealed that inadequate requirements management is associated with 60 to 65 causes of project failure. Lindquist [4] found that 71% of software projects failed due to poor management of RCM activities, and Sirvio and Tihinen's survey study [9] of European software firms also reported that 40% of software projects failed due to inadequate requirements management. These statistics highlight the significant impact of requirements changes management on the success of global software projects. Multiple research studies have proposed various models and frameworks to enhance the RCM process. However, most of these frameworks focus on improving the RCM process for single-sited environments [4]. The existing RCM models are inadequate to address the complexities of the GSD environment. Hence there is a dire need to develop RCM standards and models explicitly tailored for GSD projects.

Kamal et al. [4] highlighted an additional research gap that exists beyond the lack of RCM models for global software development (GSD). Specifically, there is a dearth of studies that investigate the RCM process within the context of agile software development in GSD environments with the aim of proposing new models, tools or techniques [4]. Global software development (GSD) and agile software development (ASD) are recent trends in modern software engineering practices [10] that effectively complement each other. According to VersionOne of the 14th annual State of the Agile report, a significant majority (71%) of respondents reported that their organizations employ agile methodologies in the GSD paradigm [10]. Agile methodologies are increasingly applied to GSD projects to facilitate the dynamic management of requirement changes in software products. Therefore the lack of frameworks or models that can provide guidance on implementing an agile requirement

change management process for distributed software development poses a significant research gap that requires attention from researchers.

1.2 Research Objectives

The research objectives of the study are as follows:

- To identify and analyze the limitations of existing RCM models in the literature.
- To propose a comprehensive and robust requirements change management model that addresses the limitations of existing models and is tailored for agile software development in the global software development paradigm.
- To assess the efficacy of the proposed model in enhancing the requirements change management process within the agile GSD paradigm by conducting an empirical study with RCM industry experts.

1.3 Structure of the Report

The rest of the project report is organized as follows:

Chapter Two provides a summary of the research background and related work in the field of study.

Chapter Three outlines the research methodology employed in the study.

Chapter Four provides a detailed explanation of the different phases comprising the proposed Agile Requirements Change Management (ARCM-GSD) model.

Chapter Five presents the data analysis and reports the results derived from the analysis.

Chapter Six discusses the improved ARCM-GSD Model, which is derived from expert recommendations.

Chapter Seven provides a discussion of the research findings in relation to the related work in the field.

Chapter Eight provides a conclusion to the study and outlines potential areas for future research.

Chapter Two

Background and Related Work

2.1 Background

2.1.1 Global Software Development (GSD)

Global software development (GSD), also known as global software engineering (GSE) is a paradigm in which experts from diverse geographical locations, speaking different languages, having varied cultural backgrounds, and operating in different time zones, collaborate on software development activities [11]. A formal definition of GSD [11] is as

"practices where geographically distributed collaborators are mutually engaged and thus interdependent in (both software engineering and) software development activities with the aim of designing, programming, and implementing an IT-system."

Collaborative practices in global software development can take the form of outsourcing (a collaboration between different organizations), offshoring (collaboration within the same organization), or a combination of both [11]. Globalization of the software industry has gained popularity over the last two decades due to its potential to provide multiple benefits for software development projects. The adoption of GSD offers access to a large pool of skilled workers, a round-the-clock development cycle, economic gains, development cost reduction, faster time-to-market, proximity to the global market, and the quick formation of virtual teams [6] [2] [12]. However, the adoption of GSD also poses several challenges in

communication, coordination, and control of the software development process [10], which are risk factors to the success of software projects. Despite these difficulties, most software development organizations continue to explore the potential advantages of GSD and work to overcome the associated problems.

2.1.2 Requirements Change Management (RCM)

Software projects are prone to requirement changes at all stages of the software development life cycle, from requirements gathering to post-deployment [2]. Software systems are subject to evolution due to dynamic and unstable requirements [11]. According to Nurmuliani [13], requirement changes can be defined as

"the tendency of requirements to change over time reacting to the evolving needs of customers, stakeholders, organizations, and the work environment."

A variety of factors can cause requirement changes. Modifications to requirements can be motivated by dynamic customer needs, improved system understanding, organizational needs, project objectives, market competition, and technological advancements [2]. Additionally, according to Ambler, requirement changes are demanded due to frequently changing stakeholder needs, missing a critical requirement in the initial phase, system bugs translating into new requirements and inadequate understanding of initial needs [11].

Managing rapidly changing software requirements is a primary requirements engineering practice to deliver high-quality software that meets client demands [11]. Successful requirements engineering incorporates techniques to handle requirements volatility [11]. The process of requirement change management (RCM) involves the interpretation, analysis, management, control, tracing, and documentation of changes in requirements [12]. Effective requirements change management (RCM) is a collaborative process that relies on communication and coordination between project stakeholders [9]. Poor RCM practices can result in various issues, such as increased software cost, schedule delays, unstable requirements, and

extended testing cycles, ultimately leading to project failure and negatively impacting the business [9].

2.1.3 Agile Software Development (ASD)

Agile methodology was introduced to respond to the business demand for faster web and mobile application development [10]. Agile is a development process that adopts an incremental and iterative approach, beginning with a planning phase and continuing to the deployment phase. Agile software development is a more lightweight process than traditional development methods like the waterfall model, as it emphasizes iterative and incremental development that delivers software faster in smaller chunks and incorporates feedback from stakeholders at earlier stages of the process. As a result, agile guarantees higher customer satisfaction as it delivers working software features within a shorter duration [10]. The significant benefits of adopting agile methods include frequent delivery of working software, adaptive response to changing requirements, feature prioritization based on evolving business needs, and faster and frequent feedback from stakeholders [10].

2.1.4 Requirements change management (RCM) process in agile global software development (AGSD)

Agile methodologies were initially developed for collocated software development teams [10], as agile emphasizes effective communication and teamwork. However, multiple recent studies have proven that agile methods are effective at mitigating challenges faced in GSD environments [10]. And since both agile software development (ASD) and global software development (GSD) are significant software industry trends that will continue to proliferate, there is a growing need for further investigation into how these two methodologies can effectively work together [10]. Understanding the benefits and challenges of combining these two approaches will help establish improved software development practices and optimize

the success rates of GSD projects [10].

Agile development, as well as requirements change management, are both team-oriented processes and rely heavily on communication and coordination between stakeholders [4]. Despite the similarities between both approaches, the adoption of GSD can pose a challenge to implementing change management in an agile development environment [4]. Agile practices do not emphasize project documentation, and insufficient documentation can cause complexities in the change management process [4]. Another key challenge faced by the distributed agile teams during the RCM process is the lack of frequent informal communication among stakeholders due to the geographical separation between offshore sites, which is essential to build trust among agile teams [4].

2.2 Related Work

The dynamic nature of software systems makes them prone to frequent requirement changes. And adequately managing these changing requirements is the main focus and top priority of competitive businesses to remain relevant in the global software market. Requirement change management is one of the most complex and intricate processes in requirement engineering. Consequently, researchers have invested significant efforts over the years to develop and establish standards and frameworks for effectively managing requirement changes.

During the early years of change management, various models were proposed, such as the Ince model [14], which consists of five phases: "change request," "rejection," "batch," "implementation," and "updating." However, these models were found to be missing critical phases, such as understanding the need for change and change verification, to ensure the efficient functioning of the software system [15]. Other models, such as V-like [14], S.A. Bohner [16], CHAM [17], and S.A. Ajila [18], also neglected the change impact analysis phase, while the spiral model was missing change decision, and validation phases [15]. Due to the limitations of these models, more comprehensive RCM frameworks were developed.

Bhatti et al. [8] developed an RCM model for collocated development teams with the following stages "initiate," "receive," "evaluate," "approve or disapprove," "implement," and "configure." The model aims to manage constant requirement changes throughout the project implementation [8]. Bhatti et al. briefly mention the involved stakeholders, artifacts and activities for every phase. But the model ignores critical steps like change verification and notifying stakeholders/clients after change completion [4]. Keshta et al. [19] and Niazi et al. [20] developed models catering to the Capability Maturity Model Integration (CMMI) Level 2 specific practice - SP 1.3 "manage requirements changes" for single site development. The model by Keshta et al. [19] is divided into six primary stages: "initiate," "validate," "implement," "verify," "update," and "release." It is explicitly designed for small and medium-sized organizations and cannot handle requirements changes for large organizations and firms with distributed work locations. Niazi et al. model [20] consists of five phases: "request," "validate," "implement," "verify," and "update." The model does not describe the communication process and the user roles performing different phases in a single-site development setting, which makes it even more unsuitable for globally distributed teams [4].

The intensive communication needed for the requirements change management process makes it a challenging activity for collocated software teams [2]. With software companies transforming into global software enterprises due to economic and strategic benefits [2], the associated RCM challenges increase multifold times due to a lack of interaction and trust among team members. Over the years, researchers have developed models to overcome such RCM and GSD challenges.

Akbar et al. [2] proposed the AZ-Model of RCM for resolving communication issues during the requirement change management process in the GSD paradigm. It consists of three phases: “coordination,” “analysis,” and “development and implementation.” The model emphasizes special project management practices and allocates fixed time to each phase to complete GSD projects under time and budget [2]. The model covers major RCM activities

and describes how communication and coordination activities should occur between GSD teams. However, the model still overlooks certain essential RCM activities like requirement categorization and prioritization. Notably, Akbar et al. [2] did not assert the effectiveness of the AZ-Model in agile development projects. Minhas et al. [12] proposed an RCM framework to mitigate coordination challenges stemming from cultural differences, time zones, and language barriers that exist within GSD teams. The change management framework features support for multiple languages, allowing for communication and understanding to take place despite linguistic barriers. The primary focus of the framework is on time boxing and the voting process by the Change Control Board (CCB). However, the framework overlooks important aspects of the RCM process, including the understanding of the need for change, change impact analysis, change verification/validation, and informing the change initiator, stakeholders, and clients. Khan et al. [21] suggested a framework for managing RCM in the GSD context. The framework covers limited steps in the RCM process, including "Change Initiation," "Change Evaluation," "Change Decision," and "Change Implementation," and is missing crucial steps like change analysis, archiving rejected changes for future reference, verifying implemented changes and informing stakeholders.

