Received June 28, 2019, accepted August 9, 2019, date of publication August 21, 2019, date of current version September 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936664

Comparative Analysis of Requirement Change Management Challenges Between In-House and Global Software Development: Findings of Literature and Industry Survey

SAJID ANWER[1], LIAN WEN[1], ZHE WANG[1], AND SAJJAD MAHMOOD[2]

[1]School of Information and Communication Technology, Griffith University, Brisbane, QLD 4111, Australia

[2]Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Sajid Anwer (sajid.anwer@griffithuni.edu.au)

This work was supported by Griffith University, Brisbane, QLD, Australia.

ABSTRACT Requirement changes are inevitable, and Requirement Change Management (RCM) is a complex process in software development. In-house software development and Global Software Development (GSD) are two widely used development approaches and there is a need to explore the RCM commonalities and differences in the two development approaches. The primary objective of this study is to identify the challenges that influence RCM in both approaches. First, we have implemented Systematic Literature Review (SLR) and identified 9 challenges that impact the general RCM process and 3 more challenges related to RCM with GSD. Second, we have conducted a questionnaire survey based on SLR results and collected feedback from 69 industry practitioners. The survey result indicates that there are four out of nine challenges, namely *impact analysis, requirement traceability, requirement dependency,* and *system instability* having the same impact in both in-house and GSD approaches. On the other hand, *cost/time estimation, artifacts documents management, user involvement, requirement consistency,* and *requirement prioritization* need more attention while implemented in GSD paradigm. Furthermore, regarding two important project management structures in GSD, *centralized project structure* and *distributed project structure,* the survey results reveal that all challenges have same impact except *user involvement* and *change control board management,* which are more important in centralized project structure. Lastly, the result from t-test indicates that both data sets retrieved from SLR and survey are close to each other. This study distinguishes RCM challenges in in-house and GSD approaches and in the context of two prominent project management structures followed in GSD projects. It would assist researchers by providing potential research directions and industry professionals to understand and implement RCM in different context more efficiently.

INDEX TERMS Requirement change management (RCM), global software development (GSD), challenges, global project structures, systematic literature review (SLR).

I. INTRODUCTION

Requirements Engineering (RE) is a critical phase in Software Development Life Cycle (SDLC), and *requirements development* and *requirements management* are the two key activities performed in this phase. Requirements development deals with requirements elicitation and specification, while requirements management deals with requirements analysis and requirements change [1], [2]. Requirements evolution can be defined as ''the tendency of requirements to change over time in response to evolving needs of customers, stakeholders, organizations and work environment'' [3]. Software development is a dynamic process and it is difficult to specify all the system requirements in the start as the requirements are subject to change. Factors such as customer needs, change in business goals, and government regulations contribute significantly to requirements changes. A study in the United Kingdom reported that 40% of the total software development process problems were related to the RE process [4]. Similarly, a study conducted by Standish (2017) revealed

The associate editor coordinating the review of this article and approving it for publication was Roberto Nardone.

that requirements change increased the project cost by three times and project time by two times [5]. Therefore, managing requirement change is crucial for the success of the project.

Requirements Change Management (RCM) is a complex process and driven by many factors such as organizational policies, market trends, and operational environments [6]. In recent years, several RCM models have been proposed in literature to improve RCM [7]–[9]. At the same time, a few reviews have been carried out to explore the different aspects of the RCM models [10]–[12]. There are two prominent software development approaches in practice namely in-house software development and Global Software Development (GSD). A GSD project is carried out by multiple teams in various locations of the world [13], [14]. The GSD paradigm offers many benefits including low cost development, access to skilled and quality workforce, and follow-the-sun development approach etc. [15]. However, GSD paradigm has failed to realize the anticipated outcomes, and got 45% projects success rate compared to 61% for co-located teams [16]. There are many reasons for these failures including cultural, temporal and communication issues [17]–[19], particularly, project management challenges across the borders. Hence the question arises, what are the differences and similarities between RCM in the two software development approaches? The comparative analysis between RCM challenges in in-house and GSD would assist practitioners to understand and implement RCM in different context more efficiently.

Furthermore, when people move from in-house software development to GSD, project management would become more challenge due to geographical and cultural differences [20]. In addition, there are two main types of global project management structures namely distributed (with local coordinators) structure and centralized structure [13]. To understand their impacts on RCM challenges is also interesting because it will help GSD practitioners to construct more suitable project structure for their projects.

Despite the importance of this problem, few studies have been found in literature to explore the comparison of the challenges associated with RCM, and the two different software development approaches. Similarly, little research has been reported to compare the impact of different project management structure in RCM challenges for GSD projects. The objective of this study is to identify and compare the challenges associated with RCM in both in-house and GSD paradigms. To identify the challenges, we have applied a Systematic Literature Review (SLR) and then conducted an industry survey. To accomplish the objective, we compile the following research questions:

RQ1: What are the challenges of RCM in in-house software development as reported in the literature?

Motivation: This question provides the starting point of this study through identifying RCM challenges of in-house software development approach reported in the literature.

RQ2: What are the challenges of RCM in in-house software development as identified in industry?

Motivation: To support findings of RQ1 and to analyze industry practices related to RCM in in-house approach, a survey is developed to collect data from industry professionals based on their experience.

RQ3: What are the challenges of RCM in GSD projects, as studied in the literature?

Motivation: This question enhanced the scope of this study and identified RCM challenges specifically related globally distributed projects.

RQ4: What are the challenges of RCM in GSD projects as identified from industry?

Motivation: To support findings of RQ3 and to analyze industry practices related to RCM process, a questionnaire survey was developed to collect data from industry professionals working on GSD projects based on their experience.

RQ5: What are the similarities and differences between RCM challenges in in-house software development and in GSD?

Motivation: The literature hasn't discussed the relationships between RCM challenges, and the two software development approaches. This research gap motivates us to tackle this question through industry survey.

RQ6: What are the similarities and differences of RCM challenges between centralized and distributed project management structures?

Motivation: Similarly, this question hasn't been addressed in literature, therefore, we try to find the answer through industry survey.

RQ7: Are there any differences between the challenges identified from the literature and the industry survey?

Motivation: This question helps people to realize the gap between research and the industry regarding RCM challenges.

The following part of this paper is organized as: section 2 introduces the background and related work, section 3 describes the research methodology of this paper. Section 4 presents the results and section 5 provides some discussions and implications. Section 6 shows the limitations of this study and finally, conclusions and future research directions are given in section 7.

II. BACKGROUND AND RELATED WORK

A. GLOBAL PROJECT MANAGEMENT STRUCTURE

Software development environment is continuously changing. Globalization, innovation, and market trend has dramatically impacted the software development environment [21]. The GSD has been growing steadily as many organizations aim to take advantage of using highly skilled workforce at a relatively reduced cost. Furthermore, GSD has the potential to reduce project's time to market by using different time zones to organize a 24/7 development model [13]. Many organizations that have tried GSD failed because of misunderstanding of requirements, poor global relationships among clients and vendors, high costs and overall poor services [22].

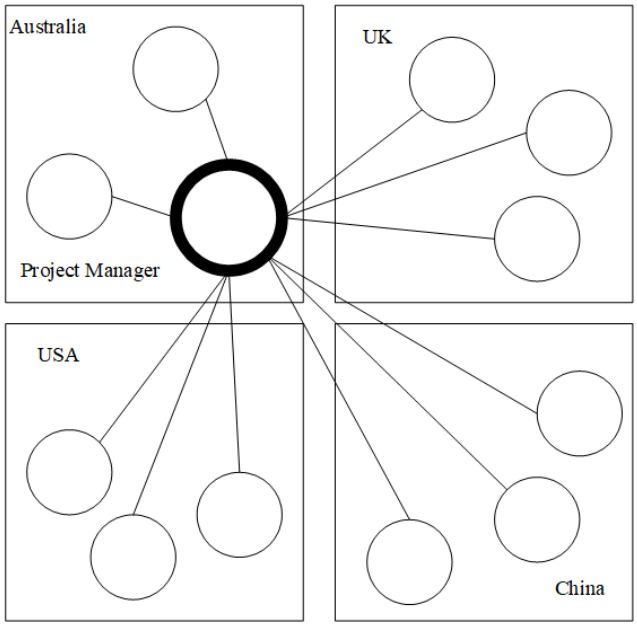


FIGURE 1. Centralized global project structure adopted from [13].

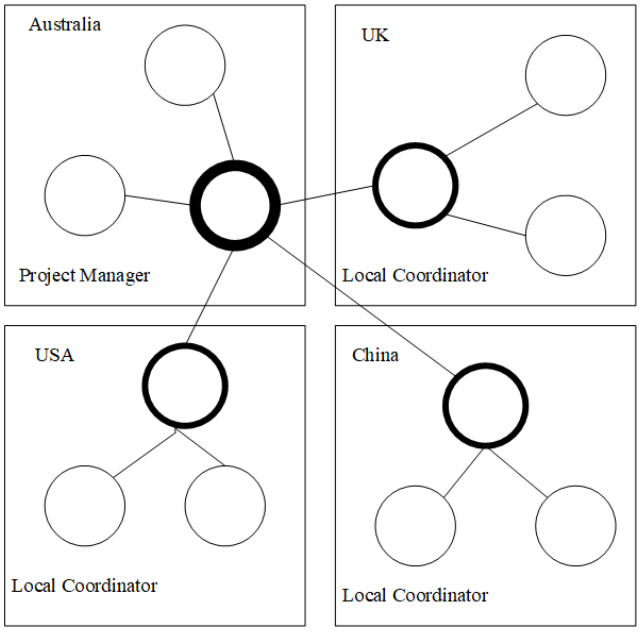


FIGURE 2. Distributed global project structure adopted from [13].

Global project structure is another crucial factor that has an impact on project management approach. The project size, organizational structure, maturity level of an organization in undertaking GSD projects, and experience of development team members working on GSD projects dominate the selection of project structure among different projects structures. There are two main types of global project structures namely, centralized project management and distributed with local coordinators [13]. In centralized project management, as shown in Figure 1, all or most of the team members report directly to a project manager who sits at one of the GSD sites and is responsible for most of the coordination and control task through collaborative tools. On contrary, in the distributed with local coordinator, as shown in Figure 2, the team members report directly to their local coordinators, who performs planning and execution of allocated task and report to the project manager at regular intervals.

B. REQUIREMENT CHANGE MANAGEMENT

This subsection briefly discusses the existing work related to RCM both in in-house and GSD.

A number of RCM models has been proposed in existing research. For example, Nurmuliani *et al.* [3] proposed an RCM model that covers most of the components of RCM process, including change impact analysis and possible causes of change. In another study, Bhatti *et al.* [23] proposed a six phase RCM process for both small and complex systems. However, both models missed some integral elements of RCM process, such as schedule adjustment according to the new change and updating of affected artifacts.

Imtiaz *et al.* [24] proposed an Requirement Change (RC) process model in a context of roles, activities, and artifacts involved in the RCM, and they validated the model completeness with pre and post conditions of each model activity. They used Unified Modelling Language (UML) to design system artifacts. However, their study missed the RC repository aspect used to keep record of all changes in RCM. Similarly, Tomyim and Pohthong [25] conducted a pilot study to address the RC, and they also used UML to model system components, however they did not discussed the change implementation and change verification and validation phase in RCM.

Niazi *et al.* [9] presented a five-step RCM model that implements CMMI level 2 practices and empirically validated their approach. Their model covers key elements in RCM including request initiation, request analysis, request implementation, artifacts updating and, verification and validation. However, communication of implemented change to concerned stakeholders, the last step of any RCM model is missing from their approach. In another study, Khan et al. [26] introduced a batch processing approach that overcomes the deficiencies in above-mentioned approach. However, this approach did not cover the change impact analysis in RCM.

Hussain *et al.* [7] proposed a technique to handle requested change in an informal way instead of following a defined formal process, means that, they will simply start implementing the requested change without the approval of change control board and storing the change in a change request pool. Their proposed approach is only suitable to handle graphical user interface problems, and it seems to be difficult to generalize for a complete problem domain. In another approach, Ahmed *et al.* [27] proposed an RCM model with the aim to minimize the impact of requested changes. In the first phase, they categories the requirements into different groups based on expert opinions. In the second phase, they analyze if a change is inevitable then try to minimize the change impact on other baselines (cost, schedule, scope etc.).

Overall, they cover most of the integral parts of RCM, however, subjectivism is involved in the first phase, which limits the applicability of the proposed approach.

RCM in GSD Projects: Li and Ali [8] proposed an ontology-based requirement management model for GSD projects. They use ontology to formalize the project information including functional requirements, non-functional requirements, project schedule, project budget, project teams' information (location, roles, contact details). In this approach they discussed RCM in the context of GSD paradigm. However, the communication framework developed and used in this study is very generic, specially the communication and collaboration need for RCM problem of GSD projects was not fully addressed. Similarly, another study proposed by Li and Ali [28], implemented a three-stage RCM model in GSD. They used an online shopping system case study to validate their approach. Several metrics were used to measure the effectiveness of the proposed model and to compare it with other models. The results show the practical significance of proposed approach and address some of the important aspects of RCM. However, the proposed approach lacks in two aspects, first, some essential elements in RCM such as change verification and validation are missing. Secondly, communication mechanism across different GSD sites and the consideration of different project management approaches in GSD are missing from the proposed approach as well.

Prikladnicki *et al.* [29] conducted a case study to address the requirements management challenges in GSD environment. The case study was conducted on two projects developed in a CMMI level 2 organization. They identified several challenges such as communication and collaboration, lack of trust, lack of face to face meetings etc. They also discussed RCM as an important aspect of requirement management in GSD in the context of above-mentioned challenges. They discussed a few of best practices to overcome the challenges of soft skills training, work standardization, and well-defined processes etc.

Mateen and Amir [30] proposed a framework to address RCM issues in GSD. The main motive of the proposed framework is to provide clear and unambiguous understanding of project knowledge among geographically dispersed stakeholders. They used semi-formal Unified Modeling Language (UML) diagrams to represent the project requirements and other related information. The proposed model addresses the change management in GSD environment, however, lacks some critical aspects. Firstly, tasks including change verification and validation, communication, and collaboration are not discussed. Secondly, the semi-formal notation UML, which is used to develop consistent understanding of project requirements among different stockholders, has some issues based on the nine principles evaluation [31]. A number of studies have been conducted to evaluate the UML notations including use case diagrams [32] and state chart diagrams [33]. The results show that the same UML notation may have different interpretations among different users. The same deficiencies exist in another framework [34] proposed to manage requirement change in GSD paradigm.

Lloyd *et al.* [35] developed a tool to manage the requirements change in distributed agile development. They used feature trees to represent project requirements. Eleven factors evaluation criteria are used to assess the proposed model and supporting tools. The proposed model is defined at an abstract level and missed some critical aspects such as change evaluation, change implementation, and change verification and validation. They only discussed change categorization, change implementation decision, and traceability.

C. NEED FOR EVIDENCE-BASED STUDY TO IDENTIFY CHALLENGES OF RCM FOR GSD PROJECTS

Nowadays, the central role of software intensive systems in everyday life emphasis the need of evidence-based software engineering (EBSE). EBSE helps researchers to ensure that their research is addressing the needs of industry practitioners and all concerned stockholders, and it also helps practitioners to make rational decisions about new techniques and emerging technology [36], [37]. Many researchers have performed empirical studies to understand the factors and challenges related to GSD. More recently, several systematic literature reviews and mapping studies has been carried out in the area of GSD. For example, Kroll et al. [38] conducted an SLR to review the challenges and best practices associated with GSD process. In another study, Niazi *et al.* [20] presented an SLR to discuss the challenges and best practices related to project management of GSD projects, and Hanssen *et al.* [39] conducted an SLR to study the application of agile methodologies in GSD projects.

