A FRAMEWORK FOR REQUIREMENTS CHANGE MANAGEMENT IN DISTRIBUTED AGILE DEVELOPMENT

By

SYEDA FARWA BATOOL

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

March, 2022

A Framework For Requirements Change Management In Distributed Agile Development

By

SYEDA FARWA BATOOL

BSSE, National University of Modern Languages, Islamabad, 2017

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In **Software Engineering**

To
FACULTY OF ENGINEERING & COMPUTER SCIENCES

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

NATIONAL UNIUERSITY OF MODERN LANGUAGES FACULTY OF ENGINEERIING & COMPUTER SCIENCE

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

THESIS TITLE: A Framework for Requirements Change Management in Distributed Agile Development

Submitted by: Syeda Farwa Batool **Registration #:** 005 MSSE/Ibd/F18

Master of Science in Software Engineering

Degree name in full

Software Engineering

Name of Discipline

Dr. Huma Hayat

Name of Research Supervisor

Signature of Research Supervisor

Dr. Basit Shahzad

Name of Dean (FE&CS)

Signature of Dean (FE&CS)

Prof. Dr. Muhammad Safeer Awan

Name of Pro-Rector Academics

Signature of Pro-Rector Academics

March 28th, 2022

Date

AUTHOR'S DECLARATION

I Syeda Farwa Barool

Daughter of Syed Israr Hussain

Registration # 005 MSSE/Ibd/F18

Discipline Software Engineering

Candidate of **Master of Science in Software Engineering (MSSE)** at the National University of Modern Languages do hereby declare that the thesis **A Framework for Requirements Change Management in Distributed Agile Development** submitted by me in partial fulfillment of MSSE degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and the degree revoked.

____

Signature of Candidate

Syeda Farwa Batool

Name of Candidate

28th March 2022

Date

ABSTRACT

A Framework for Requirements Change Management in Distributed Agile Development

Requirements engineering (RE) is a salient phase in any software development project. Requirements keep changing in today's software industry due to increased size and complexities. Therefore, an efficient requirements change management (RCM) process is vital for the success of any project. Distributed software development (DSD) has become a norm now and agile methods are being widely used in DSD to counter changing requirements. Agile methods and DSD, being opposite in nature to each other, present new challenges when they are incorporated together in distributed agile development (DAD). Therefore, an efficient RCM process is the need to today's software industry. This research study has been conducted to fill this gap by presenting a framework for RCM in DAD.

Systematic Literature Review (SLR) has been conducted to identify influencing factors that affect RCM process. Grounded theory is then applied for the analysis to categorize the resultant influencing factors. The resulting categories and the influencing factors for each category have been validated through expert review. After the expert review, a survey is conducted to prioritize the results according to their significance during the RCM process. Finally, a framework has been proposed to conduct the RCM process in DAD based on the prioritized categories and their prioritized influencing factors.

The study concludes that RCM is vital for successful DAD projects and the proposed framework provides a systematic and scaled solution to conduct the RCM process in an effective manner. The application of proposed framework at a wider scale in the industry is a potential future work of this research study. Automating the framework is another interesting dimension for the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DSD	-	Distributed Agile Development
GSD	-	Behavior Driven Development
DAD	-	Distributed Agile Development
RCM	-	Requirements Change Management
AGSD	-	Agile Global Software Development
RE	-	Requirements Engineering

LIST OF APPENDICES

ACKNOWLEDGEMENTS

First, I wish to express my gratitude and deepest thanks to Almighty Allah, who made this study possible and successful. This study would not be accomplished unless the honest espousal that was extended from several sources for which I would like to express my sincere thankfulness and gratitude. Yet, there were significant contributors for my attained success and I cannot forget their input, especially my research supervisors, Dr. Huma Hayat and Dr. Noman Malik, who did not leave any stone unturned to guide me during my research journey. Their thought-provoking ideas, scholarly supervision and attention towards research have contributed a lot in successful completion and enrichment of this thesis in time and best manner.

I am extremely thankful to the respected coordinator Dr. Raheel for his encouragement and contribution. He has always been a constant source of motivation and his invaluable feedback was the driving factor for thesis completion.

I would also like to pay special tribute to my Parents, it’s because of them whatever and wherever I am, it’s just because of them and their selfless sacrificial life and their great efforts with hard work, pain and unceasing prayers have enabled me to reach at this point in life.

Finally, the completion of this thesis could not have been accomplished without the endless support of Zohaib who has always been an inspiration for me. I would like to express my deepest and sincere gratitude for his continuous guidance and motivation and for being there for me throughout this journey.

DEDICATION

This thesis work is dedicated to my parents, family members and friends who have been a constant source of support and motivation during the challenges, who have always loved and believed in me. This work is also dedicated to my teachers throughout my educational career, who have always encouraged me and whose good examples have taught me to work hard for the things that I aspire to achieve.

CHAPTER 1

INTRODUCTION

1.1 Overview

Software development has been shifted from traditional in-house development to geographically distributed environment over the past couple of decades [1-4]. The reason of this transition is the bundle of benefits offered by global software development (GSD) including access to larger pool of highly skilled developers, increased working hours, cost savings, and reduced time to market [2-4]. However, along with these benefits, the software industry has faced new challenges which did not exist in traditional development; lack of communication, coordination and control being the fundamental reasons for those challenges [5-7]. Though the challenges faced by distributed development are huge, yet GSD is continuously gaining popularity [3,4,8]. According to statistics, investment on distributed development has increased from 100 billion US$ in 2010 [1] to 442 billion US$ in 2014 [9].

Requirement's engineering is a fundamental phase of any software project and requirements keep changing throughout the development [10-11]. According to research, 20% to 50% requirements change till the software is ready for delivery [12]. In some cases, the percentage is alarmingly high to 90% if requirements engineering phase is not conducted properly [13-14].

Software size is getting huge in today's products and with the increase in size, the complications have also increased [15]. This has added to the problem of managing requirements and the subsequent changes [15]. Thus, managing the changes in the requirements is a difficult task for collocated software development [8]. The difficulty level increases exponentially when the development teams are geographically dispersed in globally distributed software development environment [8]. According to the research, approximately 20% of the distributed software projects get cancelled and 50% fail in the first year of their development [1]. Furthermore, 50% to 80% such projects fail to achieve the targeted objectives [16]. Such failures result in delayed delivery and loss of billions of dollars every year [17-18]. Literature states that inefficient management of changes in requirements is a significant reason of such failures [19-21].

Agile methods are specifically designed to cope up with changes rapidly [22]. They have been developed for use in changing conditions [22]. Use of agile methods in global software development projects is increasing to deal with the rapid changes faced in GSD projects [23]. Software distribution to the global sites has become a norm to lessen the development costs and to choose the required remote teams in less budget from the developing countries [3,4]. In order to manage requirements change process in a better way, the software firms have started using agile methods [23]. Agile methods develop the product in parts with a continuous interaction among stakeholders that helps to make the changes in time [22]. This also helps to meet the continuously changing user and market demands [24]. However, when agile methods are used in GSD projects, the complexities of software distribution increase even further due to the contradicting nature of agile and GSD [25]. While GSD is mainly focused on explicit knowledge and more documentation, agile methods make use of tacit knowledge and less documentation [25]. There are certain factors which affect the software development in distributed agile development (DAD) thus hampering the efficient Requirements Change Management (RCM) in DAD projects [26]. It is important to find out a suitable way to mitigate those influencing factors which affect the software development in DAD [27]. Therefore, there is need of more research to efficiently manage requirements change in DAD setup [27-28].

1.2 Problem Statement

Requirements Engineering is a critical and central part in any software project [16]. A software project's realization is directly dependent on eliciting and implementing all the all the right requirements [16]. Software requirements keep changing throughout the software development process and therefore, an efficient and effective requirement change management process is required to guarantee the effective development of the software product [14]. This is a challenging task in traditional software development and the complexity increases even further in GSD context [9]. GSD is a widely accepted trend for software industry with its underlying advantages but along with the benefits, it puts forth certain new challenges and increased complexities in completing different software development activities [8]. Many traditional methods and practices have proven inefficient in GSD because the GSD teams are dispersed at different locations with different time zones, language differences and cultural differences [8]. The complexities increase even further when agile methods are incorporated in GSD [11]. These issues directly affect the requirements change management process and there is a need of identifying suitable method for the process of RCM in DAD environment [23]. Therefore, this research study aims at investigating the influencing factors (IFs) that affect efficient RCM in DAD context to provide a framework for effective RCM in DAD.

1.3 Aims and Objectives

This research is aimed to scrutinize the influencing factors for requirements change management process in DAD to provide a suitable way of managing requirements change in DAD. Hence, the objectives of this research are,

- To identify influencing factors from the literature which affect RCM in DAD
- To categorize and prioritize the identified influencing factors for RCM in DAD
- To present a framework for systematic and smooth RCM process in DAD by focusing on the most significant aspects based on the priority level of identified categories and their respective influencing factors.

1.4 Research Questions

Three research questions for the Requirement change management in distributed agile development are RQ1: What are the influencing factors which affect Requirements Change Management process in Distributed Agile Development? RQ2: How to categorize and prioritize the identified influencing factors for Requirements Change Management in Distributed Agile Development? RQ3: What is the suitable way to deal with the influencing factors to manage requirements change in Distributed Agile Development? presented with respective instrumentation and analysis in Table 1.1.

Table 1.1: Research Questions

Research Questions	**Instrumentation**	**Analysis**
RQ1: What are the influencing factors which affect Requirements Change Management process in Distributed Agile Development?	SLR	Grounded Theory
RQ2: How to categorize and prioritize the identified influencing factors for Requirements Change Management in Distributed Agile Development?	SLR, Survey	Grounded Theory
RQ3: What is the suitable way to deal with the influencing factors to manage requirements change in Distributed Agile Development?	Derived	Brainstorming

The table 1.1 lists down the three research questions devised for this study. Along with the research questions, the instruments which are used to answer each question as well as the analysis method incorporated for each question has also been shown in the Table 1.1.

1.5 Scope of the research

The identification of influencing factors during the process of Requirements Change Management in the context of Distributed Agile Development is the scope of this study.

1.6 Contributions of the research

Contributions of this research are as follows:

- A categorized and prioritized list of influencing factors for RCM process in DAD.
- A framework for Requirement Change Management process in Distributed Agile Development.

1.7 Significance of research

This study is significant to the body of knowledge and software engineering practitioners as it provides.

- A prioritized list of categories and their subsequent influencing factors to help practitioners manage the requirements changes in a better way.

- The proposed framework to scale and systemize the RCM process in DAD projects.

1.8 Layout of Thesis

Outline of thesis is shown in Figure 1.1.

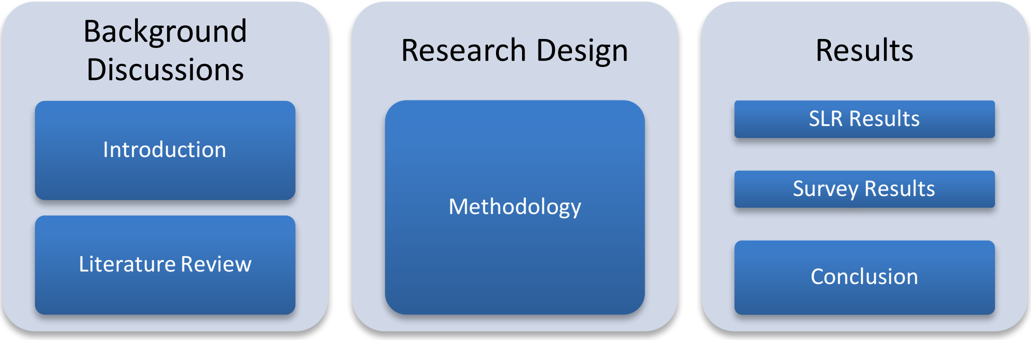


Figure 1.1: Thesis Outline

The thesis structure is divided into three main steps as shown in the Figure 1.1. In the first step Backgrounds and Discussions are presented, whereI ntroduction is explained in chapter 1 and chapter 2 Literature Review is explained. In the next section named as Research Design section, Chapter 3 Research methodology has been presented. In the Final section of Results, Chapter 4 SLR Results, Chapter 5 Survey Results and Chapter 6 Conclusion have been included.

1.9 Summary

The thesis structure is divided into three main steps as shown in the Figure 1.1. In the first step Backgrounds and Discussions, chapter 1 Introduction and chapter 2 Literature Review is explained. In the next section named as Research Design section, Chapter 3 Research methodology has been presented. In the Final section of Results, Chapter 4 SLR Results, Chapter 5 Survey Results and Chapter 6 Conclusion have been included.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter discusses the background study and literature review of the research areas of this thesis which are global software development, agile methods, requirements change management and distributed agile development.

2.2 Distributed Software Development (DSD)

The term distributed software development (DSD) also referred as distributed software engineering (DSE), globally distributed software development (GDSD), geographically dispersed software development (GDSD) or simply global software development (GSD) is based on team members geographically distributed at different locations working on the same project. The following subsections describe the domain of distributed software development in detail.

2.2.1 History

The term DSD has been devised later but IBM and other big giants of industry are distributing their software since 1960s [29]. Contract programming has been in existence since 1970's [30]. Evolution of personal computers has increased the distribution of software since 1990s [31]. This transformation in the software industry has caused the distribution of software globally [32].

Crowdsourcing approach has been successfully applied to a wide range of applications such as Stack Overflow, Linux, YouTube, Wikipedia, Recaptcha, GoogleEarth and Yahoo Answers! [25], [31]. Another example is the creation of encyclopedia, which was developed by a pool of 70,000 participants which supports 290 languages with appropriately 35 million articles [5]. Moreover, famous software companies Facebook, Google, Microsoft, Apple and Netflix provides streamlined bug bounties regarding suspicious activities [32].

2.2.2 What is DSD

The concept of DSD is evolved from contract programming and outsourcing in the 1970's [30]. As defined by Sangwan et al [33], DSD is "software development that uses teams from multiple geographic locations" [33]. Outsourcing or offshore are two ways for collaboration between these development teams. Outsourcing or inter-organizations is the practise of contracting a process such as product development to a third-party or an external organisation while offshoring is moving the development processes by a corporation to another one [33].

DSD is not the same as "normal" or "collocated" development [29]. DSD features qualities that are distinct from collocated development, aside from the many collaboration mechanisms. DSD has a set of characteristics known as distribution factors, which are the environmental aspects, exclusive to DSD. These include [34-35]:

i. Multisource — participation in a collaborative project by several collaborative members involved in a joint project characterised by many collaboration partners.

ii. Location Differences — partners are spread out at different locations.

iii. Time differences - defined by the degree of overlap in working hours.

iv. Cultural differences — the degree to which people are socially, ethnically, and culturally compatible.

v. Language differences is defined by the level of language proficiency.

vi. Contextual differences – organisational fit level (diverse work practices and maturity in the process).

vii. Legal and Political barriers – Laws and political situations of different countries.

Fundamentally, DSD attributes the ecological distribution of people working on a production of a product involving lots of software [35]. That is why DSD is also referred as Global software development (GSD) or Globally distributed software development (GDSD) or Geographically dispersed software development (GDSD) or Global software engineering (GSE) which are all the alternate names of GSD.

Furthermore, as the topographical distance surges, the possibility of temporal distance rises, ultimately resulting into more noticeable and evident cultural differences [35].

2.2.3 GDSD Trends

Offshoring has gained popularity in the late 1990s because of a shortage in labour in the United States with more than 30% companies developing their software in distribution [36]. The source of offshored work is still dominated by Europe and the United States. Offshoring in Europe, on the other hand, is lower than in the US. Only 5% of European enterprises are expected to be offshoring and seems like this pattern is not likely to alter over the next few years [36].

India is regarded as a big contributor with regards to providing personnel for development due to its huge pool of skilled workforce [37]. There are much lower wage expenses than in first-world countries such as the US [37]. Ma J. et al [38] found that software exports in India were higher than China. Russia, Ireland, Brazil, and Singapore have been stated to be the major players in exporting software as a part of GDSD [37-39]. Those in Eastern Europe, such as Estonia, Latvia, and Poland, as well as countries in South America, such as

Brazil, are increasingly participating as global suppliers of software development jobs [39]. However, compared to India or China [39], their skilful labour pool is not as big and low-priced.

2.2.4 DSD Benefits

The prospective benefits of DSD endeavours are highly enticing, which explains the massive increase of such projects throughout the twentieth century [40]. Literature has covered a wide range of reported benefits, from the most insignificant to the most major [37-40]. Lesser costs, timely deliveries and access to bigger pool of skilled workers as well as increased development hours are some major benefits of DSD [40].

2.2.5 DSD Complexities

Aside from the advantages, there are some disadvantages of using DSD. When compared to traditional development, issues in DSD are heightened to a bigger and broader scale [34-35]. These are mostly initiated by the difficulties faced by DSD setup. The global factors addressed in Section 2.1.2 are related to these complexities. As noted in [41], these factors present problems to GDSD, particularly in communication, coordination, and control methods.

Communication is an important part of any SDLC, but it is more significant in DSD because of the teams located at different locations [41-44]. It's the glue that holds “coordination” and “control” together [43].

Coordination is a process of incorporating charge with each structural division such that each division adds to the whole goal [43]. This process unites the entire organisation [45]. DSD, like any software development, needs coordination, but it necessitates it much more because activities are spread across time, geography, and cultural boundaries.

Mintzberg [46] debates three chief mechanisms of coordination: (1) mutual adjustment i.e., regular information exchange amongst members of a team (2) direct supervision, which is based on instructing and observing project activity by one person and (3) standardisation i.e., the implementation of certain protocols. It's challenging to select the right coordinating mechanisms in DSD. Carmel claims that standardization is the best strategy for DSD coordination [29].

The practice of conforming to methods, rules, regulations, and standards is referred to as control. [43]. The control process refers to the organization and presenting processes ready to ensure that a development project is on track to meet its objectives [43]. Control techniques might take formal (overt policies and application of standards) or informal (co-worker agreements) methods [41].

Keeping project control in DSD is complex [41]. When some portions of a software company's work are given to external bodies as a part of outsourcing, the company can face difficulties in regulating lose control of its corporate procedures [43]. Cultural differences can aggravate this problem [36]. Moreover, cultural distance might lead to differing perceptions, affecting control procedures [41].

2.3 Agile Methods

Agile is a set of development practises aimed at increasing the efficacy of organizations [47]. It entails self-organizing and cross-functional teams working together with their clients to uncover necessities and build solutions [48-54]. It promotes flexible preparation, incremental growth, initial supply, and consistent enhancement, as well as it inspires flexible arrangements to alterations in needs, resource accessibility, and understanding of the difficulties to be resolved [50-55].

The 2001 Manifesto for Agile Software Development proved vital in the popularity of agile. [51] The manifesto's ideas and concepts were originated from wide different development paradigms, including Scrum and Kanban. [52-53].

2.3.1 History

Iterative and Incremental Development Methodologies have started in 1957 [56]. Evolutionary Project Management [57-58] and Adaptive Development [59] have gained popularity in 1970s [60]. In 2005, the PM Declaration of Interdependence has been written as an update to PM principles to guide SPM using agile approaches [61]. The Agile movement has published the Guide to Agile Practices, calling it "the Agile Glossary" in 2016 [62].

Scott Ambler has summed up the guide as [63]:

- Tools and procedures are necessary but having experienced individuals working together proficiently is more imperative.
- Effective documentation is valuable, but it is secondary to the actual objective which is development.
- A contract is necessary, but it is no alternative for working directly with clients to understand their requirements.
- A project strategy is necessary, but it should be flexible enough to welcome changes at any stage of SDLC.

The Agile movement, founded by some of the manifesto's writers, is an organisation that encourages development based on the manifesto's objectives [64].

2.3.2 Agile software development principles

"The Manifesto for Agile Software Development is based on twelve principles [65]

- Customer satisfaction by early and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Deliver working software frequently (weeks rather than months)
- Close, daily cooperation between businesspeople and developers
- Projects are built around motivated individuals, who should be trusted

- Face-to-face conversation is the best form of communication (co-location)
- Working software is the primary measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity—the art of maximizing the amount of work not done—is essential
- Best architectures, requirements, and designs emerge from self-organizing teams
- Regularly, the team reflects on how to become more effective, and adjusts accordingly"

2.3.3 Iterative, incremental, and evolutionary

All SDLC phases are performed by a cross-functional team during each iteration and results are shared with relevant stakeholders upon completing each iteration [66]. This reduces total risk and permits the product to respond swiftly to alterations [66]. The aim is to produce a ready-to-use release with few issues at the end of each one [67]. It is possible that releasing a product or adding new features will necessitate multiple revisions [68].

2.3.4 Face-to-face communication

Co-location is a concept that states that colleagues in the same group should be located together to strengthen the team's identity and enhance communication. [70] This permits for in-person engagement, ideally utilising a whiteboard, which reduces time consumption while responding to queries through other mediums such as phone, email, online chatting etc. [71]. All groups ought to incorporate a client agent, regardless of whatever development technique is used. Stakeholders approve on this individual to perform on their behalf, and he or she makes a pledge to be accessible to developers for queries during the iteration [72]. Stakeholders and the clients' agent analyse progress and re-examine objectives after each iteration to maximise return on investment (ROI) [73]. The technique is commonly referred to as a "Customer Centred Methodology" because of the significance of stakeholder satisfaction, as seen by recurrent engagement and review at each phase's conclusion. [74].

2.3.5 Very short feedback loop and adaptation cycle

In a quick meeting, members of a team tell each other the activities of the preceding day to share their iteration objective, their intention and planning to today's purposes and any obstructions they perceive in the way of the objective [75].

2.3.6 Quality focus

To enhance quality and accelerate product development, certain tools and techniques are frequently employed [76]. This is premised on including quality from the start and the ability to develop and establish software to clients at any time, or at each iteration's conclusion [77].

2.3.7 Philosophy

Agile software development, in contrast to traditional software engineering, focuses on complicated systems and development with versatile and dynamic features. Early on, precise approximations, firm strategies, and projections can be difficult [78]. These fundamental ideas, as well as past manufacturing experiences gained through years of triumphs and malfunctions, have influenced agile development's preference [79].

2.3.8 Adaptive vs. predictive

Adaptive approaches concentrate on rapidly adapting to varying circumstances [80]. When a project's requirements alter, an adaptive team must adjust as well. It's tough for an adaptable team to predict the precise future events [80]. The more its far away, the more ambiguous an adaptive method's prediction is [80]. It is difficult for an adaptable team to predict their upcoming activities; they can only say what features they're planning in the next month

[81]. Predictive approaches, on the other hand, concentrate on thoroughly analyzing and planning the future while considering the known dangers [81].

2.3.9 Method tailoring

'Method tailoring describes the concept of method adaptation. Methods can be adjusted utilising a variety of instruments in practise. To modify software development approaches, generic process modelling languages like Unified Modelling Language can be utilised [82].

2.4 Distributed Agile Development

Distributed agile development (DAD) refers to the application of agile methods in a dispersed environment [83-84]. The idea is to take use of each methodology's distinct advantages [85-86]. Dispersed development permits companies to generate software by deliberately establishing groups in several locations throughout the world, effectively producing software around the clock [87-88]. Agile development instead provides more transparency, ongoing input, and tractability [89]. Agile software development principles give structures for improved communication, which is a critical component of working in a dispersed environment [89].

2.4.1 History

Many organizations have begun to distribute software operations to more economically appealing places [90]. This process has commenced in the 1990s, but its strategic significance has been grasped in the 2000s [90]. Many early relevant investigations have also been conducted in this period [91].

The Agile Manifesto marks a shift away because of which, the question "Can remote software development be agile?" inevitably arises [92]. In 2006, one of the first inclusive reviews aiming at answering this question has been published. [93] They have discovered that

“Agile helps in addressing three C challenges of GSD as well as to build trust among GSD teams if applied correctly” after researching three firms. In 2014, a Systematic Literature Review has been conducted to recognize the major challenges in adapting agile to operate in a remote environment. [94] A similar SLR has been conducted in 2019 [95]. In addition, an overall review of the subject has been completed [96].

Overall, DAD is a fast-paced industry. Research on all aspects continues, revealing that it provides exclusive chances and benefits over more conventional ways, but not without its own set of obstacles and hazards.

2.4.2 Opportunities

Implementing agile principles tends to have a good impact on group communication [97]. Sprint reviews are considered as a great way to strengthen external correspondence while also allowing colleagues or stakeholders to divide data concerning structures and precursor requirements [97]. Agile approaches also promote constant communication and delivery of programming deliverables, which helps to develop trust between the many teams involved in the process [98]. In this vein, embracing agile methods in a dispersed setting has proven to be beneficial to the project's value and implementation [99].

2.4.3 Individuals with incapacities and mobility limitations

As previously said, Distributed Agile Development environment places a higher value on efficiency than on physical attendance [99]. People with disabilities gain from this because they have the liberty to work in a setting that is relaxing for them while still contributing to the deliverable [99]. This situation also applies when a worker is unable to commit to the office hours; this way, they can accomplish duties from home without having a negative impact on the delivery [99].

2.4.4 Increased levels of prosperity

Operating in a dispersed agile environment warrants more profitability and welfare for both employees and organizations [99]. This is because the task is divided among several individuals throughout the world, so there is not much pressure on a single person to do it [99]. As a result, physical as well as mental health are ensured [99]. Furthermore, because several people do their share and the project goes through several revisions, the final work quality is improved, which is helpful to the organization [99]. Consequently, the scenario benefits both the corporation and its staff [99].

2.4.5 Extensive pool of HR

Availability of choosing the team from larger pool of developers across the globe is a great benefit for organizations [99]. This highlights the importance of all HRs working together as a team to urge collaborations throughout the organizations, as well as communicating with stakeholders to prioritize deliverables [99].

2.4.6 Reduced office space

Because the Distributed Agile Development encourages remote working, there is no longer a demand to enlarge office space to accommodate more personnel [99]. Time and geographic limits are no longer barriers. "For certain organisations, particularly service businesses, such constraints are no longer important, and in fact, they've been turned to their advantage." Having a worldwide 'distributed' workforce, for example, naturally permits 24-hour service and operation. Furthermore, other job-related issues are not a huge problem because employees have the freedom to work from wherever they like [99]. In some ways, this is advantageous because it saves a significant amount that would otherwise be spent on extra expenditures and Employees in Distributed workplaces spend less time and energy on mundane office tasks and more time with clients. [99].

2.4.7 Challenges

Because of the compatibility difficulties [100], DAD faces some significant challenges some of which are documentation, different time zones, pair programming, distribution of work and training [101].

2.5 Requirements Management Process

Requirements management begins with preparation, which determines the level of management that is required [102]. Each need is given an exclusive ‘identifier' after development so that it can be matched with the requirements [102]. Following the identification of requests, requirements are traced [103]. Its goal is to guarantee that all needs are thoroughly understood and accounted for in test plans and test cases [104]. The following are some of the benefits of requirements tracing [105-106].

- It checks to see if user necessities have been applied and thoroughly tested.
- It allows users to comprehend the implications of varying necessities.

Traceability matrix is usually used for this purpose. It's worth noting that a traceability matrix is more useful when there are fewer requirements to manage. Traceability matrices, on the other hand, are costly to maintain when an enormous system with many needs is being constructed. This is because huge requirements are difficult to manage [105]. As a result, large systems tracked data is recorded in a "requirements database," where each prerequisite is overtly related to adjacent requirements. This allows you to see how a variation in one criterion impacts the various components of the system you're building [106].

2.5.1 Requirements Change Management

When requirements change request is received, the process of requirements change management starts. This method has the benefit of ensuring that revisions to proposals are handled uniformly and in an organized way. Many requirements management operations are related to software configuration management activities.

For modifications to requirements, efficient requirements change management process goes through several stages. Below is a list of these stages [107].

- Problem analysis and change specification: The whole procedure starts with the recognition of issues in relation to the needs. To determine whether the amendment is justified, the issue is examined. The results of the investigation are sent to the 'change requester,' who is then given a more detailed requirement change suggestion.
- Change analysis and costing: The impact of a requested alteration on the prerequisite is evaluated using traceable data. Following the analysis, a judgement is taken about whether variations should be implemented or not.
- Change implementation: Finally, the requirements document, system design, and implementation are updated. Changes to the requirements document can be made without considerable rewriting because it is organised in this way. Changeability in the document is achieved by reducing external references and making document portions modular. Separable portions of the document can be altered and replaced without affecting the rest of the content.

2.6 RCM in DAD

As eminent from the above discussion, DAD is exponentially increasing in today's software industry and RCM is one of the most significant parts of any software project. Being much more challenging than traditional or collocated agile software development, RCM in DAD becomes even more challenging and that's why this research studies focuses on the ways to efficiently conduct RCM in DAD [100-107].

2.7 Related Work

There have been several systematic literature reviews presented in the literature targeting challenges or practices for Requirements Change Management or on Global Software

Development, but no single review exists in the domain of Requirement Change Management in DAD which was the focus of this research study. A summary of related work studies has been presented in Table 2.1.

Table 2.1: Related Work Studies

Year & Ref #	**Author**	**Paper**	**RCM**	**Agile**	**DAD**
2018 [108]	Jayatilleke et al.	A systematic Review of Requirement Change Management	No	No	Yes
2016 [109]	Hussain et al.	Current Challenges of Requirement Change Management	No	No	Yes
2015 [110]	Yaseen et al.	Critical challenges for requirement implementation in context of global software development: A systematic literature review	Yes	No	No
2019 [111]	Jain et al.	A systematic literature review on global software development life cycle	Yes	No	No
2019 [112]	Schon et al.	Agile Requirements Engineering: A systematic literature review	No	Yes	No
2017 [113]	Inayat et al.	A systematic literature review on Agile requirements engineering practices and challenges	No	Yes	No
2012 [114]	Bano et al.	Causes of requirement change - A systematic literature review	No	No	Yes
2012 [115]	Juan et al.	Preliminary Results of a systematic review on requirements evolution	No	No	Yes

2010 [116]	Alves et al.	Requirements engineering for software product lines: A systematic literature review	No	No	Yes
2009 [117]	Walia et al.	A systematic literature review to identify and classify software requirement errors	No	No	Yes
2019 [118]	Akbar et al.	Success factors influencing requirements change management process in global software development	Yes	No	Yes
2019 [119]	Akbar et al.	Towards the Guidelines for Requirements Change Management in Global Software Development: Client-Vendor Perspective	Yes	No	Yes

As shown in Table 2.1, Jayatilleke et al. [108] have presented a SLR in 2018 on requirements change management. They have discussed several techniques and practices for RCM, but the study does not identify the challenges. Also, the study does not target DAD environment. Hussain et al. [109], in 2016, have conducted a literature review and have presented the challenges of RCM. This study also does not focus on DAD as well as they do not present the practices to address the identified challenges.

Yaseen et al. [110], in 2015, have conducted a SLR to identify critical challenges for requirements implementation in GSD. The study focuses on requirements implementation and not on RCM.

In 2015, Jain et al. [111] have conducted a SLR to identify challenges that occur during global software development life cycle (GSDLC). They have identified challenges in different phases of GSDLC and have presented practices or tools to address those challenges. Their research focus on the overall phases of GSD life cycle and not on the RCM in DAD.

In 2016, Schon et al. [112] have conducted a SLR on agile requirements engineering. They have presented methodologies for requirements engineering in Agile and an overview to manage requirements in Agile software development. They do not discuss RCM or DAD. Inayat et al. [113], in 2015, have also conducted a similar systematic literature review to present challenges and practices for agile requirements engineering. Again, the authors focus on requirements engineering process in agile software development and not on RCM in DAD setup.

Bano et al. [114], in 2012, have presented a systematic literature review in which the authors have identified causes of requirements change. They stress upon the need to further explore the area. Similarly, Juan et al. [115] in the same year have conducted another SLR targeting the area of requirements evolution. In 2010, Alves et al. [116] have conducted a SLR on RE for software product lines to identify problems and future research gaps. Another SLR has been conducted by Walia et al. [117] in 2009 to identify and classify errors during the SRE process. Again, these reviews do not discuss RCM processes or RCM in DAD.

The closest literature review to this research study is conducted by Akbar et al. in 2019 [118]. In their review, the authors have identified the influencing factors during RCM process in GSD setup which lead to the success of software development. The authors also suggest that there exists research on requirements changes but the area of requirements change management is still lacking attention and needs to be studied more as well as they do not discuss RCM in the context of DAD [118]. Their study differs from this one because this study aims to identify the influencing factors for RCM process in DAD context. Another relevant literature review, conducted by Akbar et al. in 2019, presents the guidelines for RCM in GSD [119]. The authors have found out several practices for RCM in GSD and then categorize them in client and vendor categories to help practitioners choose the practices. Again, the review only focuses on the practices and do not consider the challenges. Hence their review also differs from this study.