The Global RCM (GRCM) developed by Hussain et al. [7] is an RCM model that provides a detailed description of the roles, activities, and artifacts involved in each stage of the RCM process, making it highly suitable for GSD. The model suggests the use of collaborative technology to enhance communication and coordination and is easy to understand and implement in GSD organizations. It comprehensively outlines the necessary communication and coordination actions for each phase of the RCM process. However, the model lacks intrinsic details regarding the development phases, and it does not specify how to prioritize or categorize different requirement changes. The GRCM model was designed and evaluated for the waterfall development process. Qureshi et al. [3] proposed a conceptual model to mitigate communication and coordination problems faced by globally dispersed teams during software change management. The model consists of three key phases: "identification and

categorization of communication and coordination challenges," "identification and allocation of mitigation practices," and "implementation of mitigation strategies." The evaluation of the model by GSD experts implies that it effectively decreased communication challenges via suggested mitigation practices.

Some models have been developed to help organizations evaluate the maturity of their requirement change management process before moving forward with global software development [1]. These models help software firms understand their RCM process better and suggest improvements [1]. Akbar et al. [22] designed a model to evaluate the readiness of organizations willing to adopt requirements change management processes in GSD. The model has five levels - "Initial," "Basic," "Managed," "Information Sharing," and "Continuous Improvement." Each level is mapped with RCM best practices identified through a systematic mapping study [22]. The proposed RCMRM model provides a guideline for organizations to assess and improve their RCM process in the distributed development environment [22]. Similarly, SRCMIMM [1], a software requirement change management and implementation model presents a way for GSD organizations to assess the RCM process. The SRCMIMM model levels are based on RCM challenges, success factors and best practices identified through a systematic literature review and questionnaire survey.

The RCM frameworks discussed above in the literature mainly focused on improving RCM activities in single-site development environments, with only a few frameworks considering the global development aspect. Moreover, existing studies reveal a significant research gap in RCM models designed for agile software development in the GSD context [4]. Many of these models were designed for step-by-step waterfall development and are not suitable to meet the needs of the iterative agile development process.

Shehzadi et al. [23] proposed a change requirement management (CRM) framework for agile development that is specifically designed for single-site organizations. However, this framework is not suitable for managing requirement changes in GSD organizations, and it does not provide guidance on how to integrate the RCM process with agile activities and

user roles. The CRM framework considers a change request requiring less than 20 hours of work as a small change and recommends implementing it immediately without considering the sprint capacity.

As organizations increasingly shift towards global software development and embrace agile methodologies, there is a growing need for comprehensive RCM frameworks that can effectively manage evolving software requirements. This study seeks to address this need by designing a robust RCM framework that can support Agile GSD firms in efficiently managing continuous changes in software requirements.

Chapter Three

Research Methodology

The purpose of this study was to gain an understanding of the requirements change management process for agile software development within a global context and to propose a comprehensive agile RCM model for globally distributed teams. The steps outlined in Figure 3.1 were followed to achieve the research objective. The study comprised ten steps, ranging from conducting a literature analysis of the selected research domain to presenting the research results. A thorough examination of existing RCM models and frameworks for Global Software Development was performed, and the limitations of these models were identified. The analysis revealed that current RCM models do not adequately address the needs of GSD teams that adopt agile methodologies. The literature was also reviewed to determine the challenges, success factors, and best practices associated with the Agile RCM process in GSD.

Based on the literature review findings, a theoretical Agile RCM model for GSD was devised to address the shortcomings of existing models. This model aims to provide a comprehensive solution for globally distributed teams seeking to manage requirement changes within an agile software development environment.

3.1 Research Steps

The study follows a research approach similar to [3], which is discussed in the sections that follow.

Figure 3.1 Research steps

3.1.1 Decide the Research Domain

The research study was initiated by exploring the existing literature on the RCM process in the global paradigm. Multiple research studies were analyzed to understand the RCM process in the context of GSD. Based on the literature analysis of multiple studies, a research gap was highlighted, which indicates that the agile RCM process has not received enough attention from researchers [4]. This gap in the literature suggests a need for more research to understand the challenges while adopting the agile RCM process in GSD. Conducting additional studies could provide valuable insights about best practices for organizations looking to implement an agile RCM process in their GSD practices. Further analysis revealed a lack

of RCM models/frameworks that support agile development at distributed work locations.

3.1.2 Analyze Existing Literature

Choosing Digital Repositories

This step involved thoroughly analyzing previous research studies focused on agile RCM in the GSD paradigm. To meet the research objectives, identifying relevant literature is crucial. Therefore to ensure comprehensive coverage of relevant literature, the study used digital libraries, including ACM Digital Library, IEEE Xplore, Scopus, and ScienceDirect. These databases are widely recognized as reputable sources for academic literature publication. Additionally, we searched Google Scholar to ensure all crucial studies were included.

Search String

The selected research papers focused on the challenges of requirements change management (RCM) and global software development (GSD) and suggested models or frameworks to tackle these difficulties. This step aimed to uncover RCM models and frameworks in literature and comprehend the RCM processes they encompass [3]. Furthermore, the study necessitated a comprehensive literature review on agile methodologies applied to both RCM and GSD in order to identify the best practices, challenges, and success factors within this domain. The search string used to identify studies from the literature is similar to [4]:

("requirements change management" OR "RCM" OR "requirements management" OR "requirements changes" OR "requirements volatility" OR " requirements change management practices " OR "effect of requirements change management" OR "impact of requirements change management") AND ("Agile" OR "Extreme Programming" OR "XP" OR "SCRUM" OR "Kanban") AND ("GSD" OR "Global software development" OR "Distributed software development" OR "Outsourcing" OR "Offshore software development" OR "Collaborative software engineering" OR "Multisite software development" OR "Global software teams" OR

"Collaborative software development").

Inclusion Criteria

The inclusion criteria for selecting studies is adopted from [4]:

- The study publication format must be a book chapter, conference paper, or journal article.
- The study must define the RCM process in the GSD paradigm.
- The study must discuss the application of agile methodologies to the RCM process.
- The study results should be based on empirical evaluation.

Exclusion Criteria

The exclusion criteria used to select studies is taken from [4]:

- The article lacks a detailed description of the Agile RCM process.
- The final study is considered in case of duplicate publications.
- Studies written in a language other than English are excluded.
- The publishing source of the study was not reliable.

Quality Evaluation

The quality evaluation criteria were designed based on input from the existing studies [5] [4].

- Does the selected study outline the RCM process?
- Does the selected study discuss RCM activities in the GSD paradigm?
- Does the selected study explore the application of agile methodologies to the RCM process?

- Does the selected study propose an RCM model?

Each study was evaluated based on its response to the quality evaluation questions. A study that answers a question receives a score of 1; if it partially answers, the score is 0.5; and if it fails to answer the question, a score of 0 is assigned. The selected studies, along with the total score, are documented in Appendix-A. A 50% QE score was established as the threshold for selecting studies [4]. A total of 29 research studies were selected from the literature, which are listed in Appendix A.

3.1.3 Identify Limitations of Existing RCM Models/Frameworks

The literature suggests that multiple models and frameworks were developed to resolve challenges in requirements change management for distributed software development. In this step, the existing models and frameworks were meticulously analyzed to uncover their limitations. Additionally, based on the analysis results, the study proposed a conceptual agile requirements change management (ARCM-GSD) model specifically for collaborative software development.

3.1.4 Design ARCM-GSD model

The study proposed a novel theoretical agile requirements change management (ARCM-GSD) model to address the lack of research on managing requirements changes in the GSD context using agile methods. Additionally, the model suggests solutions to overcome the limitations of the existing RCM models and frameworks.

3.1.5 Select Industry Experts

The proposed ARCM-GSD model was assessed by software industry experts with a minimum of 5 years of experience in RCM and agile in GSD, selected from diverse countries. The evaluation involved collecting feedback through surveys and interviews, which were recorded

for further analysis. The potential experts were identified using LinkedIn, Facebook, and Research-Gate. The bibliographic information of industry experts is listed in Appendix C.

3.1.6 Prepare Survey and Interview Questions

A survey was developed to validate the proposed ARCM-GSD model, following the guidelines from relevant studies [3]. The survey was divided into two parts: respondents' background information and ARCM-GSD model evaluation. The respondent information section included questions about personal details, geographical demographics, work experience and organizational details. The ARCM-GSD model evaluation was further divided into open-ended and closed-ended questions. The close-ended questions evaluate the model's design, ease of use, suitability for agile and GSD, coverage of RCM activities, and the importance of a geographically distributed change control board. Likert scale was applied to obtain the experts' opinions, with five options: "strongly agree," "agree," "neutral," "disagree," and "strongly disagree" [4]. In the open-ended section, the experts were asked to discuss the limitations of the ARCM-GSD model and suggest improvements. They were also asked about RCM challenges and the tools used for collaboration in RCM. The survey was tested by two experts, and the design was improved based on their suggestions before sending the survey to a larger population.

Semi-structured interview questions were designed to gain further insight into the proposed ARCM-GSD model. The participants were encouraged to provide more detailed responses to the questions, specifically open-ended questions where the experts comprehensively explained the model's limitations and suggested improvements. Through these interviews, we aimed to gather richer information to complement the survey and further validate the ARCM-GSD model. The detailed survey and interview questions can be viewed in Appendix B.

3.1.7 Model Assessment: Survey, Interviews, and Comparative Analysis

The survey was distributed to potential RCM GSD experts using a snowball sampling technique [24] through LinkedIn, Facebook, and Research-Gate [4] [3]. The snowball sampling technique [24] is an effective and convenient method for reaching potential respondents. The online survey feedback was collected from January 2023 to March 2023.

Subsequently, the study entailed conducting semi-structured interviews with RCM experts who had consented to participate. The objective of conducting interviews was to obtain detailed insights and feedback regarding the proposed ARCM-GSD model, as well as to gather information about the latest trends in the industry related to the RCM process. The experts were asked to assess the design of the ARCM-GSD model, identify its limitations, and suggest improvements. Additionally, challenges encountered within the scope of agile RCM in Global Software Development and collaborative tools and technologies utilized for team communication and coordination were also examined.

Lastly, a comparative analysis was conducted to evaluate the ARCM-GSD model against established RCM models found in the literature. By employing a comparative methodology, as outlined by [23], we were able to assess whether the ARCM-GSD model addresses the limitations of traditional RCM models and identify opportunities for further improvement. The results of the comparative analysis are discussed in detail in Chapter Seven.