Effective software requirements management is a key factor in work distribution and plays a critical role in system success [40]. Jayatilleke and Lie [12] conducted an SLR to investigate existing research/literature on causes of changes, processes, and techniques designed to manage requirements change. Formal and semi-formal processes of RCM have been critically evaluated. Similarly, Khan *et al.* [41], [42], used techniques such as SLR and questionnaire to investigate the communications risks of RCM process in GSD projects. They reported that, in the presences of geographical, socio-cultural, and temporal differences, communication and coordination is crucial in RCM process of globally distributed projects. Shafiq *et al.* [43] presented an empirical study to explore different aspects of requirements management and requirement change management in GSD and proposed a specialized project management technique to handle RCM problem in GSD projects.

Recently, Akbar *et al.* [11] conducted an SLR to investigate the success factors for RCM in GSD. They found 23 success factors including change acceptability, update requirements, information sharing etc. Similarly, in another study [44], they conducted SLR to investigate the challenging factors that negatively impacts RCM in GSD. They identified 15 challenging factors including lack of trust among GSD sites, etc. Both studies show the practical significance of

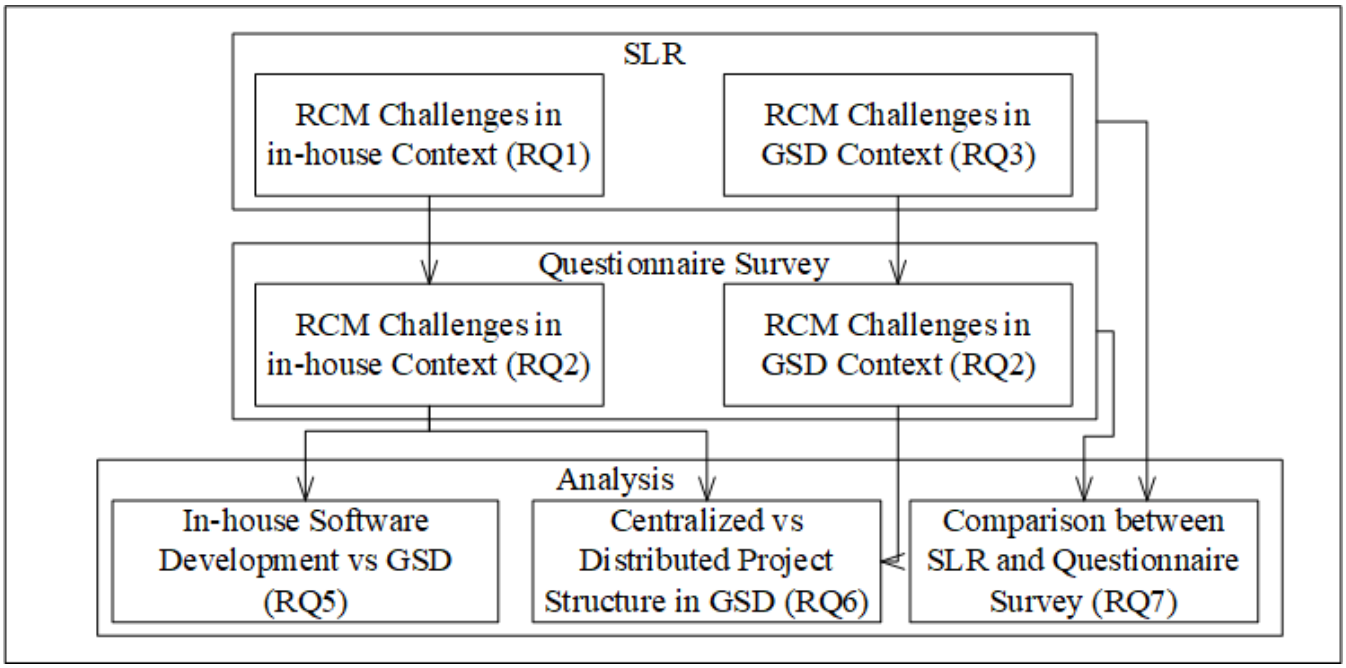


FIGURE 3. Research methodology.

RCM in GSD. However, there are some limitations in their study. Firstly, their results are more related to project management perspective [45], and missed the more general RCM domain; secondly, different project management structures involve in GSD projects are not discussed in their studies. Lastly, their research only collected data from literature without considering industry practitioner's opinions. In another study [46], they identified some challenges of RCM in GSD; these challenges includes reusability, change activity management, software artifacts management, and change automation. However, they only used ordinary literature review technique, which is not as systematic as SLR, and missed some relevant papers.

In Summary, even though some studies have already been carried out in this area, there are certain limitations that need to be addressed. First, how are the RCM challenges related to both in-house and GSD approach? Second, how is RCM process implemented in context of different project management structures such as distributed and centralized, largely followed in GSD projects? In this study, we will analyze practitioner's feedback and report results to answer these two open questions.

III. RESEARCH METHODOLOGY

We used two step approach to conduct this research as shown in Figure 3. In the first step, we used SLR technique to survey the literature published in the public domain and identify key challenges that impact RCM in in-house software development (RQ1), and additionally challenges that impact RCM in GSD (RQ3). In the second step, we used the results from the first step to develop questionnaire survey and collect feedback from industry practitioners (RQ2, RQ4). After that, we analyzed industry practitioners' feedback to explore different aspects of RCM challenges both in in-house and GSD paradigm (RQ5 and RQ6). Finally, we compared the data collected from the first two steps. (RQ7).

A. DATA COLLECTION VIA SYSTEMATIC LITERATURE

Systematic literature review is the most commonly used approach of evidence-based software engineering [37]. SLR is formally planned and systematically executed, and it provides guidelines to identify, analyze and interpret all available evidence with reference to specific research question [47], [48]. SLR is recommended to review published literature; it helps to collect evidence and identify research gaps through well-defined process. In this research, we followed Kitchenham and Charters [49] guidelines to execute an SLR process that contains three main phases: defining a protocol, conducting the protocol, and reviewing the protocol. In the first step, an SLR protocol was written to outline the complete process, and our protocol consisted of following elements, (i) identification of research questions, (ii) search strategy, (iii) study selection, (iv) quality assessment, and (v) data extraction and synthesis. The first element has been introduced in the introduction section, and the other elements are included in following parts of this section. The SLR was undertaken by a team of three researchers, one student and two academic staff members. To reduce personal biasness and improve SLR results reliability, inter-rater reliability test (Kendall's coefficient of concordance (W)) [50] was performed in all study selection phases.

TABLE 1. Keyword synonyms.

Keyword	Synonyms
Challenges	Challenges, problems, difficulties, complications, obstacles, barriers, hurdles, risks
Requirement Change Management	Requirement change, requirement volatility, requirement creep, requirement change management, requirement change difficulties, requirement change analysis, requirement change identification/type, requirement change models/processes
Global Software Development	Global software development, global project management, GSD, Offshore software development, distributed software development, offshore outsourcing global software engineering, distributed software engineering, GSE
In-house software development	In-house software development, Onshore software development, onsite software development

1) SEARCH STRATEGY

The search strategy for the SLR is based on the following four steps.

1) Construct search terms by identifying keywords from population, intervention, outcome and experimental design [49]. The results are:
   Population: Global software development, In-house software development.
   Intervention: Requirement change management challenges or barriers.
   Outcome: List of challenges in RCM of in-house and GSD projects.
   Experimental design: Systematic literature review, empirical studies, expert opinion.
2) Find synonyms of keywords. We used well reputed academic electronic databases to validate our keywords. The list of potential synonyms of each keyword is shown in Table 1.
3) Use boolean operators to connect major terms. In this step, we used Boolean operator OR to connect synonyms of each keyword and AND operator to connect major terms or keywords.
   CHALLENGES: ''Challenges'' OR ''problems'' OR ''difficulties'' OR ''complications'' OR ''obstacles'' OR ''barriers'' OR ''hurdles'' OR ''risks''
   REQUIREMENT CHANGE MANAGEMENT: ''Requirement change'' OR ''Requirement Volatility'' OR ''Requirement Creep'' OR ''Requirement Change management'' OR ''Requirement change difficulties'' OR'' Requirement change analysis'' OR ''Requirement change identification/type'' OR ''requirement change models/processes''
   GLOBAL SOFTWARE DEVELOPMENT: ''Global software development'' OR ''global project management'' OR ''GSD'' OR ''Offshore software development'' OR ''distributed software development'' OR ''offshore outsourcing'' OR ''Global Software Engineering'' OR ''Distributed Software Engineering'' OR ''GSE''
   IN-HOUSE SOFTWARE DEVELOPMENT: ''In-house software development'' OR Onshore software development'' OR ''onsite software development''
   By using AND operator, we define search strings for both in-house RCM process challenges and RCM challenges in GSD context.
   For RCM challenges in In-house software development:
   ''Challenges'' OR ''problems'' OR ''difficulties'' OR ''complications'' OR ''obstacles'' OR ''barriers'' OR ''hurdles'' OR ''risks''AND
   ''Requirement change'' OR ''Requirement Volatility'' OR ''Requirement Creep'' OR ''Requirement Change management'' OR ''Requirement change difficulties'' OR'' Requirement change analysis'' OR ''Requirement change identification/type'' OR ''requirement change models/processes'' AND
   ''In-house software development'' OR Onshore software development'' OR ''onsite software development''
   For RCM challenges in GSD context:
   ''Challenges'' OR ''problems'' OR ''difficulties'' OR ''complications'' OR ''obstacles'' OR ''barriers'' OR ''hurdles'' OR ''risks''AND
   ''Requirement change'' OR ''Requirement Volatility'' OR ''Requirement Creep'' OR ''Requirement Change management'' OR ''Requirement change difficulties'' OR'' Requirement change analysis'' OR ''Requirement change identification/type'' OR ''requirement change models/processes'' AND
   ''Global software development'' OR ''global project management'' OR ''GSD'' OR ''Offshore software development'' OR ''distributed software development'' OR ''offshore outsourcing'' OR ''Global Software Engineering'' OR ''Distributed Software Engineering'' OR ''GSE''
4) Verify search terms in electronic databases. In this step, some papers that are relevant to our research questions used to verify the search terms. The resources searched in this step include specific research databases, journals and conference proceedings. Based on the available access, the following electronic academic databases were used to search relevant primary studies. Because these research sources differ in their search mechanisms, we customized the search strings listed in previous step accordingly.
   - IEEE Access. https://ieeexplore.ieee.org
   - Science Direct. http://www.sciencedirect.com/
   - Springer Link. http://link.springer.com/
   - ACM Digital Library. http://dl.acm.org
   - Google Scholar. https://scholar.google.com/

2) STUDY SELECTION

The inclusion and exclusion criteria were used to select the primary studies retrieved from the academic databases and

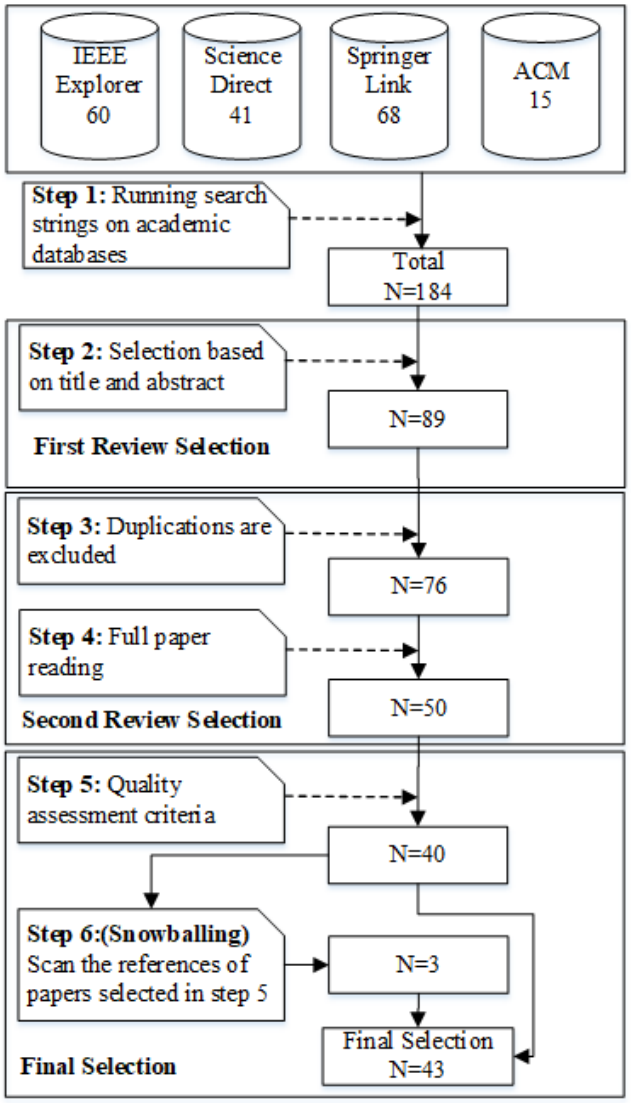


FIGURE 4. SLR process steps and number of studies in each step.

other electronic resources. The primary studies published or available online before 30, June 2018 were included in this research. The criteria used for including and excluding the primary studies are as follows:

Inclusion criteria:

- Publications that directly linked to our research questions.
- In case of duplications, the most completed version is included.
- Publication written in English.

Exclusion criteria:

- Peer-reviewed papers only- we excluded position papers, keynotes, panel discussions, editorials etc.
- Publications written in non-English.
- Publications without bibliographic information.

The number of studies selected at each stage of this SLR is shown in Figure 4. In the start, the search strings were executed on selected digital libraries, and 189 studies were retrieved. In the second step, 89 studies were selected based on paper title and abstract. The studies that cannot be decided based on their titles and abstracts are also retained for the next round of inspection. In the next step, duplicate studies (13) were excluded. In the fourth step, we shortlisted 50 out of 76 primary studies based on the full paper text. we only included the papers that are relevant to our research questions. In the next step, 40 studies were shortlisted based on quality assessment criteria. In the sixth step, we applied snowballing technique [56] to scan the references of the 40 papers selected in order to select more relevant studies. We found 15 more papers and we applied the same selection process on them and finally 3 papers were selected as primary studies. Unique identifiers were assigned to all papers, which are listed in Appendix A. A quality assessment had been applied for all the papers. The quality assessment evaluated the credibility and relevance of primary studies. The questions used to assess the quality of primary studies are shown in Table 2.

TABLE 2. Quality assessment criteria.

#	Questions	Possible Answers
1	Is there a rationale for why the study was undertaken? [51]	Y=1 N=0 P=0.5
2	Is the research goals are clearly reported? [52]	Y=1 N=0 P=0.5
3	Is the proposed technique clearly described? [53]	Y=1 N=0 P=0.5
4	Is the research results clearly described? [54]	Y=1 N=0 P=0.5
5	Is there is explicit discussion about the limitations of this research? [55]	Y=1 N=0 P=0.5

3) DATA EXTRACTION AND SYNTHESIS

In the data extraction step, two authors extracted the data using a pre-designed data extraction form and the third author validated the extracted data. A coding scheme based on grounded theory [57] was used to review the literature and conceptualize the RCM challenges.

We identified, labelled and grouped the related challenges to general categories and calculated the frequency. Furthermore, similar or related challenges were semantically compared and grouped under relevant categories.

Data Synthesis was performed and a list of RCM challenges from selected 43 studies were created. Initially 18 challenges for RCM process in in-house software development and additionally 6 RCM challenges for GSD were identified and are shown in appendix B.

Three researchers carefully reviewed the identified list of challenges and tries to reduce any biasedness and improve results validity independently. The initial list of 24 challenges were carefully reviewed and grouped into 12 main categories. The grouping of challenges was done based on the context in which those challenges were discussed in primary studies. For example, ''impact analysis'' and ''change consequences'' were grouped together in one category, as they were discussed in the same context of impact analysis.

B. DATA COLLECTION VIA QUESTIONNAIRE SURVEY

An empirical survey is an appropriate research methodology for collecting qualitative and quantitative data from a large group of participants by using techniques such as questionnaire or interviews [58], [59]. Based on the SLR results, we developed a questionnaire survey to ask industry professionals about the challenges that affect the RCM process in general (in-house) and specially in GSD projects according to their own experience. The questionnaire was designed to elicit importance about RCM challenges from industry professional's perspective. The participants were asked to note each challenge's relative importance as either strongly agree, agree, disagree, or strongly disagree. The survey participants who are using GSD as development approach were also asked what are their project management structure, either distributed with local coordinators or centralized which is normally used in GSD projects? Furthermore, participants can provide comments or add additional challenges through open-ended questions.