Although existing studies related to RCM in agile DAD are listed in Table 2.1, However there is less discuss done on RCM in Agile DAD. This leads to conduct a study on RCM in agile DAD. The method to conduct this study is reported in chapter 3.

2.8 Summary

This chapter has discussed the background study and the work related to this research. All the software engineering disciplines involved in this research study have been discussed starting from Global Software development and then discussing Agile methods, Distributed Agile development, and Requirements Change Management before eventually explaining the RCM in DAD which is the focus of this study.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter discusses the research methodology the authors adopted to conduct this research. This research study is conducted in three phases as shown in the Figure 3.1. A systematic literature review has been selected to review the literature for this study. Published research over the past five years has been examined in five different databases to answer the research questions. The results from searching the databases have been scrutinized using grounded theory approach to classify the influencing factors for RCM in DAD and the results have been validated through expert review. A survey is then conducted to prioritize the categories and elicited influencing factors for each category. Finally, a framework has been proposed based on the findings from SLR and Questionnaire survey.

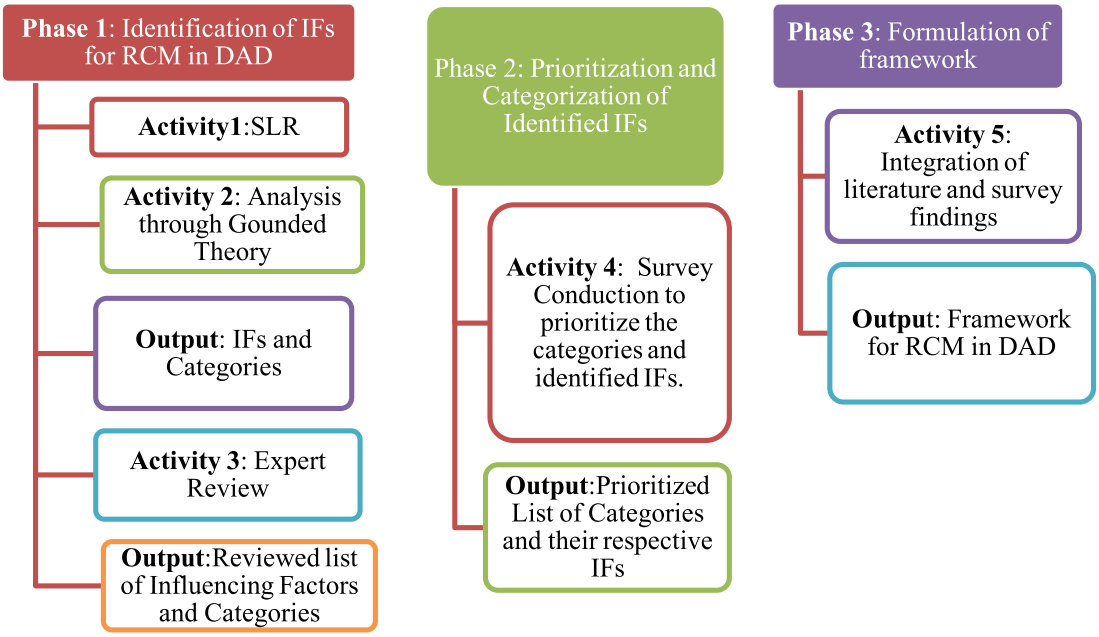

Phase 1: Identification of IFs for RCM in DAD
Activity1:SLR
Activity 2: Analysis through Gounded Theory
Output: IFs and Categories
Activity 3: Expert Review
Output:Reviewed list of Influencing Factors and Categories
Phase 2: Prioritization and Categorization of Identified IFs
Activity 4: Survey Conduction to prioritize the categories and identified IFs.
Output:Prioritized List of Categories and their respective IFs
Phase 3: Formulation of framework
Activity 5: Integration of literature and survey findings
Output: Framework for RCM in DAD

Figure 3.1: Phases of Research Methodology and it's Respective Activities

3.2 Phase 1: Identification of Influencing Factors for RCM in DAD

For identification of RCM challenges in DAD, three activities have been performed. In the first activity, SLR is conducted to identify influencing factors from the literature. Next, grounded theory is applied to analyze and categorize the identified influencing factors. In the third activity of phase 1, expert review is performed to validate the identified influencing factors and their categories.

3.2.1 Systematic Literature Review (SLR)

Systematic Literature Review (SLR) is "a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest" [120]. It is commonly referred to as systematic review. The studies which are selected as the result of SLR are said to be the Primary studies while the process of SLR is considered as secondary study [120].

SLR has been performed based on the guidelines of Kitchenham [120]. The three-step procedure includes [120]:

- Planning the review
    - Develop Review Protocol
- Conducting the review
    - Search primary studies
- Reporting the review
    - Document the results

Figure 3.2 explains the overview of the steps which have been performed in each part of the SLR.

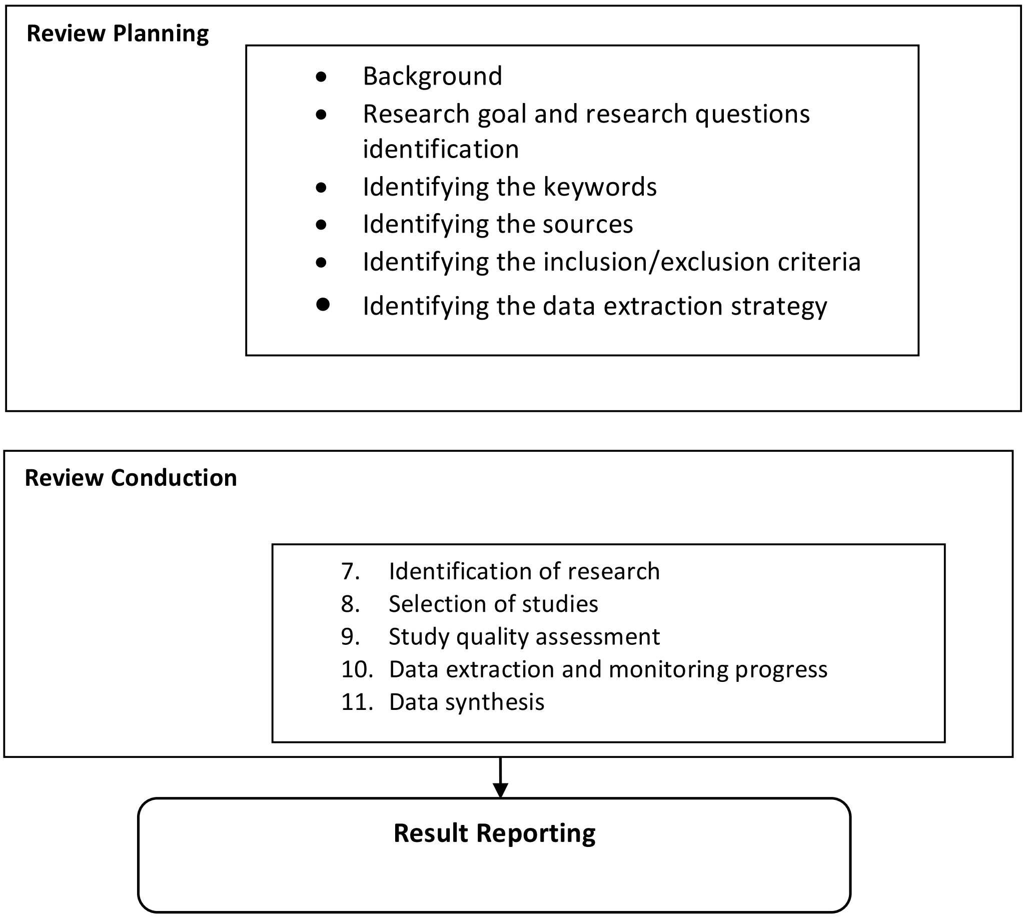
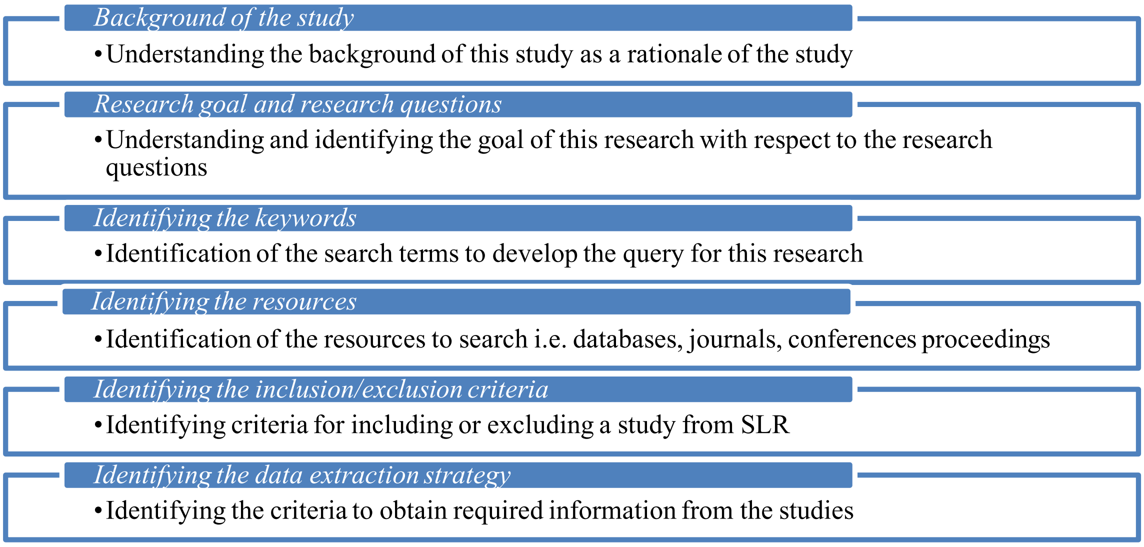

Review Planning
Background
Research goal and research questions identification
Identifying the keywords
Identifying the sources
Identifying the inclusion/exclusion criteria
Identifying the data extraction strategy
Review Conduction
7. Identification of research
8. Selection of studies
9. Study quality assessment
10. Data extraction and monitoring progress
11. Data synthesis
Result Reporting

Figure 3.2: Overview of SLR steps adopted from [120]

Review Planning comprise of the clarification of need of an SLR as well as the development of a reviewing protocol as shown in figure 3.2, whose elements have been illustrated in Figure 3.3.

Background of the study is the first step, which is used for Review Planning. To complete this step, the authors have performed the review of existing literature related to RCM in DAD. The related studies guide the authors to devise direction in Requirement change management for DAD environment.

Research objectives and Research Question is the second step, which is used for Review Planning. This step clearly explains the research goal and research questions for which this SLR is conducted. For example, the goal of the research is to come up with categorized and prioritized influencing factors that have impact during requirement change management in DAD environment.



Background of the study

- Understanding the background of this study as a rationale of the study

Research goal and research questions

- Understanding and identifying the goal of this research with respect to the research questions

Identifying the keywords

- Identification of the search terms to develop the query for this research

Identifying the resources

- Identification of the resources to search i.e. databases, journals, conferences proceedings

Identifying the inclusion/exclusion criteria

- Identifying criteria for including or excluding a study from SLR

Identifying the data extraction strategy

- Identifying the criteria to obtain required information from the studies

Figure 3.3 Review Planning Elements [120]

Thus, the research question for SLR is: "What are the influencing factors presented in the literature for RCM in DAD?"

Identifying the Keywords is the next step, which has been performed under Review Planning. In this step, the authors have identified the relevant keywords of this research.

Identifying major Terms from Research Question: Distributed agile development, Requirements change management, Influencing factors.

Identifying Alternate Spellings and Acronym for Major Terms: Global software development, distributed software development, offshore software development, dispersed software development, global software engineering, agile global development.

The final search string is formulated using all major terms and their synonyms separated by AND, OR operators as shown in Table 3.1.

Table 3.1: Search Terms

Search Terms	
1	Requirements
2	Change
3	Management
4	Agile
5	Global
6	Dispersed
7	Offshore
8	Distributed
9	Software
10	Development
11	Engineering
12	1 AND 2 AND 3
13	5 OR 6 OR 7 OR 8
14	10 0R 11
15	9 AND 14
16	AND (4 AND 13 AND 15)

Final Search term is Requirements AND Change AND Management AND (Agile AND Global OR Dispersed OR Offshore OR Distributed AND Software

Identifying the Resources is the next step in Review Planning activity. In this step, the authors have identified the data sources to retrieve the relevant research papers as shown in the Table 3.2.

Table 3.2: Databases

Search Resources (Databases)	
1	IEEE Explore
2	ACM
3	Springer Link
4	Google Scholar
5	Science Direct

ACM Digital Library, Springer Link, IEEE, Science Direct and Google Scholar have been selected as the data sources. Studies form the past five years have been searched from the above-mentioned resources. The search activity is decomposed in two layers. First layer of search is the automated search which has been performed by using the queries in selected data sources and second layer of search is manual search in which the authors have searched the references of the papers manually to ensure the completeness of search.

Identifying the Inclusion/Exclusion Criteria is the fifth step, which has been used for Review Planning. The inclusion criteria, based on three levels, is shown in the Table 3.3.

Table 3.3: Inclusion Criteria

Study Inclusion Criteria		
Step 1: Initial Criteria	1	Study is peer reviewed
	2	Full text is available
	3	Study is in English
Step 2 Abstract Level	1	Relevant by title
	2	Relevant by abstract
	3	Relevant by keywords
Step 3 Detailed Review	1	The study discusses RCM influencing factors in DAD
	2	Or Proposes/ Applies a RCM method in DAD
	3	Or compares different RCM methods in DAD

Studies in languages other than English, or not reviewed or not having full text available have been excluded in first level. After that the authors have checked the papers on basis of their title, abstract and keywords. Papers which do not have any of the keywords Software Requirement Change Management, influencing factors and Agile Global Software Development have been excluded in second level. All the remaining papers have been studied in detail according to the third step of Table 3.3 at the third stage. The study exclusion criteria are based on the factors presented in Table 3.4.

Table 3.4: Study Exclusion Criteria

Study exclusion criteria	
1	Articles just showing table of contents
2	OR the study that are repeated in our datasets
3	OR the study doesn’t discuss RCM in DAD context in any way.

The studies only showing table of contents have been excluded. Repetitive studies in multiple databases have been exclude and studies not discussing Requirements Change Management in DAD in any way have been excluded according to the criteria of Table 3.4.

The primary studies have been selected after applying the above-mentioned inclusion/ exclusion criteria on the search results of every search resource. After the application of detailed study inclusion/ exclusion criteria, the remaining studies have been examined thoroughly. Finally, cross check has been performed by a secondary reviewer. This provides a final list of primary studies.

The authors have adopted the **quality assess criteria** checklist provided by kitchenham guidelines [121]. It is consisted of seven questions e.g., question 1: Are the aims clearly stated? These questions Applied on primary studies to check the quality of studies by scaling Yes/No/Partially. It has been used to assess the quality of shortlisted primary studies. The checklist has been shown below in the Table 3.5.

Table 3.5: Quality Assess Criteria

No.	Question	Answer
1	Are the aims clearly stated?	Yes/ No/ Partially
2	Are the findings credible and important?	Yes/ No/ Partially
3	Are the used prediction techniques clearly described and their selection is justified?	Yes/ No/ Partially
4	Is the knowledge or understanding extended by research?	Yes/ No/ Partially
5	Has the diversity of perspective and context been explored?	Yes/ No/ Partially
6	Are the links between data, interpretation and conclusion clear?	Yes/ No/ Partially
7	Is the detail/ depth/ complexity of the data conveyed?	Yes/ No/ Partially

A scale specified by Azhar et al. has been used to analyse the results [122], who also adopted Kitchenham's guidelines for quality assessment [121]. The scale is provided in the Table 3.6.

Table 3.6: Score Criteria

Answer	Score
Yes	1
No	0
Partially	0.5

Studies with higher scores have been marked as more appropriate and relevant thus concluding to have greater quality for selection.

Identifying the Data Extraction Strategy is the sixth step used for Review Planning. Table 3.7 shows the design of the form which has been used by the authors for showing the information of the study. The forms comprise of paper ID, title, author name, year of publication, publication type and selection status.

Table 3.7: Study Information Form

Data Item	**Data Item Information**	**Notes**
Id		
Title		
Author		
Year of Publication		
Publication Type	Journal/ conference/un-published	
Selection status	Exclude/Included	Reason of exclusion

The table shows all the information that has been gathered about selected studies. Multiple reviewers have extracted data from each study and then the results have been compared and settled in case of disagreement to obtain finalized data.

Once the Reviewing Protocol is completed and agreed by the reviewer (supervisor of the research) then the second step of SLR "Review Conduction" is initiated. The elements of Review Conduction are shown in Figure 3.4.

Identification of Research is the seventh step, which has been performed for Review Conduction. In this step, the authors have used the devised search terms to identify studies from the pre-selected search resources.

Selection of studies is the eighth step of SLR, which has been performed for Review Conduction. Based upon the designed inclusion/exclusion criteria, the authors have selected the primary studies.

Study Quality Assessment is the ninth step of SLR, which has been performed for Review Conduction. In this step, the authors have assessed the quality of the selected study by using check list provided in Table 3.5. Three experts have been selected to assess the quality of each paper based on criteria shown in Table 3.5. Each paper is given a score according to the score scale of Table 3.6. The results are documented which have been presented in the next chapter.

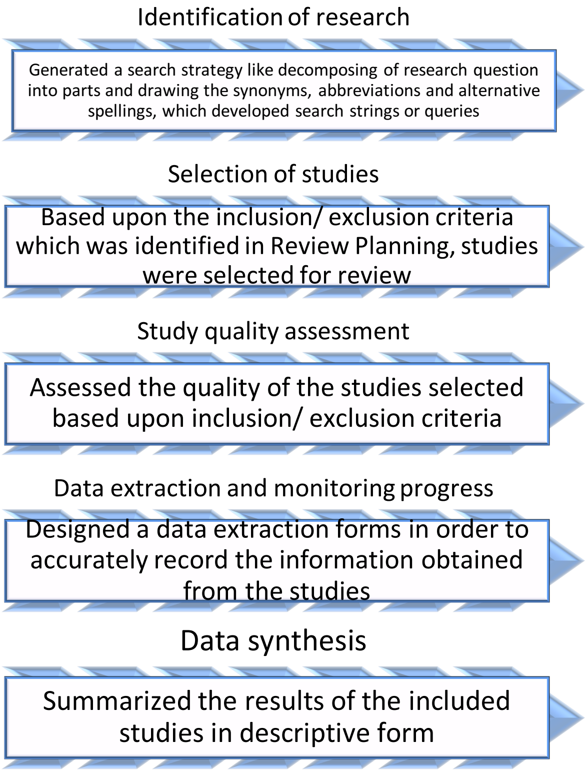

Identification of research
Generated a search strategy like decomposing of research question into parts and drawing the synonyms, abbreviations and alternative spellings, which developed search strings or queries
Selection of studies
Based upon the inclusion/ exclusion criteria which was identified in Review Planning, studies were selected for review
Study quality assessment
Assessed the quality of the studies selected based upon inclusion/ exclusion criteria
Data extraction and monitoring progress
Designed a data extraction forms in order to accurately record the information obtained from the studies
Data synthesis
Summarized the results of the included studies in descriptive form

Figure 3.4: Review Conduction Elements

Data Extraction is the tenth step of SLR, which has been performed for Review Conduction. In this step, the data extraction form as shown in Table 3.7 is used to record the data gathered from the studies. Once the influencing factors have been identified, they are reported in the form of a list. Result Reporting is important to communicate the results of SLR effectively. These results of each table have been presented in Appendix A.

Data Synthesis is the eleventh step of SLR, which has been performed for Review Conduction. This step analyzes the data which has been extracted from the studies to address the research questions. Grounded Theory is used for this purpose.

3.2.2 Grounded Theory

After the completion of data extraction from final studies, the next activity in first phase of this research is to analyse the results of SLR. The authors have chosen grounded theory for the analysis of the results. Grounded theory (GT) is a research approach that focuses on the development of theory that is 'grounded' in data that has been collected and analysed in a methodical manner. It's utilised to figure out things like social interactions and collective behaviours, which are referred to as social processes [123]. Grounded theory is used on the qualitative research data for analysis purposes [123]. It is a systematic approach which involve different coding techniques like open coding, axial coding, or selective coding to code and categorize the extracted data [123]. In using this technique, the researchers traverse back and forth where they merge the similar data and keep adding data in the existing categories which are called themes [123]. The authors have used open coding for this research.

During qualitative data analysis, open coding is the analytic process through which ideas (codes) are associated to the observed data and phenomena [123]. Open coding is an investigative process in grounded theory in which the data under observation is conceptualised (encoded) [123]. In this way, the data is classified under relevant naming conventions using the developed codes [123]. This classification is achieved by dividing the data into meaningful expressions in a single word or in a sequence of few words which usually become the category names [123]. Then the relevant codes are placed under each of the identified expressions [123].

The degree of application of open coding varies depending upon the nature of text and research [123]. For example, a code can be formed based on a line, sentence, a paragraph or a complete text [123]. However, the underlying objective of applying open coding is to analyse and to categorise the relevant text. Thus, the application of open coding process provides a characterised list of codes and categories relevant to the text which is exactly our requirement for this research study. Therefore, open coding is applied on the extracted data in this research study. During the process, each new relevant instance (codes) from the relevant text of selected studies has been encoded under a specific theme. Similar instances (codes) have been grouped together under the same theme (category). Each theme (category) as well as the codes have been assigned a name thus obtaining a set of categories and factors for each category.

3.2.3 Expert Review

After the analysis and coming up with results, the next important step is to validate the naming conventions of each category as well as to validate the relevance of each influencing factor with its respective category. Another important task at this stage is to remove duplicate influencing factors and repetitions. These tasks are performed in Activity 3 of the phase 1 using Expert Review method. The expert opinion elicitation guideline proposed by Ayyub et al., [124] and Boring, et al., [125] has been adopted for expert opinion process. Their guideline is considered as a detailed and established guideline as well as it is published research so the authors have decided to base their expert opinion phase on those guidelines.

The criterion for selection of experts is:

- Must be Software Engineering professionals either from academia or from industry.
- Must be specialized in Requirement Engineering (RE).
- Must have knowledge of DAD.
- Must have more than 5 years of experience.

Three experts of the domain, one from the academia and two from the industry, have been selected to perform this activity. At the completion of this step, the authors have obtained the final categories along with relevant influencing factors for each category. This is the end of phase 1 of this study and results of this study are used to conduct survey which is phase 2 and the method has been explained in the next section.

3.3 Survey

The second phase of this research is to prioritize the results of first phase which has been achieved by performing survey. A survey is “a system for collecting information to describe, compare, or explain knowledge, attitude, and practices or behaviour” [126]. The reason of conducting surveys is to obtain the real-time picture of the subject under discussion [127]. The survey has been conducted by developing an online questionnaire using Google forms.

The reason of adopting questionnaire as the survey method is its ability to get data from many different individuals [128]. The results of SLR are used to develop the survey questionnaire. The participants of the survey are individuals involved in different software processes in DAD projects. The survey is performed to achieve two objectives:

- To prioritize the identified categories.
- To prioritize the identified influencing factors for each category.

The survey design explained by Creswell has been used for this purpose [129]. The widely accepted and well recognized handbook to manage surveys in software engineering written by Mark Kasunic is used as guidance to conduct this survey [130].

The questionnaire is decomposed in different sections with each section representing one of the identified categories and under each section, one question represents one influencing factor. Five-point Likert scale has been used to get the answers. The results of survey have been used to prioritize the categories and influencing factors. This provides the authors with a real-time picture of the severity of each of the influencing factor as well as the importance level of each category. The steps performed to complete this phase are illustrated in Figure 3.5

Identify research objectives is the first step for survey conduction. In this step, the authors clearly describe the objective of conducting the survey. For example, the objective of conducting this survey is to prioritize the identified categories and influencing factors for RCM in DAD.

Identify and Characterize the Target Audience is the second step for survey conduction. The target audience for this research is the software engineering practitioners and academicians who are working in the domain of DAD and related to RCM activities. The handbook guidance of Mark Kasunic has been utilized to devise the questions to choose the right participants [129].

Design Sampling Plan is the third step for Survey Conduction. In this step, the authors estimate a sample size of the target audience. Once the authors have the clear picture about the size of the population then they decide the size of sample.

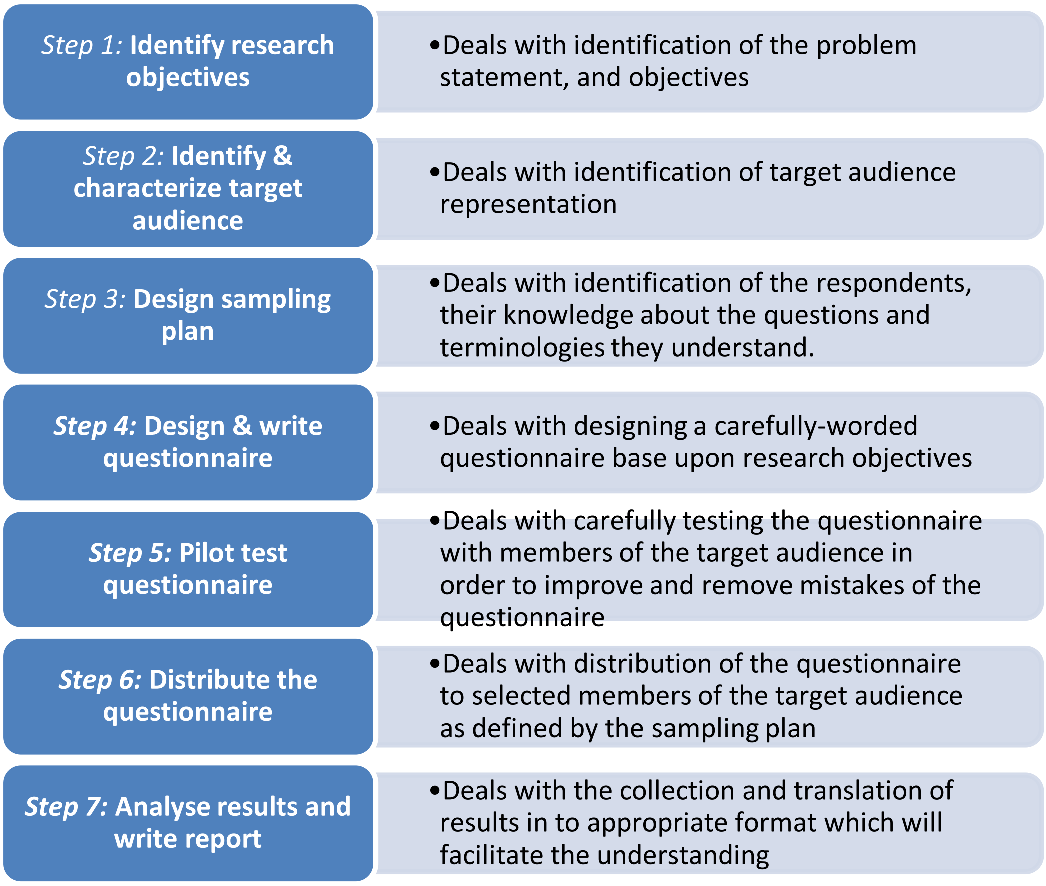

Step 1: Identify research objectives
•Deals with identification of the problem statement, and objectives
Step 2: Identify & characterize target audience
•Deals with identification of target audience representation
Step 3: Design sampling plan
•Deals with identification of the respondents, their knowledge about the questions and terminologies they understand.
Step 4: Design & write questionnaire
•Deals with designing a carefully-worded questionnaire base upon research objectives
Step 5: Pilot test questionnaire
•Deals with carefully testing the questionnaire with members of the target audience in order to improve and remove mistakes of the questionnaire
Step 6: Distribute the questionnaire
•Deals with distribution of the questionnaire to selected members of the target audience as defined by the sampling plan
Step 7: Analyse results and write report
•Deals with the collection and translation of results in to appropriate format which will facilitate the understanding

Figure 3.5: Steps involved in Survey Conduction

Table 3.8 shows the set of questions which are considered for identifying and characterizing the target audience.

Table 3.8: Questions to Identify and Characterize Target Audience [129]

Questions for Identifying and Characterizing the Target Audience
How many people are in the population we are studying? What are their jobs and responsibilities? What is the most common education level? What relevant experience do they possess? What technical abilities do they possess? What is the age range of the respondent population? Do we anticipate that they would have difficulty with using a questionnaire that is: − mailed to them? − completed using a computer via the internet? − handed to them? What can we assume about their knowledge of the domain we are studying in the survey? How much of their time can we assume they will spend completing the questionnaire?

Design and Write Questionnaire is the fourth step for Survey Conduction. In this step, the authors design and write the questionnaire for the survey. The designing and writing the questionnaire has been done by considering the objectives and the characteristics of the target audience. For designing the questionnaire, the authors focus on addressing the questions adopted from [129]. Table 3.9 shows the questions which are considered while designing the questionnaire.

Table 3.9: Questions for Designing Questionnaire, adopted from [129]

Questions for Designing a Questionnaire
How will the survey be mediated (e.g., via paper, email soft copy, Web)? How long should the questionnaire be? How should the questionnaire be structured and organized? What page design and formatting will be most effective?

Pilot Test Questionnaire is the fifth step for Survey Conduction. In this step, the authors perform the questionnaire testing by sending the questionnaire to some of the members of the target audience (Requirement Engineers). This represents the weaknesses and the problems in the questions, layout of questionnaire and technology (Web-based questionnaire). Based upon the feedback from the members of the target audience (Requirement Engineers), the authors improve the questionnaire.

Distribute the Questionnaire is the sixth step for Survey Conduction. In this step, the authors provide the questionnaire to the respondents after thorough testing of the questionnaire. The authors make sure that the respondents already know about the questionnaire to be filled by them and it is not going to become a surprise for them.

Analyze Results is the seventh step for Survey Conduction. In this step, the authors organize the data and then report the data in form of prioritized list of influencing factors and their categories for RCM in DAD environment. The results of survey have been presented in Chapter 5.

3.4 Framework for RCM in DAD

The third and final phase of this research is to present a framework to manage requirements change in DAD efficiently by effectively coping up with influencing factors and giving the required level of importance to the prioritized categories. This has been performed using derived methodology and by conducting brainstorming sessions to come up with the right framework. The devised framework has been presented in Chapter 5.

3.5 Summary

The research methodology adopted for this research study has been explained in this chapter. All three phases and their subsequent activities have been discussed. The next chapter presents the results of first phase of this study.

CHAPTER 4

RESULTS

4.1 Overview

This chapter is based on the results of first phase of this study. As discussed in Chapter 3, the first phase consisted of three activities: SLR, analysis using grounded theory and experts' review. The next sections present the results of these activities.

4.2 Results of SLR

The results of selected studies from each database after each step of inclusion/ exclusion criteria are shown in Table 4.1.

Table 4.1: Results at each step for each database

Search string	IEEE	Google scholar	ACM	Springer Link	Science Direct
((("Requirements change management" OR "Change requirements management" OR "requirements AND "Agile" AND (Global OR Distributed OR Offshore OR dispersed)) AND "Software" AND (Development OR Engineering OR Team))	Phase 1: 617 Phase 2: 232 Phase 3: 19	Phase 1: 870 Phase 2: 257 Phase3: 11	Phase 1: 13 Phase 2: 5 Phase3: 4	Phase 1: 55 Phase 2: 11 Phase 3: 2	Phase 1: 43 Phase 2: 19 Phase 3: 7

As shown in the Table 4.1, When query is applied to IEEE, 617 results have been displayed at the first phase. After applying criteria of phase 2, results have been shortlisted to 232 and after phase 3, the included paper count is 19. Similarly, the three step criteria are applied on all databases to shortlist the results according to their relevance and significance. Once, the database search is complete and results of all phases have been gathered for all five databases, the quality assess criteria is applied on the shortlisted studies to obtain the final studies as shown in Table 4.2.

Table 4.2: Quality Assessment Analysis

P #	Respondent 1							Respondent 2							Respondent 3							Out of 21
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
1	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
2	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
3	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
4	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
5	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
6	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
7	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
8	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
9	.5	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	10
10	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
11	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
12	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
13	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
14	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
15	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
16	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
17	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
18	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
19	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
20	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
21	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
22	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
23	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
24	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
25	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
26	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
27	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
28	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
29	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
30	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
31	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
32	.5	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	10
33	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
34	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
35	0	.5	1	1	0	.5	.5	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
36	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
37	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
38	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
39	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5
40	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	1	1	1	1	.5	1	.5	18
41	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
42	1	1	1	1	.5	1	.5	1	1	1	1	1	.5	.5	1	1	1	1	.5	1	.5	18
43	.5	.5	0	1	0	.5	1	1	0	.5	1	.5	0	0	1	.5	0	1	.5	0	0	9.5

The studies with score more than or equal to 11 are selected for the final list. Total number of selected studies after applying quality assessment criteria is twenty-six. The selected studies include study number 1, 2, 4, 6, 7, 8, 10, 12, 14, 16, 18, 19, 22, 24, 25, 27, 28, 30, 31, 33, 34, 36, 37, 38, 41 and 42. List of finalized results after QA are shown in Table 4.3.