3.1.8 Data Analysis

The responses from surveys and interviews were recorded. The responses were filtered to only include those from experts with industry experience in agile, RCM, and GSD. The collected data from both the open-ended and close-ended survey questions, as well as the interviews, was subjected to analysis through software tools such as Microsoft Excel [3].

3.1.9 Model Refinement and Validation through Interviews

Based on the survey and interview feedback received from industry experts, the ARCM-GSD model was revised by incorporating feasible improvements. The enhanced ARCM-GSD model was subjected to a second round of validation through follow-up interviews with RCM experts who had participated in the initial evaluation. During these follow-up interviews, the participants provided feedback on the implemented changes.

3.1.10 Report the Evaluation Results

The findings of the data analysis were utilized to evaluate the effectiveness of the ARCM-GSD model. The evaluation results encompass information regarding the experts' work location, job role, industry domain, experience, and educational background, as well as the assessment of the ARCM-GSD model [3]. A comprehensive overview of the results is presented in Chapter Five.

3.2 Ethical Considerations

As a prerequisite for obtaining ethical approval, an application was duly submitted to the Human Research Ethics Board (HREB). The study received approval from the human research ethics board with reference number 22-0204. In adherence to the ethical standards, the study participants voluntarily provided written consent to participate in the research study. The sample consent form is provided in Appendix B.2.

Chapter Four

Agile Requirements Change Management Model for Global Software Development

The literature on requirement change management models suggests a lack of capability to handle changes in a global development setting [7]. To the best of our knowledge, no existing RCM models in the literature cater to the agile development process in the GSD paradigm. This study aims to address this research gap by proposing an agile requirement change management model designed explicitly to provide a systematic guideline for conducting change management in GSD organizations that follow an agile approach. The proposed model builds upon existing RCM models/frameworks in the literature [2, 20, 7, 12, 19, 23], which include fundamental change management phases such as Initiate, Validate, Implement, Verify, Update, and Release. In addition to these phases, the model introduces new stages, such as traceability, categorization, prioritization and effort estimation of change requests, that integrate agile methods with the RCM process. The study extends the prior work by [7] [23] to design an RCM framework tailored for globally distributed agile development teams. In contrast to existing models, the ARCM-GSD model incorporates stakeholder/client communication and integration with a project tracking tool. Furthermore, the study highlights

activities, roles and artifacts (ARA) coverage for each phase of the model to improve its industrial usability [7].

4.1 ARCM-GSD Model Phases

The following section explains the agile requirement change management process steps in detail.

4.1.1 Change Initiation

Change initiation is the first step in the requirement change process. Requirement changes can be initiated by clients or stakeholders from any geographic location at any time during the agile development process through the submission of a change request. The request will include information regarding the nature of the change, the inherent business value, priority, severity and a detailed description. The change request, upon submission, will be recorded in the requirements change management database. And the request will be assigned to the change control board for a thorough evaluation [12].

4.1.2 Understanding the need for change and the change type

In each agile sprint, the Change Control Board will review the change requests stored in the database, in order of the assigned priority, given the available capacity of the sprint. The Change Control Board may involve the change initiator and the client in comprehending the proposed changes. The CCB ensures that the change request is effectively formulated and that the request captures all relevant information required for further analysis. This step also mandates identifying the type of requested change for appropriate categorization. Additionally, CCB may also revise the priority of the change based on the enhanced comprehension of the requested modifications.

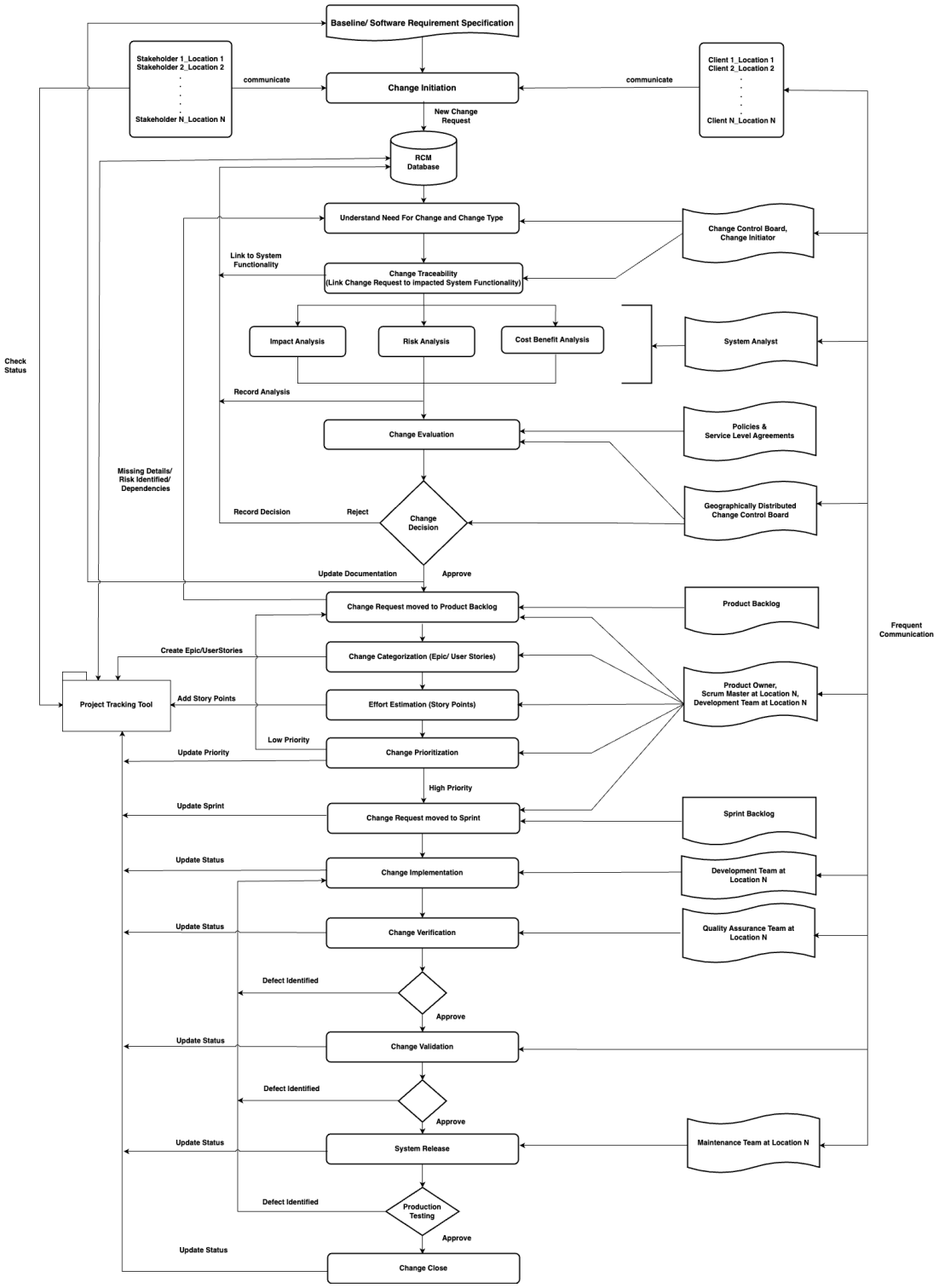

Figure 4.1 Proposed Agile RCM model for GSD

4.1.3 Change Traceability

The existing literature on RCM models has disregarded the crucial aspect of change traceability, a significant factor in tracking the evolution of system requirements. Most models treat each submitted change request as a new submission, potentially leading to resource wastage if a similar request was received previously. Multiple research studies have indicated that the issue of change traceability in requirements has emerged as a crucial challenge within the domain of Requirements Change Management [25, 26]. However, the currently prevalent RCM models and frameworks have not accorded adequate importance to tracing the new proposed changes to the system's existing functionality. This step is vital as it ensures that the proposed changes are aligned with the system's objectives. Additionally, mapping the new change request to the system functionality that it impacts would provide the Change Control Board (CCB) team with a clear understanding before evaluating the change. And this will facilitate the centralized monitoring of all proposed modifications throughout the project. Furthermore, change traceability facilitates effective impact analysis. Thus, including change traceability in the RCM process is imperative to ensure effective change implementation and maintenance of a stable software system.

Consequently, the ARCM-GSD (Agile Requirements Change Management) model incorporates a change traceability step, wherein the change control board will ascertain whether the request is new or has been previously submitted. Requests documented earlier in the backlog will be recognized as duplicates within the Requirements Change Management database. Furthermore, the newly introduced change request will be linked to the impacted system functionality in the RCM database. The change control board is responsible for establishing the traceability of the new changes with the help of the change initiator.

4.1.4 Change Analysis

Most of the existing Requirements Change Management (RCM) models found in the literature do not provide a comprehensive outline of the change analysis process. The Agile Requirements Change Management (ARCM-GSD) model, on the other hand, recommends a thorough analysis of the proposed modifications. The ARCM-GSD model separates the change analysis process into three distinct components: change impact analysis, risk analysis, and cost-benefit analysis. Change impact analysis determines the scope of the change and the impacted areas of the system, which is crucial in assessing the impact on the project in terms of expense, effort and schedule [19]. The ARCM-GSD model incorporates the assessment of potential risks associated with the implementation of a change in addition to change impact analysis. The objective of conducting a risk analysis is to recognize technical, scheduling, and budget-related risks and to develop strategies to mitigate these risks. The cost-benefit analysis within the ARCM-GSD model assesses the financial implications of the change request by weighing the expected benefits against the cost of implementation in terms of resources, time, and effort. The responsibility for the change analysis step falls on the team of system analysts, who are responsible for conducting a detailed analysis for each change request and recording the results in the Requirements Change Management database.

4.1.5 Change Evaluation and Decision

The change evaluation process involves an in-depth evaluation of the results from the change impact analysis, risk analysis, and cost-benefit analysis, along with other relevant information about the proposed change. This step aims at assessing the feasibility of the suggested changes. The information gathered from this step helps make informed decisions regarding implementing change requests. The Change Control Board is responsible for change request evaluation.

After gaining a thorough understanding through the evaluation process, the next step is to decide whether the change should be implemented. Hence the CCB team announces a formal decision based on the results of the evaluation phase [8]. The decision taken by the Change Control Board regarding the change request is formalized. If the change request is approved, the reasons for approval are recorded, and the development team responsible for implementing the change is notified [8]. On the other hand, if the change request is rejected, the reasons for rejection are documented [8]. In either case, the change decision for the request is recorded in the RCM database for future reference.