The questionnaire survey was firstly tested through a pilot study involving a list of five professionals from different organizations. Based on this pilot study, the final version of questionnaire survey was developed, as shown in appendix C. The questionnaire survey was divided into three sections: section one is related to demographic data, section two lists the challenges of RCM process in in-house software development, and section three presents the RCM challenges related to GSD projects. The participants were informed that the data would only be accessible to research team and only be used for research purposes.

1) DATA SOURCES AND ANALYSIS

In this study, the software industry practitioners who involve in managing GSD projects were the target population, however, it is always a challenging task to find suitable population frame for questionnaire survey [60], [61]. Snowballing technique [62] is used to address this difficulty and recruit participants for the questionnaire survey of this study. The research teams' personal contacts, LinkedIn groups were the initial potential participants and were asked to participate in this research. Once they agreed to participate, the link to the web-based survey was emailed to them. In the next step, these potential participants were asked to send this survey to their contacts who involved in GSD projects. We invited a total of 110 practitioners to participate in this research, and 69 of them completed the survey, giving a response rate of 63%. The responses correctness and completeness were assured through a manual review process.

The respondents came from seven different countries which includes Australia, Pakistan, India, Ireland, Saudi Arabia, United Arab Emirates, and China. These respondents' organizations were involved in business intelligence, data processing, and embedded systems. The respondents' roles in their organizations ranged from software engineer to project manager with an average experience of 5 years in in-house software development and 4 years in GSD. The demographics information of all the participants is provided in appendix D.

The frequency and percentage of each challenge was then presented in frequency tables. Frequencies were used to compare variables within and across the groups and are useful for ordinal, nominal and numeric data. Some other statistical techniques like, chi-square test, t-test of independence [63], [64] were used to analyze different aspects of participants of data.

IV. RESULTS

This section discusses the findings of Systematic Literature Review (SLR) and survey in the context of the research questions defined in the previous section. In the last subsection, a comparison of the two data sets is also discussed.

A. SLR RESULTS

The total 43 primary studies were selected from the SLR. Prior to discussing the SLR findings and analysis for each research question, we give a through overview of the general characteristics of primary studies.

1) OVERVIEW OF THE STUDIES

This subsection presents general characteristics of primary studies, including year of publication, type of source, and research method.

Figure 5 shows the number of selected studies published per year from 1996 to 2018. In the context of publication years, it is noteworthy that research related to RCM gained attention after 2005. We could find only three papers published in this domain before 2005. Another worth mentioning point is that at least one journal paper was published every year after 2008 except 2015. Lastly, we found 6 studies in 2017 and 3 studies in the first six months of 2018, which reflects researchers' growing interest in RCM domain. In the context of source type, the majority of the studies are conference papers (53%; 23 studies), followed by journal publications (40%; 17 studies), and book chapters (7%; 3 studies).

Figure 6 presents the distribution of published studies across empirical research methodologies. The primary studies selected in this research used different research strategies (literature review, case study, SLR, and survey/interview), which are commonly used in empirical software engineering domain [65], [66]. The results depict that the majority of study methodology was case study (63%; 27 studies) for empirical investigation, followed by survey/interview (16%; 7 studies), literature review (14%; 6 studies), and SLR (7%; 3 studies). It is also worth mentioning that empirical investigation, through survey/interview with industry professionals, has gained attention during the past half a decade, that means the importance of industry practitioners' feedback in software engineering research is appreciated by academics. Another important point that should be emphasized is the absence of SLR. Only three SLR have been conducted in RCM research.

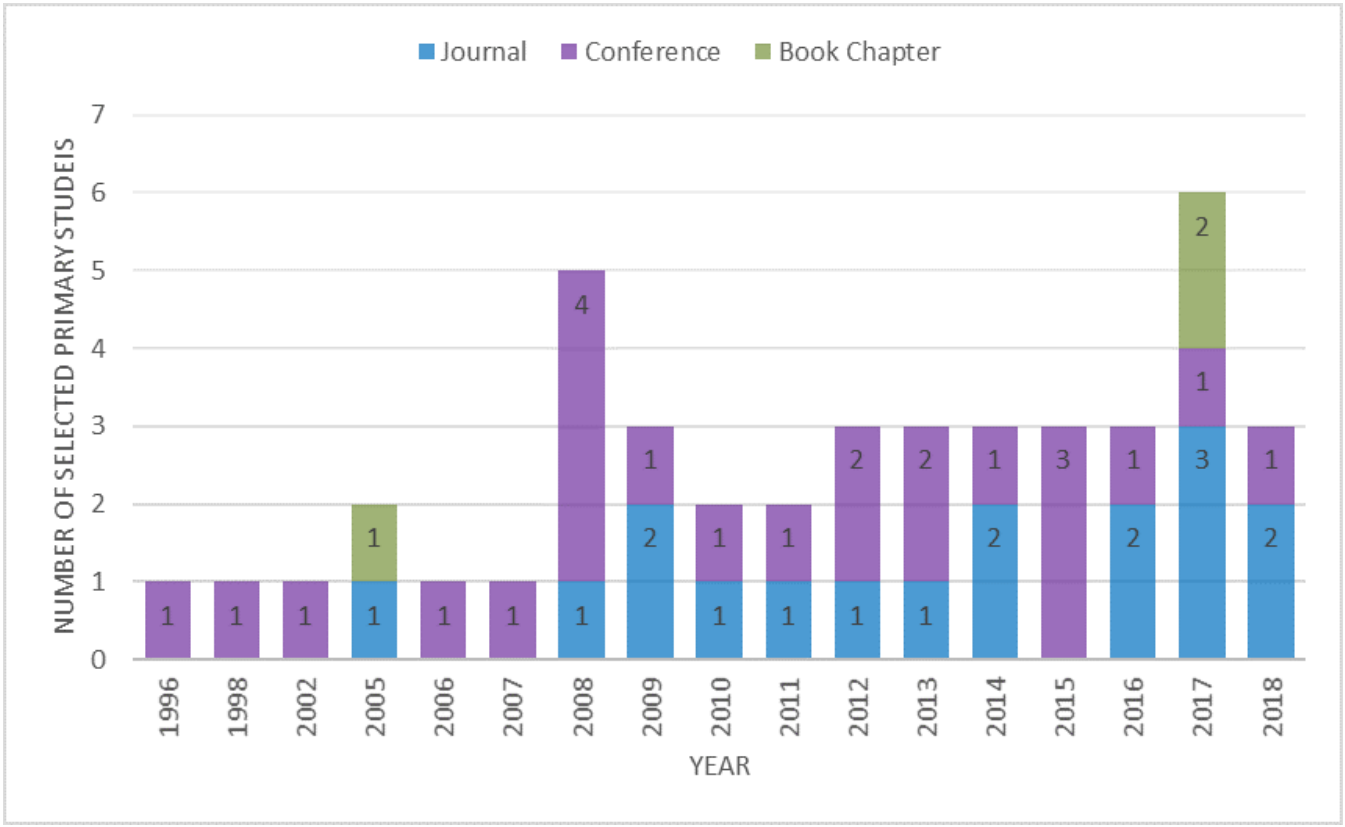


FIGURE 5. Number of selected studies published per year and their distribution over source type.

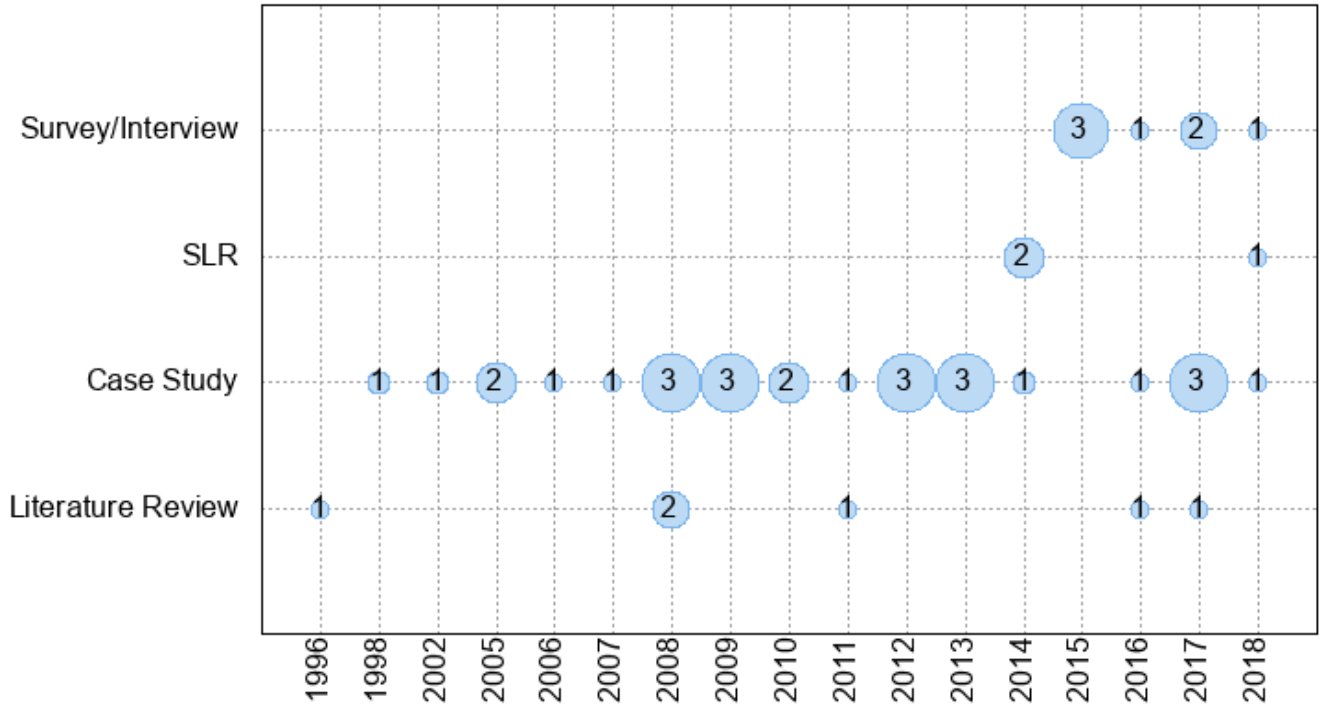


FIGURE 6. Bubble plot with year of publication and research method.

In order to reduce researchers' bias, inter-rater reliability test was performed. In this process, three independent reviewers selected a random sample of five primary studies in first selection round and performed initial selection process.

TABLE 3. Primary studies selection data.

Resource	Total Results	First Review Section	Second Review Selection	Final Selection
IEEE Xplore	60	29	17	16
ACM	15	10	06	03
Science Direct	41	16	11	10
Springer	68	36	16	14
Total	184	89	50	43

TABLE 4. RCM challenges identified via SLR.

Challenge	Frequency (n=32)	Percentage
Impact Anaysis	21	67
Cost/Time Estimation	8	25
Artifacts Documents Management	8	25
Requirements Traceability	7	22
Requirements Dependency	5	16
Requirements Consistency	4	12
Change Prioritization	2	6
User Involvement	2	6
System Instability	1	3

Similarly, the same steps were followed in the next rounds of study selection. We used the non-parametric Kendall's coefficient of concordance (W) [50] to evaluate the inter-rater agreement between reviewers. The W value range from 0 to 1, 1 indicates strong agreement and 0 indicates perfect disagreement. The value of W for randomly selected five studies from the first selection round was 0.84 ($p = 0.002$). Similarly, the value of W was 0.9 ($P = 0.04$) and 0.95 ($p = 0.03$) for the next two selection rounds respectively, and in the snowballing process, the W value was 0.97 ($p = 0.045$). These results indicate strong agreement between the findings of primary researchers and independent reviewers.

2) SYSTEMATIC LITERATURE REVIEW FINDINGS OF RCM PROCESS IN IN-HOUSE SOFTWARE DEVELOPMENT APPROACH (RQ1)

This subsection discusses the SLR findings related to RQ1, in which we intend to explore the challenges that impact the RCM process in in-house software development. The initial automated search resulted 184 papers. During the first review phase, we reviewed the titles and abstracts of the 184 papers, and shortlisted 89 papers. In the second review phase, the full text of the 89 papers were reviewed and selected 50 papers. Duplicate papers were also removed during this phase. Then we applied quality assessment criteria listed in Table 2 on the 50 papers; the papers failed to satisfy minimum quality score 50% [54] were excluded. The complete list of selected papers from the SLR and their corresponding quality scores are given in appendix F. We finally shortlisted 43 papers shown in Table 3. Among them, 32 papers deal with in-house (general) RCM challenges and 11 of them with RCM in the GSD. In this research, we have identified 9 challenges (shown in Table .4) that impact RCM process in in-house software development.

Among the challenges, the most cited challenge for RCM process in in-house context is impact analysis (67%). In RCM process, once proposed change has been identified, further analysis is required to understand the consequences of requested change on the software system, such as new system states, consistency with existing business goals and impact on other operational constraints [7], [67]. Bohner and Arnold defined impact analysis as "the activity of identifying consequences, including the side effects and ripple effects of a change". Impact analysis helps to understand the protentional effects of requested change before the actual change is implemented [69]. Incorrect understanding of a proposed change could increase project cost or even leads to system failure.

Cited by 25% of the primary studies, cost and time estimation, which is usually carried out at the beginning of a project, is a critical aspect of project management. However, the proposed change also impacts the defined project schedule and estimated cost and introduce extra cost to implement the requested change. Cost and time estimation are collectively considered as effort estimation; however, the transformation of these aspects with each other is not a straight forward process [70], [71]. The first step of this process is to calculate the software size, which is the most important aspect impacting effort estimation. Different techniques, such as functional point analysis and line of code etc., can be used to calculate software size [72]. After that, man hours are calculated based on project size, and lastly the number of man hours is multiplied by an hourly rate to calculate the total effort required to implement the proposed change.

Artifacts documents management is another key challenge, which is studied and cited by 25% of existing research. Software Development Life Cycle (SDLC) consists of several phases and each phase output, such as specification document, and design document, etc.) is recorded as a phase product. In RCM process, each phase product requires modification as a result of the proposed change, in order to maintain consistency across all artifact documents [73]. The management of SDLC phases product, such as requirement document, design document, source code, and testing document, is crucial, especially if a change occurs in the late phases of SDLC, such as during testing.

Requirements traceability is another key challenge faced in RCM process cited by 22% of primary studies. Requirements traceability can be formally defined as "the ability to describe and follow the life of a requirement in both forward and backward direction" [74]. Traceability analysis is one of the efficient ways to understand the impact of proposed change and is used for impact analysis [75]. Requirements traceability also helps to understand the dependency between requirements, which is another key challenge of RCM process, and it was cited by 16% of primary studies.

Requirements consistency is another key challenge that impacts RCM process, cited by 12% of the primary studies. Consistency analysis happens during the change analysis phase, which is usually executed after change identification. Requirements consistency can be defined in a number of

TABLE 5. RCM challenges analysis in the context of empirical studies.

Challenge	Empirical Studies Classification				Primary Studies
	Case Study	Survey/ In-terview	SLR	Literature Review	
Impact Anaysis	13	3	1	4	A2, A9, A10, A11, A12, A13, A14, A15, A17, A20, A22, A23, A24, A25, A30, A31, A32, A33, A34, A35, A39
Cost/Time Estimation	3	1	1	3	A2, A3, A10, A14, A20, A25, A33, A43
Artifacts Documents Management	5	1	0	2	A6, A10, A16, A23, A24, A28, A32, A43
Requirements Traceability	5	1	0	1	A1, A17, A27, A31, A33, A34, A35
Requirements Dependency	3	0	0	2	A2, A5, A9, A13, A25
Change Prioritization	2	0	0	0	A1, A2
User Involvement	0	0	1	1	A16, A20
System Instability	0	0	1	0	A20

ways, such as ''not two or more requirements in a specification contradict with each other'' [76], and ''requirements should be understood precisely in the same way to every person who reads it'' [77]. Researchers have used a number of techniques (including semi-formal i.e. using UML diagrams, formal i.e. first order logic, and pure logic etc.) to address this issue. In requirement evolution, either new requirements, or changes in existing requirements make requirements consistency one of the major issues in this process [78].