Table 4.3: List of Studies after QA

Databases	Phase 1	Phase 2	Phase 3	QA	Total Final Studies
IEEE	617	232	19	10	**26**
Google Scholar	870	257	11	6	
ACM	13	5	4	3	
Springer Link	55	11	2	1	
Science Direct	43	19	7	6	

Final studies with their unique paper IDs have been presented in Table 4.4. The first column "S. No." in the Table represents the number count of final papers. Paper ID column shows the unique ID given to each final paper while paper title column describes the title of each final paper.

Table 4.4: List of Final Studies

S.NO	Paper ID	Paper Title
01	P1	A Supporting Tool for Requirements Change Management in Distributed Agile Development
02	P2	Moving from traditional to agile software development methodologies also on large, distributed projects.
03	P3	Categorization of risk factors for distributed agile projects.
04	P4	Scrum Requirements Engineering Practices and Challenges in Offshore Software Development.
05	P5	Empirical studies of geographically distributed agile development communication challenges: A systematic review.
06	P6	Improving offshoring of low-budget agile software development using the dual-shore approach: an auto ethnographic study
07	P7	Quality Requirements in Large-Scale Distributed Agile Projects A Systematic Literature Review
08	P8	Quality requirements challenges in the context of large-scale distributed agile: An empirical study
09	P9	Toward successful agile requirements change management process in global software development: a client–vendor analysis
10	P10	Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development
11	P11	Experiences from the Design of an Artefact Model for Distributed Agile Project Management
12	P12	Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature

13	P13	Agile Software development in distributed team Enhancement Techniques
14	P14	A Framework for Transitioning of Traditional Software Development Method to Distributed Agile Software Development
15	P15	An Empirical Study on Lean and Agile Methods in Global Software Development
16	P16	Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective
17	P17	Toward an Agile Approach to Managing the Effect of Requirements on Software Architecture during Global Software Development
18	P18	Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process
19	P19	A Novel Framework for Change Requirement Management (CRM) In Agile Software Development
20	P20	An exploratory study in communication in Agile Global Software Development
21	P21	Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review
22	P22	A systematic review of distributed Agile software engineering
23	P23	An Empirical Investigation on Effort Estimation in Agile Global Software Development
24	P24	Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a global company
25	P25	Communication Network in an Agile Distributed Software Development Team
26	P26	A Risk Management Framework for Distributed Agile Projects

Data extraction tables for each final study are created which have been provided in the Appendix. For each final study, the information recorded in the data extraction table include the Database from which the paper is taken, its unique ID, Title, its author (s) names, year of publication, publication type and influencing factors which are presented in that paper.

4.3 Data Synthesis Result

Data synthesis is the next activity in the first phase of study. Once the data is extracted from the primary studies, it is then analysed and synthesized to help us answer certain questions like:

- What is the RCM influencing factors in DAD?
- How the influencing factors can be categorized for RCM in DAD?

This has been performed using grounded theory and the application of grounded theory to the extracted data is shown in the Table 4.5.

Table 4.5: Application of Grounded Theory

Paper #	**Coding**	
	Text in Paper	**Themes (Categories)**
P1	P. 7, Section 5, Line 2-5.	Understanding Requirements Change, Communication
P2	P. 7, Section 3, Paragraph entitled as "Enhancement Requests and Content changes"	Understanding Requirements Change, Impact Analysis, Progress Measure, Team Role
P3	P. 5, Section 3.3.1.3, Line 12-15	Understanding Requirements Change, Progress Measure, Management Role
P4	Pp 3-4, Sections 2.2.1, 2.2.2, 2.2.4	Communication, Progress Measure, Understanding Requirements Change
P5	Pp 8-10, Section 4.5.1, Paragraphs entitled as "Team configuration, Customer communication, Organization factors, Human factors"	Understanding Requirements Change, Impact Analysis, Communication
P6	P.4, Section 2.1.4, P.6 Section 2.3.3 P.12, Section 4.5	Understanding Requirements Change, Impact Analysis
P7	P.6, Table 1, Row 2	Team Role, Progress Measure
P8	P.5, Section 4.5.1	Understanding Requirements Change, Communication
P9	P.4, Table 2	Communication, Management Role, Team Role, Progress Measure, Impact Analysis
P10	Page 6, Table 1	Management Role, Communication, Progress Measure, Team Role,
P11	P3, Figure 1	Understanding Requirements Change, Impact Analysis

P12	P1, Abstract, Line 6-9. Section 1, Line 5-7	Progress Measure
P13	P2, Section 4, Bullet 2. P4. Paragraph 1.	Progress Measure, Team Role
P14	P1, Abstract, Line 10. P2, Table 1, Row 3 and 4.	Team Role, Communication, Management Role
P15	P1, Section 1B, Line 16-22 P2, Section 2, Paragraph 4, Line 7-14	Impact Analysis, Team Role, Progress Measure
P16	P1, Section 1, Paragraph 1, Line 5-11 Section 1, Paragraph 2, Line 1-4	Impact Analysis, Communication, Team Role
P17	P.2, Section 2.1, Paragraph 1	Understanding Requirements Change, Impact Analysis, Management Role
P18	P.1, Section 1, Paragraph 2, Line 3-5	Team Role, Progress Measure
P19	P.1, Section 1, Paragraph 1	Communication
P20	P.1, Section 1, Paragraph 2	Communication
P21	P.12, Section 5.4.1, Line 2-4	Communication
P22	P.18, Section 6.2.1.2, Paragraph 2	Management Role, Progress Measure
P23	P.23, Section 4F, “Requirements Related Challenges”	Communication
P24	P.4, Section 4C, Paragraph 2	Team Role, Management Role
P25	P4, Section 5, Last paragraph	Communication
P26	P9, Section 5.1.1, Paragraph 3, Line 5-7. Paragraph 5, Line 6-10	Understanding Requirements Change, Impact Analysis, Management Role

In the “paper #” section, unique paper ID of each paper is written. In the coding column, there are two subsections, text in paper and Themes (categories). Text in paper shows the instance of relevant text where it exists in the respective paper while themes are the derived categories to which the text of each paper belongs. It is important to note that there are more than one themes for a certain text in a paper.

Table 4.6: Study Wise Distribution of IFs

Study #	**Influencing Factors**
P1	IF1: Different geographical locations of the development teams IF2: Communication between development team IF3: Knowledge management

P2	IF 4: Enhancement Requests and Content Changes
	IF 5: Time Schedule
	IF 6: Project Cost

P3	IF 7: Software Development Life Cycle
	IF 8: Project Management
	IF 9: Group Awareness
	IF 10: External Stakeholder Collaboration
	IF 11: Technology Setup

P4	IF 12: Face to face communication
	IF 13: Customer Involvement and Interaction
	IF 14: Managing Requirement change

P5	IF 15: Team Configuration
	IF 16: Customer Communication
	IF 17: Organizational Factors
	IF 18: Human Factors

P6	IF 19: Responding to change over following a plan IF 20: Customer Collaboration IF 21: Cost Estimation and extra payment for changes
P7	IF 22: Skilled developers
P8	IF 23: Teams coordination and communication

P9	IF 24: Effective communication
	IF 25: Trust building
	IF 26: Roles and responsibilities
	IF 27: Vision and goal for change
	IF 28: Training and monitoring
	IF 29: Geographically distributed CCB (change control block)
	IF 30: Resistance management
	IF 31: Formalised relationship between development teams
	IF 32: Skilled human resources
	IF 33: Overseas site response
	IF 34: Clear change management strategy
	IF 35: Requirement traceability
	IF 36: Change management process awareness
	IF 37: Governance and control of RCM activities
	IF 38: Standard and process for RCM
	IF 39: Change identification and validation
	IF 40: Continuous organisational support
	IF 41: Conducting social events
	IF 42: Synchronisation of work at different sites
	IF 43: Parallel project testing and feedback

P10	IF 44: Project Administration
	IF 45: Coordination
	IF 46: Software Methodology

	IF 47: HR management
	IF 48: Technology Factors

P11	IF 49: Resources (Time and cost)
	IF 50: Change request

P12	IF 51: Rapid response to changes in requirements rather than following a prescribed plan IF 52: Collaboration between the software developer and user
P13	IF 53: Welcome requirements change, even at the end of development IF 54: Positive team role

P14	IF 55: Support quick and flexible response to change
	IF 56: Team coordination
	IF 57: Project leader

P15	IF 58: Autonomy and Decision-Making Power
	IF 59: Choosing the Right Method

P16	IF 60: Tight Time and Constraint
	IF 61: Coordination
	IF 62: Communication
	IF 63: Lack of Tools

P17	IF 64: Reporting the Change
	IF 65: Analysing the Possible Impact of the Change
	IF 66: Implementing the Change

P18	IF 67: Importance of Individuals and their Interactions
	IF 68: Early Delivery
	IF 69: Collaboration with the customers
	IF 70: Effective responses to change requirements

P19	IF 71: Communication
	IF 72: Documentation

P20	IF 72: Tools for Communication
P21	IF 73: Communication of changes in requirements
P22	IF 74: Strong leadership
P23	IF 75: Documentation of requirements
P24	IF 76: Clearly defined team roles
P25	IF 77: Communication in teams with large time difference

P26	IF 78: Fixed Costs
	IF 79: Speed of communication
	IF 80: Ease of communication

Study wise distribution of identified Influencing Factors for RCM in DAD from the 26 final selected studies has been presented in Table 4.6. A total of 80 influencing factors have been identified after analysing the studies using grounded theory.

Following six categories are devised after careful and detailed analysis of collected data from each paper.

Table 4.7: Resultant Categories

Resultant Categories
1: Understanding Requirements Change (URC) 3: Management Role (MR) 4: Team Role (TR) 5: Progress Measure (PM) 6: Communication (C)

Category wise distribution of identified influencing factors for questionnaire and further analysis is presented in Table 4.8.

Table 4.8: Category Wise Distribution of Identified Influencing Factors

Study #	**Influencing Factors**	**Category wise distribution and questionnaire formulation**
P1	IF1: Different geographical locations of the development teams IF2: Communication between development team IF3: Knowledge management	URC C

P2	IF 4: Enhancement Requests and Content Changes
	IF 5: Time Schedule
	IF 6: Project Cost
		PM
		TR
		URC

P3	IF 7: Software Development Life Cycle
	IF 8: Project Management
	IF 9: Group Awareness
	IF 10: External Stakeholder Collaboration
	IF 11: Technology Setup
		URC
		PM
		MS

P4	IF 12: Face to face communication
	IF 13: Customer Involvement and Interaction
	IF 14: Managing Requirement change
		C
		PM, TR
		URC

P5	IF 15: Team Configuration
	IF 16: Customer Communication
	IF 17: Organizational Factors
	IF 18: Human Factors
		URC
		IA

P6	IF 19: Responding to change over following a plan IF 20: Customer Collaboration IF 21: Cost Estimation and extra payment for changes	URC IA
P7	IF 22: Skilled developers	URC, TR, PM
P8	IF 23: Teams coordination and communication	URC, C

P9	IF 24: Effective communication
	IF 25: Trust building
	IF 26: Roles and responsibilities
	IF 27: Vision and goal for change
		C
		MR
		MR
		TR, MR, URC

	IF 28: Training and monitoring
	IF 29: Geographically distributed CCB (change control block)
	IF 30: Resistance management
	IF 31: Formalised relationship between development teams
	IF 32: Skilled human resources
	IF 33: Overseas site response
	IF 34: Clear change management strategy
	IF 35: Requirement traceability
	IF 36: Change management process awareness
	IF 37: Governance and control of RCM activities
	IF 38: Standard and process for RCM
	IF 39: Change identification and validation
	IF 40: Continuous organisational support
	IF 41: Conducting social events
	IF 42: Synchronisation of work at different sites
	IF 43: Parallel project testing and feedback
		TR
		C
		C
		TR
		C
		
		TR
		C, PM
		IA, MR
		IA
		IA, URC
		MR, TR, IA
		IA
		IA
		MR
		MR
		C
		C, IA

P10	IF 44: Project Administration
	IF 45: Coordination
	IF 46: Software Methodology
	IF 47: HR management
	IF 48: Technology Factors
		MR
		C, TR
		PM
		TR, MR
		C

P11	IF 49: Resources (Time and cost)
	IF 50: Change request
		URC, IA

P12	IF 51: Rapid response to changes in requirements rather than following a prescribed plan IF 52: Collaboration between the software developer and user	PM
P13	IF 53: Welcome requirements change, even at the end of development IF 54: Positive team role	PM, TR

P14	IF 55: Support quick and flexible response to change
	IF 56: Team coordination
	IF 57: Project leader
		TR, C, MR

P15	IF 58: Autonomy and Decision-Making Power
	IF 59: Choosing the Right Method
		IA, TR, PM

P16	IF 60: Tight Time and Cost Constraint
	IF 61: Coordination
	IF 62: Communication
	IF 63: Lack of Tools
		IA, C, TR

P17	IF 64: Reporting the Change
	IF 65: Analysing the Possible Impact of the Change
	IF 66: Implementing the Change
		URC, IA, MR

P18	IF 67: Importance of Individuals and their Interactions
	IF 68: Early Delivery
	IF 69: Collaboration with the customers
	IF 70: Effective responses to change requirements
		TR, PM

P19	IF 71: Communication	C

	IF 72: Documentation	
P20	IF 72: Tools for Communication	C
P21	IF 73: Communication of changes in requirements	C
P22	IF 74: Strong leadership	MR, PM
P23	IF 75: Documentation of requirements	C
P24	IF 76: Clearly defined team roles	TR, MR
P25	IF 77: Communication in teams with large time difference	C

P26	IF 78: Fixed Costs
	IF 79: Speed of communication
	IF 80: Ease of communication
		URC, IA, MR

Once the analysis is complete, the resultant categories and identified influencing factors are sent for expert review.

4.4 Expert Review Results

After the categories are devised and all influencing factors have been distributed in their respective categories, the results are sent to the experts for expert review with an additional column of description where each identified factor is described for the better understanding of the experts as shown in the Table 4.9. This is done to validate the results and naming conventions as well as to eliminate the repetitions.

Table 4.9: Results after Expert Review

Categories	Influencing Factors	Description
Understanding Requirements Change (URC) This category includes influencing factors related to the requirements change requests, processing the change, its implementation and reporting as well as the knowledge management between geographically dispersed teams and collaboration with all	IF1: Different geographical locations of the development teams	Teams are located at different geographical locations in distributed agile development (DAD). It is important to understand that which teams are involved/ will be affected during a specific requirement change request.

stakeholders.		
	IF2: Knowledge Management	Knowledge management is also a salient aspect during SDLC. It is pertinent to focus on KM issues to clearly understand the requirements change process.
	IF 3: Enhancement Requests and Content Changes	Enhancement requests and content changes trigger requirements change process. So, it is necessary to clearly understand these requirement changes.
	IF4: External Stakeholder Collaboration	Stakeholders’ collaboration is one of the most important aspect to clearly understand requirements changes. Therefore, this IF falls in the category of URC.
	IF 5: Managing Requirement change	Before starting to implement the requirements change (RC) process, it is important to plan how the change will be managed throughout the SDLC and between the relevant distributed teams. This includes the collaboration of central management and the involved agile virtual teams. Therefore, this IF falls under the URC category.
	IF 6: Vision and goal for change	Defining the vision and goal for change is also an inevitable IF for smooth understanding of requirements change.
	IF 7: Reporting the Change	Reporting every change is a primary IF to understand RC during SDLC.
	IF 8: Responding to change over following a plan	An important IF during URC is to decide whether to respond to some specific change request or to continue sticking to the plan.
	IF 9: Implementing the Change	Once its decided to the respond to the change, how

		the change will be implemented is also an important IF to understand requirements change.
Impact Analysis (IA) This category belongs to the group of factors related to the impact analysis of the change. This includes human factors, organizational factors, resources (time, cost, tools) and response measures.	IF 10: Human Factors	Human factors to measure the impact analysis include the team members' willingness to implement the change, their effectiveness in the change process, current project situation, managers-teams' relations as well as inter-teams and intra-teams' relations.
	IF 11: Organizational Factors	Another important IF to measure the Impact analysis is the organizational behavior and attitude towards welcoming the changes.
	IF 12: Cost Estimation and extra payment for changes	One of the important IF for impact analysis is the additional cost estimation to implement the change as well as having the capacity to pay for the changes if required other than the already allocated budget.
	IF 13: Parallel project testing and feedback	Parallel project testing and feedback between the teams plays a vital role to analyze the impact of every requirements change.
	IF 14: Resources (Time and cost)	Availability of resources are vital for every change. Tight schedules often end up in failure to implement change or delays in delivery as well as increasing the overall project cost. Thus, this is an important IF in impact analysis.
	IF 15: Analyzing the Possible Impact of the Change/ change request	What kind of change/ change request has arrived and how it's going to affect the overall project is a vital IF to measure IA. Therefore, analyzing the possible impact of every

		change/ change request is important in every project. It must not be ignored.
	IF 16: Lack of Tools	Lack of tools hamper the smooth change implementation. unavailability of tools and their impact on the development is vital. Therefore, it is an important IF in IA category.
	IF 17: Effective responses to change requirements	Effective responses are very important to avoid delays and to implement changes. Therefore, the response time matters in impact analysis.
	IF 18: Fixed Costs	Sometimes projects have fixed costs. In such scenario, the impact analysis becomes vital to make decisions.
Management Role (MR) All the management related influencing factors belong to this category including schedule and budget decisions, tools and technology decisions, assigning roles and responsibilities, choosing the right team and tools, and performing other administrative tasks.	IF 19: Time Schedule	Management needs to devise clear and practical time schedules including having margins for changes.
	IF 20: Project Cost	Management shall do the cost estimation keeping in view the possible additional changes during the development.
	IF 21: Project Management	Project management plays a vital role during RCM. Managers shall be flexible to adopt changes as well as they should provide the appropriate environment to the teams.
	IF 22: Technology Setup	Making sure to provide the necessary technology is also an important

		management's task
	IF 22: Trust building	Management shall play its role to build trust within teams and external stakeholders to implement changes.
	IF 23: Roles and responsibilities	Management shall clearly define roles and responsibilities of every individual.
	IF 24: Training and monitoring	Management shall provide proper training and monitoring.
	IF 25: Resistance management	Management shall cope up with resistance to the change at their level as well as at teams' level.
	IF 26: Skilled human resources	Management shall provide required skilled human resources.
	IF 27: Clear change management strategy	Management shall devise clear and precise RCM strategy.
	IF 28: Change management process awareness	Management itself shall be aware about RCM process as well as it shall create awareness among teams and external stakeholders.
	IF 29: Conducting social events	Social events play positive role in collaboration. This impact positively on development including the implementation of changes
	IF 30: Project leader	Project leader shall be defined in time. In DAD, project is divided in small dispersed teams. Leaders shall be chosen in time for every task and module.
	IF 31: Clearly defined team roles	Just like leaders, roles of every team member shall be defined clearly.
	IF 32: Tools for Communication	Availability of tools required for proper communication shall be made available by the management.
	IF 33: Project Administration	Management shall administer the project

		throughout during SDLC to effectively process changes.
	IF 34: Overseas site response	Management shall make sure the timely responses from overseas site during RCM.
Team Role (TR) Team related influencing factors fall in this category which include SDLC activities, inter team relations, outside interaction and collaboration, choosing the right processes and performing respective tasks.	IF 35: Software Development Life Cycle	Teams shall be ready to implement the change if necessary, at any SDLC phase
	IF 36: Group Awareness	Different development groups shall be aware of changes.
	IF 37: Customer Involvement and Interaction	Teams shall be in continuous communication with customers with required level of involvement.
	IF38: Formalized relationship between development teams	Distributed teams shall have formal and working relationship with each other.
	IF 39: Requirement traceability	Teams shall trace requirements at every required level.
	IF 40: Team Configuration	Teams shall configure themselves according to the situations.
	IF 41: Team coordination	Team members shall continuously coordinate with each other while implementing changes.
	IF 42: Governance and control of RCM activities	Team members shall be given control RCM activities while implementing changes to the required level.
	IF 43: Standard and process for RCM	Teams shall follow standards and process requirements of RCM.
	IF 44: Change identification and	Team member shall identify the changes as well

	validation	as validate them for successful delivery.
	IF 45: Synchronization of work at different sites	Teams shall sync their work at different sites while making changes.
	IF 46: Support quick and flexible response to change	Team members shall support quick and flexible response to changes at every level of development.
	IF 47: Choosing the Right Development Method	Team members shall choose the right implementation methods.
	IF 48: Importance of Individuals and their Interactions	Every individual shall be given the due importance and respect in the team. Proper interaction must be made sure to smoothly implement changes at all required levels.
	IF 49: Early Delivery	Teams shall focus on timely delivery.
Communication Communication methods, procedures and tasks related to the communication process between different teams, and with other stakeholders)	IF 50: Face to face communication	Face to face communication becomes impossible for agile virtual teams. Whenever possible, it shall be made sure.
	IF 51: Customer Communication	Communication with customers shall be maintained throughout the development.
	IF 52: Effective communication	Effective communication techniques and platforms must be adopted by agile virtual teams.
	IF 53: Geographically distributed CCB (change control block)	Change control block shall ensure communication between agile virtual teams.
	IF 54: Communication in teams with large time difference	Teams with larger differences in time zones face more difficulty in timely communications. Efficient means and communication methods shall be devised for such cases.
	IF 55: Documentation of	Requirement changes shall

	requirements	be documented and communicated in time.
	IF 56: Speed of communication	Communication between teams shall be speedy to avoid further delays.
	IF 57: Ease of communication	Communication methods shall be easy and formal in DAD.
Progress Measure (PM) Factors related to measuring progress as well as those factors that influence the progress of the project are added to this category.	IF 58: Skilled developers	Having skilled developers increase the pace of project. Projects progresses in time.
	IF 59: Continuous organizational support	Continuous organization support helps in timely progress of project.
	IF 60: IT platforms	The timeline and effectiveness of requirements change implementation depend on the existing IT platforms an organization is using.
	IF 61: HR management	HR management is vital to measure progress of a project.
	IF 62: Strong leadership	Strong leadership makes timely decisions thus helping in better project progress.
	IF 63: Welcome requirements change, even at the end of development	Change shall be welcomed even late in the project if necessary. The progress shall be measured continuously.
	IF 64: Positive team role	Positive role of teams helps in better progress.
	IF 65: Rapid response to changes in requirements rather than following a prescribed plan	Rapid change responses increase project progress. Although this IF is relevant to IF 8, here it is being added to measure progress.
	IF 66: Collaboration between the software developer and user	Developer and user continuous collaboration helps in achieving better progress.
	IF 67: Autonomy and Decision-Making Power	Autonomy and decision-making power shall be

		given at required levels to avoid delays and failures. This increases the project progress.

After the experts' review, the final influencing factors are 67 for the six identified categories as shown in the Table 4.9. This is the end of phase 1 in this research.

4.5 Summary

This chapter presents results of all three activities of phase one of this research. The next chapter presents results of the next phases.

CHAPTER 5

SURVEY RESULTS

5.1 Overview

This chapter presents the results of survey and the development of framework.

5.2 Survey Results

The next phase of this research is survey, and this section presents the results of survey. 91 participants take part in the online questionnaire survey which is conducted through google forms. The gender distribution is shown in Figure 5.1.

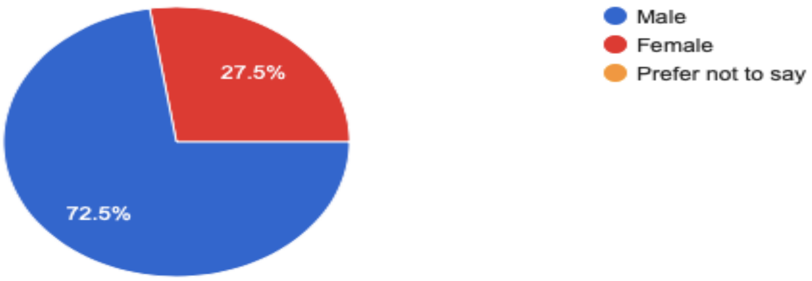


Figure 5.1: Gender-wise Distribution

As Shown in Figure 5.1, total 91 responses are gathered. Out of 91 respondents 27.5% are female and 72.5% are male.

5.2.2 Distribution of Participants against Organization Size

The distribution of participants against organization size is as follows:

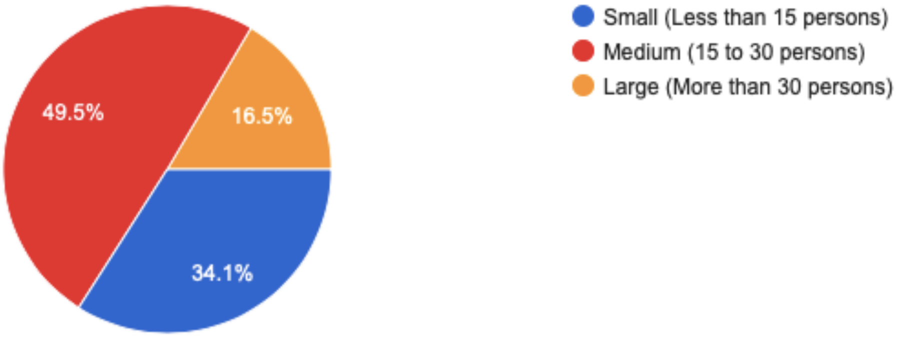


Figure 5.2: Organization Size of the Participants

As shown in Figure 5.2, total 91 responses are collected. Out of 91 respondents 34.1% have small (less than 15 persons) organization size, 49.5% have medium (15 to 30 persons) and 16.5% have large (more than 30 persons) organization size.

5.2.3 Distribution of Participants against Work Experience

The distribution of participants against work experience is:

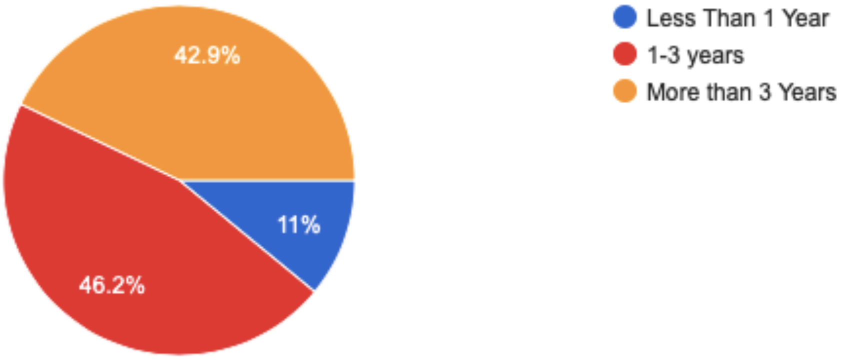


Figure 5.3: Work Experience Distribution

As shown in Figure 5.3, total 91 responses are collected. Out of 91 respondents 11% have less than 1 year of work experience, 46.2% have 1-3 years of work experience and 42.9% have more than 3 years of work experience.

5.2.4 Prioritization of Influencing Factors of Category 1: Understanding Requirements Change

There are nine influencing factors in this first category of understanding requirements change. Different geographical locations of the development teams, knowledge management, enhancement requests and content changes, external stakeholder collaboration, managing requirement change, vision and goal for change, reporting the change, responding to change over following a plan and responding the change. Total number of respondents for IF prioritization is 91.

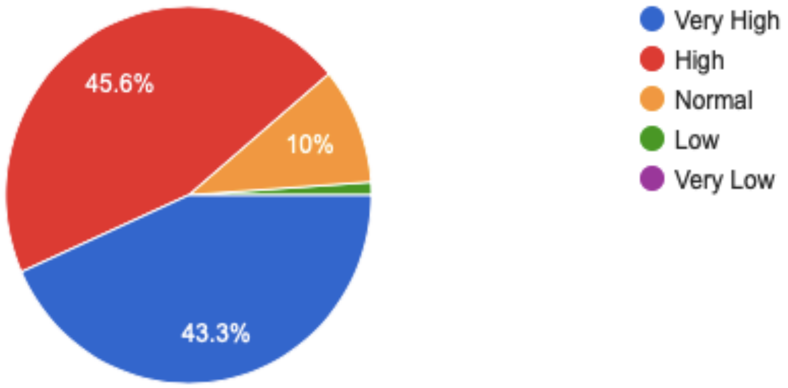


Figure 5.4: Different geographical locations of the development teams Results

The next influencing factor in this category is **Knowledge Management**. Out of 91 respondents, 52.2% participants have rated it as Very High while 32.2% have rated it as High influencing factor in this category as shown in Figure 5.5.

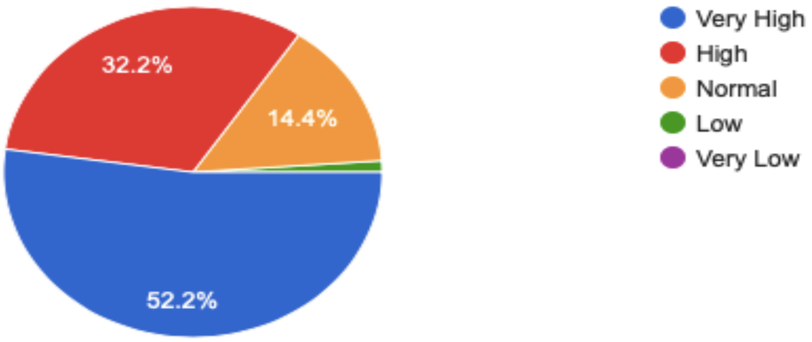


Figure 5.5: Knowledge Management Results

The next influencing factor in this category is **enhancement request and content changes**. Out of 91 respondents, 52.2% participants have rated it as Very High while 37.8% have rated it as High influencing factor in this category as shown in Figure 5.6.

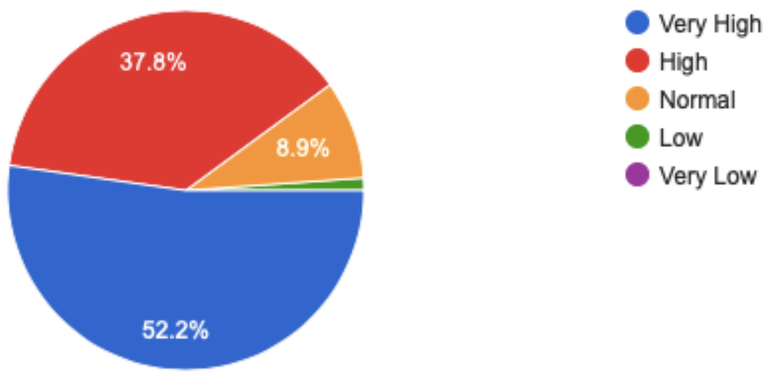


Figure 5.6: Enhancement Requests and Content Changes Result

The next influencing factor in this category is **External Stakeholder Collaboration**. 37.8% participants have rated it as Very High while 47.8% have rated it as High influencing factor in this category as shown in Figure 5.7.

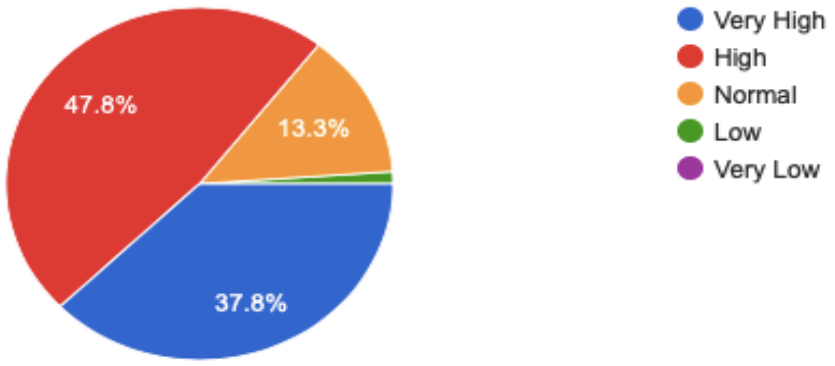


Figure 5.7: External Stakeholder Collaboration

The next influencing factor in this category is **managing requirement change**. 42.2% participants have rated it as Very High while 38.9% have rated it as High influencing factor in this category as shown in Figure 5.8.