4.1.6 Change Request to Product Backlog

In the ARCM-GSD model, approved change requests are added to the product backlog. The stakeholders review the change requests in the product backlog meeting based on the order of priority. The stakeholders discuss the implementation details and refine the change requests, ensuring that they are clear, concise, and ready to be implemented. If any missing information, identified risks or dependencies are detected during the product backlog meeting, the change request is referred back to the Change Control Board team for further analysis. The responsible stakeholders for this step include the product owner, the scrum master, and the development team.

4.1.7 Change Categorization, Effort Estimation and Prioritization

The categorization of change requests is a vital aspect of the RCM process. However, most RCM models overlook this step. This step distinguishes change requests based on their complexity and required implementation time [23], which helps in the effective and efficient management of changes. The change requests are classified into broader work units, referred to as epics or more specific work units, referred to as user stories, based on the estimated effort and implementation details for the change. A change request tagged as an epic is

further categorized into smaller user stories. In the next step, effort estimation is carried out for the change request using agile methods such as planning poker to estimate story points. Furthermore, the priority of the change request is determined, with high-priority change requests being assigned to the sprint backlog based on the available capacity within the sprint. In contrast, low-priority changes remain in the product backlog. The participating agile user roles include the scrum master, the product owner, and the development team, who may be located at dispersed locations.

4.1.8 Change Request to Sprint Backlog

Prioritized change requests are moved to the sprint backlog based on available sprint capacity. In the sprint planning meeting, the assigned team reviews the user stories in the backlog and identifies tasks and implementation details for each user story. The team will also assign user stories to individual team members. The primary stakeholders of this step are the product owner, the scrum master and the development team.

4.1.9 Change Implementation

During the sprint cycle, the designated developer(s) initiate the implementation of the change request by providing status updates at each stage to the stakeholders. Furthermore, the developer(s) conducts preliminary testing on the new changes before transferring the deliverables to the quality assurance team for comprehensive testing.

4.1.10 Change Verification and Validation

In the verification step, the implemented changes are tested against the original change request to ensure all the requested modifications are executed accurately. If any deviations are observed during verification, the development team is reassigned the change request to address the identified issues. Additionally, regression testing is carried out as part of change

verification to ensure that the new changes did not impact existing system functionality. The quality assurance team is responsible for verifying the changes in terms of functional and non-functional aspects. The status of the verification process is updated in the project tracking tool.

The subsequent step is user acceptance testing or change validation. The development team presents the developed changes to the stakeholders and client team. The stakeholders/client team tests the latest changes for compliance with system requirements. Issues identified during the process are reported to the development team for correction. Change requests approved by the stakeholders and client team are marked as ready for production release. The maintenance team is responsible for deploying the changes to production and reporting any issues that arise during production testing.

Chapter Five

Data Analysis and Results

This chapter presents the findings of the analyzed results of the proposed ARCM-GSD model, which were derived from the data collected through a questionnaire survey and interviews with industry experts. In this study, we received a total of 49 survey responses. Each response went through a manual evaluation process. Upon review, it was determined that a total of 38 responses were relevant to our study. Among the remaining 11 responses, 3 participants lacked experience in Global Software Development (GSD), 4 respondents had no prior experience in Requirement Change Management (RCM) activities, and 4 responses were incomplete. In addition, the study included semi-structured interviews with 8 practitioners of Requirement Change Management (RCM) in Global Software Development (GSD). The complete findings from both the survey and interviews are presented in the following sections.

5.1 Demographic Information of Experts

The participants of this study were industry practitioners with expertise in the field of agile development, requirements change management, and global software development. The snowball sampling technique [24] was used to identify potential industry experts with a minimum of 5 years of industry experience. The selected experts were situated in various countries, such as India (65.8%), Nepal (2.6%), the United States (13.1%), Canada (2.6%),

Australia (7.9%), Ireland (2.6%) and Germany (5.3%). Figure 5.1 illustrates the demographic distribution of the experts.

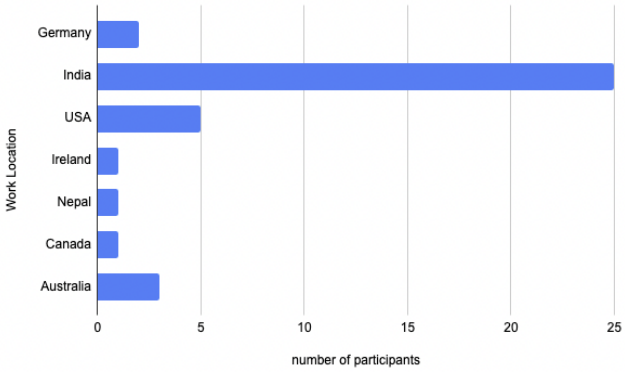

Figure 5.1 The demographic distribution of the industrial experts.

5.2 Gender Distribution of Experts

The gender distribution of experts who participated in evaluating the ARCM-GSD model was also recorded. According to the collected data, 73.7% of the survey respondents were males, while 26.3% were females. Figure 5.2 depicts the gender distribution of the industry experts.

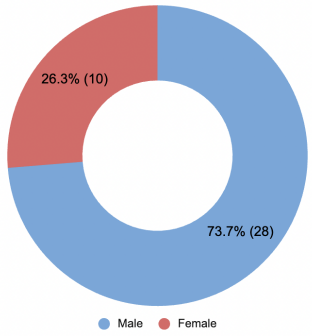

Figure 5.2 Gender distribution of the industrial experts.

5.3 Educational Background

Educational background is a critical criterion in the selection of experts evaluating the ARCM-GSD model. Hence experts with a strong educational background in computer science, software engineering, and information technology were selected for the evaluation of the ARCM-GSD model. Responses from participants who did not have an educational background in computer science, software engineering, or information technology were excluded from the analysis as they were considered not relevant to the study. Figure 5.3 illustrates the educational background of the experts chosen for the study. Based on the educational information of the survey respondents, 28.9% are from the computer science field, 36.8% are from software engineering and 34.2% have a background in information technology.

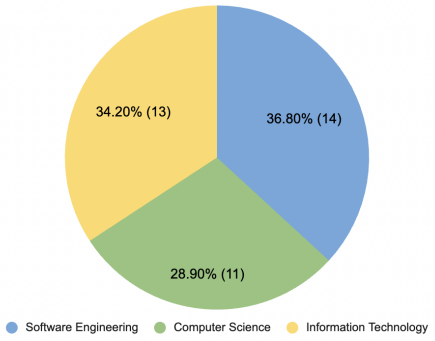

Figure 5.3 Educational background of the industrial experts.

5.4 Experience in GSD, ASD and RCM

To ensure a thorough and effective evaluation of the ARCM-GSD model, the study included only responses from software industry experts with at least four years of experience in global software development similar to [3]. Likewise, a nuanced understanding of the agile development process is critical for the comprehensive evaluation of the ARCM-GSD model, which required the study to include the participation of experts with at least four years of experience working with agile methods. Additionally, the study also ensured that the par-

ticipants had an excellent understanding of the requirement engineering and requirements change management processes. Specifically, only experts with prior experience participating in software requirements engineering and requirements change management activities were included in the study. The selected experts had software industry experience ranging from 5 to 26 years.

5.5 Professional Designation of Experts

The survey included experts with diverse job roles to ensure a comprehensive evaluation of the ARCM-GSD model from various aspects of the software development life cycle. Since different agile user roles are concerned with different phases in the SDLC, this approach aimed to provide a well-rounded evaluation of the model. The job designation of the industry experts is shown in Figure 5.4. The breakdown comprised of Project managers (34.2%), Chief Technology Officers (2.6%), Scrum masters (7.9%), System architects (5.3%), Business analysts (5.3%), DevOps engineers (5.3%) and Developers (39.5%).

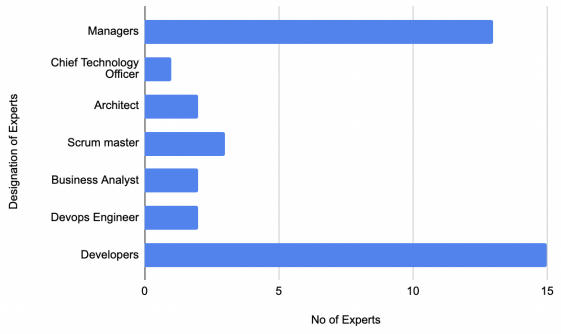

Figure 5.4 Professional designation of the industrial experts.

5.6 Organizational size

The objective of the study was to gather feedback from a range of respondents representing organizations of varying sizes as determined by employee strength. The distribution of

experts among organizations is shown in Figure 5.5. None of the experts belonged to organizations with 0 to 10 employees, 10.5% were affiliated with organizations that had between 10 to 49 employees, 18.4% were associated with organizations that had between 50 to 249 employees, and the majority of 71% were affiliated with organizations that had more than 250 employees.

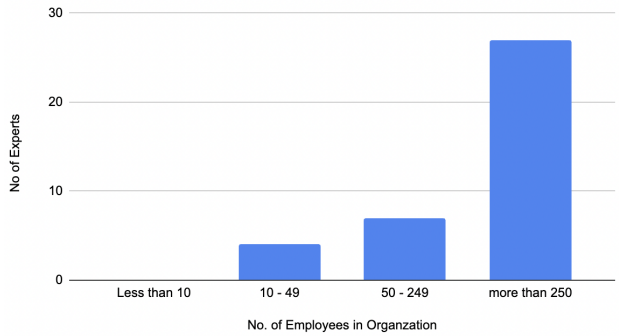

Figure 5.5 Distribution of experts based on the organization's employee strength.

5.7 Business Domain

Experts were selected from diverse organizational domains to avoid biased feedback from any particular industry type. The selected survey respondents were from business domains such as finance, eCommerce, banking, telecommunications, aviation, healthcare, business intelligence, business analytics, insurance, automobile and supply chain.

5.8 ARCM-GSD Model Assessment

The ARCM-GSD model was validated by assessing factors such as ease of use and understandability, suitability for agile and GSD, RCM activity coverage, and inclusion of CCB members from geographically dispersed locations. A five-point Likert scale comprising the following options: strongly agree, agree, neutral, disagree, and strongly disagree, was employed in this process [4]. To facilitate the interpretation of industry expert opinions,

the aforementioned five options were categorized into broader categories such as positive (strongly agree and agree), negative (strongly disagree and disagree) and neutral [4].