Change prioritization cited by 6% of the primary studies, is crucial to meet deadline and business goals. Every system requirement contributes to strategic business goals and deliver some financial value to the organization. Prioritization is measured based on the urgency, impact, and risk involved with the proposed change. Prioritization of proposed changes is very important in RCM, particularly when strategic business goals are depended on a given time frame [79].

User involvement is another key challenge of RCM, that is cited by 6% in existing research. According to standish report (2014), user involvement is top ranked software project success factor among other 10 success factors [5]. RCM requires user feedback, especially when the requested change proposed by one of the system users. User involvement plays a critical role in successful execution of an RCM process and ultimately in project success [81]. Finally, system instability is another key challenge that impacts the RCM process and referenced in 3% of the primary studies. A requested change can be easily handled before a system is put in the live environment, however, the RCM process becomes cumbersome, when the system is already in live environment. The key success indicator of an RCM process is to provide uninterrupted services to the customers during change implementation process.

Table 5 presents RCM challenges in in-house software development, their source primary studies, and the corresponding empirical study strategies. In our study, the impact analysis is the highest cited challenge. We found 62% out of 21 papers that mentioned impact analysis as a key challenge faced in RCM process using case study for empirical investigation. Similarly, other empirical techniques, such as survey/interview, literature review, and SLR, were used by 14%, 19%, and 4% of the primary studies for empirical investigation respectively. Cost/time estimation is the second most cited challenge, and 38% of the primary studies that listed cost/time estimation as a key challenges of RCM process used case study and literature review for empirical investigation. Oher empirical techniques, such as interview/survey and SLR were used by 12% of the primary studies for empirical investigation.

TABLE 6. RCM-GSD challenges identified via SLR.

Challenge	Frequency (n=11)	Percentage
Communication and Coordination	10	91
Knowledge Management and Sharing	8	73
Change Control Board Management	2	18

3) SYSTEMATIC LITERATURE REVIEW FINDINGS of RCM IN GSD CONTEXT (RQ3)

This subsection discusses the challenges that specifically impact RCM in GSD (shown in Table 6). In the previous subsection, we have identified nine challenges, which are general RCM challenges and related to in-house software development. In this subsection, we discuss three more challenges that are only relevant to RCM in GSD. In total, we need to consider twelve challenges, while implementing proposed changes in GSD. The GSD has been increasingly used for developing software systems efficiently and effectively by capitalizing the talent pool across the world [82]. However, there are certain issues, such as time zone difference etc., that overshadow these benefits.

In our study, communication and coordination, cited by 91% of the primary studies, is the highest cited challenge that impacts RCM in GSD. Ineffective communication in software development process is one of main reasons for software projects failure [83]. In GSD, communication and coordination are usually discussed in two different contexts; one is communication between different team members working on system development and the other one is communication between client and development teams [84]. The geographical, cultural, and social differences makes communication and coordination process more difficult while implementing RCM in GSD projects [42].

TABLE 7. RCM-GSD challenges analysis in the context of empirical studies.

Challenge	Empirical Studies Classification				Primary Studies
	Case Study	Survey/Interview	SLR	Literature Review	
Communication and Coordination	6	2	2	0	A4, A18, A19, A21, A26, A29, A36, A37, A41, A42
Knowledge Management and Sharing	6	1	1	0	A4, A18, A21, A36, A37, A38, A41, A42
Change Control Board Management	1	1	0	0	A21, A29

Knowledge management and sharing, cited by 73% of the primary studies, is another key challenge that impacts RCM in GSD. In RCM, the development teams may reside in different parts of the globe, work on the same proposed change, and need to access software artifacts with precise, accurate, and unified understating. The geographical and cultural differences between development teams and clients makes this process very cumbersome [85]. The development teams also need to communicate and collaborate with the client who propose new requirements or modify existing requirements.

Finally, change control board management, cited by 18% of the primary studies, is another challenge that impacts an RCM process in GSD projects. In in-house software development, project managers, along with other members, act as a change control board and accomplish the proposed change approval process. However, In GSD, projects are usually managed under one of two project management structures: centralized or distributed with local coordinators and the formation of change control board would be different. The issues, such as who will be included in the CCB and how CCB will work, need to be addressed for RCM to success in GSD context.

Furthermore, Table 7 presents RCM challenges in GSD, their source of the primary studies, and corresponding empirical study strategies. In this research, the communication and coordination is the highest cited challenge faced in RCM in GSD. We found 60% out of 10 papers used case study for empirical investigation, while survey/interview and SLR where applied by 20% of the papers respectively. Knowledge management and sharing is the second most cited challenge and 75% of the primary studies that listed knowledge management and sharing as a key challenge used case study for empirical investigation. Oher empirical techniques, such as interview/survey and SLR, were used by 12% of primary studies for empirical investigation.

B. QUESTIONNAIRE RESULTS AND ANALYSIS

We conducted a questionnaire survey to compare SLR findings with industry practices. We performed number of different analyses based on the industry practitioners' feedback.

1) INDUSTRY SURVEY FINDINGS OF RCM PROCESS (RQ2)

This subsection presents industry practitioners' opinions about RCM challenges. We received feedback from 69 industry practitioner's and summary of the feedback is shown in Table 8. The participants' responses were divided into two groups: positives responses and negative responses. Positive feedback indicates that the listed challenges influence RCM, while negative feedback shows that the listed challenge has no impact on RCM. Because these challenges are common to both in-house and GSD development, we have considered all participants' observations in frequency analysis.

TABLE 8. RCM process challenges analysis based on questionnaire survey.

Challenge	Organizations' Observation (n=69)					
	Positive			Negative		
	SA	A	%	D	SD	%
Impact Analysis	23	46	100	0	0	0
Cost/Time Estimation	30	33	91	6	0	9
Requirements Traceability	16	47	91	6	0	9
System Instability	23	40	91	5	1	9
Requirements Dependency	18	43	88	8	0	12
Change Prioritization	20	37	83	11	1	17
User Involvement	24	31	80	13	1	20
Requirements Consistency	23	31	78	15	0	22
Artifacts Documents Management	13	41	78	14	1	22

Note: Strongly Agree (SA); Agree (A); Disagree (D); Strongly Disagree (SD)

More than 90% of the respondents agreed that impact analysis, cost/time estimation, requirements traceability, and system instability are the key challenges that impact RCM in in-house software development. For example, one of the participants supported his positive response for impact analysis with the following comment:

"The success/failure of an RCM process heavily depends upon the understanding of requested change impact on other baselines such as cost, time, artifacts documents and other requirements. We used number of different techniques such as cross-matrix, trees to understand the impact of proposed change". Team Lead Similarly, cost/time estimation is another key challenge, and received 91% of positive response from the participants. Cost and time estimations are normally used interchangeably in software engineering as a key factor that determines the project success/failure. One of the participants supported his response with the following comment:

"Cost/Time estimation is always a challenging task in software development process, and it becomes more difficult in the requirement change process. In RCM process, it is very challenging to estimate time for requested change with normally used techniques such as functional point analysis, line of code etc. Therefore, we normally use combination of different techniques to measure time for proposed change." Project Manager The above feedback indicates that there is need to develop customized technique that can be used in RCM process.

TABLE 9. RCM-GSD challenges analysis based on questionnaire survey data.

Challenge	Organizations' Observation (n=69)					
	Positive			Negative		
	SA	A	%	D	SD	%
Communication and Co-ordination	27	18	100	0	0	0
Knowledge Management and Sharing	13	31	98	1	0	2
Change Control Board Management	17	21	84	6	1	16

Note: Strongly Agree (SA); Agree (A); Disagree (D); Strongly Disagree (SD)

Requirements traceability is another key challenge, and 91% of respondents agreed that it impacts an RCM process. In industry, some of the participants considered it as a supportive element of impacts analysis, but most agreed that requirements traceability itself needs attention is RCM, and it helps to understand the requirement life and scope both forward and backward direction.

Furthermore, system instability is another key challenge, and received 91% of positive response. This challenge becomes more critical if the change request comes after the system is put in the live environment at the client side. It is crucial for the vendor to keep the system functioning and to provide uninterrupted services to the system users and other stakeholders. One of the participants supports his positive response with the following comment:

"It is very challenging for us to control behavior of the system during RCM process. we normally try to implement proposed change without affecting system working, but we put system offline, if the requested change impacts key functionally or key requirements of the system." Development Lead.

Similarly, requirements dependency, change prioritization, and user involvement received 88%, 83%, and 80% positive response respectively. One of the participants support his positive response about change prioritization with the following comments:

"It is very important to decide the implementation plan for proposed change in RCM process, and we usually use dependency maps to prioritize the requested changes". Requirements Manager.

2) INDUSTRY SURVEY FINDINGS OF RCM IN GSD CONTEXT (RQ4)

This subsection presents industry practitioners' opinions about the RCM challenges that are specific to GSD projects. The listed challenges are additional to the RCM process challenges discussed in the previous subsection as general RCM challenges. We received data from a total of 69 industry practitioners and 45 of them are involving in GSD. A summary of the data is presented in Table 9.

It is interesting to note that industry practitioners support our research findings from SLR with two out of three challenges receiving more than 90% of positive response. All the participants agree that communication and coordination is the key challenge for RCM in GSD. The key difference between in-house software development and GSD is the geographical or physical location of development teams, which makes communication and coordination crucial for project success in GSD. One of the respondents supported his positive response with the following comment:

"Communication and coordination is key success factor in GSD projects. We always try to minimize the impact of time zone and geographical difference by using variety of communication media such as teleconferences, instant messaging". Project Manager.

Similarly, knowledge management and sharing, and change control board management received (98%) and (84%) of positive response respectively from industry practitioners. One interesting comment we have received as:

"We usually struggled to share and convey the similar understanding of software artifacts between different development teams resides at different part of the globe. We used different cloud-based tools such as AWS cloud9 for sharing software artifacts". Team Lead.

3) INDUSTRY SURVEY FINDINGS ANALYSIS BASED ON IN-HOUSE AND GSD APPROACH (RQ5)

This subsection discusses industry practitioners' feedback on the two widely used software development approaches, namely in-house software development and GSD. In the survey, a demographic field asked about experiences with RCM in both development approaches. We applied the chi-square test of independence to compare the two categorial values from a single population. The chi-square results are shown in Table 10. We analyzed the data based on the following hypothesis:

Null Hypothesis: There is no significant association between the identified list of RCM process challenges and software development approaches.

The comparison of RCM process challenges from the in-house software development and GSD approach indicates that there are more differences than similarities, as shown in Table 10. The p-value for impact analysis, requirement dependency, requirement traceability, and system instability are greater than 0.05, which indicates that there is no relationship between these RCM process challenges and development approaches; therefore, we will accept the null hypothesis.

On the other hand, the p-value for cost/time estimation, artifacts documents management, requirement consistency, requirement prioritization, and user involvement are less than 0.05, which indicates that these RCM process challenges are different in different software development approaches; therefore, we will reject the null hypothesis. The industry practitioners' feedback also reveals that these RCM challenges require extra effort while working in GSD context. The results show that 95% of practitioners are either strongly agreed or agreed that cost/time estimation is more challenging in GSD, while the same opinion was given by 83% of practitioners working on RCM problems in in-house

TABLE 10. Chi square test results of industry data (in-house vs GSD).

Challenge	In-house (n=24)				GSD (n=45)				Chi-square test (linear-by-linear association) α=0.05		
	SA	A	D	SD	SA	A	D	SD	X2	df	*p*-value
Impact Analysis	11	13	0	0	13	32	0	0	1.952	1	0.162
Cost/Time Estimation	**7**	**13**	**4**	**0**	**23**	**20**	**2**	**0**	**4.493**	**1**	**0.034**
Artifacts Documents Management	**4**	**10**	**9**	**1**	**9**	**31**	**5**	**0**	**4.993**	**1**	**0.025**
Requirements Traceability	5	18	1	0	11	29	5	0	0.058	1	0.81
Requirements Dependency	7	12	5	0	11	31	3	0	0.387	1	0.534
Requirements Consistency	**7**	**7**	**10**	**0**	**16**	**24**	**5**	**0**	**3.919**	**1**	**0.048**
Change Prioritization	**6**	**9**	**8**	**1**	**14**	**28**	**3**	**0**	**5.246**	**1**	**0.022**
User Involvement	**6**	**9**	**8**	**1**	**18**	**22**	**5**	**0**	**5.546**	**1**	**0.019**
System Instability	7	15	2	0	16	25	3	1	0.049	1	0.825

Note: Strongly Agree (SA); Agree (A); Disagree (D); Strongly Disagree (SD)

TABLE 11. Chi square test results of industry data (centralized vs global project structure).

Challenge	Centralized (n=21)				Distributed (n=24)				Chi-square test (linear-by-linear association) α=0.05		
	SA	A	D	SD	SA	A	D	SD	X2	df	*p*-value
Impact Analysis	8	13	0	0	5	19	0	0	1.588	1	0.208
Cost Estimation	9	10	2	0	14	10	0	0	2.026	1	0.155
Artifacts Documents Management	4	16	1	0	5	15	4	0	0.37	1	0.543
Requirements Traceability	5	14	2	0	6	15	3	0	0.01	1	0.919
Requirements Dependency	6	13	2	0	5	18	1	0	0.022	1	0.882
Requirements Consistency	7	12	2	0	9	12	3	0	0.004	1	0.951
Change Prioritization	5	14	2	0	9	14	1	0	1.249	1	0.264
User Involvement	**12**	**8**	**1**	**0**	**6**	**14**	**4**	**0**	**4.968**	**1**	**0.026**
System Instability	8	11	2	0	8	14	1	1	0.145	1	0.703
Communication and Coordination	**16**	**5**	**0**	**0**	**11**	**13**	**0**	**0**	**4.205**	**1**	**0.04**
Knowledge Management and Sharing	4	17	0	0	9	14	1	0	0.931	1	0.335
Change Control Board Management	**11**	**9**	**1**	**0**	**6**	**12**	**5**	**1**	**5.244**	**1**	**0.022**

Note: Strongly Agree (SA); Agree (A); Disagree (D); Strongly Disagree (SD)

software development. Furthermore, 89% of the industry practitioners in GSD are either strongly agreed or agreed that artifacts documents management is more challenging in GSD than in in-house software development approach, where 58% practitioners in in-house development gave the same feedback. Similarly, 89%, 90% and 89% of industry practitioners either strongly agreed or agreed that requirements consistency, change prioritization, and user involvement are more challenging while working on RCM in GSD projects, in contrast to in-house software development, where 58%, 63% and 63% of industry practitioners gave the same opinion for these three RCM process challenges.

4) INDUSTRY SURVEY FINDINGS ANALYSIS BASED ON CENTRALIZED AND DISTRIBUTED GLOBAL PROJECT STRUCTURE (RQ6)

This subsection discusses industry practitioners' feedback in the context of global project structures used for GSD projects. In the questionnaire survey, a demographic field asked for the corresponding organizational management structure (i.e. centralized or distributed) mostly followed in GSD projects. The gathered data reflects the practitioners experience for centralized and distributed structured organizations. In the survey, we received data from a total of 69 participants, and 45 of them are working in GSD. In GSD, 21 of them are working in centralized structure and 24 are working in distributed with local coordinators project structure. We applied the chi square test of independence on those 45 participants' feedback to compare the two categorical variables (centralized or distributed) from a single data set. Chi square results are shown in Table 11. We analyzed our data based on the following hypothesis:

Null Hypothesis: There is no significant association between the identified list of RCM process challenges and GSD project management structure.