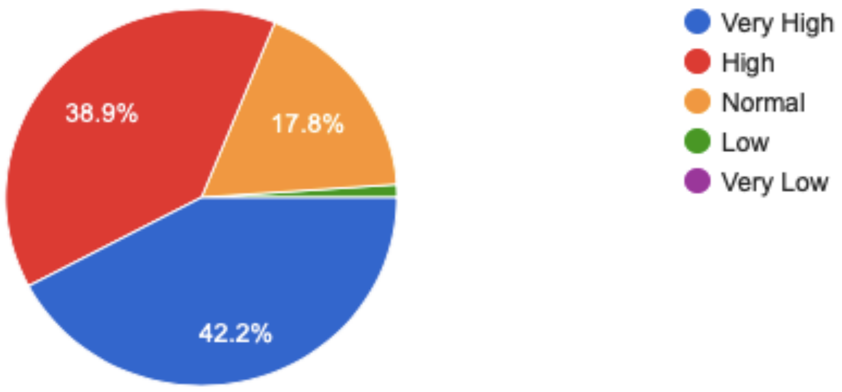


Figure 5.8: Managing Requirement change Result

The next influencing factor in this category is **vision and goal for change**. 54.4% participants have rated it as Very High while 30% have rated it as High influencing factor in this category as shown in Figure 5.9.

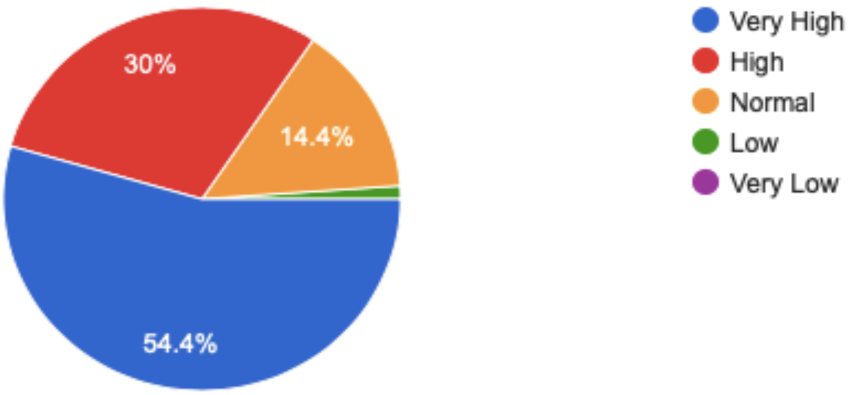


Figure 5.9: Vision and goal for change Result

The next influencing factor in this category is **reporting the change**. Out of 90 respondents 48.9% participants have rated it as Very High while 38.9% have rated it as High influencing factor in this category as shown in Figure 5.10.

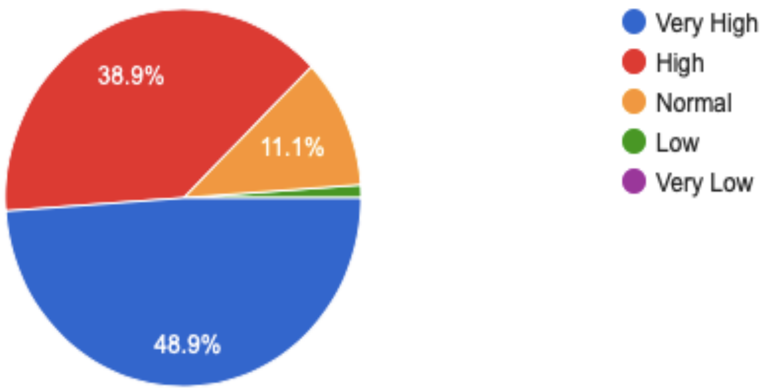


Figure 5.10: Reporting the Change Result

The next influencing factor in this category is **Responding to change over following a plan Result.** Out of 90 respondents, 48.4% participants have rated it as Very High while 44% have rated it as High influencing factor in this category as shown in Figure 5.11.

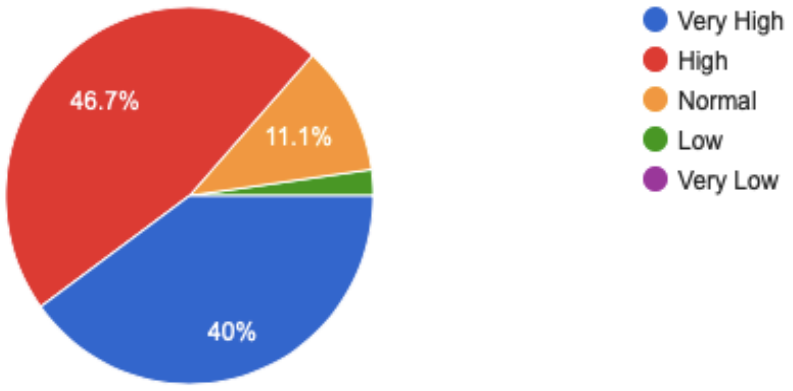


Figure 5.11: Responding to change over following a plan Result

The next influencing factor in this category is **implementing the change**. Out of 90 respondents, 48.4% participants have rated it as Very High while 44% have rated it as High influencing factor in this category as shown in Figure 5.12.

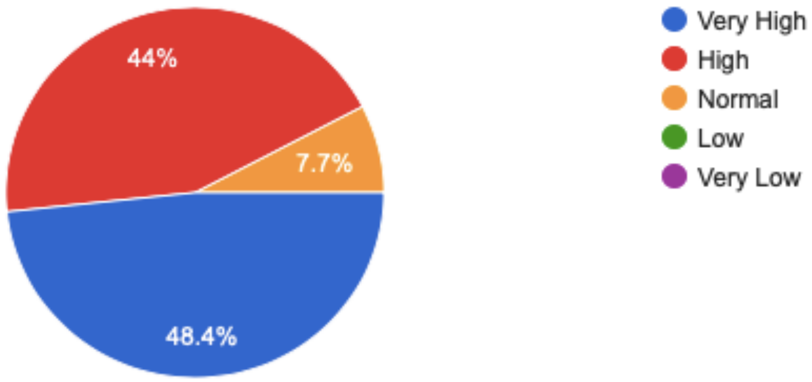


Figure 5.12: Implementing the Change Result

Complete List of prioritized factors for URC category are shown in Figure 5.13.

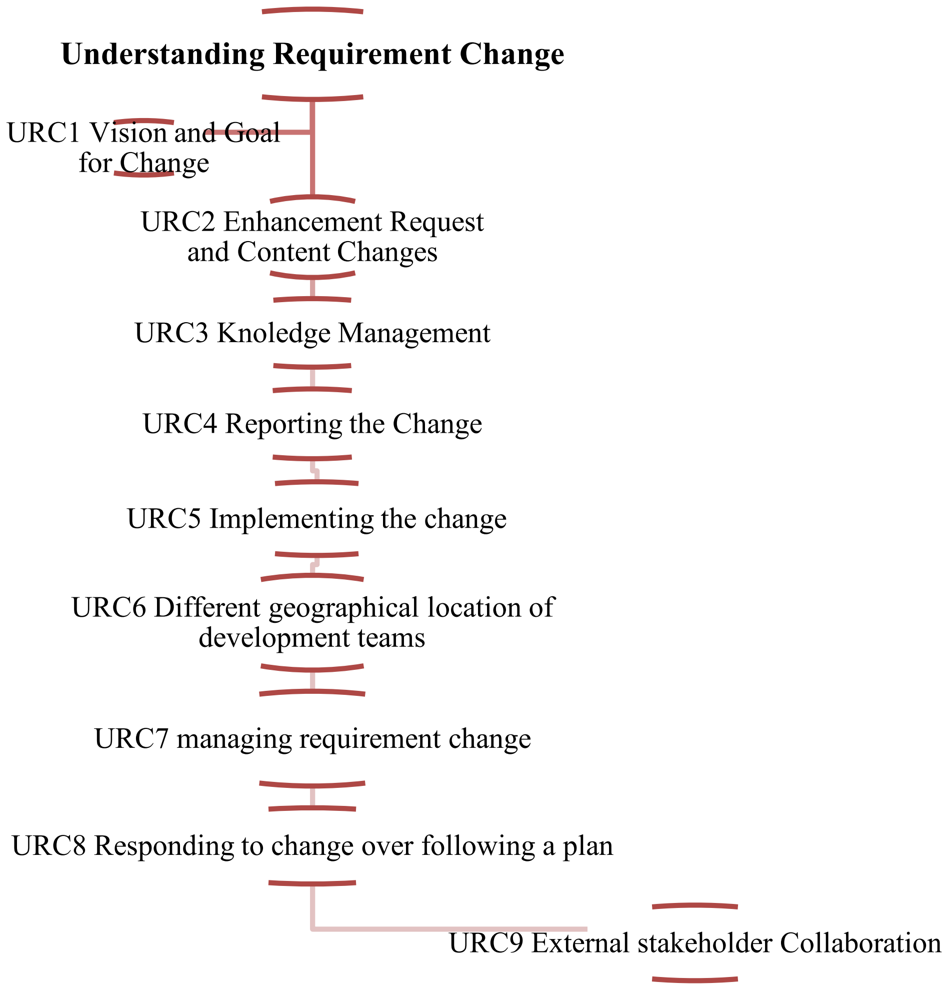

Understanding Requirement Change
URC1 Vision and Goal for Change
URC2 Enhancement Request and Content Changes
URC3 Knoledge Management
URC4 Reporting the Change
URC5 Implementing the change
URC6 Different geographical location of development teams
URC7 managing requirement change
URC8 Responding to change over following a plan
URC9 External stakeholder Collaboration

Figure 5.13: Prioritized List of URC Category

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.13. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.13 represents the initials "URC" of the category Understanding Requirements Change and numbers represent the priority level from high to level low for each influencing factor.

- **Overall Category Rating**: 71.07
- **Priority Number**: 3

5.2.5 Prioritization of Influencing Factors of Category 2: Impact Analysis

There are nine influencing factors in this category as well. Human factors, organizational factor, Cost Estimation and extra payment for changes Result, parallel project testing and feedback, resources (time and cost), Analysing the Possible Impact of the Change/ change request Result, lack of tool results, Effective Response to Change Requirements Result and Fixed Costs result.

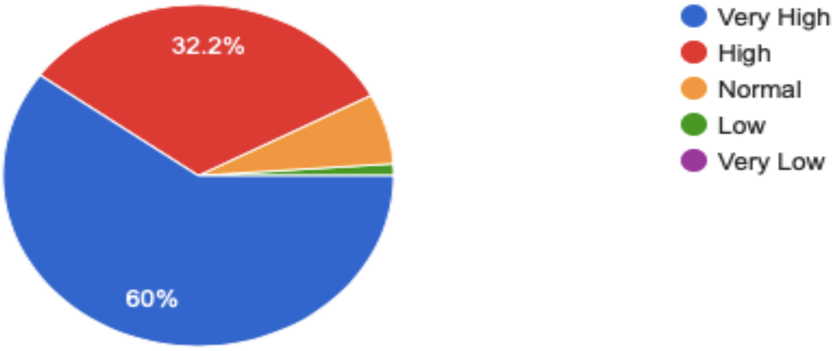


Figure 5.14: Human Factors Result

The first influencing factor in impact analysis result category is **human factors**. Out of 90 respondents, 60% participants have rated it as Very High while 32.2% have rated it as High influencing factor in this category as shown in Figure 5.14.

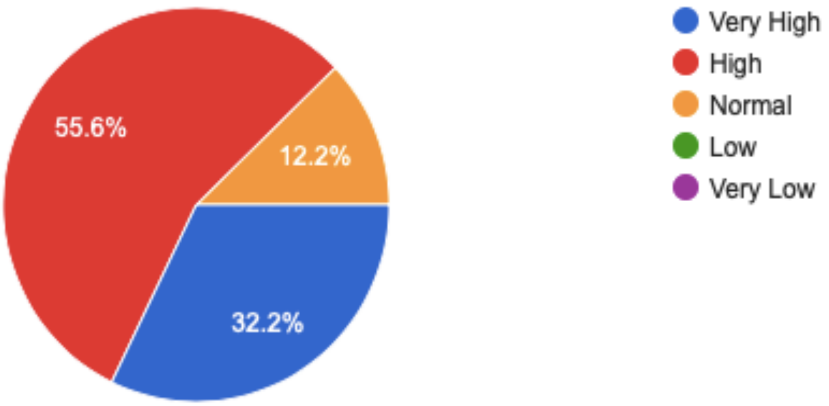


Figure 5.15: Organizational Factors Result

The next influencing factor in this category is **organizational factors**. Out of 90 respondents, 32.2% participants have rated it as Very High while 55.6% have rated it as High influencing factor in this category as shown in Figure 5.15.

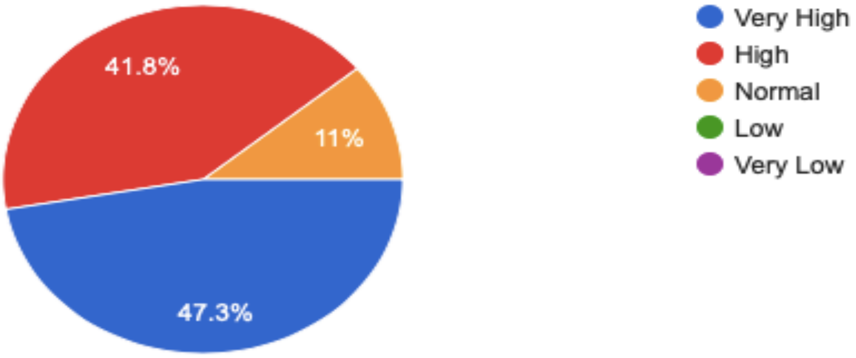


Figure 5.16: Cost Estimation and extra payment for changes Result

The next influencing factor in this category is **cost estimation for change**. Out of 90 respondents, 47.3% participants have rated it as Very High and 41.8% have rated it as High influencing factor in this category as shown in Figure 5.16.

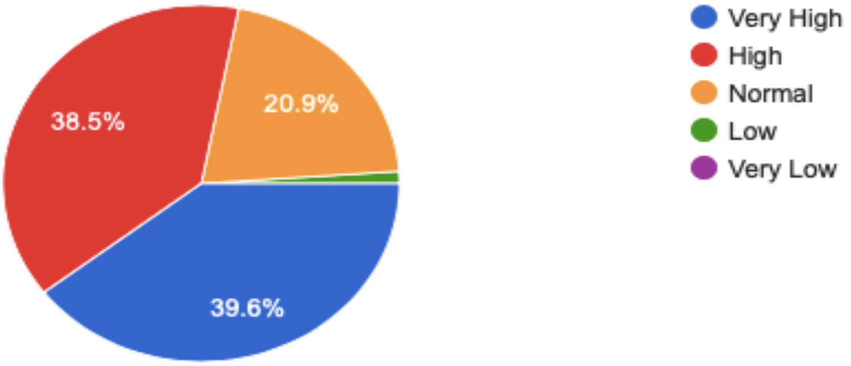


Figure 5.17: Parallel project testing and feedback

The next influencing factor in this category is **parallel project testing and feedback**. Out of 90 respondents, 39.6% participants have rated it as Very High while 38.5% have rated it as High influencing factor in this category as shown in Figure 5.17.

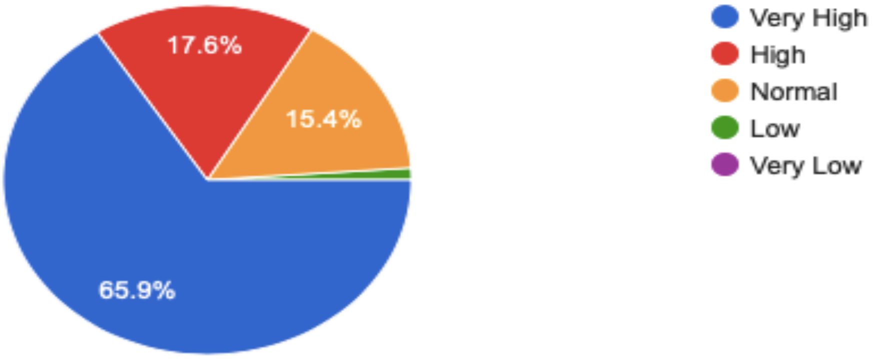


Figure 5.18: Resources (Time and cost) Result

The next influencing factor in this category is **resources**. Out of 90 respondents, 65.9% participants have rated it as Very High while 17.6% have rated it as High influencing factor in this category as shown in Figure 5.18.

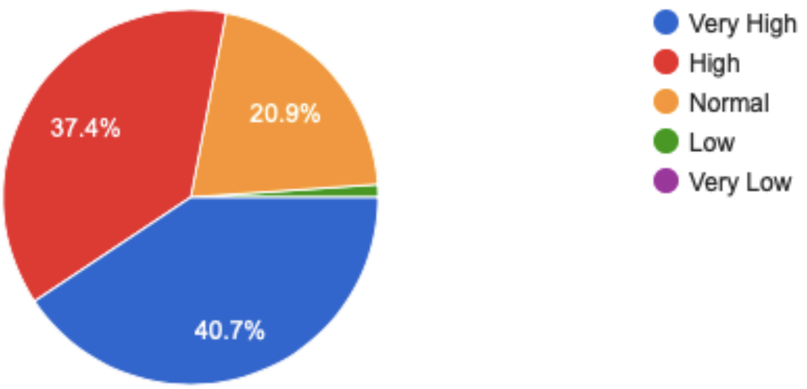


Figure 5.19: Analysing the Possible Impact of the Change/ change request Result

The next influencing factor in this category is **analyzing the possible impact of the change/change request**. Out of respondents 40.7% participants have rated it as Very High while 37.4% have rated it as High influencing factor in this category as shown in Figure 5.19.

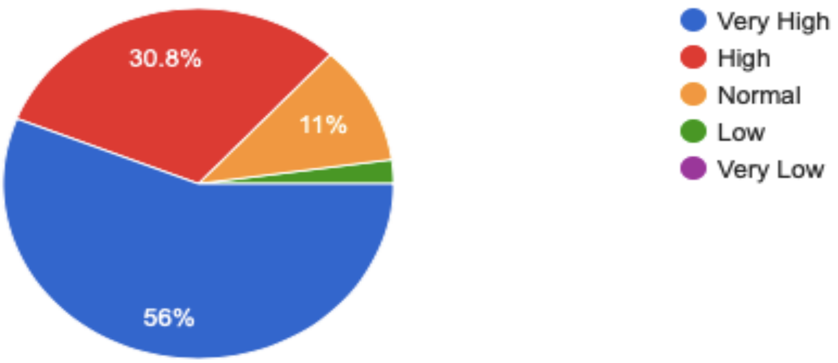


Figure 5.20: Lack of Tools Result

The next influencing factor in this category is **lack of tools**. Out of 90 respondents, 56% participants have rated it as Very High while 30.8% have rated it as High influencing factor in this category as shown in Figure 5.20.

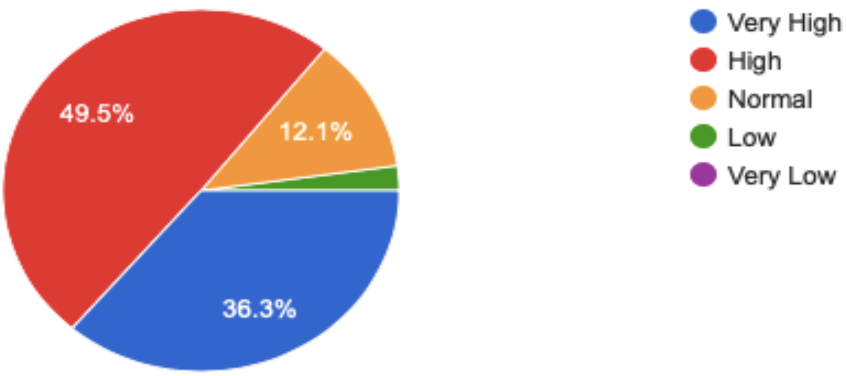


Figure 5.21: Effective Response to Change Requirements Result

The next influencing factor in this category is **effective response to change requirements** results. Out of 90 respondents, 36.3% participants have rated it as Very High while 49.5% have rated it as High influencing factor in this category as shown in Figure 5.21.

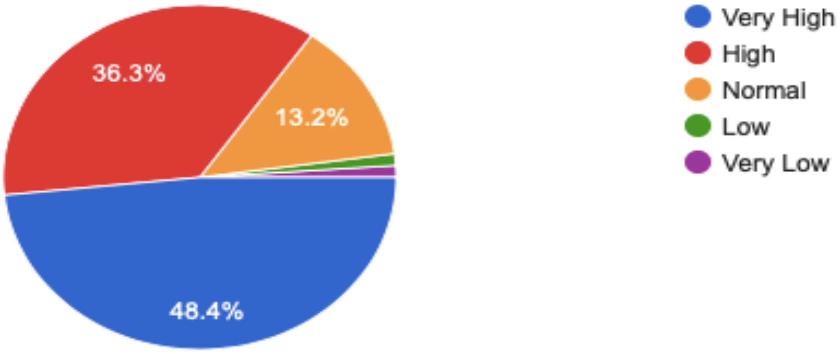


Figure 5.22: Fixed Costs result

The next influencing factor in this category is **fixed costs** results. Out of respondents, 48.4% participants have rated it as Very High while 36.3% have rated it as High influencing factor in this category as shown in Figure 5.22.

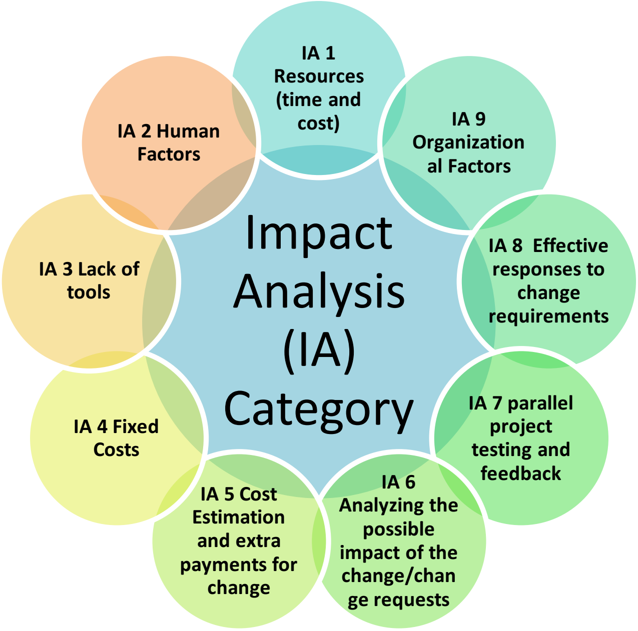

Impact Analysis (IA) Category
IA 1 Resources (time and cost)
IA 2 Human Factors
IA 3 Lack of tools
IA 4 Fixed Costs
IA 5 Cost Estimation and extra payments for change
IA 6 Analyzing the possible impact of the change/chan ge requests
IA 7 parallel project testing and feedback
IA 8 Effective responses to change requirements
IA 9 Organization al Factors

Figure 5.23: Prioritized Influencing Factors of Impact Analysis Category

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.23. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.23 represents the initials "IA" of the category Impact Analysis and numbers represent the priority level from high to level low for each influencing factor.

- **Overall Category Rating**: 71.07
- **Priority Number**: 2

5.2.6 Prioritization of Influencing Factors of Category 3: Management Role

There are 17 influencing factors in Management role category. Time schedule, project cost, project management, technology setup, trust building, roles and responsibilities, training and monitoring, skilled human resources, clear change management strategy, change management process awareness, conducting social events, clearly defined team roles, project leader, tools for communication results and overseas site response.

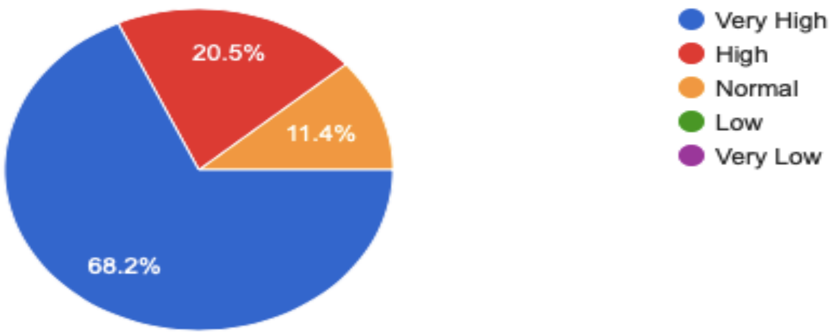


Figure 5.24: Time Schedule Result

The 1st influencing factor in this category is **time schedule**. Out of 90 respondents, 68.2% participants have rated it as Very High while 20.5% have rated it as High influencing factor in this category as shown in Figure 5.24.

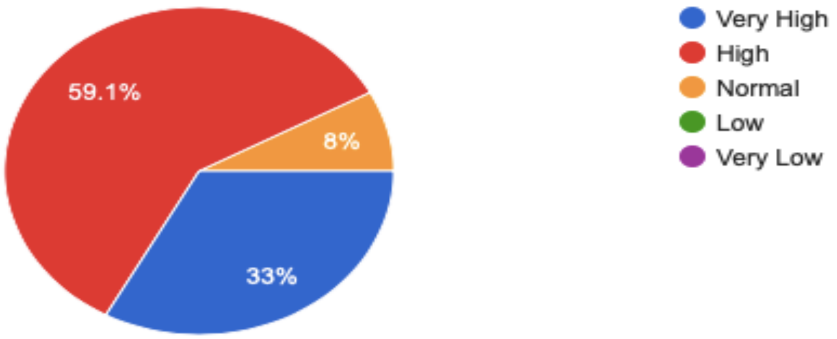


Figure 5.25: Project Cost result

The next influencing factor in this category is **project cost**. Out of 90 respondents, 33% participants have rated it as Very High while 59.1% have rated it as High influencing factor in this category as shown in Figure 5.25.

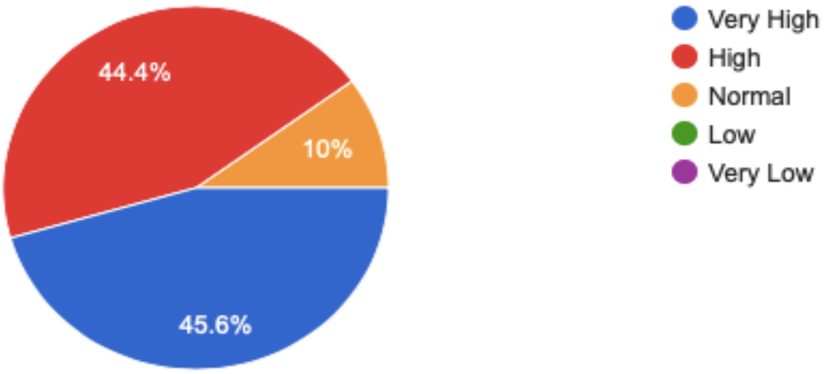


Figure 5.26: Project Management Result

The next influencing factor in this category is **project management**. Out of 90 respondents, 45.5% participants have rated it as Very High while 44.4% have rated it as High influencing factor in this category as shown in Figure 5.26.

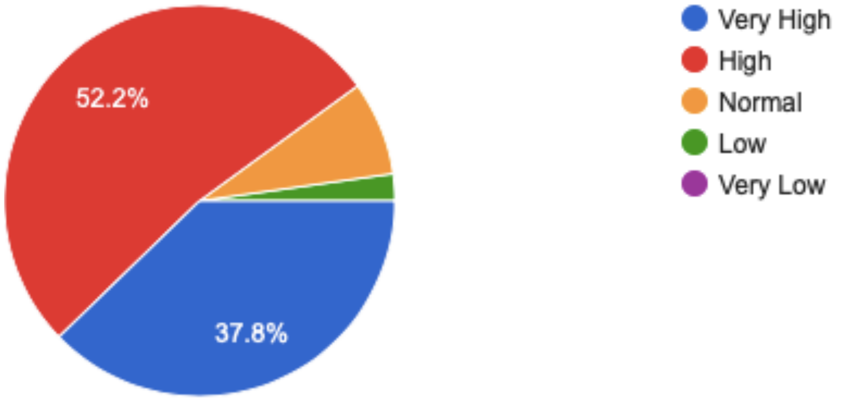


Figure 5.27: Technology Setup Result

The next influencing factor in this category is **technology setup**. Out of 90 respondents, 37.8% participants have rated it as Very High while 52.2% have rated it as High influencing factor in this category as shown in Figure 5.27.

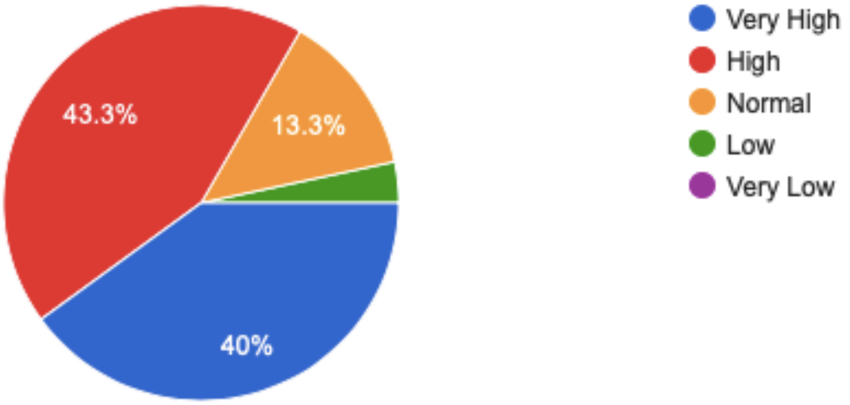


Figure 5.28: Trust Building Result

The next influencing factor in this category is trust building. Out of 90 respondents, 40% participants have rated it as Very High while 43.3% have rated it as High influencing factor in this category as shown in Figure 5.28.

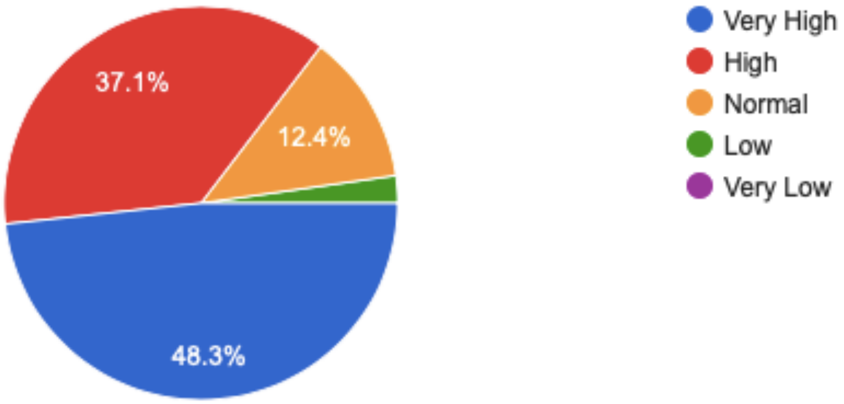


Figure 5.29: Roles and Responsibilities Result

The next influencing factor in this category is **roles and responsibilities**. Out of 90 respondents, 48.3% participants have rated it as Very High while 37.1% have rated it as High influencing factor in this category as shown in Figure 5.29.

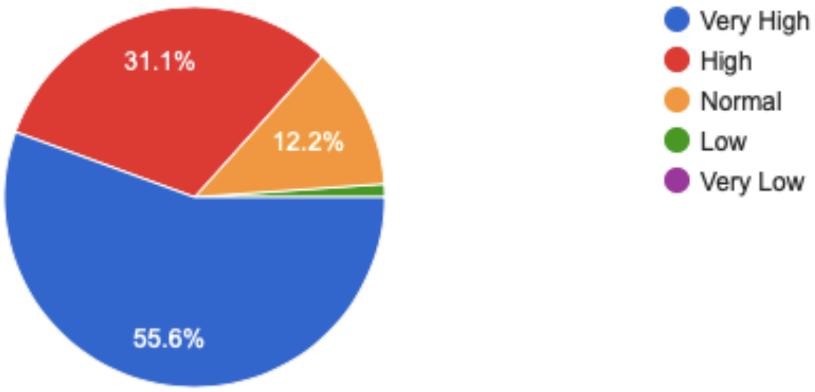


Figure 5.30: Training and Monitoring Result

The next influencing factor in this category is **training and monitoring results**. Out of 90 respondents, 55.6% participants have rated it as Very High while 31.1% have rated it as High influencing factor in this category as shown in Figure 5.30.

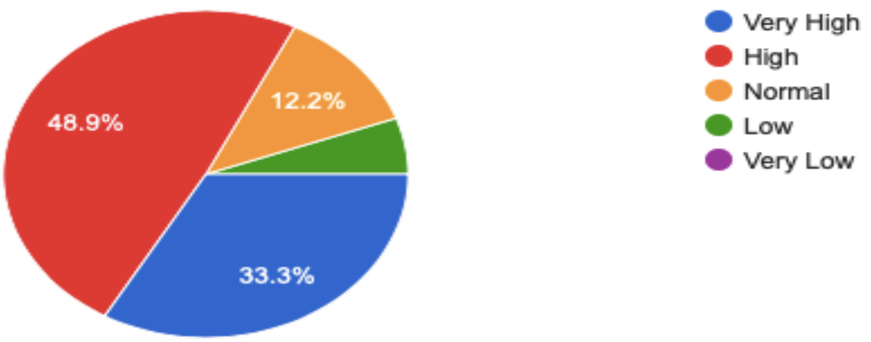


Figure 5.31: Resistance Management Result

The next influencing factor in this category is **resistance management**. Out of 90 respondents, 33.3% participants have rated it as Very High while 48.9% have rated it as High influencing factor in this category as shown in Figure 5.31.