No. Respondents = 38								
Model Assessment	**Positive**			**Neutral**		**Negative**		
	S.A	**A**	**%**	**N**	**%**	**D**	**S.D**	**%**
RCM model is easy to understand and implement	12	20	84	6	16	0	0	0
RCM model is suitable for Agile software development	16	18	89	4	11	0	0	0
RCM model is suitable for Global Software Development	13	20	87	5	13	0	0	0
The model accounts for all the major Requirement Change Management activities	19	16	92	3	8	0	0	0
Do you agree that GSD organizations will benefit from having requirements change control board representatives from all geographically distributed sites.	17	15	84	5	13	1	0	3
S.A=Strongly agree, A=Agree, S.D=Strongly disagree, D=Disagree, N=Neutral								

Table 5.1 ARCM-GSD model assessment

The findings presented in table 5.1 illustrates that a significant majority of experts (84%) concur that the ARCM-GSD model is comprehensible and easy to implement. Furthermore, 89% and 87% of respondents opined that the ARCM-GSD model is well-suited for agile methodologies and global software development, respectively. The majority of experts (92%) agreed that the ARCM-GSD model encompasses all requisite RCM activities. Additionally, 84% of experts agreed that GSD organizations would benefit from incorporating requirements change control board representatives from all geographically dispersed locations, with only one expert expressing disagreement. The analyzed results in table 5.1 illustrate that values of positive responses (agree + strongly agree) are significantly greater than negative responses (disagree + strongly disagree), which indicates a favourable positive reaction towards the ARCM-GSD model from the industry.

	No. Respondents=38							
	Poor	%	Average	%	Good	%	Excellent	%
ARCM-GSD model design evaluation	0	0	2	5	22	58	14	37

Table 5.2 ARCM-GSD design evaluation

Moreover, the expert evaluations (table 5.2) of the ARCM-GSD design revealed that 37% of experts deemed it as excellent, while 58% rated it as good. A small proportion, 5%, regarded the design as average, and notably, none of the experts classified the design as poor.

5.9 Collaborative Software Tools

The experts were queried about the collaborative software tools employed in their respective organizations that facilitate collaboration among distributed teams. The expert feedback indicates the utilization of a multitude of collaborative software tools. As shown in Figure 5.6, the GSD organizations make use of a diverse range of software tools, including Jira (36.8%), Teams (23.0%), Confluence (13.8%), Trello (4.6%), Slack (4.6%), SharePoint (3.4%), ServiceNow (3.4%), Git (3.4%), Gerrit (1.1%) and many others.

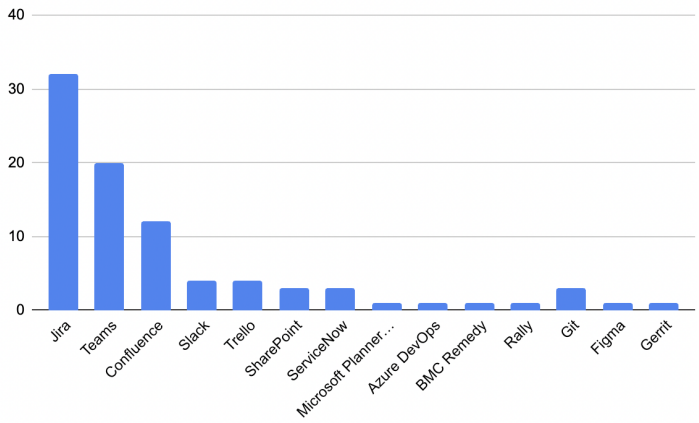

Figure 5.6 Collaborative software tools used by industry experts.

5.10 ARCM Challenges in GSD Organizations

The participants were asked to provide insights into the challenges distributed agile teams face during the requirement change management process. As per the results, 76.3% of the respondents identified communication and coordination as the most significant challenge for the RCM process in the GSD domain. Additionally, knowledge management (44.7%), RCM process factors (39.4%), technology factors (10.5%), and project administration (28.9%) were also cited as other critical challenges in the field. Figure 5.7 provides a visual representation of the ARCM challenges faced by the GSD teams.

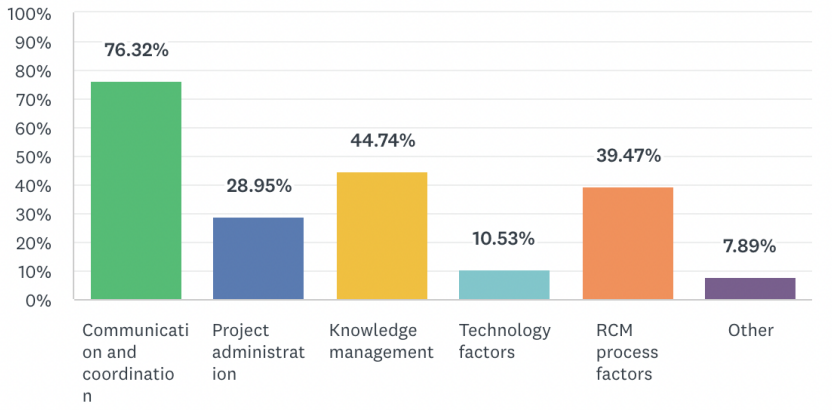

Figure 5.7 Challenges reported in the agile RCM process in GSD.

Chapter Six

Improved ARCM-GSD Model

6.1 Expert Recommended Improvements

The experts were requested to provide recommendations for improving the ARCM-GSD model through surveys and interviews. Six RCM practitioners suggested that the ARCM-GSD model can be simplified by reducing the number of steps or merging similar steps to streamline the change management process, making the model more efficient and easier to understand and implement. Two experts recommended consolidating the RCM database and project tracking tool into a unified system to reduce redundancy. One expert suggested that introducing a project manager role can make the ARCM-GSD model more relevant to a broader array of software companies by minimizing the number of individuals involved in the decision-making phases and establishing a more structured chain of command, enabling better communication and coordination among different stakeholders. An expert suggested that the ARCM-GSD model should be evaluated for compatibility with third-party RCM tools. Another expert proposed that the ARCM-GSD model could be enhanced by incorporating a fast-track process for high-priority change requests or those necessitated by production issues.

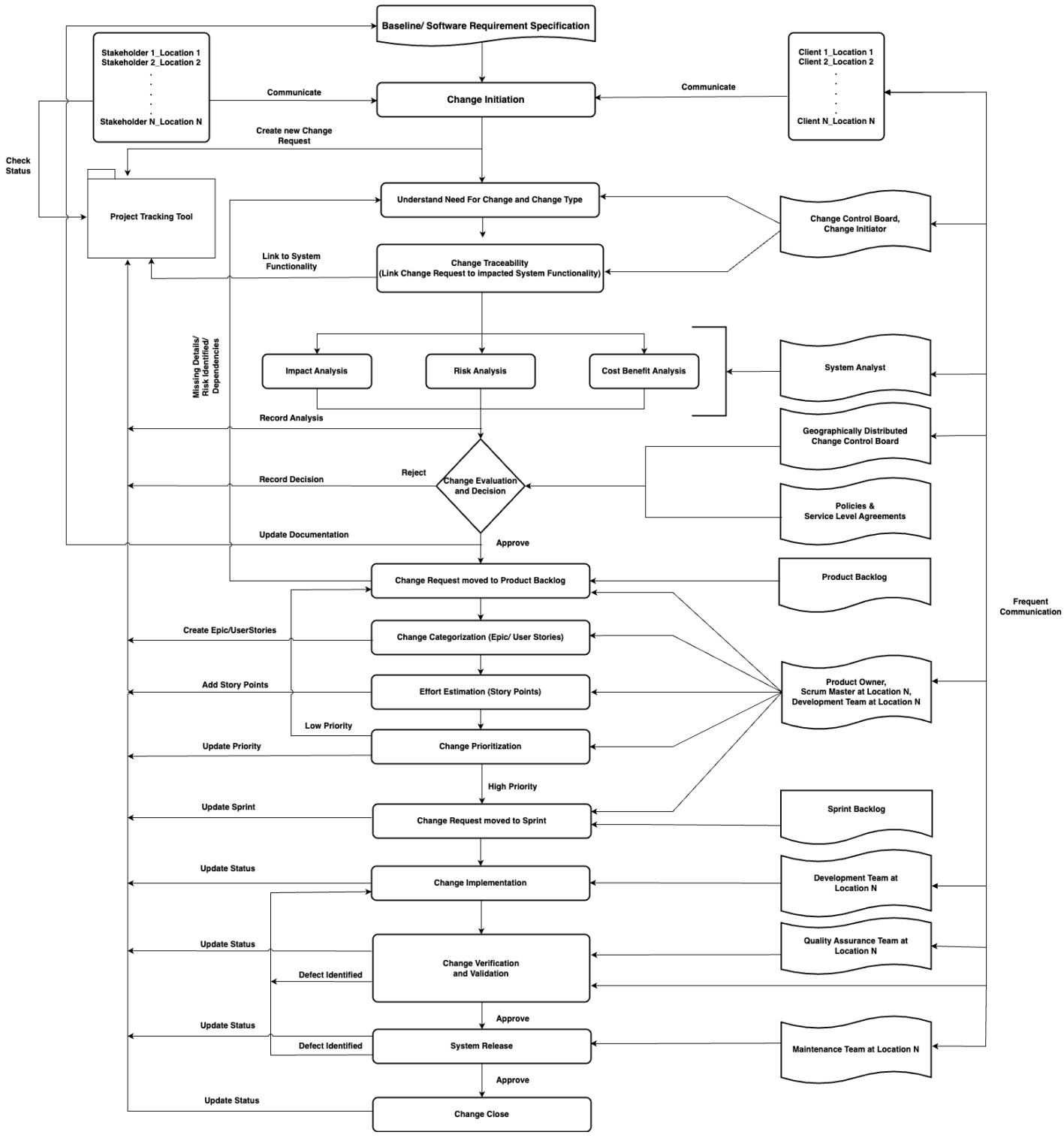

Figure 6.1 Improved ARCM-GSD Model

6.2 Improvements to ARCM-GSD Model

To enhance the effectiveness and practicality of the ARCM-GSD model, feedback from RCM practitioners has been carefully considered and integrated into the model as illustrated in Figure 6.1. The following revisions were implemented in the ARCM-GSD model:

- The change evaluation and change decision phases were merged, as the change control board oversees both steps.
- The RCM database and project tracking tool were combined into a unified system to eliminate redundancy.
- The change verification and validation phases were unified, as both stages entail distinct testing aspects of the system.
- System release and production testing were merged into a single phase, termed system release, which encompasses the release of changes into production and system testing.