A comparison of both general RCM challenges and challenges specific in GSD, from the centralized and distributed project management structure, indicates that there are more similarities than differences between the two GSD project management structures. The *p*-value for impact analysis, cost/time estimation, requirement traceability, artifacts documents management, requirement dependency, requirement consistency, change prioritization, system instability, and knowledge management and sharing is greater than 0.05. Therefore, we accept the null hypothesis and infer that these RCM challenges are independent to the two different project management structures.

On the other hand, the *p*-values for user involvement, communication and coordination, and change control board management are 0.026, 0.040, and 0.022 respectively. The p-value for user involvement, communication and coordination, and change control board management is less than 0.05, which indicates the significance of the results; therefore, we reject the null hypothesis. The 95% of industry practitioners who adopted centralized project management structure are either strongly agreed or agreed that user involvement is more

TABLE 12. Comparison of two data sets of RCM process challenges.

Challenge	SLR (n=32)	%	Rank	SA (n=69)	%	Rank
Impact Analysis	21	67	1	23	33	3
Cost/Time Estimation	8	25	2	30	43	1
Artifacts Documents Management	**8**	**25**	**3**	**13**	**19**	**9**
Requirements Traceability	**7**	**22**	**4**	**16**	**23**	**8**
Requirements Dependency	5	16	5	18	26	7
Requirement Consistency	4	12	6	23	33	5
Change Prioritization	2	6	7	20	29	6
User Involvement	**2**	**6**	**8**	**24**	**35**	**2**
System Instability	**1**	**3**	**9**	**23**	**33**	**4**

TABLE 13. Comparison of two data sets of RCM-GSD challenges.

Challenge	SLR (n=11)	%	Rank	SA (n=45)	%	Rank
Communication and Coordination	10	91	1	27	60	1
Knowledge Management and Sharing	8	73	2	13	29	3
Change Control Board Management	2	18	3	17	47	2

challenging compared with distributed project management structure, where 80% of industry practitioners gave the same opinion. Furthermore, 95% of industry professionals who followed centralized project management approach are either strongly agreed or agreed that change control board management is more challenging than in distributed project management structure, where 75% of industry practitioners give the same opinion. Similarly, 100% of industry practitioners are either strongly agreed or agreed that communication and coordination is a challenging factor in both centralized and distributed project management structures followed in GSD projects.

5) COMPARISON OF TWO DATA SETS (RQ7)

This subsection compares the two data sets from SLR and the survey using T-test of independence. In previous sections, we discuss the list of challenges that impacts the RCM process in in-house, and RCM in GSD from both the published literature and the survey. In the survey, the participants were asked to give their opinion for each challenge by choosing one of the four options: strongly agree, agree, disagree, or strongly disagree. In Table 12, we present the rank of each RCM process challenge based on the SLR and the survey results. We only take percentage of strongly agreed option from the survey results. Furthermore, Table 13 shows each RCM challenge in GSD context and questionnaire survey results.

The comparison shows that there are some similarities and differences between SLR and the survey results, as shown in Table 12. A critical analysis of two data sets shows that the researchers and industry practitioners agree on the key challenges that impact RCM process in in-house and RCM in GSD. The majority of the RCM process challenge identified from the literature received the similar response from industry practitioners. However, artifacts documents management ranks third in SLR data, while ranks ninth in the survey data.

TABLE 14. Group statistics of RCM-in-house challenges.

	Type	N	Mean	Std. Deviation	Std. Error Mean
Challenge	SLR	9	20.22	19.44	6.48
	Survey	9	30.44	7.13	2.38

TABLE 15. Group statistics of RCM-GSD challenges.

	Type	N	Mean	Std. Deviation	Std. Error Mean
Challenge	SLR	3	60.67	38.03	21.96
	Survey	3	45.33	15.57	8.99

Furthermore, user involvement ranks 8th in SLR data, but ranks 2nd in questionnaire data. Similarly, system instability ranked ninth in SLR data but ranked fourth in questionnaire data. Furthermore, the list of challenges particularly in GSD received a similar response from industry practitioners, as shown in Table 13. However, change control board management ranked third in SLR data, but ranked second in questionnaire data. We applied independent t-test to quantify the significance of similarities between the challenging factor identified from SLR and questionnaire survey. Our hypothesis is as follows:

Null hypothesis: The population variances of two data sets (SLR and questionnaire survey) are equal.

In this study, we have two data sets (data from the SLR and data from the survey) for two different categories (general RCM challenges and challenges for GSD only). Accordingly, we performed two different independent t-test to compare both data sets for both categories. The descriptive statistics of the two data sets used for this study for both categories are shown in Table 14 and Table 15 respectively, whereas Table 16 and Table 17 show the independent sample t-test results.

The t-test assumes that the variability of each group is approximately equal. This assumption will be verified by using Levene's test significant level; as the p-value of Levene's test for general RCM is greater than 0.05 (0.142>0.05); therefore, this assumption is verified. Now we will analyze the p-value of t-test for equality of means against equal variance assumed, and the p-value of test is 0.158 as shown in Table 16, which is greater than 0.05. As a result, we will accept our null hypothesis and conclude that these two data sets (SLR and questionnaire survey) tend to be very close to each other, and the difference in the both data sets is simply a result of statistical factors.

Similarly, the p-value of Levene's test for RCM challenges in GSD context is greater than 0.05 (0.153>0.05): therefore, this assumption is verified. Now we will analyze the p-value of t-test for equality of means against equal variance assumed. The p-value of t-test is 0.553, as shown in Table 17, which is greater than 0.05. As a result, we will accept our null hypothesis and conclude that these two data sets (SLR and questionnaire survey) tend to be very close to each other, and the difference in the both data sets is simply a result of statistical factors.

TABLE 16. Independent samples t-test of RCM-in-house challenges.

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Challenge	Equal variances assumed	2.37	0.142	-1.48	16	0.158	-10.22	6.902	-24.85	4.41
	Equal variances not assumed			-1.48	10.1	0.169	-10.22	6.902	-25.58	5.13

TABLE 17. Independent samples t-test of RCM-GSD challenges.

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Challenge	Equal variances assumed	3.11	0.153	0.646	4	0.553	15.33	23.73	-50.54	81.21
	Equal variances not assumed			0.646	2.65	0.570	15.33	23.73	-66.10	96.77

TABLE 18. Summary of results.

Research question	Summary of answers
RQ1: What are the challenges of Requirement Change Management process in in-house software development as commonly studied in the literature?	• Impact analysis
RQ3: What are the challenges of Requirement Change Management process in GSD projects, as commonly studied in the literature?	• Communication and coordination • Knowledge management and sharing
RQ7: Are there differences between the challenges identified from the literature and questionnaire survey?	Researches and practitioners agreed that impact analysis and cost/time estimation impact RCM process in in-house and additionally communication and coordination and knowledge management and sharing impact RCM process in GSD domain

V. DISCUSSION

A. CRITICAL RCM CHALLENGES

In this research, we identified a list of challenges that impact RCM in both in-house and GSD approach. To analyze the significance of challenges, we used the following criteria: the challenge is critical if it is cited in literature with frequency of greater than or equal to 50%, and similarly, a challenge considered significant, if it is answered as strongly agree by more than 90% of the survey participants. The similar criteria are followed in existing research [40], [86], [87]. Table 18 summarizes the key findings of this research based on the literature and industry feedback. In RQ1, we identified only impact analysis as a critical challenge that impacts RCM in in-house software development context, however, there are some other challenges such as cost/time estimation and artifacts documents management which have frequency 25% and cannot fulfil the criticality criteria but important for RCM process [12]. With reference to RQ2, no RCM process challenge have a frequency greater than 90%, however, cost/time estimation have 43% positive response from industry practitioners and important in RCM. It is also worth mentioning that, user involvement received second highest positive response from industry practitioners as compared to literature with ranked at the 8th position. In summary, impact analysis, cost/time estimation, artifacts documents management and user involvement are the key challenges that are important and should be managed with high priority in RCM in in-house software development approach.

Furthermore, in RQ3, communication and coordination and knowledge management and sharing are cited in more than 50% of primary studies and important for RCM in GSD context [41]. On the other hand, in questionnaire survey (RQ4), although no challenge satisfies the criticality criteria however, it is wort mentioning that, change control board management received 47% of positive response and ranked 2nd as compared to literature where only two primary studies cited as a key challenge. In summary, all three challenges are important and should be considered while implementing RCM in GSD context. With reference to RQ7, the results indicate that industry practitioners are in aligned with research and reveals that impact analysis, cost/time estimation are key challenges of RCM process in inhouse and additionally communication and coordination and knowledge management and sharing are the challenging factor faced during RCM in GSD domain.

B. SOFTWARE DEVELOPMENT APPROACH-BASED ANALYSIS

In-house software development and GSD are the two most widely followed development approaches in software development industry. It is interesting to note that, the existing

TABLE 19. Comparison of two data sets for RCM-In-house challenges.

Challenge	In-house n=24	%	Rank	GSD n=45		Rank
Impact Analysis	11	46	1	13	29	6
Cost/Time Estimation	7	29	2	23	51	1
Requirements Dependency	7	29	3	11	24	7
Requirement Consistency	7	29	4	16	35	3
System Instability	7	29	5	16	35	4
Change Prioritization	6	25	6	14	31	5
User Involvement	6	25	7	18	40	2
Requirements Traceability	5	21	8	11	24	8
Artifacts Documents Management	4	17	9	9	20	9

literature discusses RCM process only in the context of in-house software development, and there is a lack of research on how these challenges affect RCM process in GSD paradigm. The questionnaire-based survey presented in this research is the first attempt to address the important research gap identified in SLR.

In RQ5, we have tried to understand the characteristics of different RCM challenges when they are implemented in GSD paradigm. In the questionnaire survey, we received data of 24 participants who are using in-house software development paradigm and 45 participants who are using GSD paradigm. Accordingly, we have compared survey data between the two development approaches. The chi-square test results indicate that, there are more differences than similarities between the RCM process development approaches. The influence of impact analysis, requirements dependency, requirements traceability, and system instability remain same regardless of development approach. On the other hand, cost/time estimation, requirement consistency, change prioritization, artifacts documents management and user involvement are more challenging in GSD than in in-house software development. We believe that this is due to the fact that to exchange and to synchronize information is much more difficult in GSD than in in-house software development. For example, cost/time estimation is heavily influenced by different time zones of GSD teams which is not true in in-house software development. Furthermore, requirements consistency, and artifacts documents management require proper coordination and precise understanding of the system requirements and other software artifacts between teams. Therefore, it is more challenging in GSD due to cultural, geographical differences. Similarly, user involvement can be easily managed in in-house software development paradigm where teams are resides at one physical location as compared to GSD paradigm in which teams are working in different time zones.

Moreover, we have ranked RCM challenges based on the feedback received from both type of participants as shown Table 19. It is worth mentioning that, most challenges are relatively equally important and have same rank in both development approaches, except impact analysis which ranked the 1st in in-house software development while ranked the 6th in GSD. Similarly, user involvement ranked 7th in in-house software development while ranked 1st in GSD paradigm data. In summary, these results present the relative importance of RCM challenges in both development approaches. Impact analysis and cot/time estimation are more important in in-house software development as compared to GSD in which user involvement, cost/time estimation and requirement consistency are more important.

Furthermore, software methodologies such as lean, agile, iterative and waterfall are widely used in software development industry. Lean and agile methodologies are more successful as compared to iterative and waterfall [88], [89]. In this study, we performed comparatives analysis between RCM challenges and most widely used software development approaches namely in-house software development and GSD. However, there is a need to explore how RCM will be implemented, when these development methodologies (lean, agile, iterative, and waterfall) will be followed in in-house software development and GSD.

C. GLOBAL PROJECT MANAGEMENT STRUCTURE-BASED ANALYSIS

In RQ5, we have investigated the association of RCM process challenges and development approaches namely in-house software development and GSD paradigm. In RQ6, we have further analyzed RCM challenges based on different project structures in GSD. The management structure of GSD projects can be either centralized or distributed with local coordinators based on the project size, complexity and other factors [13]. It is important to note that, the existing literature discuss RCM for general project management approach followed in GSD projects and there is a lack on how these challenging factors are impacted by different project management structures. The industry practitioners feedback collected through our survey assists us to address this research gap identified in SLR.

In RQ6, the survey results indicate that there are more similarities than difference between RCM challenges in different project management structures. Most challenges have same impact on RCM process regardless of management structure except user involvement and change control board management, which are more challenging in centralized project management structure as compared to distributed structure. We believe that this is due to the difference between centralized and distributed project management structure followed in GSD projects. For example, in a distributed project structure, user will communicate with one person on each site, who works as the site coordinator and is responsible to communicate and collaborate with the project manager. On the other hand, in a centralized project structure, all the team members working on different sites report directly to the project manager who is solely responsible for all tasks. Therefore, all users need to communicate directly with project manager, who may reside at a different development site with

a different time zone. Furthermore, communication and coordination are equally important in both project management structures followed in GSD projects.

D. IMPLICATIONS OF THIS RESEARCH

The objective of this study is to identify the similarities and differences of RCM challenges between in-house software development and GSD approach. Furthermore, we analyze RCM challenges in the context of project management structures followed in GSD. Based on these two-comparative analyses, the recommendation for researchers and practitioners are as follows:

1) In in-house software development, impact analysis, cost/time estimation, and artifacts documents management are more important, and project managers should give more attentions on those challenges in RCM.
2) Impact analysis is one of the key challenges reported by the primary studies. Impact analysis helps to understand consequences of proposed changes. Incorrect understanding of proposed change consequences may increase project cost, postpone the delivery, and ultimately cause project failure. We believe that, the use of formal and semi-formal languages such as description logic and behavior trees will assist researchers and practitioners to better understand the requirements change and develop tools/techniques for impact analysis.
3) Cost/time estimation is another key challenge that impacts RCM process. A number of techniques like function point analysis, line of code etc. are used to calculate project time and cost at the start of the project. However, industry practitioners' data reveals that existing techniques such as function point analysis, line of code for software size and cost/time estimation are not suitable in RCM process. Accordingly. there is a need to understand and develop techniques that can be used to estimate and adjust project cost/time because of proposed requirement changes. Hence, researchers should pay attention to develop customized techniques that can be used in RCM for cost/time adjustment and estimation.
4) Requirement consistency is another key challenge that impacts RCM process. Requirement consistency may emerge due to change in existing requirements or proposing new requirements. Industry survey reveals that there is a need to develop techniques/tools using formal languages for this task. We believe that the use of formal and semi-formal languages such as description logic and behavior trees could be a suitable solution.
5) Most of the studies did not consider system instability as a key challenge for RCM as compared to industry opinions. Hence, we assert that there is an important and urgent need for sufficient research to be conducted to fully understand system instability in RCM.
6) Communication and coordination, knowledge management and sharing, user involvement are the key challenges and must be managed with high priority in RCM in GSD projects.
7) Knowledge management and sharing is one of the key challenges of RCM process in GSD context. The cultural, social and geographical differences make knowledge management more challenging and increase the need of suitable notation to record the requirements. We believe that behavior trees, as a semi-formal design notation, will assist practitioners to convey precise and clear understanding of system requirements among GSD teams.
8) The comparative analysis based on industry data between RCM challenges and software development approaches lays a foundation for future research directions. We strongly suggest that, more attention should be paid by researchers to report RCM process in the context of different software development approaches.
9) The comparative analysis between RCM challenges and project structures lays a foundation for GSD practitioners to make better choice of project structure based on project size and the nature of a project. The GSD practitioners should consider user involvement and change control board while using centralized project structure in GSD projects. This analysis also asserts GSD researchers to report RCM process in the context of these two project management structures in GSD projects.
10) The rank-based analysis lays a foundation for future research and will help researchers to focus and direct their research in RCM domain such as develop techniques for higher ranked challenges both in in-house software development and GSD.