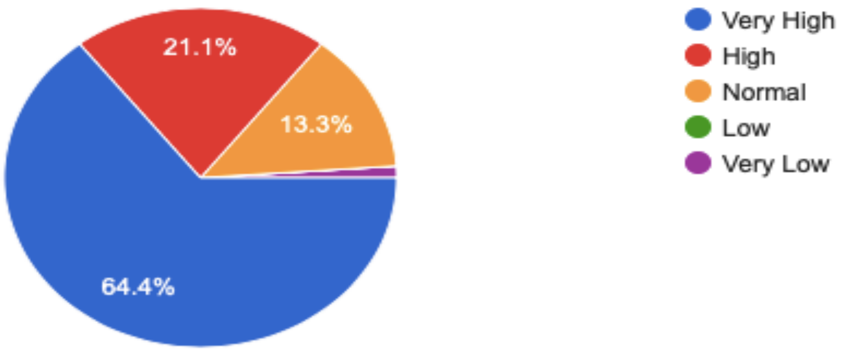


Figure 5.32: Skilled Human Resources Result

The next influencing factor in this category is **skilled human resource**. Out of 90 respondents, 64.4% participants have rated it as Very High while 21.1% have rated it as High influencing factor in this category as shown in Figure 5.32.

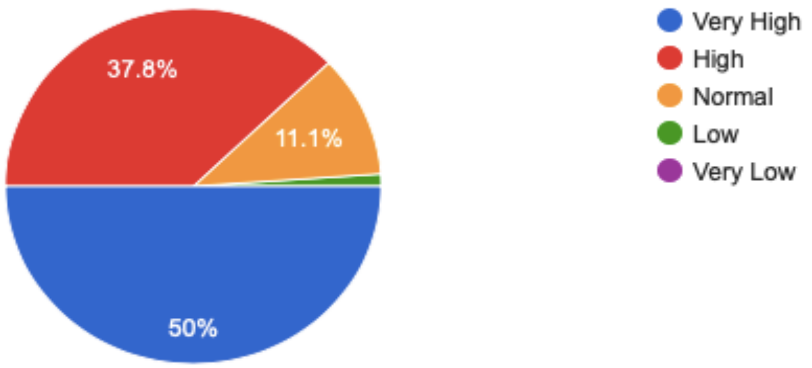


Figure 5.33: Clear Change Management Strategy Result

The next influencing factor in this category is **clear change management strategy**. Out of 90 respondents, 50% participants have rated it as Very High while 37.8% have rated it as High influencing factor in this category as shown in Figure 5.33.

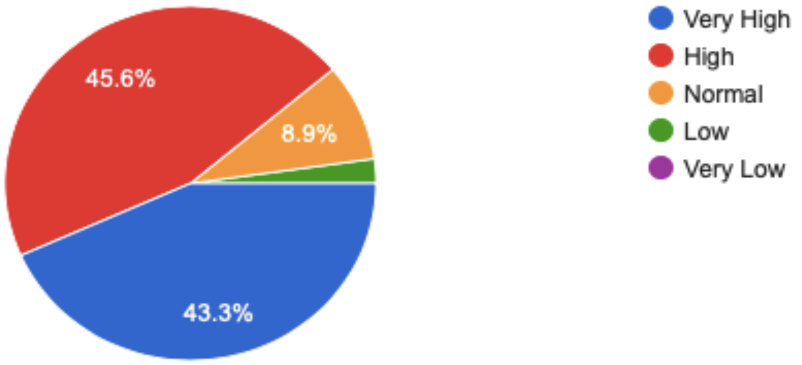


Figure 5.34: Change Management Process Awareness Result

The next influencing factor in this category is **change management process awareness**. Out of 90 respondents, 43.3% participants have rated it as Very High while 45.6% have rated it as High influencing factor in this category as shown in Figure 5.34.

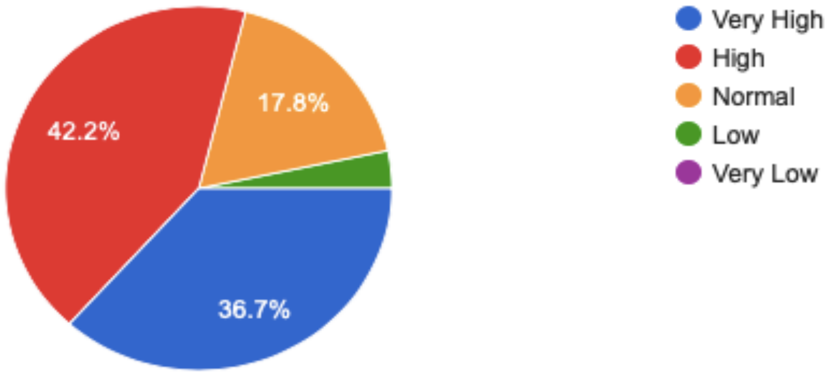


Figure 5.35: Conducting Social Events Result

The next influencing factor in this category is **conducting social events**. Out of 90 respondents, 36.7% participants have rated it as Very High while 42.2% have rated it as High influencing factor in this category as shown in Figure 5.35.

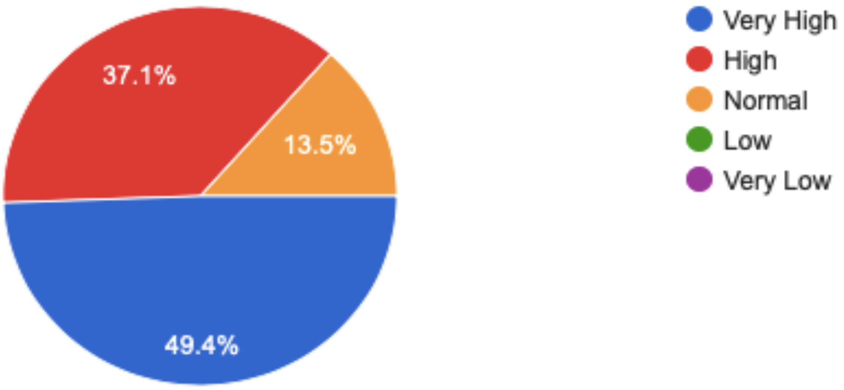


Figure 5.36: Project Leader Result

The next influencing factor in this category is **project leader**. Out of 90 respondents, 49.4% participants have rated it as Very High while 37.1% have rated it as High influencing factor in this category as shown in Figure 5.36.

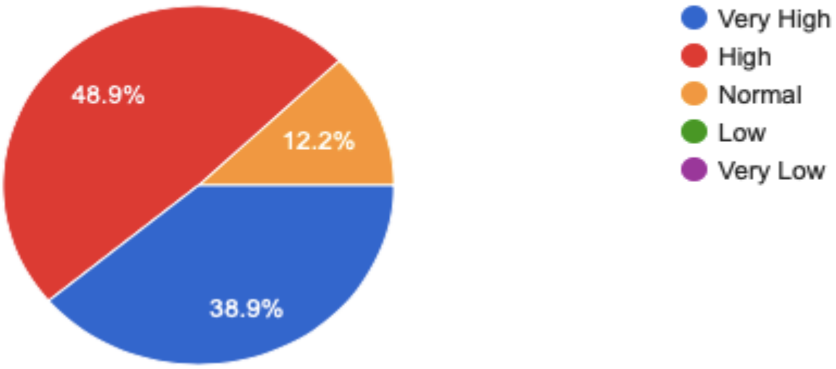


Figure 5.37: Clearly defined Team Roles Result

The next influencing factor in this category is **clearly defined team roles**. Out of 90 respondents, 38.9% participants have rated it as Very High while 48.9% have rated it as High influencing factor in this category as shown in Figure 5.37.

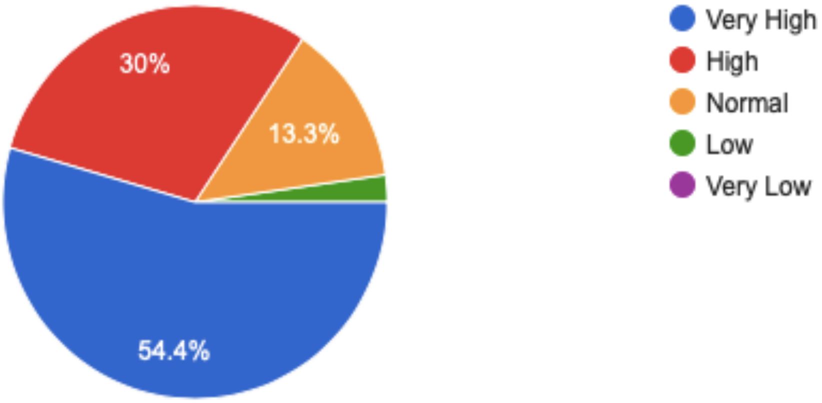


Figure 5.38: Tools for Communication Result

The next influencing factor in this category is **tools for communication** results. Out of 90 respondents, 54.4% participants have rated it as Very High while 30% have rated it as High influencing factor in this category as shown in Figure 5.38.

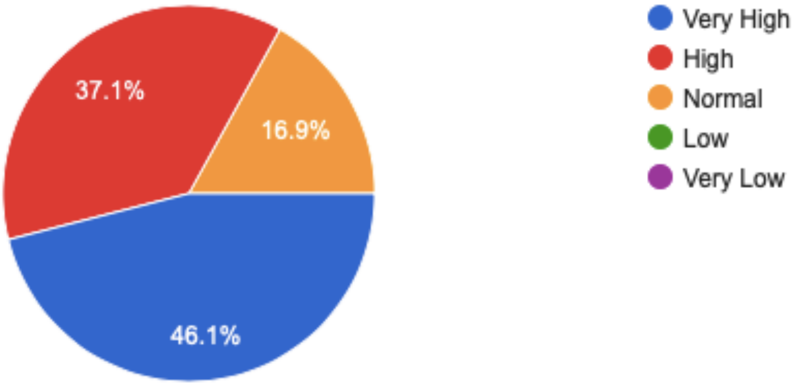


Figure 5.39: Project Administration Result

The next influencing factor in this category is **project administration**. Out of 90 respondents, 46.1% participants have rated it as Very High while 37.1% have rated it as High influencing factor in this category as shown in Figure 5.39.

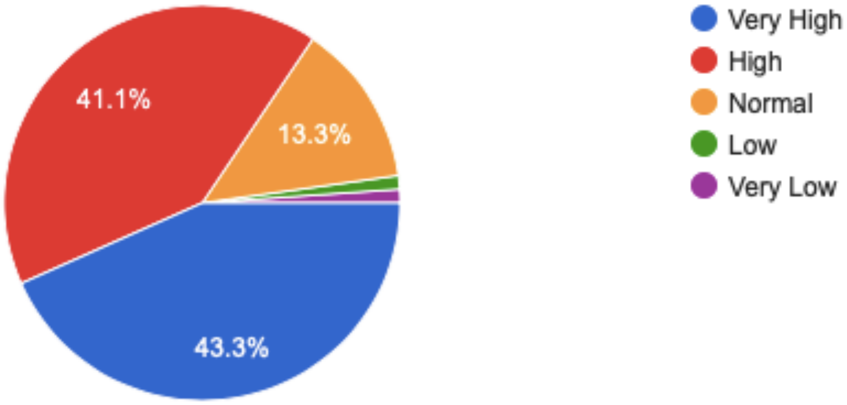


Figure 5.40: Overseas Site Response Result

The next influencing factor in this category is **overseas site response**. Out of 90 respondents, 43.3% participants have rated it as Very High while 41.1% have rated it as High influencing factor in this category as shown in Figure 5.40.

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.41. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.41 represents the initials "MR" of the category Management Role and numbers represent the priority level from high to level low for each influencing factor.

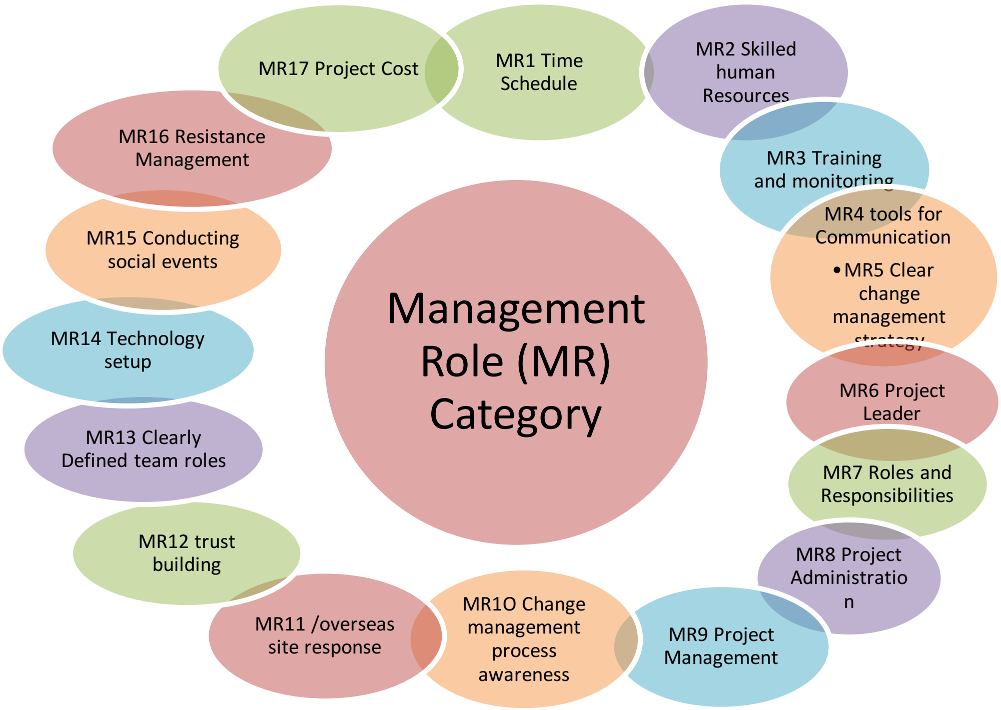

Management Role (MR) Category
MR1 Time Schedule
MR2 Skilled human Resources
MR3 Training and monitorting
MR4 tools for Communication
•MR5 Clear change management strategy
MR6 Project Leader
MR7 Roles and Responsibilities
MR8 Project Administratio n
MR9 Project Management
MR1O Change management process awareness
MR11 /overseas site response
MR12 trust building
MR13 Clearly Defined team roles
MR14 Technology setup
MR15 Conducting social events
MR16 Resistance Management
MR17 Project Cost

Figure 5.41: Prioritized Influencing Factors of Management Role Category

- **Overall Category Rating**: 69.5
- **Priority Number**: 4

5.2.7 Prioritization of Influencing Factors of Category 4: Team Role

There are 15 influencing factors in team and role category are as follows: software development life cycle, group awareness, customer involvement and interaction, formalized relationship between development teams, requirement traceability, team configuration, team coordination, was governance and control of RCM activities, standard and process for RCM, change identification and validation, synchronization of work at different sites, support quick and flexible response to change, choosing the right development method, importance of individuals and their interactions and early delivery.

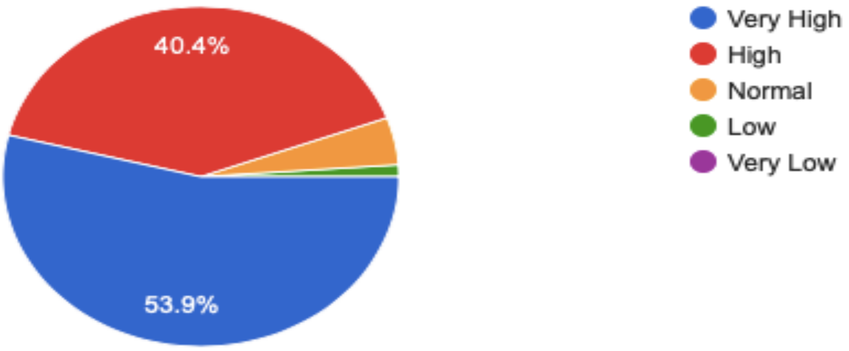


Figure 5.42: SDLC Result

The 1st influencing factor in this category is **software development life cycle**. Out of 90 respondents, 53.9% participants have rated it as Very High while 40.4% have rated it as High influencing factor in this category as shown in Figure 5.42.

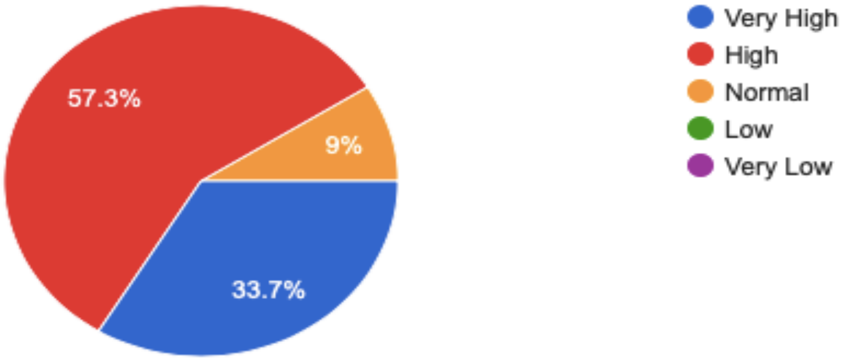


Figure 5.43: Group Awareness Result

The next influencing factor in this category is **Group awareness**. Out of 90 respondents, 33.7% participants have rated it as Very High while 57.3% have rated it as High influencing factor in this category as shown in Figure 5.43.

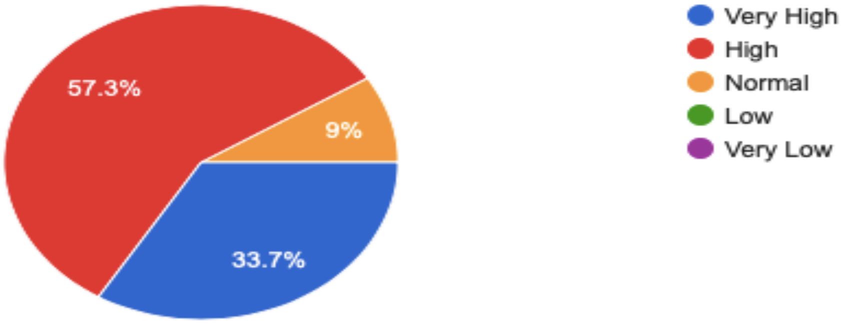


Figure 5.44: Customer Involvement and Interaction Result

The next influencing factor in this category is **customer involvement and interaction**. Out of 90 respondents, 50% participants have rated it as Very High while 34.4% have rated it as High influencing factor in this category as shown in Figure 5.44.

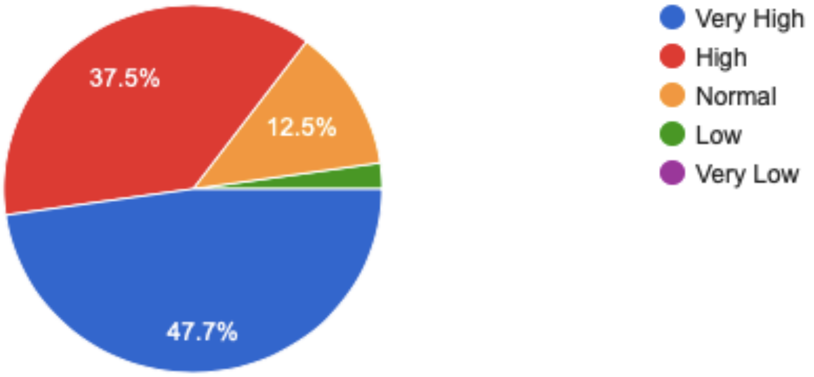


Figure 5.45: Formalized Relationship between Development Teams Result

The next influencing factor in this category is **formalized relationship between development teams**. Out of 90 respondents, 47.7% participants have rated it as Very High while 37.5% have rated it as High influencing factor in this category as shown in Figure 5.45.

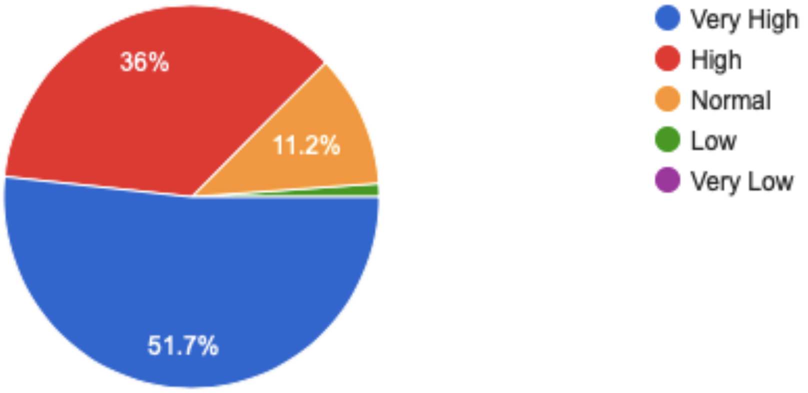


Figure 5.46: Requirement Traceability Result

The next influencing factor in this category is **requirement traceability**. Out of 90 respondents, 51.7% participants have rated it as Very High while 36% have rated it as High influencing factor in this category as shown in Figure 5.46.

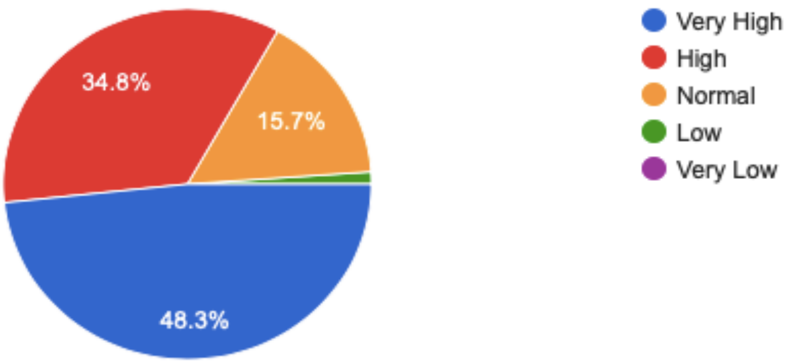


Figure 5.47: Team Configuration result

The next influencing factor in this category is **team configuration**. Out of 90 respondents, 48.3% participants have rated it as Very High while 34.8% have rated it as High influencing factor in this category as shown in Figure 5.47.

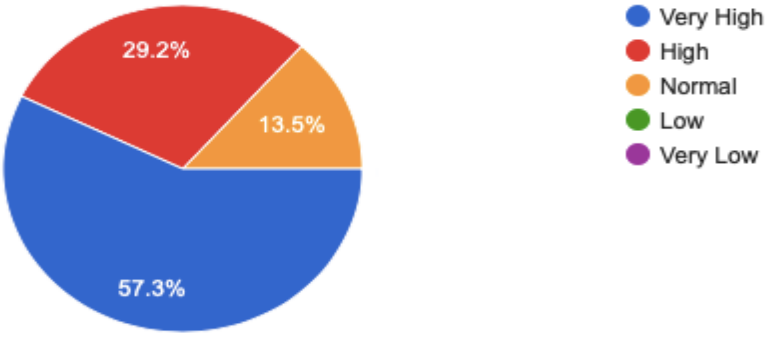


Figure 5.48: Team Coordination Result

The next influencing factor in this category is **team coordination**. Out of 90 respondents, 57.3% participants have rated it as Very High while 29.2% have rated it as High influencing factor in this category as shown in Figure 5.48.

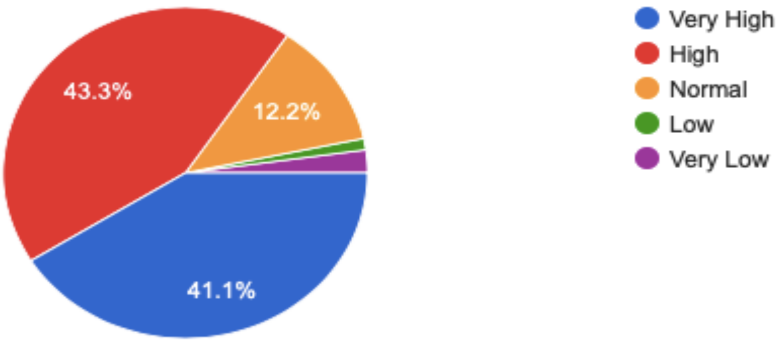


Figure 5.49: Governance and Control of RCM Activities Result

The next influencing factor in this category is **governance and control of RCM activities**. Out of 90 respondents, 41.1% participants have rated it as Very High while 43.3% have rated it as High influencing factor in this category as shown in Figure 5.49.

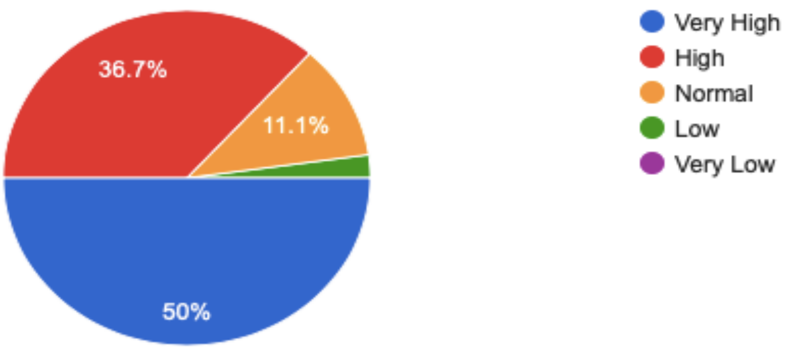


Figure 5.50: Standard and Process of RCM Result

The next influencing factor in this category is **standard and process for RCM**. Out of 90 respondents, 50% participants have rated it as Very High while 36.7% have rated it as High influencing factor in this category as shown in Figure 5.50.

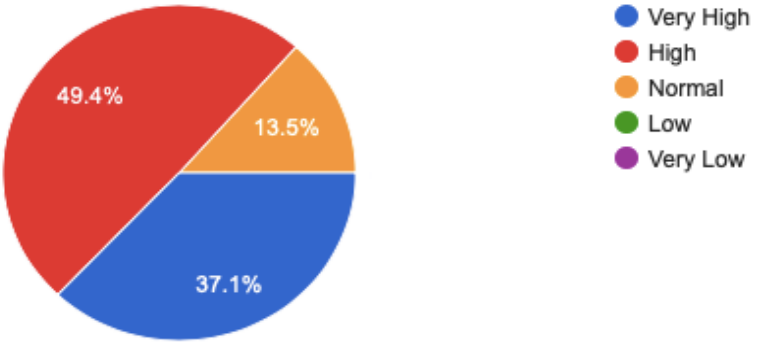


Figure 5.51: Change Identification and Validation Result

The next influencing factor in this category is **change identification and validation**. Out of 90 respondents, 37.1% participants have rated it as Very High while 49.4% have rated it as High influencing factor in this category as shown in Figure 5.51.

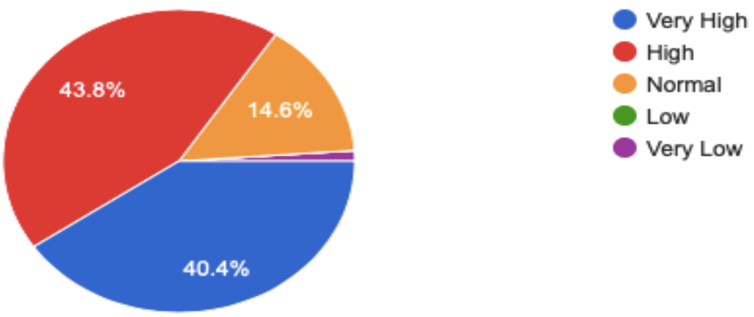


Figure 5.52: Sync of Work at Diff Sites Result

The next influencing factor in this category is **synchronization of work at different sites**. Out of 90 respondents, 40.4% participants have rated it as Very High while 43.8% have rated it as High influencing factor in this category as shown in Figure 5.52.

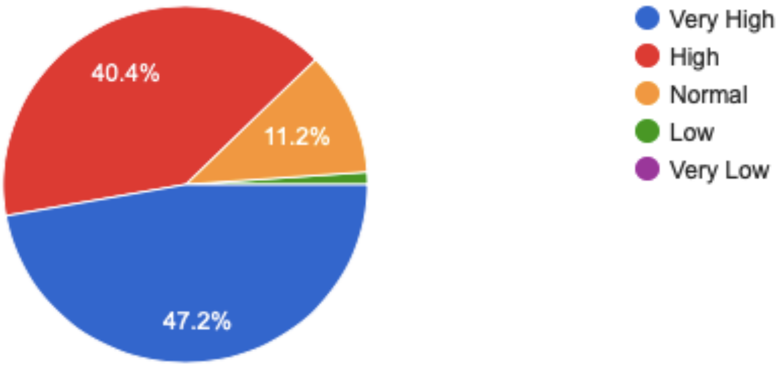


Figure 5.53: Support Quick and Flexible Response to Change Result

The next influencing factor in this category is **support quick and flexible response to change**. Out of 90 respondents, 47.2% participants have rated it as Very High while 40.4% have rated it as High influencing factor in this category as shown in Figure 5.53.

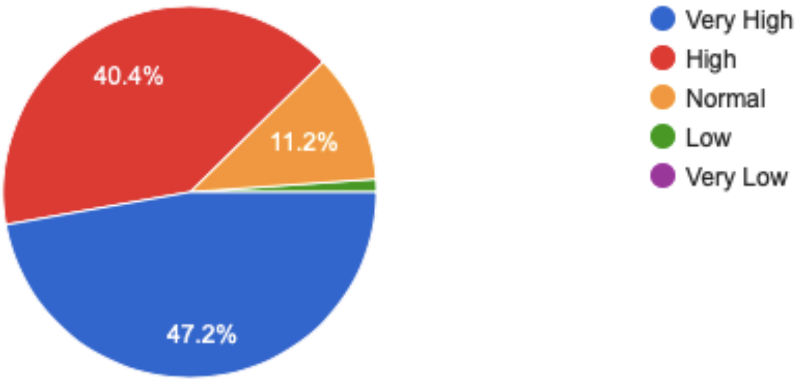


Figure 5.54: Choosing the Right Dev Method Result

The next influencing factor in this category is **choosing the right development method**. Out of 90 respondents, 44.9% participants have rated it as Very High while 42.7% have rated it as High influencing factor in this category as shown in Figure 5.54.

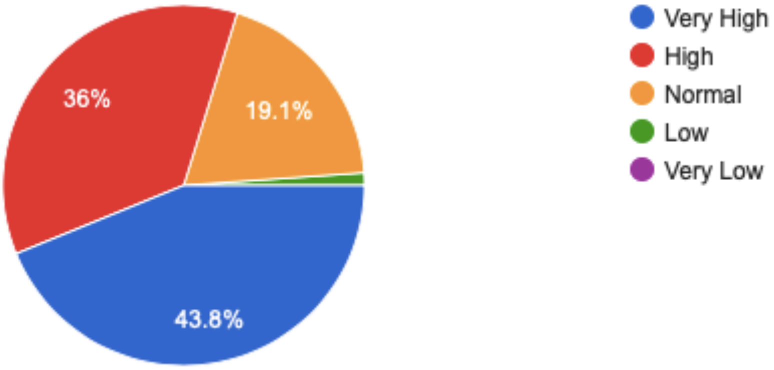


Figure 5.55: Imp of Individuals and Their Interactions Result

The next influencing factor in this category is **importance of individuals and their interactions.** Out of 90 respondents, 43.8% participants have rated it as Very High while 36% have rated it as High influencing factor in this category as shown in Figure 5.55.

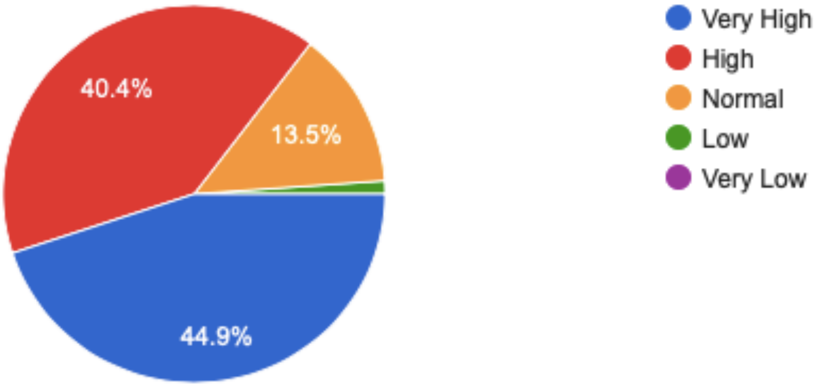


Figure 5.56: Early Delivery Result

The next influencing factor in this category is **early delivery**. 44.9% participants have rated it as Very High while 40.4% have rated it as High influencing factor in this category as shown in Figure 5.56.