6.3 Validation of Improved ARCM-GSD Model

The revised ARCM-GSD model, which integrated the expert recommendations, underwent a subsequent validation round to assess the impact of the revisions. For this purpose, six RCM experts participated in follow-up interviews to evaluate the improvements incorporated into the model. All of the six RCM experts acknowledged that the refinements significantly enhanced the model's comprehensibility and minimized redundancy. The validation process substantiated the successful incorporation of expert feedback and reinforced the model's suitability for managing requirement changes within the context of agile global software development projects.

Chapter Seven

Discussion

7.1 ARCM-GSD Model Comparison with Existing Models

The study utilizes a comparative methodology, as outlined by [23], which focuses on comparing the proposed ARCM-GSD model with established RCM models in the literature. The results from the comparative analysis played a crucial role in determining whether the ARCM-GSD model successfully addressed the limitations of traditional RCM models while also uncovering opportunities for further refinement. This approach provided a comprehensive evaluation of the novel ARCM-GSD model, highlighting its strengths and potential areas of improvement compared to existing models/frameworks. The findings will be used for future iterations and improvement of the model.

	[7]	**[2]**	**[23]**	**[8]**	**[12]**	**[21]**	**[20]**	**[19]**	**ARCM-GSD**
Is the model suitable for Global Software Development (GSD)?	Y	Y	N	N	Y	N	N	N	Y
Is the model suitable for Agile Software Development (ASD)?	N	N	Y	N	N	N	N	N	Y
Does the model mention activities, roles and artifacts (ARA)?	Y	N	N	N	N	N	N	N	Y
Does the model demonstrate integration with project tracking tools?	Y	N	N	N	N	N	N	N	Y
Does the model facilitate client/stakeholder/team communication?	Y	Y	Y	Y	Y	N	N	N	Y

Y=Yes, N=No

Table 7.1 Comparative evaluation of RCM models

Table 7.1 presents a comparison of RCM models in the literature with the proposed ARCM-GSD model. The table evaluates the suitability of each model for Global Software Development (GSD) and Agile Software Development (ASD), whether each model specifies activities, roles, and artifacts (ARA), as well as each model's integration with project tracking tools and ability to facilitate communication. From the comparison, it is evident that the ARCM-GSD model stands out as the only model that is suitable for both GSD and ASD, and it explicitly mentions activities, roles, and artifacts for each phase, enhancing its comprehensibility and practicality in industry settings. In contrast, other models in the comparison either cater to GSD or ASD individually or do not provide detailed information on activities, roles, and artifacts. The comparative analysis revealed that only ARCM-GSD and GRCM models exhibit compatibility with project tracking tools, which is crucial in monitoring the status of changes. Finally, a limited number of models facilitate communication among clients, stakeholders and development teams.

RCM Phases	**[7]**	**[2]**	**[23]**	**[8]**	**[12]**	**[21]**	**[20]**	**[19]**	**ARCM-GSD**
Change Traceability	N	N	N	N	N	N	N	N	Y
Understanding Need for Change	Y	Y	N	N	N	Y	Y	Y	Y
Change Analysis	N	Y	Y	N	N	N	Y	Y	Y
Change Evaluation & Decision	Y	Y	N	Y	Y	Y	Y	N	Y
Change Categorization	N	N	Y	N	N	N	N	N	Y
Change Prioritization	N	N	Y	N	N	N	N	N	Y
Effort Estimation	N	N	N	N	N	N	N	N	Y
Change Implementation	Y	Y	Y	Y	Y	Y	Y	Y	Y
Change Verification and Validation	Y	Y	Y	N	N	Y	Y	Y	Y
Requirements Backlog Management	Y	N	Y	N	N	Y	Y	N	Y
Y=Yes, N=No									

Table 7.2 Comparative analysis of RCM phases

Table 7.2 provides a comparative analysis of RCM activities addressed by nine RCM models, including the proposed ARCM-GSD model. The table evaluates whether a specific RCM model incorporates a particular phase, with "Y" denoting "yes" (the model includes the phase) and "N" representing "no" (the model does not include the phase).

The ARCM-GSD model presents an innovative addition to the RCM process by incorporating a change traceability phase, a crucial element that existing RCM frameworks have largely neglected. The change traceability phase facilitates the systematic tracking of system requirements across the entire project duration. Although change traceability has been highlighted as a key challenge within the RCM process [25, 26], it remains unaddressed by alternative models. Similarly, the ARCM-GSD model introduced effort estimation as a phase, which allows for an assessment of the required resources to implement the requirement change. Contrarily, extant RCM frameworks do not adequately delineate resource evaluation strategies. The ARCM-GSD model integrates change categorization and prioritization phases specifically tailored for agile global software development (AGSD), as initially proposed by [23] in the context of single-site development. By assimilating these phases, the ARCM-GSD model provides a structured approach to organize and prioritize system changes. The ARCM-GSD model emphasizes the importance of maintaining and organizing requirement change backlog and monitoring the status of changes at every step through integration with project tracking tools. ARCM-GSD incorporates a broader range of RCM activities than other alternative models, suggesting that it provides a more comprehensive approach to coping with evolving requirements.

7.2 Limitations of the ARCM-GSD Model

The proposed ARCM-GSD model is subject to several limitations. Firstly, the model has only been validated through expert assessments and lacks empirical validation in real-world scenarios. Secondly, the model has been designed specifically for globally distributed agile

development teams and may not be applicable to other development environments or contexts, thereby limiting its generalizability. Thirdly, the scalability of the model in large-scale software projects remains unknown, as it has not been tested for such scenarios. Next, the model may not be suitable for organizations with highly customized change management processes or those that use different project tracking tools, indicating limited flexibility. These limitations need to be addressed in future work to improve the effectiveness and usability of the proposed model.

7.3 Study Implications

For industry RCM practitioners, the ARCM model offers a comprehensive and structured approach to managing requirements change in a global software development setting. It can improve product quality, reduce costs, and increase customer satisfaction by emphasizing stakeholder/client communication, traceability, categorization, prioritization, and effort estimation.

Chapter Eight

Conclusion and Future Work

Software requirement changes have increasingly become dynamic and frequent, with clients or stakeholders necessitating software modifications at any stage throughout the software development process [2]. Consequently, effective management of requirement changes is crucial to the success of software projects [6], particularly in GSD, as challenges associated with RCM intensify when software teams are geographically dispersed. The prevalence of globalization and the adoption of lightweight, agile approaches to development suggest the need for effective RCM frameworks that enable GSD organizations to efficiently manage volatile software requirements [2]. A literature review of existing RCM models and frameworks suggests that most of these models are designed for single-site development and are unsuitable for GSD. Moreover, to the best of our knowledge, none of these models/frameworks explicitly address the needs of the iterative agile development process.

To address the research gap, this study proposed a novel Agile Requirement Change Management (ARCM-GSD) model, a solution designed to address the unique challenges within the agile GSD paradigm by seamlessly integrating agile methodologies with RCM activities. The ARCM-GSD model incorporates the change traceability phase into the RCM process, enabling comprehensive tracking of software requirements. Additionally, the model integrates change categorization and prioritization [23] phases in conjunction with other standard RCM phases. The ARCM-GSD model specifies activities, roles, and artifacts for

each phase of the model to enhance its applicability in industrial settings [7]. ARCM-GSD model has been validated through an empirical study involving surveys and interviews with industry experts in the field of agile development, RCM, and GSD and also through comparative analysis by comparing ARCM-GSD with existing RCM models. The analysis results indicate that the ARCM-GSD model was well received by industry experts, with a significantly higher number of positive responses compared to negative responses. Furthermore, expert evaluations of the ARCM-GSD design revealed that the majority of experts (94.6%) rated it as either excellent or good, with no experts classifying the design as poor.

8.1 Future Work

Future work for the project involves evaluating the efficacy and scalability of the ARCM-GSD model in an industrial setting through case studies with GSD organizations. The findings from the case studies will offer valuable insights into the model's efficacy, limitations, areas for improvement and applicability towards diverse business domains within GSD. Additionally, potential future work on the ARCM-GSD model can be derived from the feedback provided by the experts. For instance, integrating a project manager role into the model, as also suggested by other RCM studies [2] [27], could streamline the decision-making process and facilitate communication and coordination among stakeholders. Moreover, future research could explore enhancing the model by incorporating a fast-track process for high-priority change requests or those necessitated by production issues, as suggested by an industry expert. Additionally, further research can focus on finding ways to integrate ARCM-GSD with agile development and third-party RCM tools such as Jira, Trello and Confluence to increase its accessibility and usability in the industry.

REFERENCES

[1] Muhammad Azeem Akbar et al. "SRCMIMM: the software requirements change management and implementation maturity model in the domain of global software development industry". In: *Information Technology and Management* (2022), pp. 1–25.

[2] Muhammad Azeem Akbar et al. "AZ-Model of software requirements change management in global software development". In: *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*. IEEE. 2018, pp. 1–6.

[3] Saim Qureshi et al. "A Conceptual Model to Address the Communication and Coordination Challenges During Requirements Change Management in Global Software Development". In: *IEEE Access* 9 (2021), pp. 102290–102303.

[4] Tahir Kamal et al. "Identification and prioritization of agile requirements change management success factors in the domain of global software development". In: *IEEE Access* 8 (2020), pp. 44714–44726.

[5] Arif Ali Khan and Muhammad Azeem Akbar. "Systematic literature review and empirical investigation of motivators for requirements change management process in global software development". In: *Journal of Software: Evolution and Process* 32.4 (2020), e2242.

[6] Saim Qureshi, Saif Ur Rehman Khan, Javed Iqbal, et al. "A study on mitigating the communication and coordination challenges during requirements change management in global software development". In: *IEEE Access* 9 (2021), pp. 88217–88242.

[7] Waqar Hussain and Tony Clear. "GRCM: a model for global requirements change management". In: (2012).

[8] Muhammad Wasim Bhatti et al. "A methodology to manage the changing requirements of a software project". In: (2010), pp. 319–322.

[9] Muhammad Azeem Akbar et al. "Success factors influencing requirements change management process in global software development". In: *Journal of Computer Languages* 51 (2019), pp. 112–130.