VI. LIMITATIONS

We applied combined SLR and questionnaire approach to identify key challenges that impact RCM process in both inhouse and GSD approaches. One limitation of SLR is incompleteness. The results depend upon the keywords we used for key terms and publication databases (science direct, IEEE explorer, springer link, and ACM) used to find primary studies relevant of our research questions. However, we mitigated this risk of incompleteness in the search terms by using alternative synonyms to build search stings. Furthermore, with increasing of publications related to this topic, we may miss some recent publications at the time of consolidating the results of the SLR. Another possible limitation of SLR is the frequency calculation of identified challenges. We calculated frequency of each challenges based on the grounded theory-based coding scheme, which provides an analytical approach to identify, label and group related challenges into one category. We used inter-rater reliability tests to reduce the impact and this limitation and researchers bias. Nevertheless, we believe that our presented results are comprehensive and cover most of relevant published literature.

With respect to the questionnaire survey, one possible limitation is that some participants may be lack of experience to respond survey questions. In our study, we try to choose participants who had either higher degree in computer science or related fields and experience related to requirement management in industry projects to mitigate this risk. Another potential limitation of questionnaire-based study is the ambiguity in the survey questions. To minimize this limitation, the first author is always available on skype and email during the study to clarify any potential ambiguities. In this paper, we used standard statistical techniques such as chi-square test, t-test of independence to either reject or accept the hypothesis. Furthermore, to mitigate construct validity threat, we used standard scale in survey design, which is largely used in reported research [48].

Another potential limitation of questionnaire-based studies lies in their external validity. This limitation is mainly due to the low participation rate and difficulty in choosing true random sample. We address this limitation by using LinkedIn, mailing list, and industrial contacts, and use snowballing technique to engage more participants in this study [61]; and finally, we managed to receive 69 useable responses. Although, this is a low participation rate, but it will help to understand trends in the data [58]. Therefore, we believe that our results at least will assist industry practitioners to make rational decisions during RCM process.

VII. CONCLUSION & FUTURE WORK

Requirement change management is a key activity in requirements engineering phase and profoundly determines project success or failure. The different aspects of RCM challenges have been explored in existing research, however the need to understand this process from different development approaches such as in-house software development and GSD, has motivated us to conduct this study. We used SLR and questionnaire-based survey approaches to identify the key challenges that impact RCM in both in-house and GSD.

Through both approaches, we have identified 9 challenges for RCM process in-house software development and 3 additional challenges that are specific to RCM process in GSD. Among these challenges, impacts analysis, cost/time estimation, artifacts documents management, requirements traceability are top ranked in RCM process in in-house context, while communication and coordination, knowledge management and sharing are important in GSD projects. Furthermore, chi-square test shows that cost/time estimation, requirement consistency, change prioritization, artifacts documents management, and user involvement are more challenging in GSD as compared to in-house software development. Similarly, user involvement, and change control board management are more challenging in centralized project management structure than distributed project management structure in GSD projects. The t-test of independence used for comparative analysis of both data sets, reveals that research and industry are aligned and share the same opinions regarding challenges that impact RCM.

For future work, we plan to conduct further empirical study to understand the inter-dependencies between the key challenges and their impacts on the success or failure of a project. We also plan to develop techniques to better identify and manage identified challenges such as impact analysis, requirement consistency. As a result, the project managers and development teams can implement RCM in both in-house software development and global software development more efficiently and effectively.

APPENDIX A
LIST OF PRIMARY STUDIES IN SLR

A01: S. Ahn and K. Chong, "Requirements change management on feature-oriented requirements tracing," in International Conference on Computational Science and Its Applications, 2007, pp. 296-307: Springer.

A02: S. Jayatilleke, R. Lai, and K. Reed, "A method of requirements change analysis," Requirements Engineering, vol. 23, no. 4, pp. 493-508, 2018.

A03: A. AlSanad and A. Chikh, "The Impact of Software Requirement Change-A Review," in New Contributions in Information Systems and Technologies: Springer, 2015, pp. 803-812.

A04: M. Kausar and A. Al-Yasiri, "Using Distributed Agile Patterns for Supporting the Requirements Engineering Process," in Requirements Engineering for Service and Cloud Computing: Springer, 2017, pp. 291-316.

A05: I. Navarro, N. Leveson, and K. Lunqvist, "Semantic decoupling: reducing the impact of requirement changes," Requirements engineering, vol. 15, no. 4, pp. 419-437, 2010.

A06: L. Lin and J. H. Poore, "Pushing requirements changes through to changes in specifications," Frontiers of Computer Science in China, vol. 2, no. 4, pp. 331-343, 2008.

A07: J. J. Lin and Y.-S. Lin, "Research and Development of a CMMI-Compliant Requirement Management System for Software Engineering," in International Conference on Computer Supported Cooperative Work in Design, 2007, pp. 76-86: Springer.

A08: K.-D. Mu, W. Liu, Z. Jin, J. Hong, and D. Bell, "Managing software requirements changes based on negotiation-style revision," Journal of Computer Science and Technology, vol. 26, no. 5, p. 890, 2011.

A09: M. A. Chauhan and C. W. Probst, "Architecturally Significant Requirements Identification, Classification and Change Management for Multi-tenant Cloud-Based Systems," in Requirements Engineering for Service and Cloud Computing: Springer, 2017, pp. 181-205.

A10: D. Damian, J. Chisan, L. Vaidyanathasamy, and Y. Pal, "Requirements engineering and downstream software development: Findings from a case study," Empirical Software Engineering, vol. 10, no. 3, pp. 255-283, 2005.

A11: J. Lockerbie, N. A. M. Maiden, J. Engmann, D. Randall, S. Jones, and D. Bush, "Exploring the impact of software requirements on system-wide goals: a method using satisfaction arguments and i* goal modelling," Requirements Engineering, vol. 17, no. 3, pp. 227-254, 2012.

A12: M. Heindl and S. Biffl, "Modeling of requirements tracing," in Balancing Agility and Formalism in Software Engineering: Springer, 2008, pp. 267-278.

A13: T. O. de Jesus and M. S. Soares, "An Event-Based Technique to Trace Requirements Modeled with SysML," in International Conference on Computational Science and Its Applications, 2017, pp. 145-159: Springer.

A14: B. B. Chua, D. V. Bernardo, and J. Verner, "Criteria for estimating effort for requirements changes," in European Conference on Software Process Improvement, 2008, pp. 36-46: Springer.

A15: T. Gorschek and M. Svahnberg, "Requirements experience in practice: Studies of six companies," in Engineering and Managing Software Requirements: Springer, 2005, pp. 405-426.

A16: C. Ting, "The Control and Measure of Requirements Stability in Software Project," in International Conference on Information and Management Engineering, 2011, pp. 387-394: Springer.

A17: J. Shah and N. Kama, "Issues of Using Function Point Analysis Method for Requirement Changes During Software Development Phase," in Asia Pacific Requirements Engeneering Conference, 2017, pp. 156-163: Springer.

A18: N. Ali and R. Lai, "A method of requirements change management for global software development," Information and Software Technology, vol. 70, pp. 49-67, 2016.

A19: A. A. Khan, S. Basri, and P. Dominc, "A proposed framework for communication risks during RCM in GSD," Procedia-Social and Behavioral Sciences, vol. 129, pp. 496-503, 2014.

A20: S. Jayatilleke and R. Lai, "A systematic review of requirements change management," Information and Software Technology, vol. 93, pp. 163-185, 2018.

A21: D. Lloyd, R. Moawad, and M. Kadry, "A supporting tool for requirements change management in distributed agile development," Future Computing and Informatics Journal, vol. 2, no. 1, pp. 1-9, 2017.

A22: R. A. Aziz and B. Wong, "The interplay between requirements relationships knowledge and requirements change towards software project success: an assessment using partial least square (PLS)," Procedia Computer Science, vol. 46, pp. 732-741, 2015.

A23: A. Goknil, I. Kurtev, K. van den Berg, and W. Spijkerman, "Change impact analysis for requirements: A metamodeling approach," Information and Software Technology, vol. 56, no. 8, pp. 950-972, 2014.

A24: C.-Y. Chen and P.-C. Chen, "A holistic approach to managing software change impact," Journal of Systems and Software, vol. 82, no. 12, pp. 2051-2067, 2009.

A25: R. Sugden and M. Strens, "Strategies, tactics and methods for handling change," in ecbs, 1996, p. 457: IEEE.

A26: A. A. Khan, J. Keung, S. Hussain, and K. E. Bennin, "Effects of geographical, socio-cultural and temporal distances on communication in global software development during requirements change management a pilot study," in Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on, 2015, pp. 159-168: IEEE.

A27: N. Assawamekin, "An ontology-based approach for multiperspective requirements traceability between analysis models," in Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on, 2010, pp. 673-678: IEEE.

A28: L. Lin, S. J. Prowell, and J. H. Poore, "The impact of requirements changes on specifications and state machines," Software: Practice and Experience, vol. 39, no. 6, pp. 573-610, 2009.

A29: M. Shafiq et al., "Effect of project management in requirements engineering and requirements change management processes for global software development," IEEE Access, 2018.

A30: A. Teka, N. Condori-Fernandez, I. Kurtev, D. Quartel, and W. Engelsman, "Change impact analysis of indirect goal relations: Comparison of NFR and TROPOS approaches based on industrial case study," in 2012 IEEE Model-Driven Requirements Engineering Workshop (MoDRE 2012), 2012, pp. 58-67: IEEE.

A31: M. Weber and J. Weisbrod, "Requirements engineering in automotive development-experiences and challenges," in Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, 2002, pp. 331-340: IEEE. A32: W. Hussain, D. Zowghi, T. Clear, S. MacDonell, and K. Blincoe, "Managing Requirements Change the Informal Way: When Saying 'No' is Not an Option," in Requirements Engineering Conference (RE), 2016 IEEE 24th International, 2016, pp. 126-135: IEEE.

A33: F. Mokammel, E. Coatanéa, M. Bakhouya, F. Christophe, and S. Nonsiri, "Impact analysis of graph-based requirements models using PageRank algorithm," in Systems Conference (SysCon), 2013 IEEE International, 2013, pp. 731-736: IEEE.

A34: S. Imtiaz, N. Ikram, and S. Imtiaz, "Impact analysis from multiple perspecties: Evaluation of traceability techniques," in Software Engineering Advances, 2008. ICSEA'08. The Third International Conference on, 2008, pp. 457-464: IEEE.

A35: I. Keshta, M. Niazi, and M. Alshayeb, "Towards Implementation of Requirements Management Specific Practices (SP1. 3 and SP1. 4) for Saudi Arabian Small and Medium Sized Software Development Organizations," IEEE Access, vol. 5, pp. 24162-24183, 2017.

A36: A. A. Khan, S. Basri, and P. Dominic, "Communication risks in GSD during RCM: Results from SLR," in Computer and Information Sciences (ICCOINS), 2014 International Conference on, 2014, pp. 1-6: IEEE.

A37: A. Khatoon, Y. H. Motla, M. Azeem, H. Naz, and S. Nazir, "Requirement change management for global

software development using ontology," in Emerging Technologies (ICET), 2013 IEEE 9th International Conference on, 2013, pp. 1-6: IEEE.

A38: Y. Hafeez et al., "A requirement change management framework for distributed software environment," in Computing and Convergence Technology (ICCCT), 2012 7th International Conference on, 2012, pp. 944-948: IEEE.

A39: Y. Jin, J. Zhang, P. Ma, W. Hao, S. Luo, and Z. Li, "Applying pagerank algorithm in requirement concern impact analysis," in Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, 2009, vol. 1, pp. 361-366: IEEE.

A40: W. Lam, V. Shankararaman, S. Jones, J. Hewitt, and C. Britton, "Change analysis and management: a process model and its application within a commercial setting," in Application-Specific Software Engineering Technology, 1998. ASSET-98. Proceedings. 1998 IEEE Workshop on, 1998, pp. 34-39: IEEE.

A41: V. Sinha, B. Sengupta, and S. Chandra, "Enabling collaboration in distributed requirements management," IEEE software, vol. 23, no. 5, pp. 52-61, 2006.

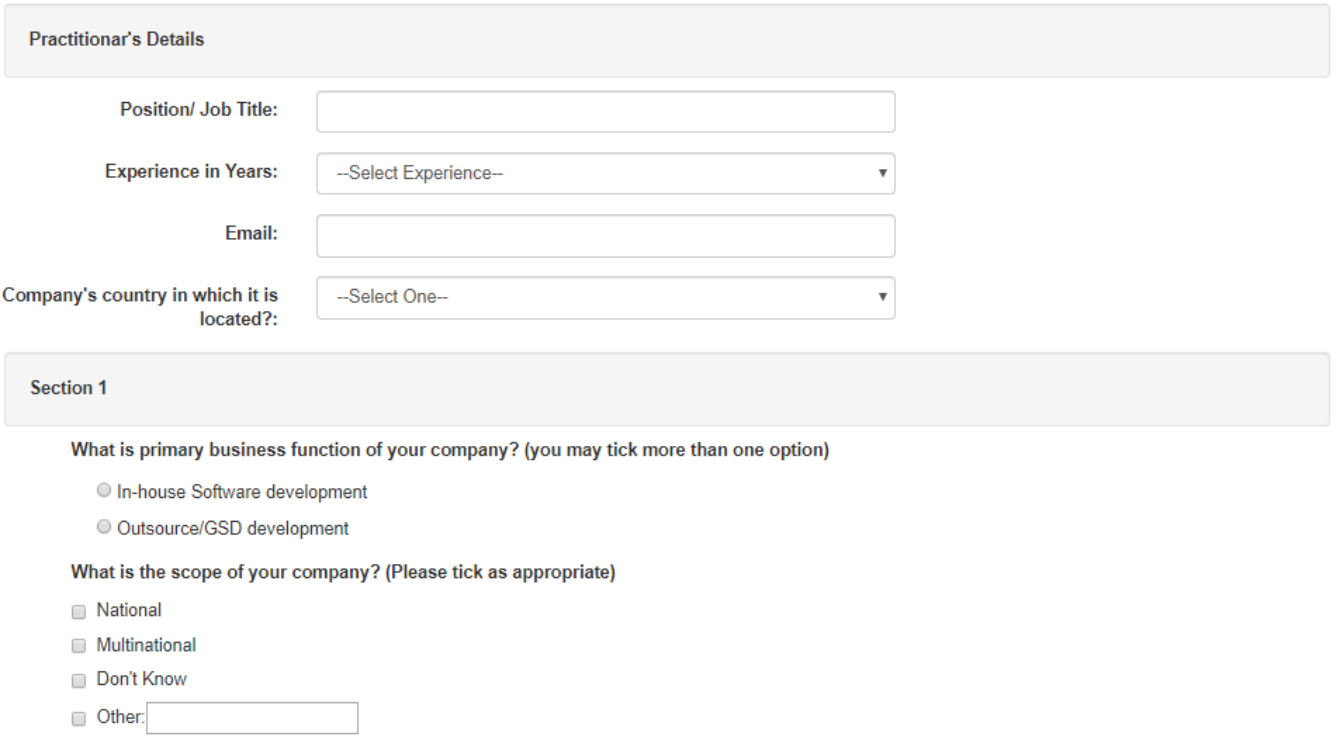
A42: R. Lai and N. Ali, "A requirements management method for global software development," AIS: Advances in Information Sciences, vol. 1, no. 1, pp. 38-58, 2013.

A43: H. Ahmed, A. Hussain, and F. Baharom, "Current challenges of requirement change management," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 8, no. 10, pp. 173-176, 2016.

APPENDIX B
REQUIREMENT CHANGE MANAGEMENT CHALLENGES CATEGORIES IDENTIFIED VIA SLR

List of challenges for RCM process in in-house software development	
Final list of RCM Challenges	RCM Challenges – sub categories
Impact analysis	• Impact analysis • Change consequences
Cost/Time estimation	• Cost estimation • Time estimation • Effort estimation • Change cost
Requirements traceability	• Requirements traceability
Artifacts documents management	• Artifacts documents management • Artifacts documents updation • SDLC products management • Documents consistency management
Requirement dependency	• Requirements dependency • Requirements inter-dependency
Requirements consistency	• Requirements consistency • Change conflicts with existing requirements
Change prioritization	• Change prioritization
User involvement	• User involvement
System instability	• System instability
List of RCM challenges for GSD projects	
Communication and coordination	• Communication and coordination • Coordination control
Knowledge management and sharing	• Knowledge management • Knowledge sharing • Use of similar terminology
Change control board management	• Change control board management

APPENDIX C
QUESTIONNAIRE SURVEY



Practitionar's Details

Position/ Job Title:

Experience in Years: --Select Experience--

Email:

Company's country in which it is located?: --Select One--

Section 1

What is primary business function of your company? (you may tick more than one option)

- In-house Software development
- Outsource/GSD development

What is the scope of your company? (Please tick as appropriate)

- National
- Multinational
- Don't Know
- Other:

What type of Project Management Model typically used in your organization for GSD projects?