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.57. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.57 represents the initials "TR" of the category Team Role and numbers represent the priority level from high to level low for each influencing factor.

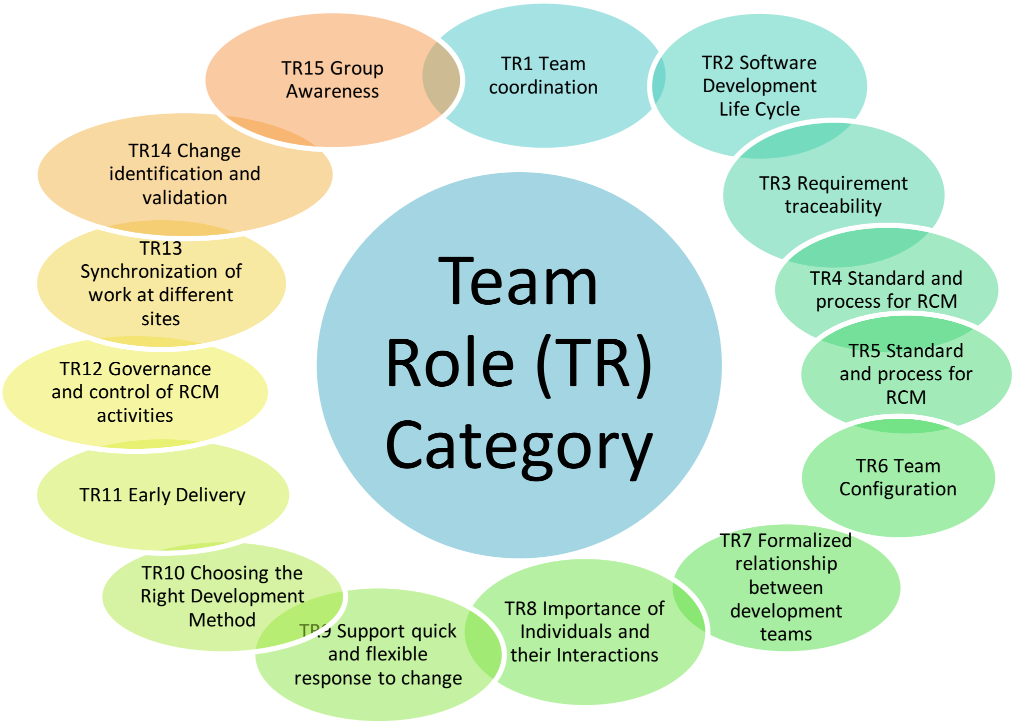

Team Role (TR) Category
TR1 Team coordination
TR2 Software Development Life Cycle
TR3 Requirement traceability
TR4 Standard and process for RCM
TR5 Standard and process for RCM
TR6 Team Configuration
TR7 Formalized relationship between development teams
TR8 Importance of Individuals and their Interactions
TR9 Support quick and flexible response to change
TR10 Choosing the Right Development Method
TR11 Early Delivery
TR12 Governance and control of RCM activities
TR13 Synchronization of work at different sites
TR14 Change identification and validation
TR15 Group Awareness

Figure 5.57: Prioritized Influencing Factors of Team Role Category

- **Overall Category Rating**: 69.2
- **Priority Number**: 5

5.2.8 Prioritization of Influencing Factors of Category 5: Communication

There are 8 influencing factors in communication category are as follow: face to face communication, customer communication, effective communication, geographical distributed change control block, documentation of requirements, communication in terms with large time difference, speed of communication and ease of communication.

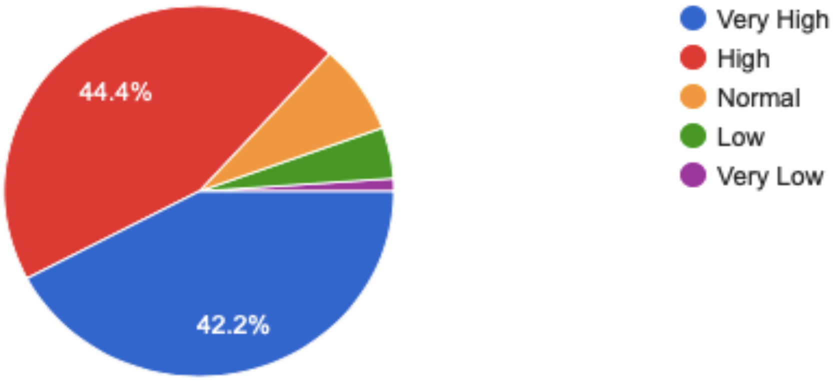


Figure 5.58: F2F Communication Result

The next influencing factor in this category is **face to face communication**. Out of 90 respondents, 42.2% participants have rated it as Very High while 44.4% have rated it as High influencing factor in this category as shown in Figure 5.58.

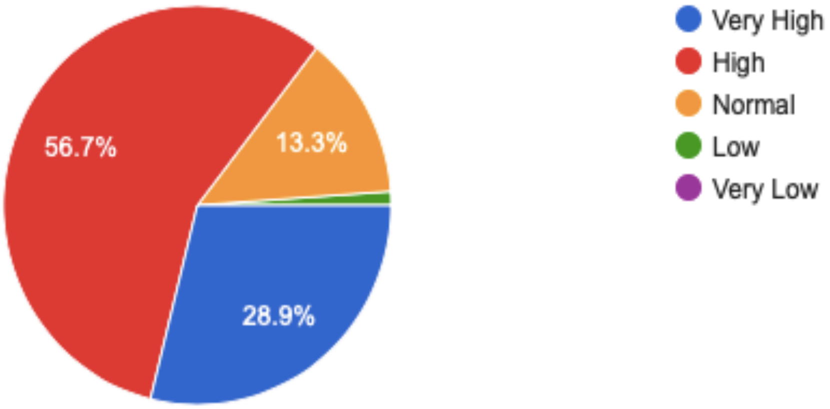


Figure 5.59: Customer Communication Result

The next influencing factor in this category is **customer communication**. Out of 90 respondents, 28.9% participants have rated it as Very High while 56.7% have rated it as High influencing factor in this category as shown in Figure 5.59.

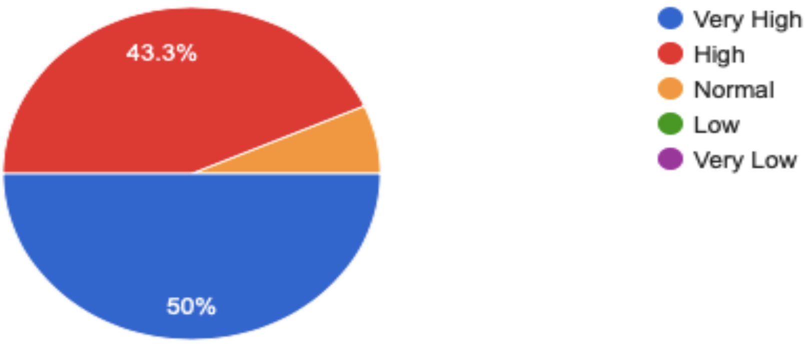


Figure 5.60: Effective Communication Result

The next influencing factor in this category is **effective communication**. Out of 90 respondents, 50% participants have rated it as Very High while 43.3% have rated it as High influencing factor in this category as shown in Figure 5.60.

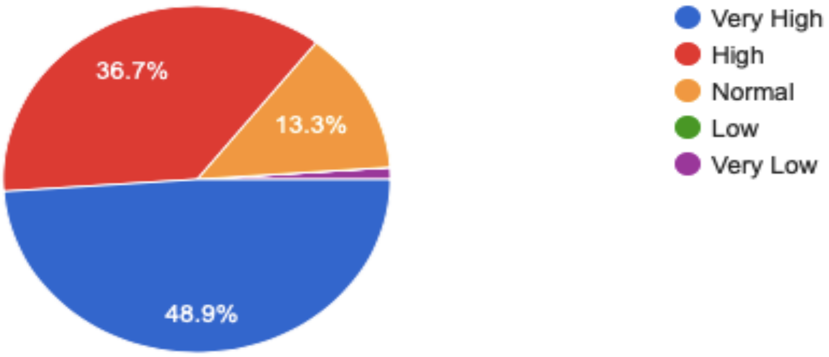


Figure 5.61: Geographically Distributed Change Control Block Result

The next influencing factor in this category is **geographical distributed change control block**. Out of 90 respondents, 48.9% participants have rated it as Very High while 36.7% have rated it as High influencing factor in this category as shown in Figure 5.61.

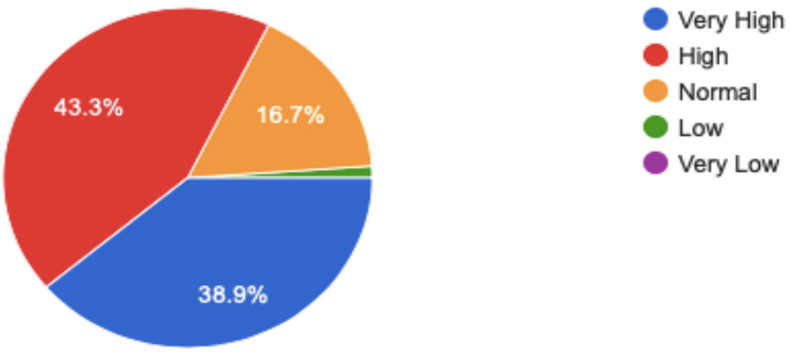


Figure 5.62: Communication in Team with Large Time Diff Result

The next influencing factor in this category is **communication in terms with large time difference**. Out of 90 respondents, 38.9% participants have rated it as Very High while 43.3% have rated it as High influencing factor in this category as shown in Figure 5.62.

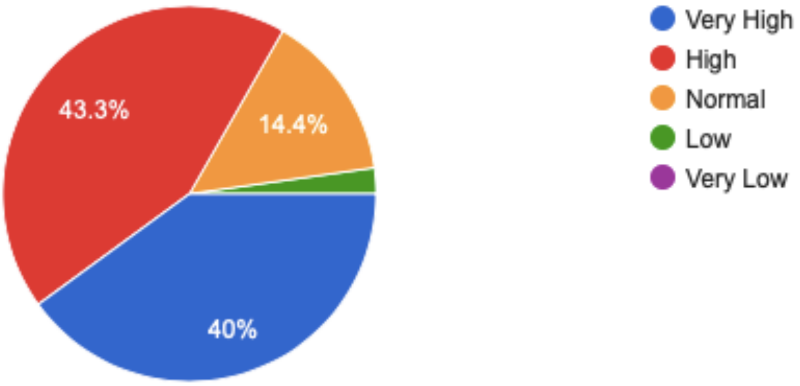


Figure 5.63: Doc of Requirements Result

The next influencing factor in this category is **documentation of requirements**. Out of 90 respondents, 40% participants have rated it as Very High while 43.3% have rated it as High influencing factor in this category as shown in Figure 5.63.

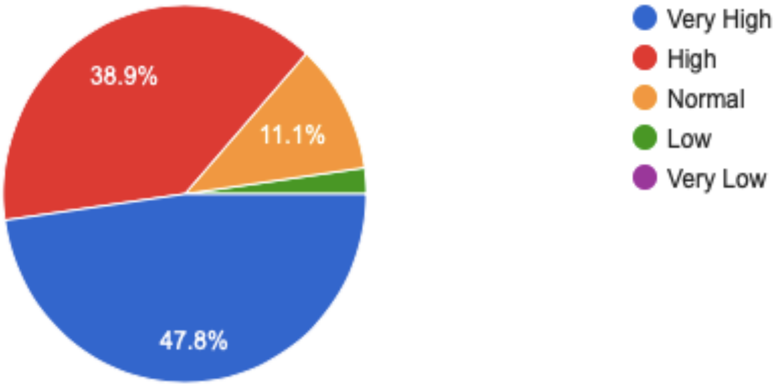


Figure 5.64: Speed of Communication Result

The next influencing factor in this category is **speed of communication**. Out of 90 respondents, 47.8% participants have rated it as Very High while 38.9% have rated it as High influencing factor in this category as shown in Figure 5.64.

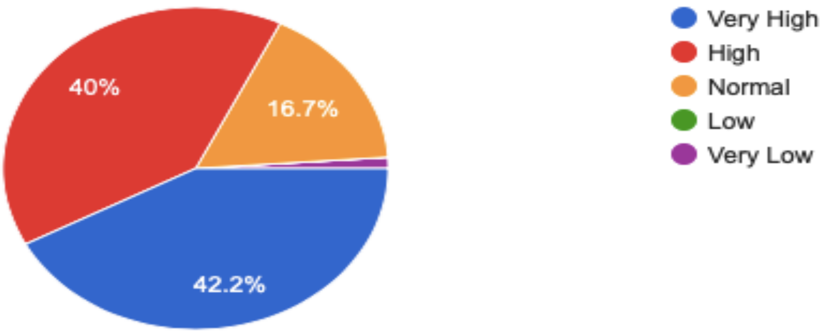


Figure 5.65: Ease of Communication Result

The next influencing factor in this category is **ease of communication**. Out of 90 respondents, 42.2% participants have rated it as Very High while 40% have rated it as High influencing factor in this category as shown in Figure 5.65.

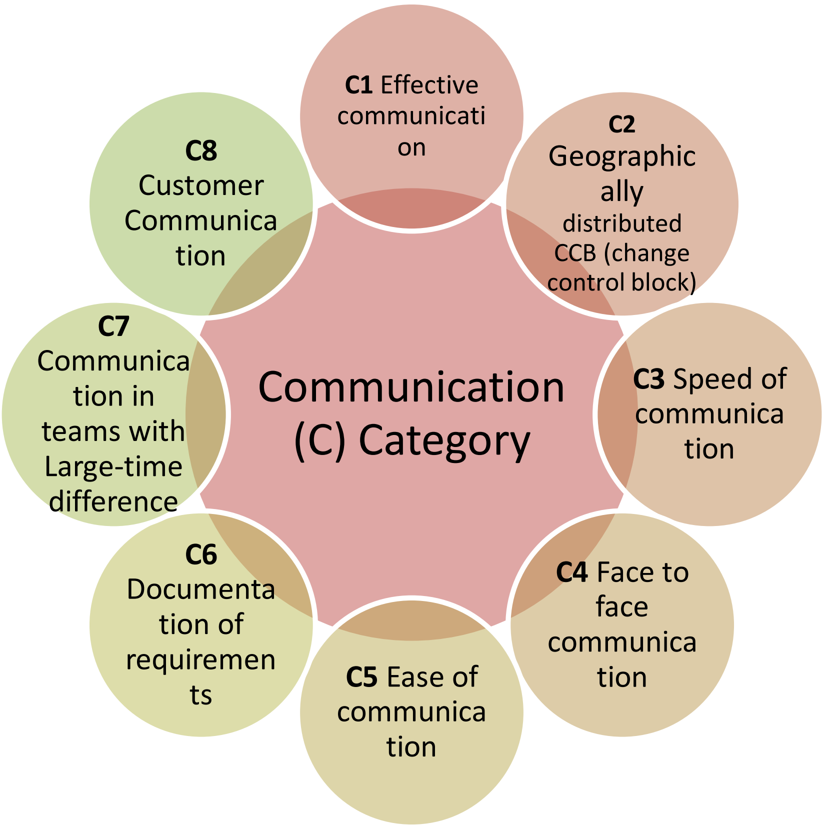

Communication (C) Category
C1 Effective communication
C2 Geographically distributed CCB (change control block)
C3 Speed of communication
C4 Face to face communication
C5 Ease of communication
C6 Documentation of requirements
C7 Communication in teams with Large-time difference
C8 Customer Communication

Figure 5.66: Prioritized Influencing Factors of Communication Category

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.66. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.66 represents the initials "C" of the category Communication and numbers represent the priority level from high to level low for each influencing factor.

- **Overall Category Rating**: 63.54
- **Priority Number**: 6

5.2.9 Prioritization of Influencing Factors of Category 6: Progress Measure

There are 9 influencing factors in process measure category are as follow: skilled developers, continuous organization support, IT platforms, HR management, welcome requirements change, even at the end of development, positive team role, rapid response to change in requirements rather than following a prescribed plan and collaboration between the software developer and user.

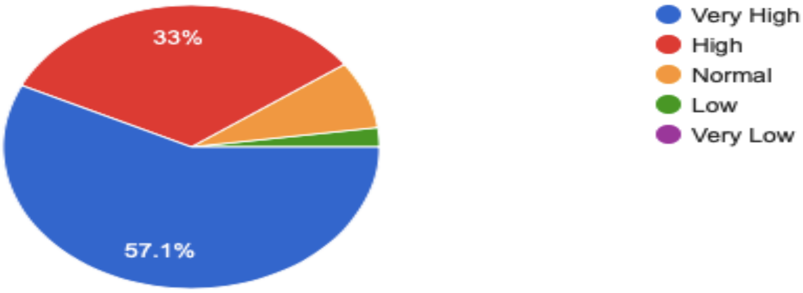


Figure 5.67: Skilled Developers Result

The next influencing factor in this category is **skilled developers**. Out of 90 respondents, 57.1% participants have rated it as Very High while 33% have rated it as High influencing factor in this category as shown in Figure 5.67.

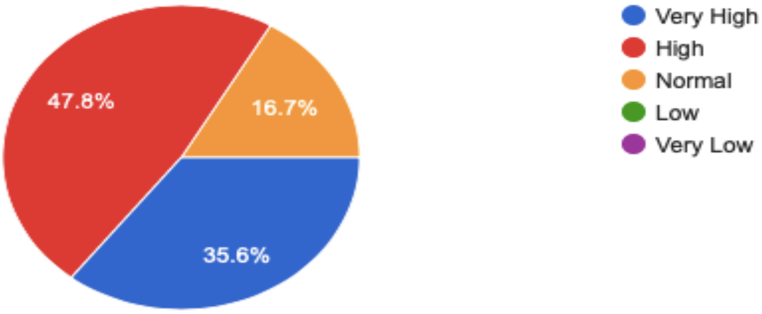


Figure 5.68: Continuous Organizational Support Result

The next influencing factor in this category is **continuous organizational support**. Out of 90 respondents, 35.6% participants have rated it as Very High while 47.8% have rated it as High influencing factor in this category as shown in Figure 5.68.

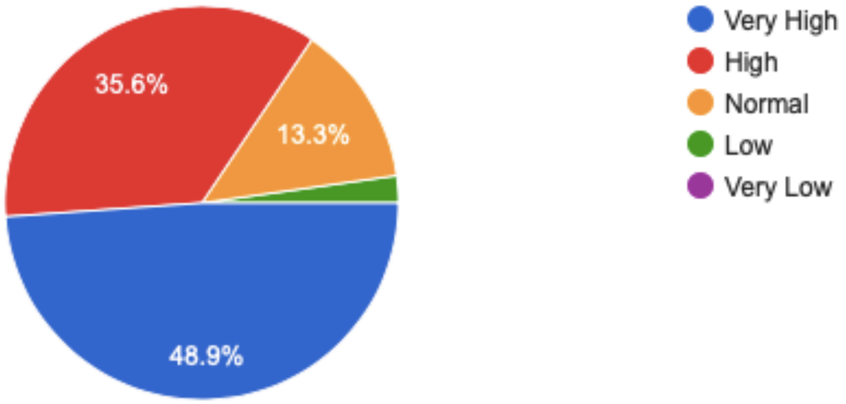


Figure 5.69: IT Platforms Result

The next influencing factor in this category is **IT platforms**. Out of 90 respondents, 48.9% participants have rated it as Very High while 35.6% have rated it as High influencing factor in this category as shown in Figure 5.69.

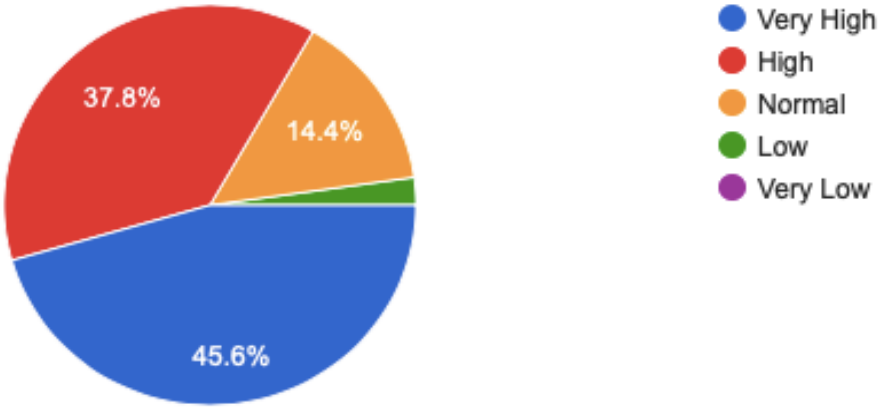


Figure 5.70: HR Management Results

The next influencing factor in this category is **HR management**. Out of 90 respondents,45.6% participants have rated it as Very High while 37.8% have rated it as High influencing factor in this category as shown in Figure 5.70.

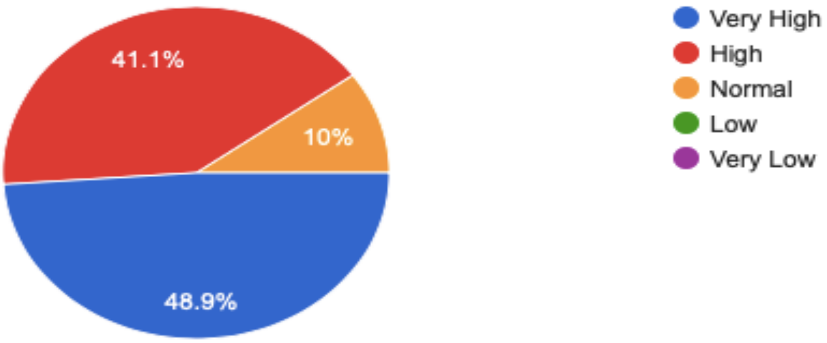


Figure 5.71: Strong Leadership Result

The next influencing factor in this category is **strong leadership**. Out of 90 respondents, 48.9% participants have rated it as Very High while 41.1% have rated it as High influencing factor in this category as shown in Figure 5.71.

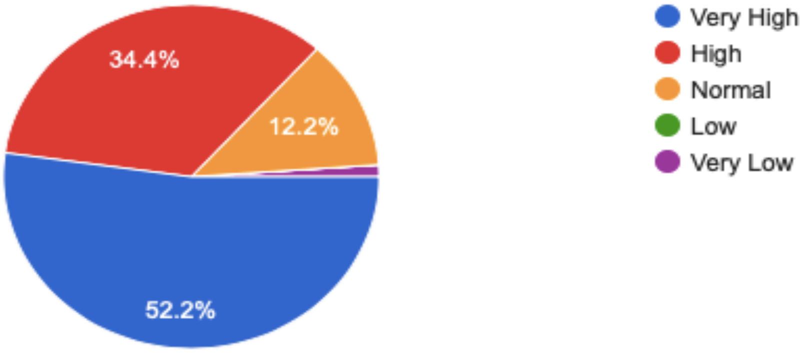


Figure 5.72: Welcome Requirement Change Even at the End of Dev Result

The next influencing factor in this category is **welcome requirements change, even at the end of development**. Out of 90 respondents, 52.2% participants have rated it as Very High while 34.4% have rated it as High influencing factor in this category as shown in Figure 5.72.

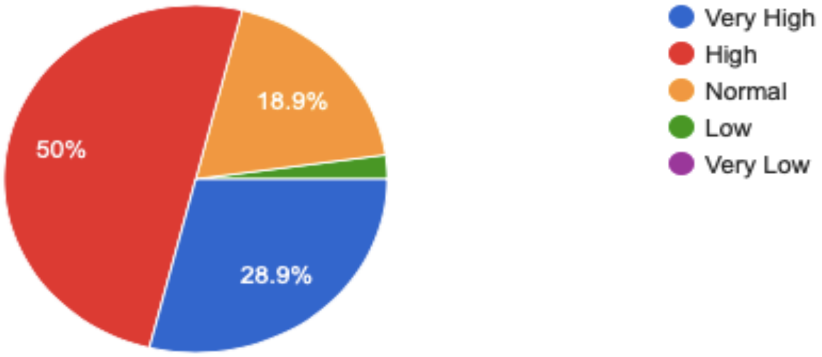


Figure 5.73: Positive Team Role Result

The next influencing factor in this category is **positive team role**. Out of 90 respondents, 28.9% participants have rated it as Very High while 50% have rated it as High influencing factor in this category as shown in Figure 5.73.

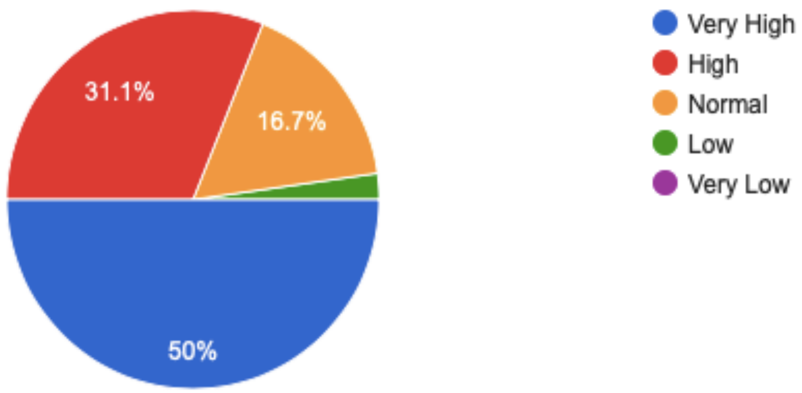


Figure 5.74: Rapid Response to Change in Requirements rather than following a Prescribed Plan Result

The next influencing factor in this category is **rapid response to change in requirements rather than following a prescribed plan**. Out of 90 respondents, 50% participants have rated it as Very High while 31.1% have rated it as High influencing factor in this category as shown in Figure 5.74.

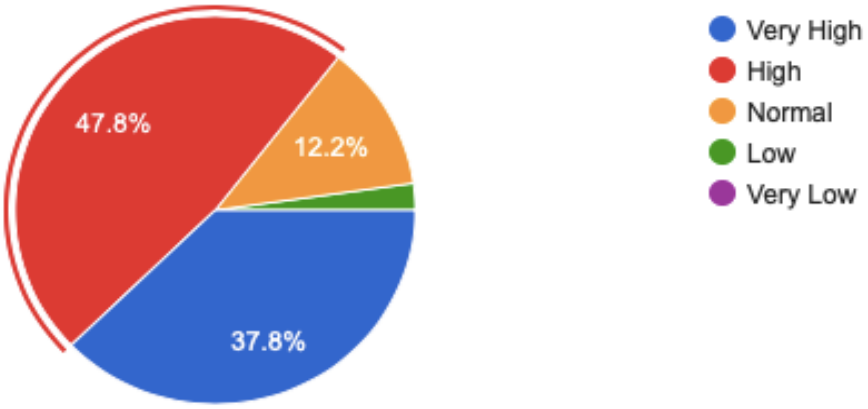


Figure 5.75: Collaboration between the Software Dev and the User Result

The next influencing factor in this category is **collaboration between the software developer and user**. 37.8% participants have rated it as Very High while 47.8% have rated it as High influencing factor in this category as shown in Figure 5.75.

Based on the results of each influencing factor's rating, a prioritized list is generated for this category which is shown in Figure 5.76. Each influencing factor has been given a new Id with initials of its category and its priority number in the list. This is done for the easy tracking of each influencing factor. For example, the column "New Id" in the Figure 5.76 represents the initials "PM" of the category Progress Measure and numbers represent the priority level from high to level low for each influencing factor.

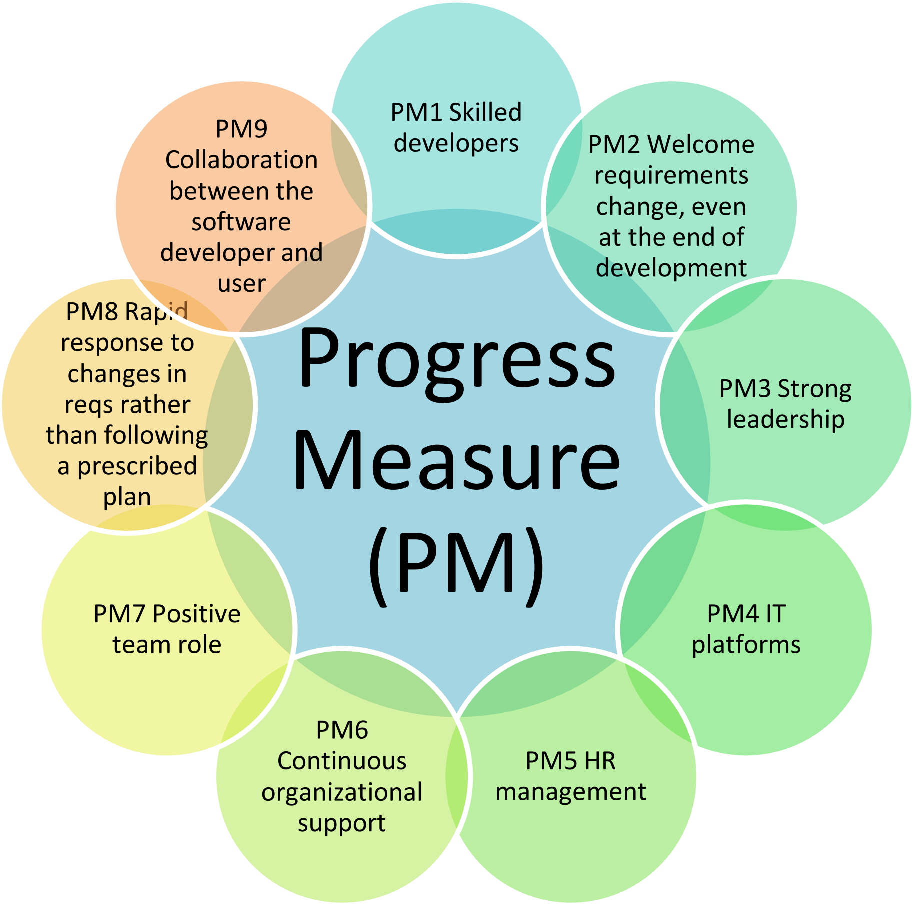

Progress Measure (PM)
PM1 Skilled developers
PM2 Welcome requirements change, even at the end of development
PM3 Strong leadership
PM4 IT platforms
PM5 HR management
PM6 Continuous organizational support
PM7 Positive team role
PM8 Rapid response to changes in reqs rather than following a prescribed plan
PM9 Collaboration between the software developer and user

Figure 5.76: Prioritized Influencing factors of Progress Measure Category

- **Overall Category Rating**: 75.5
- **Priority Number**: 1

After the categories are prioritized and a prioritized list of influencing factors for each category is developed, finally a complete prioritized list of all categories and their respective prioritized influencing factors is obtained which is illustrated in Table 5.1.

Table 5.1: Final Prioritized List of Categories and their Prioritized Influencing Factors

Unique ID	Prioritized Influencing Factors
CATEGORY 1: PROGRESS MEASURE	
PM 1	Skilled developers
PM 2	Welcome requirements change, even at the end of development
PM 3	Strong leadership
PM 4	IT platforms
PM 5	HR management
PM 6	Continuous organizational support
PM 7	Positive team role
PM 8	Rapid response to changes in requirements rather than following a prescribed plan
PM 9	Collaboration between the software developer and user
CATEGORY 2: IMPACT ANALYSIS	
IA 1	Resources (Time and cost)
IA 2	Human Factors
IA 3	Lack of Tools
IA 4	Fixed Costs
IA 5	Cost Estimation and extra payment for changes
IA 6	Analyzing the Possible Impact of the Change/ change request
IA 7	Parallel project testing and feedback
IA 8	Effective responses to change requirements
IA 9	Organizational Factors
CATEGORY 3: UNDERSTANDING REQUIREMENTS CHANGE	
URC 1	Vision and Goal for change
URC 2	Enhancement requests and Content changes
URC 3	Knowledge Management
URC 4	Reporting the change
URC 5	Implementing the change
URC 6	Different geographical locations of the development team
URC 7	Managing Requirement Change
URC 8	Responding to change over following a plan
URC 9	External Stakeholder Collaboration
CATEGORY 4: MANAGEMENT ROLE	
MR 1	Time Schedule
MR 2	Skilled human resources
MR 3	Training and monitoring
MR 4	Tools for Communication
MR 5	Clear change management strategy
MR 6	Project leader
MR 7	Roles and responsibilities
MR 8	Project Administration
MR 9	Project Management
MR 10	Change management process awareness
MR 11	Overseas site response
MR 12	Trust building

MR 13	Clearly defined team roles
MR 14	Technology Setup
MR 15	Conducting social events
MR 16	Resistance management
MR 17	Project Cost
CATEGORY 5: TEAM ROLE	
TR 1	Team coordination
TR 2	Software Development Life Cycle
TR 3	Requirement traceability
TR 4	Standard and process for RCM
TR 5	Customer Involvement and Interaction
TR 6	Team Configuration
TR 7	Formalized relationship between development teams
TR 8	Importance of Individuals and their Interactions
TR 9	Support quick and flexible response to change
TR 10	Choosing the Right Development Method
TR 11	Early Delivery
TR 12	Governance and control of RCM activities
TR 13	Synchronization of work at different sites
TR 14	Change identification and validation
TR 15	Group Awareness
CATEGORY 6: COMMUNICATION	
C 1	Effective communication
C 2	Geographically distributed CCB (change control block)
C 3	Speed of communication
C 4	Face to face communication
C 5	Ease of communication
C 6	Documentation of requirements
C 7	Communication in teams with Large time difference
C 8	Customer Communication

5.3 Framework for RCM in DAD

The final step of this research is to propose a framework based on the findings of SLR and Survey to efficiently conduct RCM process in DAD environment. The Proposed framework is shown in Figure 5.76.