[10] Rafael Camara et al. “Agile global software development: A systematic literature review”. In: *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 2020, pp. 31–40.

[11] Waqar Hussain. “Requirements change management in global software development: A multiple case study”. PhD thesis. Auckland University of Technology, 2016.

[12] Nasir Mehmood Minhas, Atika Zulfiqar, et al. “An improved framework for requirement change management in global software development”. In: *Journal of Software Engineering and Applications* 2014 (2014).

[13] Nur Nurmuliani, Didar Zowghi, and Steven Powell. “Analysis of requirements volatility during software development life cycle”. In: *2004 Australian Software Engineering Conference. Proceedings.* IEEE. 2004, pp. 28–37.

[14] Minna Mäkäräinen. “Software change management processes in the development of embedded software”. In: (2000).

[15] Saffena Ramzan and Naveed Ikram. “Requirement change management process models: activities, artifacts and roles”. In: *2006 IEEE International Multitopic Conference*. IEEE. 2006, pp. 219–223.

[16] Shawn A Bohner et al. “Impact analysis in the software change process: a year 2000 perspective.” In: *icsm*. Vol. 96. 1996, pp. 42–51.

[17] W Lam et al. “Change analysis and management: a process model and its application within a commercial setting”. In: *Proceedings. 1998 IEEE Workshop on Application-Specific Software Engineering and Technology. ASSET-98 (Cat. No. 98EX183)*. IEEE. 1998, pp. 34–39.

[18] Samuel A Ajila. “Change management: modeling software product lines evolution”. In: *Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida*. 2002, pp. 492–497.

[19] Ismail Keshta, Mahmood Niazi, and Mohammad Alshayeb. “Towards implementation of requirements management specific practices (SP1. 3 and SP1. 4) for Saudi Arabian small and medium sized software development organizations”. In: *IEEE Access* 5 (2017), pp. 24162–24183.

[20] Mahmood Niazi et al. “A model for requirements change management: Implementation of CMMI level 2 specific practice”. In: *Product-Focused Software Process Improvement: 9th International Conference, PROFES 2008 Monte Porzio Catone, Italy, June 23-25, 2008 Proceedings 9*. Springer. 2008, pp. 143–157.

[21] Arif Ali Khan, Shuib Basri, PDD Dominic, et al. “A process model for requirements change management in collocated software development”. In: *2012 IEEE Symposium on E-Learning, E-Management and E-Services*. IEEE. 2012, pp. 1–6.

[22] Muhammad Azeem Akbar et al. "Readiness model for requirements change management in global software development". In: *Journal of Software: Evolution and Process* 32.10 (2020), e2264.

[23] Zainab Shehzadi et al. "A novel framework for change requirement management (CRM) in agile software development (ASD)". In: *Proceedings of the 9th International Conference on information communication and management*. 2019, pp. 22–26.

[24] Leo A Goodman. "Snowball sampling". In: *The annals of mathematical statistics* (1961), pp. 148–170.

[25] Sajid Anwer et al. "Comparative analysis of requirement change management challenges between in-house and global software development: Findings of literature and industry survey". In: *IEEE Access* 7 (2019), pp. 116585–116611.

[26] Muhammad Azeem Akbar et al. "Investigation of the requirements change management challenges in the domain of global software development". In: *Journal of Software: Evolution and Process* 31.10 (2019), e2207.

[27] Muhammad Shafiq et al. "Effect of project management in requirements engineering and requirements change management processes for global software development". In: *IEEE Access* 6 (2018), pp. 25747–25763.

Appendix One

Literature Review

A.1 Selected Primary Studies

Study ID	Reference	QE1	QE2	QE3	QE4	Total Score	Percentage
S1	Akbar, Muhammad Azeem, et al. "AZ-Model of software requirements change management in global software development." 2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube). IEEE, 2018.	1	1	0.5	1	3.5	0.88
S2	Hussain, Waqar, and Tony Clear. "GRCM: a model for global requirements change management." (2012).	1	1	0.5	1	3.5	0.88
S3	Qureshi, Saim, et al. "A Conceptual Model to Address the Communication and Coordination Challenges During Requirements Change Management in Global Software Development." IEEE Access 9 (2021): 102290-102303.	1	1	0.5	1	3.5	0.88
S4	Kamal, Tahir, Qinghua Zhang, and Muhammad Azeem Akbar. "Toward successful agile requirements change management process in global software development: a client–vendor analysis." IET Software 14.3 (2020): 265-274.	1	1	1	0.5	3.5	0.88
S5	Khan, Arif Ali, Shuib Basri, and P. D. D. Dominic. "A propose framework for requirement change management in global software development." 2012 International Conference on Computer & Information Science (ICCIS). Vol. 2. IEEE, 2012.	1	1	0	1	3	0.75
S6	Adjepon-Yamoah, D. E. (2016). Towards Dependable Change Management and Traceability for Global Software Development. arXiv preprint arXiv:1608.05981.	1	1	0	1	3	0.75
S7	Minhas, Nasir Mehmood, and Atika Zulfiqar. "An improved framework for requirement change management in global software development." Journal of Software Engineering and Applications 2014 (2014).	1	1	0	1	3	0.75
S8	Shafiq, Muhammad, et al. "Effect of project management in requirements engineering and requirements change management processes for global software development." IEEE Access 6 (2018): 25747-25763.	1	1	0	1	3	0.75
S9	Shehzadi, Zainab, et al. "A novel framework for change requirement management (CRM) in agile software development (ASD)." Proceedings of the 9th International Conference on information communication and management. 2019.	1	0	1	1	3	0.75

S10	AlQarni, Turki A., and Rizwan Jameel Noor Muhammad. "A unified model to manage requirement engineering for global software development." Kuwait Journal of Science 46.1 (2019).	1	1	0	1	3	0.75
S11	Hafeez, Yasir, et al. "A requirement change management framework for distributed software environment." 2012 7th International Conference on Computing and Convergence Technology (ICCCT). IEEE, 2012.	1	1	0	1	3	0.75
S12	Mateen, Ahmed, and Hina Amir. "Enhancement in the effectiveness of requirement change management model for global software development." arXiv preprint arXiv:1605.00770 (2016).	1	1	0	1	3	0.75
S13	Alsanad, Abeer Abdulaziz, Azeddine Chikh, and Abdulrahman Mirza. "Multilevel ontology framework for improving requirements change management in global software development." IEEE Access 7 (2019): 71804-71812.	1	1	0	1	3	0.75
S14	Kamal, Tahir, et al. "Identification and prioritization of agile requirements change management success factors in the domain of global software development." IEEE Access 8 (2020): 44714-44726.	1	1	1	0	3	0.75
S15	Akbar, Muhammad Azeem, et al. "Readiness model for requirements change management in global software development." Journal of Software: Evolution and Process 32.10 (2020): e2264.	1	1	0	1	3	0.75
S16	Abdullah, A. A., & Khan, H. U. (2015). FreGsd: A Framework for Global Software Requirement Engineering. JSW, 10(10), 1189-1198.	1	1	0	1	3	0.75
S17	Shameem, M., Kumar, C., Chandra, B., & Khan, A. A. (2017, December). Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client- Vendor Perspective. In Software Engineering Conference Workshops (APSECW), 2017 24th Asia-Pacific (pp. 17-24). IEEE.	1	1	1	0	3	0.75
S18	Akbar, Muhammad Azeem, et al. "SRCMIMM: the software requirements change management and implementation maturity model in the domain of global software development industry." Information Technology and Management (2022): 1-25.	1	1	0	1	3	0.75

S19	Ramzan, Saffena, and Naveed Ikram. "Requirement change management process models: activities, artifacts and roles." *2006 IEEE International Multitopic Conference*. IEEE, 2006.	1	1	0.5	0	2.5	0.63
S20	Akbar, Muhammad Azeem, et al. "Towards the guidelines for requirements change management in global software development: client-vendor perspective." IEEE Access 7 (2019): 76985-77007.	1	1	0	0.5	2.5	0.63
S21	Khan, Arif Ali, and Muhammad Azeem Akbar. "Systematic literature review and empirical investigation of motivators for requirements change management process in global software development." Journal of Software: Evolution and Process 32.4 (2020): e2242.	1	1	0	0.5	2.5	0.63
S22	Keshta, Ismail, Mahmood Niazi, and Mohammad Alshayeb. "Towards implementation of requirements management specific practices (SP1. 3 and SP1. 4) for Saudi Arabian small and medium sized software development organizations." IEEE Access 5 (2017): 24162-24183.	1	0.5	0	1	2.5	0.63
S23	Niazi, Mahmood, et al. "A model for requirements change management: Implementation of CMMI level 2 specific practice." Product-Focused Software Process Improvement: 9th International Conference, PROFES 2008 Monte Porzio Catone, Italy, June 23-25, 2008 Proceedings 9. Springer Berlin Heidelberg, 2008.	1	0.5	0	1	2.5	0.63
S24	Bhatti, Muhammad Wasim, et al. "A methodology to manage the changing requirements of a software project." 2010 International conference on computer information systems and industrial management applications (CISIM). IEEE, 2010.	1	0.5	0	1	2.5	0.63
S25	Akbar, Muhammad Azeem, et al. "Investigation of the requirements change management challenges in the domain of global software development." Journal of Software: Evolution and Process 31.10 (2019): e2207.	1	1	0	0.5	2.5	0.63
S26	Akbar, Muhammad Azeem, et al. "Success factors influencing requirements change management process in global software development." Journal of Computer Languages 51 (2019): 112-130.	1	1	0	0.5	2.5	0.63
S27	Khan, Arif Ali, Shuib Basri, and P. D. D. Dominic. "A process model for requirements change management in collocated software development." 2012 IEEE Symposium on E-Learning, E-Management and E-Services.	1	0	0	1	2	0.5

	IEEE, 2012.						
S28	Anwer, Sajid, et al. "Comparative analysis of requirement change management challenges between in-house and global software development: Findings of literature and industry survey." IEEE Access 7 (2019): 116585-116611.	1	1	0	0	2	0.5
S29	Albuquerque, Danyllo, et al. "Defining agile requirements change management: a mapping study." Proceedings of the 35th Annual ACM Symposium on Applied Computing. 2020.	1	0	1	0	2	0.5