- ☐ Centralized Project Management-- All or most of the team members report directly to project manager, who may work at other geographical site and responsible for the planning and execution of projects.
- ☐ Distributed Project Management with Local Coordinators-- All or most of the team members report to local coordinators, who are responsible for the planning and execution of sub-projects or work packages and report to project manager.

Approximately how many staff are employed by your company? (Please tick as appropriate)

- ○ Less than 25
- ○ 26-199
- ○ Greater than 200
- ○ Not Sure

Approximately how many staff are employed directly in the production/maintenance of software? (Please tick as appropriate)

- ○ Less than 25
- ○ 26-199
- ○ Greater than 200
- ○ Not sure

Approximately how many different geographical sites are used by your company?

- ○ 1-5
- ○ 6-100
- ○ Greater than 10
- ○ Not sure

What type of systems are your company concerned with? (You may tick more than one)

- ☐ Safety Critical
- ☐ Business Systems
- ☐ Telecommunications
- ☐ Real Time Systems
- ☐ Data Processing
- ☐ System Software
- ☐ Windows-based
- ☐ Embedded Systems
- ☐ Android Applications
- ☐ IOS Applications
- ☐ Other:

Section 2

2.1. Evaluation of the challenges of Software Requirement Change Management Processes
For each challenge, please select the appropriate box based on your experience in software projects.

Challenges of software requirement Change Management Process	Strongly Agree	Agree	Disagree	Strongly Disagree
Impact Analysis	○	○	○	○
Cost/Time Estimation	○	○	○	○
Requirements Traceability	○	○	○	○
Artifacts Documents Managements	○	○	○	○
Requirements Dependency	○	○	○	○
Requirments Consistency	○	○	○	○
Change Prioritisation	○	○	○	○
User Involvement	○	○	○	○
System Instability**	○	○	○	○

**System Instability-- means that when the requested change is in process, how the functionality impacted by requested change will be handeled?

2.3. Please list the challenges that you think are important for requirement change management in addition to the above challenges identified from literature.

Section 3 Global Software Development

3.1. Evaluation of the challenges of Software Requirement Change Management Processes in GSD projects.
For each challenge, please select the appropriate box based on your experience in GSD projects.

Challenges faced to manage requirement change in GSD Projects

	Strongly Agree	Agree	Disagree	Strongly Disagree
Communication & Coordination	○	○	○	○
Knowledge Management & Sharing	○	○	○	○
Change Control Board Management	○	○	○	○

3.3. Please list the challenges that you think are important for requirement change management in addition to the above challenges identified from literature in GSD projects.

APPENDIX D
PARTICIPANTS DEMOGRAPHIC DETAILS

Job Title	Experience	Company size	Number of geographic sites	Types of Systems
Project Manager	9-11 Years	Less than 25	1-5	Safety Critical, Real Time systems
Development Manager	7-8 Years	Less than 25	6-10	Business Systems, Android Applications
Software Engineer	3-5 Years	Greater than 200	1-5	Safety Critical, Business Systems, Data processing
Requirements Engineer	3-5 Years	Greater than 200	1-5	Safety Critical, Real Time systems, Data processing
System Manager	7-8 Years	Greater than 200	1-5	System Software, Android Applications
Software Engineer	3-5 Years	Greater than 200	6-10	Business Systems, Windows based, IOS Applications
Software Engineer	0-2 Years	26-199	1-5	Business Systems
Business Intelligence Engineer	3-5 Years	Less than 25	Greater than 10	Business Systems, Real Time systems, Data processing
Sr. Software Engineer	3-5 Years	26-199	6-10	Real Time systems, Android Applications, IOS Applications
BI Developer	3-5 Years	26-199	6-10	Business Systems, Data processing
Team Lead	7-8 Years	Greater than 200	6-10	Business Systems, Telecommunications, Real Time systems
Sr. PeopleSoft Consultant	7-8 Years	26-199	6-10	Business Systems, System Software, IOS Applications
Business Intelligence Solution Developer	3-5 Years	Greater than 200	6-10	Business Systems, Data processing
Sr. Developer	9-11 Years	Greater than 200	6-10	Business Systems, Data processing
Project Manager	9-11 Years	Less than 25	1-5	Safety Critical, Data processing, Android Applications, IOS Applications
Senior Software Engineer	7-8 Years	Greater than 200	6-10	Business Systems, Telecommunications, Data processing
Sr. Software Engineer	3-5 Years	26-199	1-5	Business Systems
Development Lead	7-8 Years	Less than 25	1-5	Safety Critical
Software Engineer	3-5 Years	26-199	6-10	Business Systems
Software Engineer	3-5 Years	Less than 25	1-5	Business Systems, Data processing, System Software
Software Quality Analyst	3-5 Years	Greater than 200	6-10	Business Systems, Real Time systems, Data processing
Sr. Requirements Engineer	3-5 Years	Greater than 200	Greater than 10	Safety Critical, Windows based, IOS Applications
Team Lead	7-8 Years	26-199	1-5	Business Systems
Software Engineer	3-5 Years	26-199	6-10	Real Time systems, Data processing, System Software

Software Engineer	0-2 Years	Less than 25	6-10	Business Systems, Telecommunications, Real Time systems
Sr. Software Engineer	3-5 Years	Greater than 200	Greater than 10	Safety Critical, Business Systems, Embedded Systems
Software Engineer	3-5 Years	Greater than 200	1-5	Safety Critical, Business Systems, System Software, Embedded Systems
Software Engineer	7-8 Years	Greater than 200	6-10	Business Systems, Data processing
Sr. Software Engineer	9-11 Years	Greater than 200	6-10	Real Time systems, Data processing, System Software
Team Lead	7-8 Years	Greater than 200	6-10	Business Systems, Real Time systems
Sr. Software Engineer	7-8 Years	Greater than 200	6-10	Safety Critical, Telecommunications, System Software, Embedded Systems
Software Design Engineer	3-5 Years	Greater than 200	6-10	Business Systems, Data processing, Embedded Systems
Development Lead	7-8 Years	26-199	Greater than 10	Business Systems, Real Time systems, Android Application
Project Manager	9-11 Years	26-199	6-10	Business Systems, Android Applications, IOS Applications
Team Lead	7-8 Years	26-199	6-10	Business Systems, Real Time systems
Sr. Software Engineer	7-8 Years	Less than 25		Business Systems, Data processing, System Software
Requirement Engineer	3-5 Years	Less than 25	1-5	Business Systems, Data processing, System Software
Sr. Software Engineer	3-5 Years	Less than 25	1-5	Business Systems
Sr. ISO Developer	3-5 Years	26-199	Greater than 10	Business Systems, Data processing, Android Applications, IOS Applications
Development Lead	3-5 Years	Less than 25	6-10	Business Systems, Telecommunications, Real Time systems
Software Engineer	0-2 Years	Less than 25	1-5	Business Systems, Telecommunications, Real Time systems, System Software
Project Manager	9-11 Years	Greater than 200	1-5	Business Systems, Telecommunications
Sr. PHP Developer	3-5 Years	Less than 25	Greater than 10	Business Systems, Android Applications, IOS Applications
Team Lead	7-8 Years	Greater than 200	1-5	Business Systems, Real Time systems, Data processing, Android Applications
Server Engineer	0-2 Years	26-199	1-5	Games
Sr. Software Engineer	3-5 Years	Greater than 200	No Site information in in-house	Windows based
Team Lead	7-8 Years	Greater than 200		Windows based
Sr. Requirements Engineer	7-8 Years	Greater than 200		Business Systems, Real Time systems, Android Applications
Software Engineer	3-5 Years	Greater than 200		Safety Critical, Business Systems, Real Time systems
Software Engineer	3-5 Years	26-199		Security
Project Manager	9-11 Years	26-199		System Software
Software Engineer	3-5 Years	Less than 25		IOS Applications
Development Lead	7-8 Years	26-199		Business Systems, Data processing, IOS Applications
Principal Software Engineer	9-11 Years	Greater than 200		Safety Critical, Business Systems, Real Time systems
Sr. Software Engineer	7-8 Years	Greater than 200		Business Systems, Data processing, Android Applications
Team Lead	7-8 Years	Less than 25		Business Systems, Android Applications
Software Engineer	3-5 Years	Less than 25		Safety Critical, Business Systems, Real Time systems, Data processing
Team Lead	7-8 Years	Less than 25		System Software, Windows based, Android Applications, IOS Applications
Sr. Software Engineer	7-8 Years	Less than 25		Business Systems, System Software, Android Applications, IOS Applications
Software Engineer	0-2 Years	Less than 25		Block chain
Software Engineer	3-5 Years	26-199		Business Systems, Real Time systems, Data processing, System Software
Principal Software Engineer	7-8 Years	26-199		Business Systems, Real Time systems, Data processing, System Software
Project Manager	9-11 Years	Greater than 200		Data processing, Windows based, Android Applications
Software Engineer	3-5 Years	Greater than 200		Safety Critical, Business Systems, Real Time systems, System Software
Sr. Software Engineer	3-5 Years	Greater than 200		Safety Critical, Business Systems, Real Time systems
Development Lead	7-8 Years	Greater than 200		Business Systems, Real Time systems, Data processing, Android Application
Software Engineer	3-5 Years	Greater than 200		Business Systems, Real Time systems, System Software, Android Applications
Sr. Software engineer	3-5 Years	26-199		Branch-less banking
Team Lead	7-8 Years	Greater than 200		Safety Critical, Business Systems, Real Time systems, Data processing

APPENDIX E
SLR PRIMARY STUDIES QUALITY ASSESSMENT RESULTS

ID	Paper ID	Q1	Q2	Q3	Q4	Q5	Total Score	Quality (%)
1	A01	1	0.5	0.5	1	0.5	3.5	70
2	A02	1	1	1	1	0.5	4.5	90
3	A03	1	0.5	1	0.5	0	3	60
4	A04	1	1	0.5	1	0.5	4	80
5	A05	1	0.5	1	0.5	0.5	3.5	70
6	A06	1	0.5	1	0.5	0.5	3.5	70
7	A07	1	0.5	0.5	1	0	3	60
8	A08	1	0.5	1	1	1	4.5	90
9	A09	1	0.5	0.5	1	0	3	60
10	A10	1	0.5	1	1	0.5	4	80
11	A11	0.5	1	0.5	1	0.5	3.5	70
12	A12	0.5	0.5	1	1	0.5	3.5	70
13	A13	1	0.5	1	1	0.5	4	80
14	A14	1	0.5	1	1	0.5	4	80
15	A15	0.5	0.5	1	1	0	3	60
16	A16	1	1	0.5	1.0`	0	3.5	70
17	A17	1	0.5	1	1	0.5	4	80
18	A18	1	1	1	1	0.5	4.5	90
19	A19	1	0.5	1	1	0.5	4	80
20	A20	1	1	1	1	1	5	100
21	A21	0.5	1	0.5	1	0	3	60
22	A22	1	1	1	0.5	0.5	4	80
23	A23	1	0.5	1	0.5	0.5	3.5	70
24	A24	1	0.5	1	1	0	3.5	70
25	A25	1	0.5	1	0.5	0.5	3.5	70
26	A26	1	1	1	1	0.5	4.5	90
27	A27	0.5	0.5	1	1	0.5	3.5	70
28	A28	1	1	1	1	0.5	4.5	90
29	A29	1	1	0.5	1	0.5	4.5	90
30	A30	0.5	0.5	1	1	0	3	60
31	A31	1	0.5	1	1	0.5	4	80
32	A32	1	1	1	1	0.5	4.5	90
33	A33	1	0.5	1	1	0.5	4	80
34	A34	1	0.5	1	1	0.5	4	80
35	A35	1	1	0.5	1	1	4.5	90
36	A36	1	1	1	1	0.5	4.5	90
37	A37	1	0.5	0.5	1	0.5	3.5	70
38	A38	0.5	0.5	1	1	0	3	60
39	A39	0.5	1	0.5	1	0	3	60
40	A40	1	0.5	0.5	1	0	3	60
41	A41	1	0.5	1	1	0.5	4	80
42	A42	1	1	0.5	1	0.5	4	80
43	A43	1	0.5	0.5	1	0	3.5	70
Average		0.91	0.69	0.84	0.93	0.39	3.77	

REFERENCES

[1] B. Berenbach, D. Paulish, J. Kazmeier, and A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*. New York, NY, USA: McGraw-Hill, 2009.

[2] K. Wiegers and J. Beatty, *Software Requirements*. London, U.K.: Pearson, 2013.

[3] N. Nurmuliani, D. Zowghi, and S. Powell, "Analysis of requirements volatility during software development life cycle," in *Proc. Austral. Softw. Eng. Conf.*, 2004, pp. 28–37.

[4] T. Hall, S. Beecham, and A. Rainer, "Requirements problems in twelve software companies: An empirical analysis," *IEE Proc.–Softw.*, vol. 149, no. 5, pp. 153–160, Oct. 2002.

[5] *Standish-Group International Chaos—The State of the Software Industry*, Standish Group Int., West Yarmouth, MA, USA, 2017.

[6] K. Mohan, P. Xu, L. Cao, and B. Ramesh, "Improving change management in software development: Integrating traceability and software configuration management," *Decis. Support Syst.*, vol. 45, no. 4, pp. 922–936, 2008.

[7] W. Hussain, D. Zowghi, T. Clear, S. MacDonell, and K. Blincoe, "Managing requirements change the informal way: When saying 'no' is not an option," in *Proc. IEEE 24th Int. Requirements Eng. Conf. (RE)*, Sep. 2016, pp. 126–135.

[8] N. Ali and R. Lai, "A method of requirements change management for global software development," *Inf. Softw. Technol.*, vol. 70, pp. 49–67, Feb. 2016.

[9] M. Niazi, C. Hickman, R. Ahmad, and M. A. Babar, "A model for requirements change management: Implementation of cmmi level 2 specific practice," in *Proc. Int. Conf. Product Focused Softw. Process Improvement*. Berlin, Germany: Springer, 2008, pp. 143–157.

[10] M. Bano, S. Imtiaz, N. Ikram, M. Niazi, and M. Usman, "Causes of requirement change—A systematic literature review," in *Proc. 16th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2012, pp. 22–31.

[11] M. A. Akbar, J. Sang, Nasrullah, A. A. Khan, S. Mahmood, S. F. Qadri, H. Hu, and H. Xiang, "Success factors influencing requirements change management process in global software development," *J. Comput. Lang.*, vol. 51, pp. 112–130, Apr. 2019.

[12] S. Jayatilleke and R. Lai, "A systematic review of requirements change management," *Inf. Softw. Technol.*, vol. 93, pp. 163–185, Jan. 2018.

[13] J. Binder, *Global Project Management: Communication, Collaboration and Management Across Borders*. Evanston, IL, USA: Routledge, 2016.

[14] T. Kern and L. Willcocks, "Exploring information technology outsourcing relationships: Theory and practice," *J. Strategic Inf. Syst.*, vol. 9, no. 4, pp. 321–350, 2000.

[15] E. Carmel, J. A. Espinosa, and Y. Dubinsky, "'Follow the sun' workflow in global software development," *J. Manage. Inf. Syst.*, vol. 27, no. 1, pp. 17–38, 2010.

[16] S. Ambler, "It project success rates survey results," Ambysoft, Toronto, ON, Canada, Tech. Rep., 2018.