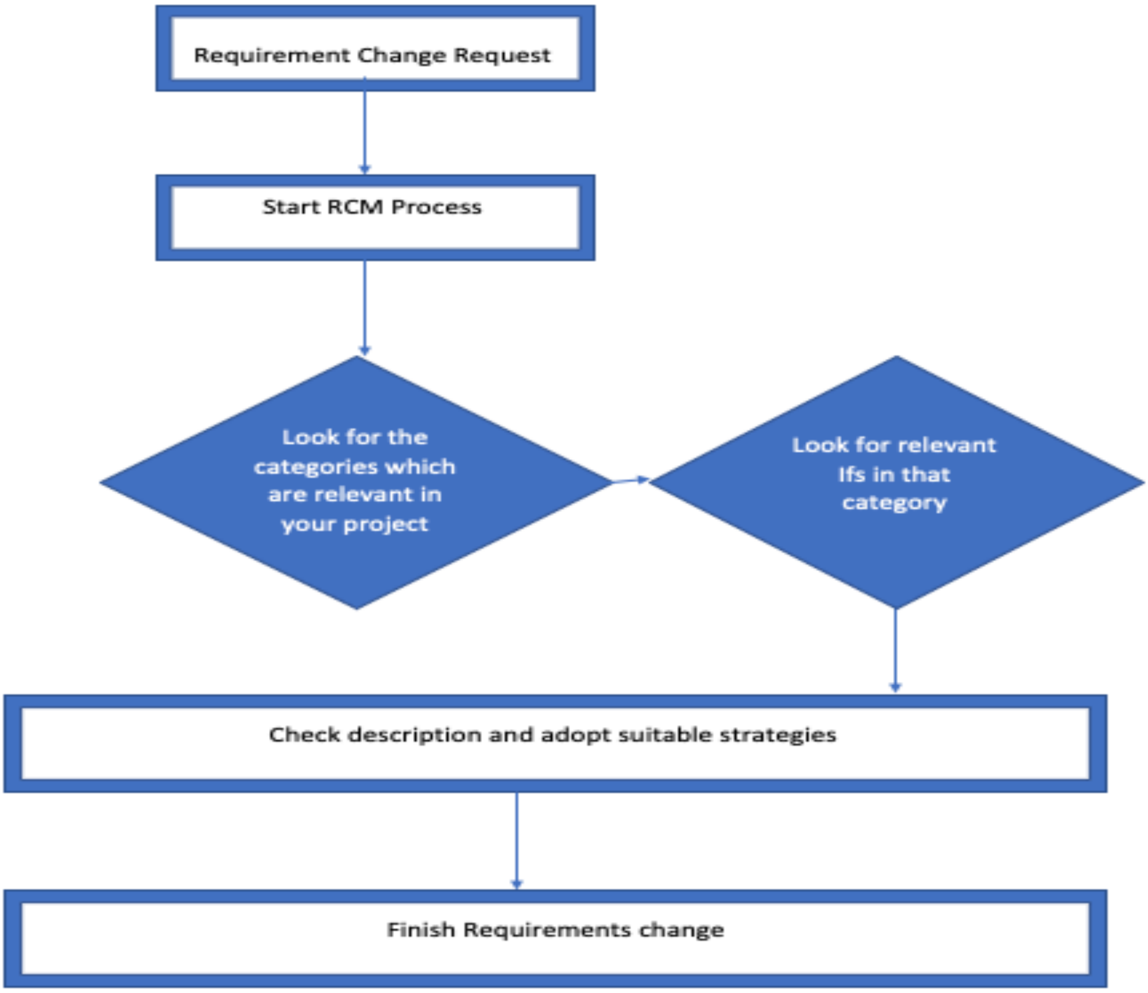

Requirement Change Request
Start RCM Process
Look for the categories which are relevant in your project
Look for relevant Ifs in that category
Check description and adopt suitable strategies
Finish Requirements change

Figure 5.77: Framework for RCM in DAD

The proposed framework is developed to help practitioners manage the requirements change process in a better way. It incorporates the results of this research study to let the software engineers focus on all the important aspects while doing the RCM process. When a requirement change request is received and RCM process is initiated, the practitioners shall look for the categories relevant to their part of work and then focus on the influencing factors of the category/ categories relevant to their task. If they are lacking at one or more factors, they shall take measures to cope up with the influencing factors to achieve their objective in a systematic manner. For example, if an individual is working as a project manager and (s)he wants to measure the progress to decide whether to go for the change or not, (s)he shall go to the Progress Measure category first. Then (s)he shall consider the influencing factors of this category one by one and decide what's suitable for the project. So, (s)he shall examine; If (s)he has the skilled developers, if (s)he can afford to welcome the change at that stage of the development, if (s)he has the strong leadership to support, if (s)he has required IT platforms available, etc. Once (s)he has considered the influencing factors, (s)he can now easily decide whether to go for the change or to reject it or to delay it for the time being. Every individual or team can make use of this framework in the similar manner by following the simple steps provided in the framework and by making use of its categories and influencing factors to perform the RCM process in a systematic and efficient manner. Thus, the framework helps its users to avoid problems, failure, or hassle later in the process by considering all the relevant factors. Therefore, this framework is quite useful for the practitioners to conduct RCM process in DAD environment.

5.4 Discussion

The study proposed a framework for RCM process in DAD setup. The framework uses the categories and influencing factors found as the result of this research study (Table 5.2). There are six categories devised for the practitioners. Progress measure category is found out to be the most important one. There are nine influencing factors in this category. Out of those nine influencing factors, Skilled developers is rated as the most highly influencing factor while "welcome change even at the end of development" is the second most highly influencing factor for RCM process in this category.

The category at the second priority in the list is Impact Analysis category. It also has nine influencing factors in total. Resources (cost and time) is the most highly influencing factor in this category which means that while deciding whether to go for change or not, the practitioners must consider if they have the required cost and time to incorporate change. Human factors and lack of tools are second and third most important influencing factors in this category respectively. This means that practitioners shall consider the human factors and the availability of tools to decide for the requirement change request.

Understanding requirement change is the third important category with a total of nine influencing factors as shown in the Table 5.2. Vision and Goal for change is the topmost important influencing factor in this category. This means that when the requirements change process is initiated, the practitioners need to first understand the vision and goal for the respective change. Enhancement requests and Content changes is the second important influencing factor which suggests that it is important to understand the enhancement and content changes request prior to practically starting the change in the development process. Knowledge management is the third highly influencing factor in this category suggesting that it is also vital to understand and incorporate efficient knowledge management mechanisms for RCM process to be successfully completed.

The fourth important category is Management Role category. It has 17 influencing factors in total with “Time schedule” being the most important one. “Skilled human resources” and “Training and monitoring” are the second and third most significant influencing factors in this category. This means that the personnel involved in the management of a software project in DAD setting must consider the time schedule, available human resources and available training and monitoring mechanisms along with other succeeding influencing factors when deciding on a requirement change process. If they are lacking in one or more factors and if they consider that lacking in those factors may affect their project negatively, then they must take certain measures to cope up with the situation. For example, if they don’t have skilled human resources to meet the requirements of the project, they must hire new suitable individuals and if they think they must provide suitable training and monitoring mechanisms for the existing employees, they must arrange such sessions to deal with the situation.

Team role is the fifth significant category in the findings with fifteen influencing factors in total as shown in Table 5.2. “Team coordination” is the most important factor in this category. “Software Development Life Cycle” and “Requirement traceability” are the next important factors in the list. Team members must focus on their coordination which becomes more complex in distributed agile development setting. They should incorporate proper coordination mechanisms to cope up with this factor. Teams need to keep track of requirements in entire software development life cycle and there should be proper traceability mechanisms to achieve RCM process in an efficient manner.

The sixth important category as the results of this study is “Communication” category with eight influencing factors. “Effective communication”, “Geographically distributed CCB (change control block)” and “speed of communication” are the top three influencing factors in this category. This suggests that the teams distributed at different locations must incorporate effective means of communication, a geographically distributed CCB and they must maintain desired speed while communicating with the distributed teams to avoid delays.

The proposed framework uses these categories and influencing factors to guide the software development teams, managers as well as higher management and chief executives to carry out RCM process properly and systematically in any DAD project. The list of categories shall help them to look for the categories relevant to their roles and responsibilities only. This helps in saving efforts and to keep focus only on the required aspects. The influencing factors of each category help the practitioners to keep their focus on all required important aspects while completing their tasks. This framework provides a scaled and systematic approach for RCM in DAD setup by providing all the necessary aspects and saves lots of efforts and resources by limiting the focus on only the required factors at the same time.

5.5 Summary

The results of second and third phases of this study have been shown in this chapter. The outcome as well as detailed discussion has also been presented. The next chapter presents the conclusion to sums up this research study.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Overview

This chapter discusses the significance of this research and potential research areas relevant to this study for the future.it also presents the concluding remarks of the authors on this research study.

6.2 Research Summary

This research is conducted to explore the requirements change management process in the domain of distributed agile development. The research is conducted to answer the following research questions:

- What are the influencing factors which affect requirements change management process in distributed agile development?
- How to categorize and prioritize the identified influencing factors for requirements change management in distributed agile development?
- What is the suitable way to deal with the influencing factors to manage requirements change in distributed agile development?

first research question. Survey is used to answer the second RQ and results of first two RQs were combined to obtain the eventual outcome which is a framework for RCM in DAD.

6.3 Fulfilment of Research Objectives

This research has the following three objectives:

- To identify influencing factors from the literature which affect RCM in DAD
- To categorize and prioritize the identified influencing factors for RCM in DAD
- To present a framework for efficient RCM in DAD

All three objectives are achieved successfully. A categorized list of IFs is developed to fulfil objective one. Then, it is categorized to satisfy objective two and in the end a framework is proposed to address third objective.

6.4 Limitations of this Research

The SLR in this study is limited to the five databases over the past five years. Also, this research is based in the domain of software engineering and experts from the industry are chosen as the participants. Thus, the number of participants might be less. Covid-19 is another hindrance in getting more participants. Another limitation of this study is that the proposed framework is not validated in the industry due to the short of time and resources to conduct this research study. This might as well be a potential future work of this study.

6.5 Significance and Future Work

This research would be quite significant and useful for the software engineering professionals working in Distributed Agile Development environment to manage requirements change process in an efficient and systematic manner. The proposed framework, categories and influencing factors help to scale and systemize the RCM process for any DAD project at any phase of SDLC. The potential future work would be to validate the proposed framework in a real-time DAD project. A couple of case studies in DAD projects would reveal the uses and benefits of this research. Another potential future work in this domain would be the real-time validation of the categories and its influencing factors in the industry.

The research adds significant value to the area of requirements change management in the distributed agile development environment. The results of this study are novel and open many new dimensions for the researchers to work on in the coming years.

6.6 Conclusion

Distributed agile development has increased exponentially over the recent years round the globe. Companies all around the world, no matter what the size is, are making use of DAD to get aided from the benefits of DAD. Distributed software development has already become a norm for software industry and agile has always gained popularity in the software industry since its existence. Being different in the fundamental nature but having separate benefits, both agile and distributed software development have been the focus of attention in the academia as well as in the industry. That's the prime reason for the increasing use of DAD in software development.

Requirement's engineering is one of the most salient and central phase of any modern day's software development project. Software projects being built today are huge and extensive where requirements keep changing in the real time. Therefore, the need to have proper requirements change management mechanisms is inevitable. This becomes even more necessary and vital in DAD due to the very nature of DAD environment. This research study is an effort to fulfil this need by proposing a RCM framework for DAD as well as by scaling and systemizing the RCM process by providing categories and influencing factors which are to be used with the framework. This research study might be a tiny drop in the ocean but it sure is one step forward towards the advancement of modern days' software engineering.

REFERENCES

[1] C. Ebert, “The dark side: challenges. Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing”; 19-25; 2011.

[2] D. Damian and D. Moitra, “Global software development: How far have we come?” IEEE Software, vol. 23, no. 5, p. 17, 2006, doi: 10.1109/MS.2006.126.

[3] R. E. Grinter, J. D. Herbsleb, and D. E. Perry, “Geography of coordination: Dealing with distance in R&D work,” Proc. Int. ACM Siggr. Conf. Support. Gr. Work, pp. 306–315, 1999.

[4] J. D. Herbsleb and D. Moitra, “Global software development,” IEEE Software, vol. 18, no. 2, pp. 16–20, 2001, doi: 10.1109/52.914732.

[5] M. A. Akbar et al., “Success factors influencing requirements change management process in global software development,” J. Computer. Lang., vol. 51, no. August 2018, pp. 112–130, 2019, doi: 10.1016/j.cola.2018.12.005.

[6] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, “An empirical study of global software development: Distance and speed,” Proc. - Int. Conf. Software. Eng., pp. 81–90, 2001, doi: 10.1109/ICSE.2001.919083.

[7] J. M. Bhat, M. Gupta, and S. N. Murthy, “Lessons from Offshore Outsourcing,” IEEE Software., vol. September, pp. 38–43, 2006.

[8] K. Schmid, “Challenges and solutions in global requirements engineering - A literature survey,” Lect. Notes Bus. Inf. Process., vol. 166 LNBIP, pp. 85–99, 2014, doi: 10.1007/978-3-319-03602-1.

[9] W. Hussain, “Reflections on requirements change management in global software development: A multiple case study,” Proc. - 11th IEEE Int. Conf. Glob. Software. Eng. Companion Proceedings, ICGSEW 2016, pp. 77–79, 2016, doi: 10.1109/ICGSEW.2016.25.

[10] P. Diebold, D. M. Fernández, and D. Šmite, “Summary of the 1st International Workshop on Impact of Agile Practices (Impact 2015),” in Proceedings of the 2015 International Conference on Software and System Process, 2015, pp. 181–182, doi: 10.1145/2785592.2785622.

[11] G. K. Hanssen, D. Šmite, and N. B. Moe, “Signs of agile trends in global software engineering research: A tertiary study,” Proc. - 2011 6th IEEE Int. Conf. Glob. Software. Eng. Work. ICGSE Work. 2011, pp. 17–23, 2011, doi: 10.1109/ICGSE-W.2011.12.

[12] S. W. Ambler, Disciplined Agile delivery meets CMMI, vol. 25, no. 11. 2012.

[13] I. Sommerville, "Integrated requirements engineering: A tutorial," IEEE Software, vol. 22, no. 1, pp. 16–23, 2005, doi: 10.1109/MS.2005.13.

[14] A. J. Nolan, S. Abrahão, P. Clements, and A. Pickard, "Managing requirements uncertainty in engine control systems development," Proc. 2011 IEEE 19th Int. Requirement Eng. Conf. RE 2011, pp. 259–264, 2011, doi: 10.1109/RE.2011.6051622.

[15] J. M. Verner and L. M. Abdullah, "Exploratory case study research: Outsourced project failure," Inf. Software. Technol., vol. 54, no. 8, pp. 866–886, 2012, doi: 10.1016/j.infsof.2011.11.001.

[16] D. M. Berry, K. Czarnecki, M. Antkiewicz, and M. AbdElRazik, "Requirement's determination is unstoppable: An experience report," Proc. 2010 18th IEEE Int. Requirement. Eng. Conf. RE2010, pp. 311–316, 2010, doi: 10.1109/RE.2010.44.

[17] M. Jørgensen, "Failure factors of small software projects at a global outsourcing marketplace," J. Syst. Software., vol. 92, no. 1, pp. 157–169, 2014, doi: 10.1016/j.jss.2014.01.034.

[18] D. Mishra and A. Mishra, "Research trends in management issues of global software development: Evaluating the past to envision the future," J. Glob. Inf. Technol. Manag., vol. 14, no. 4, pp. 48–69, 2011, doi: 10.1080/1097198X.2011.10856549.

[19] M. Raatikainen, T. Männistö, T. Tommila, and J. Valkonen, "Challenges of requirements engineering - A case study in nuclear energy domain," Proc. 2011 IEEE 19th Int. Requir. Eng. Conf. RE 2011, pp. 253–258, 2011, doi: 10.1109/RE.2011.6051629.

[20] Project Management Institute, "Requirements Management: A Core Competency for Project and Program Success," PMI's Pulse Prof., pp. 1–20, 2014.

[21] W. Liu, K. Q. He, J. Wang, and R. Peng, "Heavyweight semantic inducement for requirement elicitation and analysis," 3rd Int. Conf. Semant. Knowledge, Grid, SKG 2007, pp. 206–211, 2007, doi: 10.1109/SKG.2007.144.

[22] M. A. Akbar et al., "Statistical Analysis of the Effects of Heavyweight and Lightweight Methodologies on the Six-Pointed Star Model," IEEE Access, vol. 6, pp. 8066–8079, 2018, doi: 10.1109/ACCESS.2018.2805702.

[23] M. Shafiq et al., "Effect of Project Management in Requirements Engineering and Requirements Change Management Processes for Global Software Development," IEEE Access, vol. 6, no. c, pp. 25747–25763, 2018, doi: 10.1109/ACCESS.2018.2834473.

[24] M. A. Akbar, Nasrullah, M. Shafiq, J. Ahmad, M. Mateen, and M. T. Riaz, "AZ-Model of software requirements change management in global software development," 2018 Int.

Conf. Computer. Electron. Electrical. Eng. ICE Cube 2018, pp. 1–6, 2019, doi: 10.1109/ICECUBE.2018.8610964.

[25] S. Schneider, R. Torkar, and T. Gorschek, “Solutions in global software engineering: A systematic literature review,” Int. J. Inf. Manage., vol. 33, no. 1, pp. 119–132, 2013, doi: 10.1016/j.ijinfomgt.2012.06.002.

[26] G. Levin, Program Management Complexity. 2016.

[27] S. Hussain, N. Ehsan, and S. Nauman, “A strategic framework for requirements change in technical projects: Case study of a R&D project,” pp. 354–358, 2010, doi: 10.1109/iccsit.2010.5564998.

[28] A. A. Khan, S. Basri, and P. D. D. Dominic, “A propose framework for requirement change management in global software development,” 2012 Int. Conf. Comput. Inf. Sci. ICCIS 2012 - A Conf. World Eng. Sci. Technol. Congr. ESTCON 2012 - Conf. Proc., vol. 2, pp. 944–947, 2012, doi: 10.1109/ICCISci.2012.6297161.

[29] E. Carmel, Global software teams: collaborating across borders and time zones, Prentice Hall PTR, 1999.

[30] J.N. Lee, “The evolution of outsourcing research: What is the next issue,” Proceedings of the Hawaii International Conference on System Sciences, vol. 7, 2000, p. 7070.

[31] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, “Empirical evidence in global software engineering: a systematic review,” Empirical Software Engineering.

[32] J.D. Herbsleb, “Global software development,” IEEE Software, vol. 18, 2001, p. 16.

[33] R. Sangwan, M. Bass, N. Mullick, D.J. Paulish, and J. Kazmeier, Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series), Auerbach Publications, 2006.

[34] D. Šmite and J. Borzovs, “A framework for overcoming supplier related threats in global projects,” Software Process Improvement, 2006, pp. 50–61.

[35] N.B. Moe and D. Smite, “Understanding a lack of trust in global software teams: A multiple-case study,” Software Process Improvement and Practice, vol. 13, 2008, pp. 217-231.

[36] W. Aspray, F. Mayadas, and M. Vardi, Eds., Globalization and Offshoring of Software - A Report of the ACM Job Migration Task Force, Association for Computing Machinery, 2006.

[37] E.Ó. Conchúir, P.J. Ågerfalk, H.H. Olsson, and B. Fitzgerald, “Global software development: where are the benefits?” Commun. ACM, vol. 52, 2009, pp. 127-131.

[38] J. Ma, J. Li, W. Chen, R. Conradi, J. Ji, and C. Liu, “An Industrial Survey of Software

Outsourcing in China," Product-Focused Software Process Improvement, 2007, pp. 5-19.

[39] J.M. Hussey, "Managing Global Development Risk", 2007.

[40] E. Conchuir, H. Holmstrom, P. Agerfalk, and B. Fitzgerald, "Exploring the Assumed Benefits of Global Software Development," 2006 IEEE International Conference on Global Software Engineering (ICGSE'06), 2006, pp. 159-168.

[41] P.J. Agerfalk, H. Holmstrom, B. Lings, B. Lundell, and E. Conchuir, "A Framework for Considering Opportunities and Threats in Distributed Software Development," DiSD, Paris, France: 2005.

[42] M. Korkala and P. Abrahamsson, "Communication in distributed agile development: A case study," 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2007, August 27, 2007 - August 31, 2007, Lubeck, Germany: Inst. of Elec. and Elec. Eng. Computer Society, 2007, pp. 203-210.

[43] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," Software, IEEE, vol. 18, 2001, pp. 22-29.

[44] D. Damian, L. Izquierdo, J. Singer, and I. Kwan, "Awareness in the wild: Why communication breakdowns occur," International Conference on Global Software Engineering, ICGSE 2007, August 27, 2007 - August 30, 2007, Munich, Germany: Inst. of Elec. and Elec. Eng. Computer Society, 2007, pp. 81-90.

[45] D. Smite, N. Moe, and R. Torkar, "Pitfalls in remote team coordination: Lessons learned from a case study," PRODUCT-FOCUSED SOFTWARE PROCESS IMPROVEMENT, PROCEEDINGS, vol. 5089, 2008, pp. 345-359.

[46] H. Mintzberg, Mintzberg on management: Inside our strange world of organizations, 1989.

[47] Rally. "Agile with a Capital "A" Vs. agile With a Lowercase "a"". Archived from the original on 5 January 2016. Retrieved 9 September 2015.

[48] Collier, Ken W. "Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Pearson Education". pp. 121 ff. ISBN 9780321669544. What is a self-organizing team? 2011.

[49] Beck, M, et al., "Manifesto for Agile Software Development". Undefined. S2CID 109006295.

[50] "What is Agile Software Development?". Agile Alliance. 8 June 2013. Retrieved 4 April 2015.

[51] Kent Beck, Robert C. "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June2010.

[52] “Which is better – Kanban or Scrum?”, 4 March 2016

[53 Larman, Craig.” Agile and Iterative Development: A Manager's Guide”. Addison-Wesley. p. 27. ISBN 978-0-13-111155-4. 2004.

[54] V. Stray, R. Hoda, M. Paasivaara and P. Kruchten, "Agile software development practices and Success in outsourced projects: the moderating role of requirements risk", Agile Processes in software engineering and extreme programming, 2020.

[55] Lee, Gwanhoo. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility". MIS Quarterly. 34 (1): 87–114. doi:10.2307/20721416. JSTOR 20721416. S2CID 26477249.2010.

[56] Gerald M Weinberg, “Iterative and Incremental Development”, 2003, pp. 47–56.

[57] Gilb."Evolutionary Project Management (Original page, external archive)". Archived from the original on 27 March 2016. Retrieved 30 April 2017.

[58] Gilb. "Evolutionary Project Management (New page)". Retrieved 30 April 2017.

[59] Edmonds, E. A. "A Process for the Development of Software for Nontechnical Users as an Adaptive System". General Systems. 19: 215–18.1974.

[60] Gilb, Tom. "Evolutionary development". ACM SIGSOFT Software Engineering Notes. 6 (2): 17. doi:10.1145/1010865.1010868. S2CID 33902347. 1 April 1981.

[61] Martin., James. “Rapid Application Development”. ISBN 978-0-02-376775-3. 1991.

[62] Kerr., James M. “Inside RAD: How to Build a Fully Functional System in 90 Days or Less.” McGraw-Hill. p. 3. ISBN 978-0-07-034223-1.1993.

[63] "21st Century Manufacturing Enterprise Strategy: An Industry Led View". Iacocca Institute, Lehigh University, Bethlehem, PA. 1991.

[64] Presley, A."Agile Aerospace Manufacturing". Nepcon East 1995, Boston.

[65] Anderson, David. "Declaration of Interdependence". Archived from the original on 27 January 2018. Retrieved 4 October 2018.

[61] McDonald., Kent. "How You Can Help Agile Alliance Help You". Agile Alliance Blog. Retrieved 4 July 2017.

[62] "Examining the Agile Manifesto". Ambysoft Inc. Retrieved 6 April 2011.

[63] Jim Highsmith. "History: The Agile Manifesto". agilemanifesto.org.2001.

[64] Kent Beck., Robert C., et al., "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 6 June 2010.

[65] Moran., A. “Agile Risk Management”. Springer Verlag. ISBN 978-3319050072. 2014.

[66] Beck, Kent. "Embracing Change with Extreme Programming". Computer. 32 (10): 70–77. doi:10.1109/2.796139. 1999.

[67] Mergel., Ines."Agile innovation management in government: A research agenda". Government Information Quarterly. 33 (3): 516–523. doi: 10.1016/j.giq.2016.07.004. July 2014.

[68] Steven J zeil, “Software development process models”, 2006, website: cs.odu.edu

[69] Preuss, Deborah Hartmann (13 October 2006). "Study: Co-Located Teams vs. the Cubicle Farm". InfoQ. Retrieved 23 October 2018.

[70] Cockburn., Alistair. "Agile Software Development: The Cooperative Game". www.pearson.com (2nd ed.). Addison-Wesley Professional. Retrieved 23 October 2018.

[71] Jain, Parita., et al, "The Impact of Agile Software Development Process on the Quality of Software Product". 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Noida, India: IEEE: 812–815. doi:10.1109/ICRITO.2018.8748529. ISBN 978-1-5386-4692-2. S2CID 195775457. 2018

[72] Cockburn., Alistair. "Information radiator".19 June 2008

[73] Ambler., Scott. “Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process”. John Wiley & Sons. pp. 12, 164, 363. ISBN 978-0-471-20282-0. 12 April 2002.

[74] Vasiliauskas., Vidas. "Developing agile project task and team management practices". Eylean. Archived from the original on 15 September 2014. Retrieved 15 September 2014.

[75] Jeffries., Ron.” Extreme Programming installed”. Addison-Weslsy. pp. 72–147. ISBN 978-0201-70842-4. 2001

[76] Lisa Crispin., Janet Gregory. “Agile Testing: A Practical Guide for Testers and Agile Teams”. Addison-Wesley. 2009.

[77] Mitchell., Ian. “Agile Development in Practice”. Tamare House. p. 11. ISBN 978-1-908552-49-5. 2016

[78] Larman, Craig. “Agile and Iterative Development”: A Manager's Guide. Addison-Wesley. p. 27. ISBN 978-0-13-111155-4. 2004.

[79] Boehm., B. “Balancing Agility and Discipline: A Guide for the Perplexed”. Boston, MA: Addison-Wesley. ISBN 978-0-321-18612-6. Appendix A, pages 165–194. 2004.

[80] Larman, Craig. “Agile and Iterative Development: A Manager's Guide”. p. 253. ISBN 9780131111554. Retrieved 14 October 2013. 2004.

[81] Sliger., Michele. “The Software Project Manager's Bridge to Agility”. Addison-Wesley. p. 46. ISBN 978-0-321-50275-9. 2008.

[82] Park., J. S., B.” Scrum Powered by Essence. ACM SIGSOFT Software Engineering Notes”, 41(1), pp. 1–8. 2016

[83] Boehm., B., R. “Balancing Agility and Discipline: A Guide for the Perplexed”. Boston, MA: Addison-Wesley. pp. 55–57. ISBN 978-0-321-18612-6. 2004.

[84] Beck, K.” Extreme Programming Explained: Embrace Change”. Boston, MA: Addison-Wesley. ISBN 978-0-321-27865-4. 1999.

[85] W. Scott Ambler. “Supersize Me” in Dr. Dobb's Journal, 15 February 2006.

[86] Schaaf., R.J. "Agility XL Systems and Software Technology Conference 2007” Archived 13 March 2016 at the Wayback Machine, Tampa, FL. 2007.

[87] "Bridging the Distance". Sdmagazine.com. Retrieved 1 February 2011.

[88] Fowler., Martin. "Using an Agile Software Process with Offshore Development". Martinfowler.com. Retrieved 6 June2010.

[89] “Agile Processes Workshop II Managing Multiple Concurrent Agile Projects”. Washington: OOPSLA 20

[90] Jiménez., M. “Challenges and improvements in distributed software development: A systematic review”. Advances in Software Engineering, 2009.

[91] Prikladnicki., R., et al., “Patterns of evolution in the practice of distributed software development: quantitative results from a systematic review”. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12* (pp. 1-10). June 2008.

[92] Fowler., M., et al., “The agile manifesto. Software Development”, 9(8), 28-35. 2001

[93] Ramesh., B. “Can distributed software development be agile?”. Communications of the ACM, 49(10), 41-46. 2006.

[94] Razavi., A. “Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects”. In 2014 8th. Malaysian Software Engineering Conference (MySEC) (pp. 216-221). IEEE. September 2014.

[95] Ghani., I. “Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review”. KSII Transactions on Internet & Information Systems, 13(9). 2019.

[96] Shrivastava., S. V. “Distributed agile software development: A review". *arXiv preprint arXiv:1006.1955*. 2010

[97] M.Paasivaara., S. “Using Scrum in Distributed Agile Development: A Multiple Case Study”, IEEE International Conference on Global Software Engineering, p.195-204,

2009

[98] Shrivastava., S. “Distributed Agile Software Development: A Review”. Seo-chogu: Journal of computer science and engineering. 10-17. 2010

[99] https://www.knowledgehut.com/blog/agile/key-factors-to-succeed-agile-teams

[100] Shrivastava., S. “Risks in distributed agile development: A review. Procedia - Social and Behavioral Sciences”, 133, pp.417-424. 2014

[101] Shrivastava., S.” Distributed agile software development: A review”. arXiv preprint arXiv:1006.1955. 2010

[102] Stellman., Andrew. “Applied Software Project Management”. O'Reilly Media. ISBN 978-0-596-00948-9. Archived from the original on 2015-02-09. 2005

[103] “Requirements management". UK Office of Government Commerce. Retrieved 2009-11-10.

[104] “A Guide to the Project Management Body of Knowledge” (4th ed.). Project Management Institute. 2008. ISBN 978-1-933890-51-7. 2008

[105] Gotel., O. “An Analysis of the Requirements Traceability Problem Proc. of First International Conference on Requirements Engineering”, 1994, pages 94-101

[106] Gotel., Orlena., et al., “Software and Systems Traceability”. Springer London. pp. 3–22. doi:10.1007/978-1-4471-2239-5_1. ISBN 9781447122388. 1st January 2012

[107] Chemuturi M. “Requirements Engineering and Management for Software Development Projects”. doi:10.1007/978-1-4614-5377-2. ISBN 978-1-4614-5376-5. S2CID 19818654 2013

[108] Jayatilleke.S., Lai. R.; “A systematic review of requirements change management”; journal of Information and Software Technology; 2018.

[109] Hussin. A, Hussain.A, Baharom.F.; “Current Challenges of Requirement Change Management”; JTEC, 2016.

[110] Yaseen. M, Baseer. S., Sherin. S.; “Critical challenges for requirement implementation in context of global software development: A systematic literature review”; 2015 International Conference on Open Source Systems & Technologies (ICOSST); IEEE; 2015.

[111] Jain. R., Suman. U.; “A Systematic Literature Review on Global Software Development Life Cycle”; ACM SIGSOFT Software Engineering; 2015.

[112] Schon.E.M., Thomas.J., Escalona. M.J.; “Agile Requirements Engineering: A systematic literature review”; Computer Standards & Interfaces; 2017.

[113] Inayat. I., Salim. S.S., Marczak. S., Daneva. M., Shamshirband. S.; “A systematic

literature review on agile requirements engineering practices and challenges"; Computers in Human Behavior; 2015.

[114] Bano.M., Imtiaz. S., Ikram. N., Niazi .M., Usman . M.; "Causes of requirement change - A systematic literature review"; EASE; 2012.