Appendix Two

Survey and Interviews

B.1 Survey and Interview Questions

Questionnaire Survey

Section-A1 Respondents' personal detail				
Full Name (optional)				
Gender	Male	Female	Other	
Educational Background	Software Engineering	Computer Science	Information Technology	Other
Job title				
Work Location				
Have you had experience working on a software development project with team members located in multiple geographical locations? (GSD - Global Software Development)			Yes	No
Please mention work experience (Years) in Global Software Development (GSD).				
Do you have experience working in Agile Software Development (ASD)?			Yes	No
Please mention work experience (Years) in Agile Software Development				
Have you participated in Software Requirement Engineering/ Requirement Change Management (RCM) activities?			Yes	No
Please mention work experience (Years) related to RCM activities				
Total number of years in the Software Industry				
Name of Organization (optional)				
Business Domain of Organization				
Number of Employees your Organization	Less than 10	10 - 49	50 - 249	More than 250

Section B - ARCM Model Assessment

S. D. = Strongly Disagree, D = Disagree, N = Neutral, A = Agree, S. A. = Strongly Agree

Sr.no	Model Assessment Criteria	S.D.	D	N	A	S.A
1.	RCM model is easy to understand and implement.					
2.	RCM model is suitable for Agile software development					
3.	RCM model is suitable for Global Software Development					
4.	The model accounts for all the major Requirement Change Management activities.					
5.	Do you agree that GSD organizations will benefit from having requirements change control board representatives from all geographically distributed sites.					
6.	Evaluate the design of the Agile RCM model.	Poor	Average	Good	Excellent	
What are the limitations of the above Agile RCM model? Please identify any missing phases.						
Please suggest improvements to the above Agile RCM model.						
What challenges does your organization experience while managing requirement changes with distributed teams? Select all that apply.		Communication and coordination	Project administration	Knowledge management	Technology factors	RCM process factors

Which collaborative tools/technology does your organization use for distributed team collaboration?	

Sample Interview Questions:

Q No	Question
1	How frequent are software requirement changes at your organization?
2	Can you briefly explain the RCM process followed at your organization?
3	What are the limitations of the proposed Agile RCM model? Please identify any missing phases.
4	Please suggest improvements to the proposed Agile RCM model.
5	What challenges does your organization experience while managing requirement changes with distributed teams?
6	Which collaborative tools/technology does your organization use for distributed team collaboration?

B.2 Participant Consent Form

University of Victoria

Participant Consent Form

Agile Requirement Change Management for Global Software Development

You are invited to participate in a study entitled Agile Requirement Change Management for Global Software Development that is being conducted by the SEGAL research group, University of Victoria.

Daniela Damian is a professor and faculty member in the department of Computer Science at the University of Victoria, and you may contact her if you have further questions by danielad@uvic.ca.

I am Neha Koulecar, a graduate student in the department of Computer Science at the University of Victoria. I am required to conduct research as part of the requirements for a degree in the Masters in Computer Science. It is being conducted under the supervision of Dr. Daniela Damian. You may contact my supervisor at danielad@uvic.ca.

Purpose and Objectives

This research study investigates the requirement change management (RCM) process for global software development(GSD) organizations following agile methodologies. The study proposes a theoretical framework which integrates the RCM process with agile activities for the GSD paradigm. The study also aims at understanding the challenges agile teams face in managing requirement changes when software teams are distributed globally.

Importance of this Research

The research on Agile Requirement Change Management (ARCM) in the context of Global Software Development (GSD) is essential as it addresses the specific challenges distributed agile teams face in managing requirement changes. It fills a research gap in the existing literature, primarily focusing on traditional development models and lacking comprehensive approaches for agile development practices. By proposing the ARCM-GSD model, this research offers insights and guidelines that can enhance the effectiveness and success of software projects in global and agile environments, ultimately improving project outcomes and client satisfaction.

Participants Selection

Participants were selected for the ARCM-GSD study based on their expertise and experience in Agile Requirement Change Management (ARCM) within Global Software Development (GSD). The selection criteria focused on individuals with practical knowledge and involvement in managing requirement changes in distributed agile teams. Participants were recruited from various organizations that engage in GSD projects.

What is involved

If you consent to voluntarily participate in this research, your participation will include participating in surveys and interviews as part of the data collection process. The interviews will be conducted either virtually or in person, depending on your preference and availability. During the interviews, notes will be taken, and with the participant's consent, the interviews may be recorded using video conferencing technology such as Zoom. The interviews would be expected to take a minimum of 30 minutes of your

time. In case further follow-up interviews are required, this consent would be used unless it is withdrawn by the participant.

Inconvenience
Participation in this study may cause some inconvenience to you, including the time and effort participants might dedicate to participating in the survey and interviews.

Risks
There are no known or anticipated risks to you by participating in this research.

Benefits
By participating in the ARCM-GSD study, you contribute to advancing knowledge in Agile Requirement Change Management (ARCM) in Global Software Development (GSD). You also gain a deeper understanding of managing requirement changes in Agile GSD projects and benefit from practical recommendations for improving the effectiveness of the ARCM process.

Voluntary Participation
Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study, your data will be deleted from the researcher's system.

On-going Consent
To make sure that you continue to consent to participate in this research, I will request participant consent for further follow-up interviews. If participants agree, follow-up interviews will not require signing additional consent unless the consent is withdrawn.

Anonymity
In terms of protecting your anonymity, the study results will not disclose identifiable participant information. Data collected through survey and interviews will not be shared with anyone other than the mentioned researchers.

Confidentiality
Your confidentiality and the confidentiality of the data will be protected by securely storing the data collected on password-protected computers.

Dissemination of Results
It is anticipated that the results of this study will be shared with others in the following ways
Data collected will be anonymized and will be used in the project report.
The results from this study may be used in a publication for a conference/journal.

Disposal of Data
Data from this study will be disposed of by deleting from the researcher's computer within the maximum time span of three years.

Contacts
Individuals that may be contacted regarding this study include the researcher, Neha Koulecar, nehakoulecar@uvic.ca, the co-investigators/supervisors, Daniela Damian, danielad@uvic.ca.

In addition, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Human Research Ethics Office at the University of Victoria (250-472-4545 or ethics@uvic.ca).

Your signature below indicates that you understand the above conditions of participation in this study, that you have had the opportunity to have your questions answered by the researchers, and that you consent to participate in this research project.

_____	_____	_____
Name of Participant	*Signature*	*Date*

PLEASE SELECT STATEMENT only if you consent:

I consent to be identified by name / credited in the results of the study: _____ (Participant to provide initials)

I consent to have my responses attributed to me by name in the results: _____ (Participant to provide initials)

Future Use of Data

PLEASE SELECT STATEMENT:

I consent to the use of my data in future research: _____ (Participant to provide initials)

I **do not** consent to the use of my data in future research: _____ (Participant to provide initials)

I consent to be contacted in the event my data is requested for future research: _____ (Participant to provide initials)

A copy of this consent will be left with you, and a copy will be taken by the researcher.

Appendix Three

Bibliographic Information of Research Participants

C.1 Bibliographic Information

Sr. No	Job Title	Experience (Years)	Country	Educational Background	Business domain of Organization	Experience in Global Software Development	Experience in Agile Development	Experience in RE/RCM practices
1	Senior Project Manager	22	Germany	Software Engineering	Ecommerce	12	10	14
2	Manager	16	India	Software Engineering	Banking	11	13	15
3	Project Manager	26	USA	Computer Science	Ecommerce	15	15	26
4	Project Manager	18	Germany	Software Engineering	Automobile	10	10	15
5	IT Project Manager	9	India	Information Technology	Banking	5	5	5
6	Architect	18	India	Software Engineering	Retail and e-commerce	18	12	16
7	Sr. Project Manager	15	India	Information Technology	Automobile	9	12	12
8	Chief Technology Officer	6	Nepal	Computer Science	Software Development	4	5	6
9	Manager	24	India	Information Technology	Business Intelligence	12	9	20
10	Architect	21	USA	Computer Science	Telecommunication	14	14	21
11	Project Manager	15	India	Software Engineering	Banking	10	10	10
12	Manager	17	India	Software Engineering	Retail and e-commerce	13	13	17
13	Manager, Quality Assurance	15	India	Software Engineering	Banking	15	15	15
14	Product Manager	21	India	Information Technology	Finance	16	14	16
15	Manager	16	India	Computer Science	Healthcare	12	12	12
16	Project Manager	13	India	Information Technology	Ecommerce	11	11	10

17	Scrum Master and Tech BA	7+	Australia	Information Technology	Banking and Finance	7+	7+	7+
18	Scrum Master	8	India	Computer Science	Retail	8	8	8
19	Scrum Master	9	India	Software Engineering	Insurance and Banking	5	4	4
20	Senior Business Analyst	8	Australia	Software Engineering	Aviation	8	4	8
21	Business Analyst	7	India	Computer Science	Banking	7	7	7
22	DevOps Engineer	8	Australia	Information Technology	IT Consultancy	8	5	8
23	DevOps Engineer	5	USA	Computer Science	Telecommunication	4	4	4
24	Senior Software Engineer	7	USA	Computer Science	Supply chain	5	5	5
25	Associate Software Developer	8	India	Information Technology	e-commerce	8	8	8
26	Senior Software Engineer	7	Ireland	Software Engineering	FinTech	5	5	5
27	Software Developer	8	Canada	Information Technology	Train Electronics	7	6	8
28	Senior Software Engineer	9	India	Information Technology	Health and Fitness	9	9	9
29	Web developer	10	India	Computer Science	Business Intelligence	10	10	10
30	Java Developer	6	India	Software Engineering	Retail and Supply chain	6	6	6
31	Software Developer	7	India	Information Technology	Sales	7	7	7
32	Software Developer	8	India	Software Engineering	Telecommunication	6	5	6
33	Software Specialist Testing	9	India	Computer Science	Business Analytics	5	6	6
34	Developer	7	India	Software Engineering	FinTech and Ecommerce	7	7	7

35	Software Engineer	5	India	Information Technology	IT and Consulting	5	5	5
36	Senior Technical Associate	10	USA	Computer Science	Saas	6	6	8
37	Software Specialist	11	India	Information Technology	Banking	11	11	11
38	Senior Software Engineer	10	India	Software Engineering	HealthCare	8	8	8

Appendix Four

ARCM-GSD Model

D.1 ARCM-GSD Model: High-Resolution Images

The following links provide high-resolution images for the proposed ARCM-GSD model.

ARCM-GSD Model - https://tinyurl.com/2wf75bdv

Improved ARCM-GSD Model - https://tinyurl.com/3vaaddbc