[17] D. Bradstreet, "Dun & bradstreet's barometer of global outsourcing," *Dun Bradstreet*, vol. 20, Sep. 2000. [Online]. Available: http://findarticles.com/p/articles/mi_m0EIN/is_2000_Feb_24/ai_59591405

[18] L. McLaughlin, "An eye on India: Outsourcing debate continues," *IEEE Softw.*, vol. 20, no. 3, pp. 114–117, May 2003.

[19] S. Deshpande, S. Beecham, and I. Richardson, "Using the PMBOK guide to frame GSD coordination strategies," in *Proc. IEEE 8th Int. Conf. Global Softw. Eng.*, Aug. 2013, pp. 188–196.

[20] M. Niazi, S. Mahmood, M. Alshayeb, M. R. Riaz, K. Faisal, N. Cerpa, S. U. Khan, and I. Richardson, "Challenges of project management in global software development: A client-vendor analysis," *Inf. Softw. Technol.*, vol. 80, pp. 1–19, Dec. 2016.

[21] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 16–20, Mar. 2001.

[22] S. Islam, M. M. A. Joarder, and S. H. Houmb, "Goal and risk factors in offshore outsourced software development from vendor's viewpoint," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng.*, Jul. 2009, pp. 347–352.

[23] M. W. Bhatti, F. Hayat, N. Ehsan, A. Ishaque, S. Ahmed, and E. Mirza, "A methodology to manage the changing requirements of a software project," in *Proc. Int. Conf. Comput. Inf. Syst. Ind. Manage. Appl. (CISIM)*, 2010, pp. 319–322.

[24] S. Imtiaz, N. Ikram, and S. Imtiaz, "A process model for managing requirement change," in *Proc. 4th IASTED Int. Conf. Adv. Comput. Sci. Technol.*, Langkawi, Malaysia, 2008, pp. 121–128.

[25] J. Tomyim and A. Pohthong, "Requirements change management based on object-oriented software engineering with unified modeling language," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Aug. 2016, pp. 7–10.

[26] A. A. Khan, S. Basri, and P. D. D. Dominic, "A process model for requirements change management in collocated software development," in *Proc. IEEE Symp. E-Learn., E-Manage. E-Services*, Oct. 2012, pp. 1–6.

[27] Z. Ahmad, M. Hussain, A. Rehman, U. Qamar, and M. Afzal, "Impact minimization of requirements change in software project through requirements classification," in *Proc. 9th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2015, p. 15.

[28] R. Lai and N. Ali, "A requirements management method for global software development," *Adv. Inf. Sci.*, vol. 1, no. 1, pp. 38–58, 2013.

[29] R. Prikladnicki, J. Audy, and R. Evaristo, "Requirements management in global software development: Preliminary findings from a case study in a SW-CMM context," in *Proc. ICSE Int. Workshop Global Softw. Develop. (GSD)*, 2003, pp. 53–58.

[30] A. Mateen and H. Amir, "Enhancement in the effectiveness of requirement change management model for global software development," 2016, *arXiv:1605.00770*. [Online]. Available: https://arxiv.org/abs/1605.00770

[31] D. Moody, "The 'physics' of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Trans. Softw. Eng.*, vol. 35, no. 6, pp. 756–779, Nov. 2009.

[32] F. Saleh and M. El-Attar, "A scientific evaluation of the misuse case diagrams visual syntax," *Inf. Softw. Technol.*, vol. 66, pp. 73–96, Oct. 2015.
[33] S. Anwer and M. El-Attar, "An evaluation of the statechart diagrams visual syntax," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, 2014, pp. 1–4.
[34] Y. Hafeez, M. Riaz, S. Asghar, H. Naz, S. M. M. Gilani, A. Batool, M. Ahmed, and M. S. Hassan, "A requirement change management framework for distributed software environment," in *Proc. 7th Int. Conf. Comput. Converg. Technol. (ICCCT)*, 2012, pp. 944–948.
[35] D. Lloyd, R. Moawad, and M. Kadry, "A supporting tool for requirements change management in distributed agile development," *Future Comput. Informat. J.*, vol. 2, no. 1, pp. 1–9, 2017.
[36] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, vol. 4. Boca Raton, FL, USA: CRC Press, 2015.
[37] B. A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proc. 26th Int. Conf. Softw. Eng.*, 2004, pp. 273–281.
[38] J. Kroll, S. I. Hashmi, I. Richardson, and J. L. N. Audy, "A systematic literature review of best practices and challenges in follow-the-sun software development," in *Proc. IEEE 8th Int. Conf. Global Softw. Eng. Workshops*, Aug. 2013, pp. 18–23.
[39] G. K. Hanssen, D. Šmite, and N. B. Moe, "Signs of agile trends in global software engineering research: A tertiary study," in *Proc. IEEE 6th Int. Conf. Global Softw. Eng. Workshop*, Aug. 2011, pp. 17–23.
[40] S. Mahmood, S. Anwer, M. Niazi, M. Alshayeb, and I. Richardson, "Key factors that influence task allocation in global software development," *Inf. Softw. Technol.*, vol. 91, pp. 102–122, Nov. 2017.
[41] A. A. Khan, S. Basri, and P. Dominic, "Communication risks in GSD during RCM: Results from SLR," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCOINS)*, 2014, pp. 1–6.
[42] A. A. Khan, J. Keung, S. Hussain, and K. E. Bennin, "Effects of geographical, socio-cultural and temporal distances on communication in global software development during requirements change management a pilot study," in *Proc. Int. Conf. Eval. Novel Approaches Softw. Eng. (ENASE)*, 2015, pp. 159–168.
[43] M. Shafiq, Q. Zhang, M. A. Akbar, A. A. Khan, S. Hussain, F.-E. Amin, A. Khan, and A. A. Soofi, "Effect of project management in requirements engineering and requirements change management processes for global software development," *IEEE Access*, vol. 6, pp. 25747–25763, 2018.
[44] M. A. Akbar, M. Shafiq, T. Kamal, and M. Hamza, "Towards the successful requirements change management in the domain of offshore software development outsourcing: Preliminary results," *Int. J. Comput. Digit. Syst.*, vol. 8, no.3, pp. 205–215, 2019.
[45] M. A. Akbar, Nasrullah, M. Shameem, J. Ahmad, A. Maqbool, and K. Abbas, "Investigation of project administration related challenging factors of requirements change management in global software development: A systematic literature review," in *Proc. Int. Conf. Comput., Electron. Electr. Eng. (ICE Cube)*, 2018, pp. 1–7.
[46] H. Ahmed, A. Hussain, and F. Baharom, "Current challenges of requirement change management," *J. Telecommun., Electron. Comput. Eng.*, vol. 8, no. 10, pp. 173–176, 2016.
[47] B. A. Kitchenham, M. Brereton, M. Turner, M. Z. Niazi, S. Linkman, and R. Pretorius, "Refining the systematic literature review process—Two participant-observer case studies," *Empirical Softw. Eng.*, vol. 15, no. 6, pp. 618–653, 2010.
[48] M. Niazi, "Do systematic literature reviews outperform informal literature reviews in the software engineering domain? an initial case study," *Arabian J. Sci. Eng.*, vol. 40, no. 3, pp. 845–855, 2015.
[49] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007.
[50] A. Von Eye and E. Y. Mun, *Analyzing Rater Agreement: Manifest Variable Methods*. New York, NY, USA: Psychology Press, 2014.
[51] S. Mahdavi-Hezavehi, M. Galster, and P. Avgeriou, "Variability in quality attributes of service-based software systems: A systematic literature review," *Inf. Softw. Technol.*, vol. 55, no. 2, pp. 320–343, 2013.
[52] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, nos. 9–10, pp. 833–859, 2008.
[53] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.
[54] D. Dermeval, J. Vilela, I. B. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva, "Applications of ontologies in requirements engineering: A systematic review of the literature," *Requirements Eng.*, vol. 21, no. 4, pp. 405–437, 2016.
[55] W. Ding, P. Liang, A. Tang, and H. Van Vliet, "Knowledge-based approaches in software documentation: A systematic literature review," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 545–567, 2014.
[56] D. Budgen, M. Turner, P. Brereton, and B. A. Kitchenham, "Using mapping studies in software engineering," in *Proc. PPIG*, vol. 8, 2008, pp. 195–204.
[57] J. M. Corbin and A. Strauss, "Grounded theory research: Procedures, canons, and evaluative criteria," *Qualitative Sociol.*, vol. 13, no. 1, pp. 3–21, 1990.
[58] T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying software engineers: Data collection techniques for software field studies," *Empirical Softw. Eng.*, vol. 10, pp. 311–341, Jul. 2005.
[59] M. Niazi, M. A. Babar, and J. M. Verner, "Software process improvement barriers: A cross-cultural comparison," *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1204–1216, 2010.
[60] M. Spreen, "Rare populations, hidden populations, and link-tracing designs: What and why?" *Bull. Sociol. Methodol./Bulletin de Methodologie Sociologique*, vol. 36, no. 1, pp. 34–58, 1992.
[61] D. A. Dillman, J. D. Smyth, and L. M. Christian, *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Design Method*. Hoboken, NJ, USA: Wiley, 2014.
[62] S. L. Pfleeger and B. A. Kitchenham, "Principles of survey research: Part 1: Turning lemons into lemonade," *ACM SIGSOFT Softw. Eng. Notes*, vol. 26, no. 6, pp. 16–18, 2001.
[63] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *Trans. Softw. Eng.*, vol. 25, no. 4, pp. 557–572, Jul./Aug. 1999.
[64] O. F. Althuwaynee, B. Pradhan, H.-J. Park, and J. H. Lee, "A novel ensemble decision tree-based CHi-squared automatic interaction detection (CHAID) and multivariate logistic regression models in landslide susceptibility mapping," *Landslides*, vol. 11, no. 6, pp. 1063–1078, 2014.
[65] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*. London, U.K.: Springer, 2008, pp. 285–311.
[66] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012.
[67] S. Lock and G. Kotonya, "An integrated framework for requirement change impact analysis," in *Proc. 4th Austral. Conf. Requirements Eng.*, 1999.
[68] S. A. Bohner and R. S. Arnold, *Software Change Impact Analysis*, vol. 6. Los Alamitos, CA, USA: IEEE Computer Society Press, 1996.
[69] S. Ibrahim, N. B. Idris, M. Munro, and A. Deraman, "A requirements traceability to support change impact analysis," *Asian J. Inf. Tech*, vol. 4, no. 4, pp. 345–355, 2005.
[70] S. Rajper and Z. A. Shaikh, "Software development cost estimation approaches—A survey," *Indian J. Sci. Technol.*, vol. 9, no. 31, pp. 177–205, 2016.
[71] R. Wang, P. Peng, L. Xu, X.-X. Huang, and X.-L. Qiao, "A novel algorithm for software development cost estimation based on fuzzy rough set," *J. Eng. Sci. Technol. Rev.*, vol. 9, no. 4, pp. 217–223, 2016.
[72] N. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*. Boca Raton, FL, USA: CRC Press, 2014.
[73] M. Kamalrudin and S. Sidek, "A review on software requirements validation and consistency management," *Int. J. Softw. Eng. Appl.*, vol. 9, no. 10, pp. 39–58, 2015.
[74] O. Gotel and A. Finkelstein, "Extended requirements traceability: Results of an industrial case study," in *Proc. 3rd IEEE Int. Symp. Requirements Eng. (ISRE)*, Jan. 1997, pp. 169–178.
[75] N. Almasri, L. Tahat, and B. Korel, "Toward automatically quantifying the impact of a change in systems," *Softw. Qual. J.*, vol. 25, no. 3, pp. 601–640, 2017.
[76] D. Zowghi and V. Gervasi, "On the interplay between consistency, completeness, and correctness in requirements evolution," *Inf. Softw. Technol.*, vol. 45, no. 14, pp. 993–1009, 2003.
[77] S. Robertson and J. Robertson, *Mastering the Requirements Process: Getting Requirements Right*, 3rd ed. Reading, MA, USA: Addison-Wesley, 2012.
[78] T. H. Nguyen, B. Q. Vo, M. Lumpe, and J. Grundy, "KBRE: A framework for knowledge-based requirements engineering," *Softw. Qual. J.*, vol. 22, no. 1, pp. 87–119, 2014.

[79] F. Kaymaz, "Prioritisation and selection of the right business and it requirements in the software engineering process," in *Proc. Vorgehensmodelle*, 2013, pp. 87–98.
[80] *Standish-Group International Chaos—The State of the Software Industry*, Standish Group Int., West Yarmouth, MA, USA, 2014.
[81] D. Zowghi, F. da Rimini, and M. Bano, "Problems and challenges of user involvement in software development: An empirical study," in *Proc. 19th Int. Conf. Eval. Assessment Softw. Eng.*, 2015, p. 9.
[82] A. Begel and N. Nagappan, "Global software development: Who does it?" in *Proc. IEEE Int. Conf. Global Softw. Eng.*, Aug. 2008, pp. 195–199.
[83] C. K. West, "Four common reasons why projects fail," Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2010.
[84] Q. Yang, S. Kherbachi, Y. S. Hong, and C. Shan, "Identifying and managing coordination complexity in global product development project," *Int. J. Project Manag.*, vol. 33, no. 7, pp. 1464–1475, 2015.
[85] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: Barriers and solutions," *ACM Inroads*, vol. 1, no. 3, pp. 66–78, 2010.
[86] M. Niazi, M. A. Babar, and S. Ibrahim, "An empirical study identifying high perceived value practices of CMMI level 2," in *Proc. Int. Conf. Product Focused Softw. Process Improvement*. Amsterdam, The Netherlands: Springer, 2008, pp. 427–441.
[87] S. U. Khan, M. Niazi, and R. Ahmad, "Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review," *J. Syst. Softw.*, vol. 84, pp. 686–699, Apr. 2011.
[88] S. Ambler, "It project success rates survey results," Ambysoft, Toronto, ON, Canada, Tech. Rep., 2014.
[89] S. Ambler, "The non-existent software crisis: Debunking the chaos report," Ambysoft, Toronto, ON, Canada, Tech. Rep., 2014. [Online]. Available: http://www.drdobbs.com/architecture-and-design/the-non-existent-software-crisis-debunki/240165910

SAJID ANWER received the bachelor's degree in computer science from the COMSATS Institute of Information Technology, Lahore, Pakistan, and the master's degree in computer science from the King Fahd University of Petroleum and Minerals. He is currently pursuing the Ph.D. degree with the School of ICT, Griffith University. Prior to joining Griffith University, he was a Lecturer with the King Fahd University of Petroleum and Minerals, Saudi Arabia. His research interests include global software engineering, software requirements engineering, and behavior Engineering.

LIAN WEN (LARRY) received the bachelor's degree in mathematics from Peking University, the master's degree in electrical engineering from the Chinese Academy of Space Technology, and the Ph.D. degree in software engineering from Griffith University. He is also a 5 dan Go player and likes to run Marathons. He is currently a Lecturer with the School of ICT, Griffith University. Afterwards, he was a Software Engineer and the Project Manager of different IT companies. His current research interests include behavior engineering, complex systems, scale-free networks, software change and evolution, logic programming, and answer set programing.

ZHE WANG is a Lecturer with Griffith University, Australia. His research interests include semantic technologies, artificial intelligence, and ontology-based systems. His research has led to over 40 publications in influential journals and conferences.

SAJJAD MAHMOOD received the Ph.D. degree from La Trobe University, Melbourne, Australia. Prior to pursuing the Ph.D. degree, he was a Software Engineer in USA and Australia. He taught and designed a number of courses related to software engineering at KFUPM. He is currently an Associate Professor of software engineering with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. He is an Active Researcher in the field of software engineering. He has published over 55 articles in peer-reviewed journals and international conferences. He has worked as the Principal and the Co-Investigator in a number of research projects that investigate issues related to component-based software development and global software development projects. His research interests include empirical software engineering, evidence-based software engineering, component-based systems, global software development, and software process improvement in general. He received the excellence in teaching, excellence in instructional technology, and excellence in academic advising awards from KFUPM.