[115] Li. J., Zhang. H., Zhu. L., Jeffery. R., Wang. Q., Li. M.; "Preliminary Results of a systematic review on requirements evolution"; EASE; 2012

[116] Alves. V., Niu. N., Alves. C., Valenca. G.; "Requirements engineering for software product lines: A systematic literature review"; Information and Software Technology, 2010.

[117] Walia. G. S., Carver. J. C.; "A systematic literature review to identify and classify software requirement errors"; Information and Software Technology; 2009.

[118] Akbar. M. A., Sang. J., Nasrullah, Khan. A. A., Mahmood. S., Qadri . S. F., Hu. H., Xiang. H.; "Success factors influencing requirements change management process in global software development"; Journal of Computer Languages; 2019.

[119] Akbar. M. A., Sang. J., Khan. A. A., Shafiq. M.; "Towards the Guidelines for Requirements Change Management in Global Software Development: Client-Vendor Perspective"; IEEE Access; 2019.

[120] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Version, vol. 2, 2007, pp. 2007–01.

[121] B. Kitchenham, "Procedures for Performing Systematic Reviews."

[122] D. Azhar, E. Mendes, E. Mendesazu, and P. Riddle, "A Systematic Review of Web Resource Estimation". 2012.

[123] H. Noble and G. Mitchell, "What is grounded theory", BMJ Journals, Vol. 19-2, 2016.

[124] Ayyub, B.; "Elicitation of Expert Opinion for Uncertainty and Risks"; CRC Press; 2001

[125] Boring, R., Gertman, D., Joe, J., & Marble, J.; "Simplified Expert Elicitation Guideline For Risk Assessment Of Operating Events"; INL; 2005

[126] A. Fink, "How to ask survey questions", Sage Publications, Inc, 2003.

[127] C. Wohlin, M. H\öst, P. Runeson, M.C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering: an introduction, Kluwer Academic Pub, 2000.

[128] C.W. Dawson, Projects on computing and information systems: a student's guide, Addison Wesley Publishing Company, 2005.

[129] John W. Creswell, Research Design: Qualitative, Quantitative and Mixed Method approaches, USA: 2009.

[130] T. Friedman, “The world is flat: A brief history of the 21st century”, New York: Farrar, Strauss, and Giroux, 2005.

APPENDIX A

List of Final Studies

S.NO	Paper ID	Paper Title
01	P1	A Supporting Tool for Requirements Change Management in Distributed Agile Development
02	P2	Moving from traditional to agile software development methodologies also on large, distributed projects.
03	P3	Categorization of risk factors for distributed agile projects.
04	P4	Scrum Requirements Engineering Practices and Challenges in Offshore Software Development.
05	P5	Empirical studies of geographically distributed agile development communication challenges: A systematic review.
06	P6	Improving offshoring of low-budget agile software development using the dual-shore approach: an auto ethnographic study
07	P7	Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review
08	P8	Quality requirements challenges in the context of large-scale distributed agile: An empirical study
09	P9	Toward successful agile requirements change management process in global software development: a client–vendor analysis
10	P10	Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development
11	P11	Experiences from the Design of an Artefact Model for Distributed Agile Project Management
12	P12	Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature

13	P13	Agile Software development in distributed team Enhancement Techniques
14	P14	A Framework for Transitioning of Traditional Software Development Method to Distributed Agile Software Development
15	P15	An Empirical Study on Lean and Agile Methods in Global Software Development
16	P16	Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective
17	P17	Toward an Agile Approach to Managing the Effect of Requirements on Software Architecture during Global Software Development
18	P18	Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process
19	P19	A Novel Framework for Change Requirement Management (CRM) In Agile Software Development
20	P20	An exploratory study in communication in Agile Global Software Development
21	P21	Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review
22	P22	A systematic review of distributed Agile software engineering
23	P23	An Empirical Investigation on Effort Estimation in Agile Global Software Development
24	P24	Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a global company
25	P25	Communication Network in an Agile Distributed Software Development Team
26	P26	A Risk Management Framework for Distributed Agile Projects

APPENDIX B

Data Extraction Tables

Data Source	ACM Digital Library
ID	P1
Title	A Supporting Tool for Requirements Change Management in Distributed Agile Development
Author	Domia Lloyd, Ramadan Moawad, Mona Kadry
Year of Publication	12 April 2017
Publication Type	Journal
Influencing Factor	IF1: Different geographical locations of the development teams IF2: Communication between development team IF3: Knowledge management

Data Source	Science Direct
ID	P2
Title	Moving from traditional to agile software development methodologies also on large, distributed projects.
Author	Georgios Papadopoulos
Year of Publication	2015
Publication Type	Conference Paper

Influencing Factor	IF 4: Enhancement Requests and Content Changes
	IF 5: Time Schedule
	IF 6: Project Cost

Data Source	Science Direct
ID	P3
Title	Categorization of risk factors for distributed agile projects
Author	Suprika V. Shrivastava ⇑, Urvashi Rathod 1
Year of Publication	17 July 2014
Publication Type	Journal

Influencing Factor	IF 7: Software Development Life Cycle
	IF 8: Project Management
	IF 9: Group Awareness
	IF 10: External Stakeholder Collaboration
	IF 11: Technology Setup

Data Source	Google Scholar
ID	P4
Title	Scrum Requirements Engineering Practices and Challenges in Offshore Software Development
Author	V. N. Vithana
Year of Publication	April 2015
Publication Type	Journal

Influencing Factor	IF 12: Face to face communication
	IF 13: Customer Involvement and Interaction
	IF 14: Managing Requirement change

Data Source	Science Direct
ID	P5
Title	Empirical studies of geographically distributed agile development communication challenges: A systematic review
Author	Yehia Ibrahim Alzoubi , Asif Qumer Gill
Year of Publication	2016
Publication Type	Journal

Influencing Factor	IF 15: Team Configuration
	IF 16: Customer Communication
	IF 17: Organizational Factors
	IF 18: Human Factors

Data Source	Google Scholar
ID	P6
Title	Improving offshoring of low-budget agile software development using the dual-shore approach: an autoethnographic study
Author	Michael Thorkild, Nørgaard Jørgensen
Year of Publication	2015
Publication Type	Conference Paper
Influencing Factor	IF 19: Responding to change over following a plan IF 20: Customer Collaboration IF 21: Cost Estimation and extra payment for changes

Data Source	Springer Link
ID	P7
Title	Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review

Author	Wasim Alsaqaf(&), Maya Daneva
Year of Publication	2017
Publication Type	Journal
Influencing Factor	IF 22: Skilled developers

Data Source	Science Direct
ID	P8
Title	Quality requirements challenges in the context of large-scale distributed agile: An empirical study
Author	Wasim Alsaqaf, Maya Daneva
Year of Publication	2019
Publication Type	Journal
Influencing Factor	IF 23: Team's coordination and communication

Data Source	Google Scholar
ID	P9
Title	Toward successful agile requirements change management process in global software development: a client–vendor analysis
Author	Nosheen Sabahat, Faiza Iqbal, Farooque Azam, Muhammad Younus Javed
Year of Publication	2020
Publication Type	Journal

Influencing Factor	IF 24: Effective communication
	IF 25: Trust building
	IF 26: Roles and responsibilities
	IF 27: Vision and goal for change
	IF 28: Training and monitoring
	IF 29: Geographically distributed CCB (change control block)
	IF 30: Resistance management
	IF 31: Formalised relationship between development teams
	IF 32: Skilled human resources
	IF 33: Overseas site response
	IF 34: Clear change management strategy
	IF 35: Requirement traceability
	IF 36: Change management process awareness
	IF 37: Governance and control of RCM activities
	IF 38: Standard and process for RCM
	IF 39: Change identification and validation
	IF 40: Continuous organisational support
	IF 41: Conducting social events

	IF 42: Synchronization of work at different sites IF 43: Parallel project testing and feedback

Data Source	IEEE
ID	P10
Title	Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development
Author	Tahir kamal, qinghua zhang
Year of Publication	13 March 2020
Publication Type	Journal

Influencing Factor	IF 44: Project Administration
	IF 45: Coordination
	IF 46: Software Methodology
	IF 47: HR management
	IF 48: Technology Factors

Data Source	IEEE Xplore
ID	P11
Title	Experiences from the Design of an Artefact Model for Distributed Agile Project Management
Author	Henning Femmer; Marco Kuhrmann
Year of Publication	2015
Publication Type	Conference

Influencing Factor	IF 49: Resources (Time and cost)
	IF 50: Change request

Data Source	IEEE Xplore
ID	P12
Title	Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature
Author	Victor Temitayo Faniran; Abdulbaqi Badru
Year of Publication	2017
Publication Type	Conference
Influencing Factor	IF 51: Rapid response to changes in requirements rather than following a prescribed plan IF 52: Collaboration between the software developer and user

Data Source	IEEE Xplore
ID	P13
Title	Agile Software development in distributed team Enhancement Techniques
Author	Ivaturi Saikiran; Rajbala Simon
Year of Publication	2019
Publication Type	Conference
Influencing Factor	IF 53: Welcome requirements change, even at the end of development IF 54: Positive team role

Data Source	IEEE XPLORE
ID	P14
Title	A Framework for Transitioning of Traditional Software Development Method to Distributed Agile Software Development
Author	Madan Singh; Naresh Chauhan
Year of Publication	2020
Publication Type	Conference

Influencing Factor	IF 55: Support quick and flexible response to change
	IF 56: Team coordination
	IF 57: Project leader

Data Source	IEEE Xplore
ID	P15
Title	An Empirical Study on Lean and Agile Methods in Global Software Development
Author	Mohammad Abdur Razzak
Year of Publication	2016
Publication Type	Conference

Influencing Factor	IF 58: Autonomy and Decision-Making Power
	IF 59: Choosing the Right Method

Data Source	IEEE Xplore
ID	P16
Title	Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective
Author	Mohammad Shameem; Chiranjeev Kumar
Year of Publication	2017
Publication Type	Conference

Influencing Factor	IF 60: Tight Time and Constraint
	IF 61: Coordination

	IF 62: Communication
	IF 63: Lack of Tools

Data Source	Google Scholar
ID	P17
Title	Toward an Agile Approach to Managing the Effect of Requirements on Software Architecture during Global Software Development
Author	Hameed Khan, and Sultan Alyahya
Year of Publication	2016
Publication Type	Journal

Influencing Factor	IF 64: Reporting the Change
	IF 65: Analysing the Possible Impact of the Change
	IF 66: Implementing the Change

Data Source	Google Scholar
ID	P18
Title	Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process
Author	Mohammad Shameem, Rakesh Ranjan Kumar
Year of Publication	2017
Publication Type	Journal

Influencing Factor	IF 67: Importance of Individuals and their Interactions
	IF 68: Early Delivery
	IF 69: Collaboration with the customers
	IF 70: Effective responses to change requirements

Data Source	ACM Digital Library
ID	P19
Title	A Novel Framework for Change Requirement Management (CRM) In Agile Software Development
Author	Zainab Shehzadi, Farouq Ahmed
Year of Publication	2019
Publication Type	Conference

Influencing Factor	IF 71: Communication
	IF 72: Documentation

Data Source	Science Direct
ID	P20
Title	An exploratory study in communication in Agile Global Software Development
Author	AgustinYagüe JuanGarbajosa
Year of Publication	2016
Publication Type	Journal
Influencing Factor	IF 72: Tools for Communication

Data Source	Google Scholar
ID	P21
Title	Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review
Author	Imran Ghani, Angelica Lim
Year of Publication	2019
Publication Type	Journal
Influencing Factor	IF 73: Communication of changes in requirements

Data Source	ACM Digital Library
ID	P22
Title	A systematic review of distributed Agile software engineering
Author	Buturab Rizvi, Ebrahim Bagher
Year of Publication	2015
Publication Type	Journal
Influencing Factor	IF 74: Strong leadership

Data Source	IEEE Xplore
ID	P23
Title	An Empirical Investigation on Effort Estimation in Agile Global Software Development
Author	Ricardo Britto; Emilia Mendes
Year of Publication	2015
Publication Type	Conference
Influencing Factor	IF 75: Documentation of requirements

Data Source	IEEE Xplore
ID	P24
Title	Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a global company
Author	Murat Dogus Kahya; Çağla Seneler
Year of Publication	2018

Publication Type	Conference
Influencing Factor	IF 76: Clearly defined team roles

Data Source	IEEE Xplore
ID	P25
Title	Communication Network in an Agile Distributed Software Development Team
Author	Paul T. Robinson
Year of Publication	2019
Influencing Factor	IF 77: Communication in teams with large time difference

Data Source	Science Direct
ID	P26
Title	A Risk Management Framework for Distributed Agile Projects
Author	Suprika VasudevaShrivastava
Year of Publication	2017
Publication Type	Journal

Influencing Factor	IF 78: Fixed Costs
	IF 79: Speed of communication
	IF 80: Ease of communication

APPENDIX C

Study wise distribution of influencing factors

Study #	Influencing Factors
P1	IF1: Different geographical locations of the development teams IF2: Communication between development team IF3: Knowledge management

P2	IF 4: Enhancement Requests and Content Changes
	IF 5: Time Schedule
	IF 6: Project Cost

P3	IF 7: Software Development Life Cycle
	IF 8: Project Management
	IF 9: Group Awareness
	IF 10: External Stakeholder Collaboration
	IF 11: Technology Setup

P4	IF 12: Face to face communication
	IF 13: Customer Involvement and Interaction
	IF 14: Managing Requirement change

P5	IF 15: Team Configuration
	IF 16: Customer Communication
	IF 17: Organizational Factors
	IF 18: Human Factors

P6	IF 19: Responding to change over following a plan IF 20: Customer Collaboration IF 21: Cost Estimation and extra payment for changes
P7	IF 22: Skilled developers
P8	IF 23: Teams coordination and communication

P9	IF 24: Effective communication
	IF 25: Trust building
	IF 26: Roles and responsibilities
	IF 27: Vision and goal for change
	IF 28: Training and monitoring
	IF 29: Geographically distributed CCB (change control block)
	IF 30: Resistance management
	IF 31: Formalised relationship between development teams
	IF 32: Skilled human resources
	IF 33: Overseas site response
	IF 34: Clear change management strategy
	IF 35: Requirement traceability
	IF 36: Change management process awareness
	IF 37: Governance and control of RCM activities
	IF 38: Standard and process for RCM
	IF 39: Change identification and validation
	IF 40: Continuous organisational support

	IF 41: Conducting social events IF 42: Synchronisation of work at different sites IF 43: Parallel project testing and feedback

P10	IF 44: Project Administration
	IF 45: Coordination
	IF 46: Software Methodology
	IF 47: HR management
	IF 48: Technology Factors

P11	IF 49: Resources (Time and cost)
	IF 50: Change request

P12	IF 51: Rapid response to changes in requirements rather than following a prescribed plan IF 52: Collaboration between the software developer and user
P13	IF 53: Welcome requirements change, even at the end of development IF 54: Positive team role

P14	IF 55: Support quick and flexible response to change
	IF 56: Team coordination
	IF 57: Project leader

P15	IF 58: Autonomy and Decision-Making Power
	IF 59: Choosing the Right Method

P16	IF 60: Tight Time and Constraint
	IF 61: Coordination
	IF 62: Communication
	IF 63: Lack of Tools

P17	IF 64: Reporting the Change
	IF 65: Analysing the Possible Impact of the Change
	IF 66: Implementing the Change

P18	IF 67: Importance of Individuals and their Interactions
	IF 68: Early Delivery
	IF 69: Collaboration with the customers
	IF 70: Effective responses to change requirements

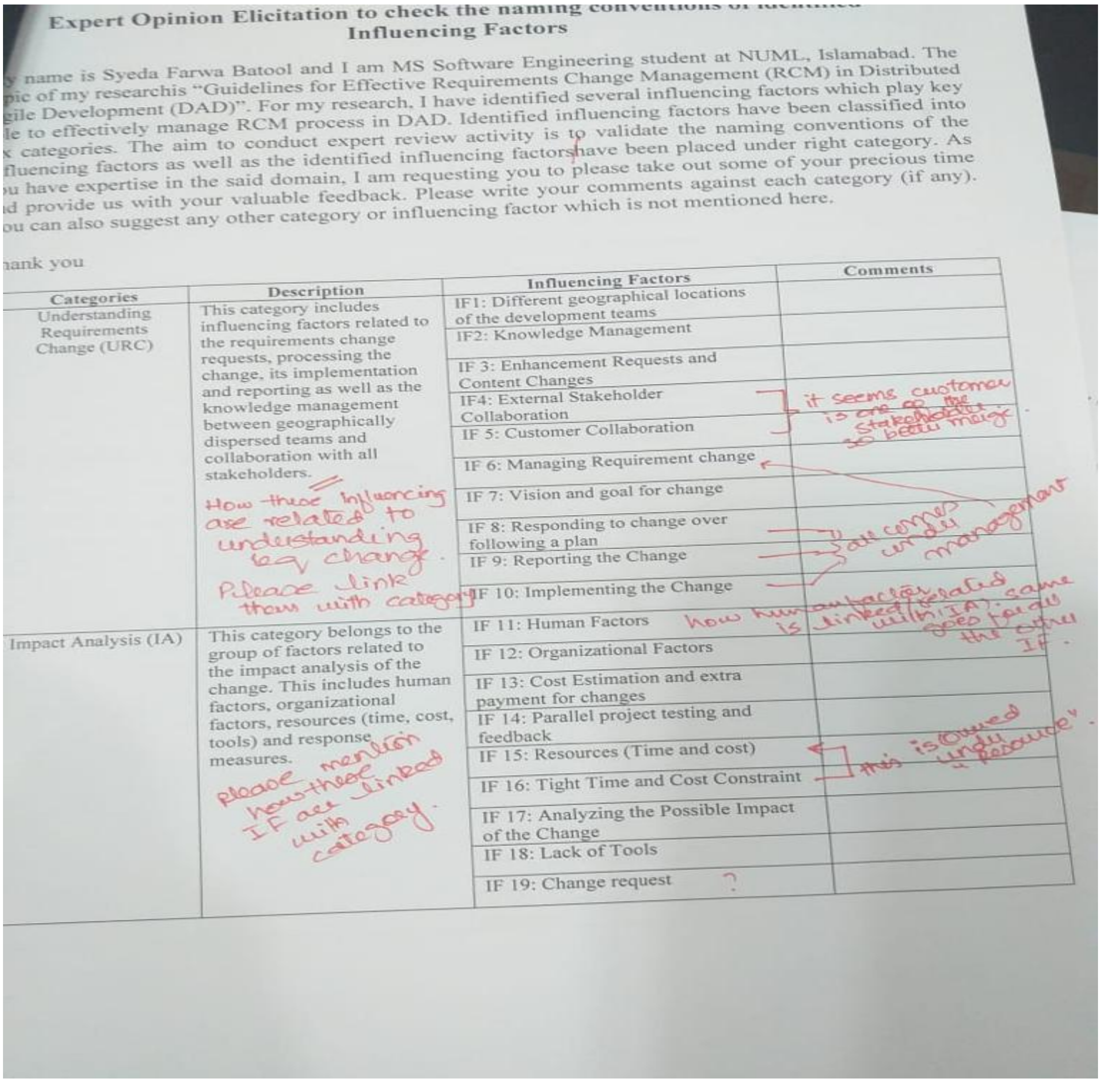
P19	IF 71: Communication
	IF 72: Documentation

P20	IF 72: Tools for Communication
P21	IF 73: Communication of changes in requirements
P22	IF 74: Strong leadership
P23	IF 75: Documentation of requirements
P24	IF 76: Clearly defined team roles
P25	IF 77: Communication in teams with Large time difference

P26	IF 78: Fixed Costs
	IF 79: Speed of communication
	IF 80: Ease of communication

APPENDIX D

Expert Evaluation Results

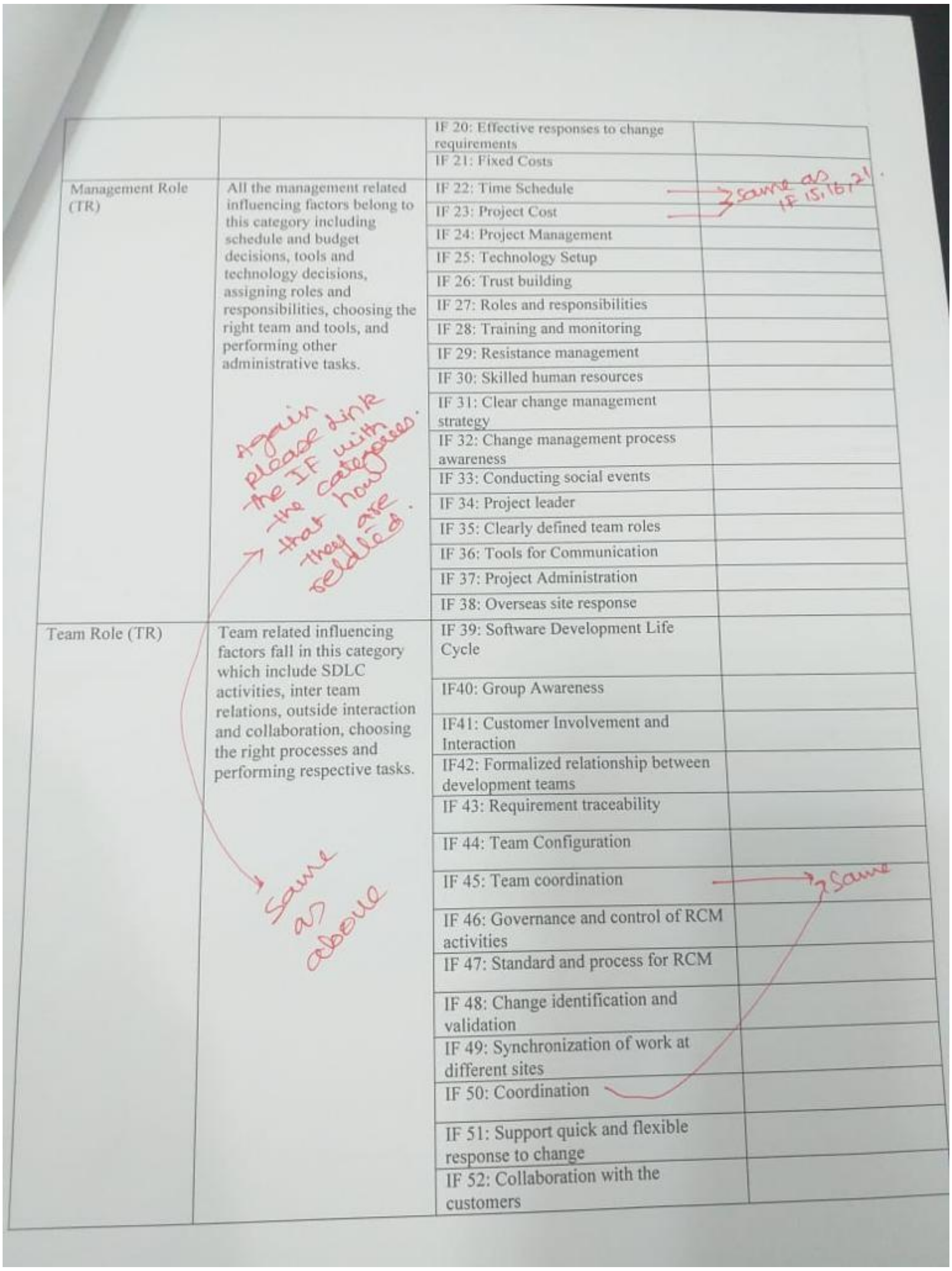


Expert Opinion Elicitation to check the naming conventions of identified Influencing Factors

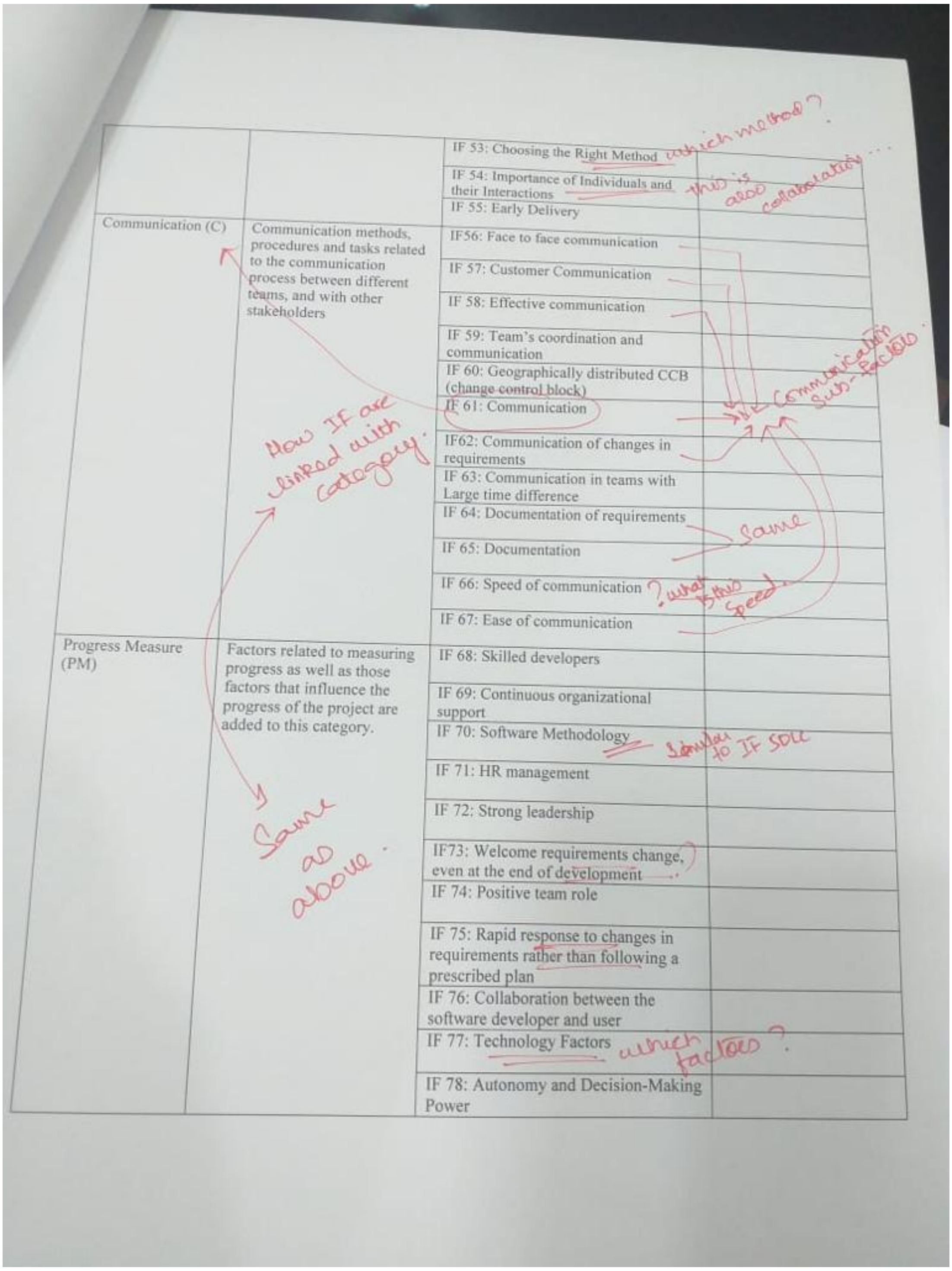
y name is Syeda Farwa Batool and I am MS Software Engineering student at NUML, Islamabad. The pic of my researchis "Guidelines for Effective Requirements Change Management (RCM) in Distributed gile Development (DAD)". For my research, I have identified several influencing factors which play key le to effectively manage RCM process in DAD. Identified influencing factors have been classified into x categories. The aim to conduct expert review activity is to validate the naming conventions of the fluencing factors as well as the identified influencing factors have been placed under right category. As ou have expertise in the said domain, I am requesting you to please take out some of your precious time d provide us with your valuable feedback. Please write your comments against each category (if any). ou can also suggest any other category or influencing factor which is not mentioned here.

hank you

Categories	Description	Influencing Factors	Comments
Understanding Requirements Change (URC)	This category includes influencing factors related to the requirements change requests, processing the change, its implementation and reporting as well as the knowledge management between geographically dispersed teams and collaboration with all stakeholders. How these influencing are related to understanding req change. Please link them with category	IF1: Different geographical locations of the development teams	
		IF2: Knowledge Management	
		IF 3: Enhancement Requests and Content Changes	
		IF4: External Stakeholder Collaboration	it seems customer is one of the stakeholder so better merge
		IF 5: Customer Collaboration	
		IF 6: Managing Requirement change	
		IF 7: Vision and goal for change	
		IF 8: Responding to change over following a plan	all comes under management
		IF 9: Reporting the Change	
		IF 10: Implementing the Change	
Impact Analysis (IA)	This category belongs to the group of factors related to the impact analysis of the change. This includes human factors, organizational factors, resources (time, cost, tools) and response measures. please mention how these IF are linked with category.	IF 11: Human Factors	how human factor is linked (related with IA). same goes for all the other IF.
		IF 12: Organizational Factors	
		IF 13: Cost Estimation and extra payment for changes	
		IF 14: Parallel project testing and feedback	
		IF 15: Resources (Time and cost)	this is covered under "Resource".
		IF 16: Tight Time and Cost Constraint	
		IF 17: Analyzing the Possible Impact of the Change	
		IF 18: Lack of Tools	
		IF 19: Change request ?	



		IF 20: Effective responses to change requirements	
		IF 21: Fixed Costs	
Management Role (TR)	All the management related influencing factors belong to this category including schedule and budget decisions, tools and technology decisions, assigning roles and responsibilities, choosing the right team and tools, and performing other administrative tasks. Again please link the IF with the categories that how they are related.	IF 22: Time Schedule	same as IF 15,16,21.
		IF 23: Project Cost	
		IF 24: Project Management	
		IF 25: Technology Setup	
		IF 26: Trust building	
		IF 27: Roles and responsibilities	
		IF 28: Training and monitoring	
		IF 29: Resistance management	
		IF 30: Skilled human resources	
		IF 31: Clear change management strategy	
		IF 32: Change management process awareness	
		IF 33: Conducting social events	
		IF 34: Project leader	
		IF 35: Clearly defined team roles	
		IF 36: Tools for Communication	
		IF 37: Project Administration	
		IF 38: Overseas site response	
Team Role (TR)	Team related influencing factors fall in this category which include SDLC activities, inter team relations, outside interaction and collaboration, choosing the right processes and performing respective tasks. same as above	IF 39: Software Development Life Cycle	
		IF40: Group Awareness	
		IF41: Customer Involvement and Interaction	
		IF42: Formalized relationship between development teams	
		IF 43: Requirement traceability	
		IF 44: Team Configuration	
		IF 45: Team coordination	same
		IF 46: Governance and control of RCM activities	
		IF 47: Standard and process for RCM	
		IF 48: Change identification and validation	
		IF 49: Synchronization of work at different sites	
		IF 50: Coordination	
		IF 51: Support quick and flexible response to change	
		IF 52: Collaboration with the customers	



		IF 53: Choosing the Right Method	which method?
		IF 54: Importance of Individuals and their Interactions	this is also collaboration...
		IF 55: Early Delivery	
Communication (C)	Communication methods, procedures and tasks related to the communication process between different teams, and with other stakeholders	IF56: Face to face communication	
		IF 57: Customer Communication	
		IF 58: Effective communication	
		IF 59: Team's coordination and communication	
		IF 60: Geographically distributed CCB (change control block)	Communication Sub-Factors.
	How IF are linked with category.	IF 61: Communication	
		IF62: Communication of changes in requirements	
		IF 63: Communication in teams with Large time difference	
		IF 64: Documentation of requirements	Same
		IF 65: Documentation	
		IF 66: Speed of communication ?	what is this speed.
		IF 67: Ease of communication	
Progress Measure (PM)	Factors related to measuring progress as well as those factors that influence the progress of the project are added to this category.	IF 68: Skilled developers	
		IF 69: Continuous organizational support	
		IF 70: Software Methodology	Similar to IF SDLC
		IF 71: HR management	
	Same as above.	IF 72: Strong leadership	
		IF73: Welcome requirements change, even at the end of development	
		IF 74: Positive team role	
		IF 75: Rapid response to changes in requirements rather than following a prescribed plan	
		IF 76: Collaboration between the software developer and user	
		IF 77: Technology Factors	which factors?
		IF 78: Autonomy and Decision-Making Power	

APPENDIX E

PROFILE DETAILS OF EXPERT REVIEWERS

Name	Domain	Exp in Yrs	Job Title	Organization	Email
Zohaib Ahmed	Academia	10+	Lecturer	Numl, Ibd Pak	zohaib.ahmed@numl.edu.pk
Salman Ahmed	Industry	10+	Software Engineer	Sourceability vertriebsgesellschaft mbH Germany	Christian.Meier@sourceability.com
Qandeel Fatima	Industry	5+	SQA Engineer	Immentia, Pakistan	Qandeelfatima6@gmail.com