IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 May 2026, accepted 20 June 2026, date of publication 25 June 2026, date of current version 8 July 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3707394

TOPICAL REVIEW

Exploring the Practices and Factors Influencing Requirements Change Management in Agile Global Software Development: A Systematic Literature Review

NEMER WAHBEH[1,2], NOVIA ADMODISASTRO[1], SAADAH HASSAN[1], SALFARINA ABDULLAH[1], AHMAD BAABAAD[3], AND NORAZLINA KHAMIS[4]

[1]Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan 43400, Malaysia
[2]Software Engineering and Information Systems, Ammam Arab University, Jordan Street, Amman 11953, Jordan
[3]Department of Management Information Systems, Administrative Sciences, Hadhramout University, Al Mukalla, Yemen
[4]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah 88400, Malaysia

Corresponding author: Nemer Wahbeh (gs65402@student.upm.edu.my)

This work was supported by the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Malaysia.

ABSTRACT Requirements change management (RCM) in agile global software development (AGSD) represents a challenging and underexplored issue. Despite an increasing number of researchers showing interest in this area, most current studies remain confusing. On the one hand, studies aim to identify "practical" (i.e., actionable) agile practices with contextual influencing factors. However, this confusion has generated conceptually vague results, making it difficult for both researchers and practitioners to derive clear insights and actionable strategies from the findings. These limitations represent a barrier to both the theoretical accumulation of RCM knowledge and providing practitioners with useful guidance on the topic. This study attempts to fill this gap by employing a systematic literature review (SLR) for the 2016–2026 timeframe. The literature review followed Kitchenham and Charters' guidelines and used the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 reporting guidelines. In total, 55 primary studies were identified, coded, and examined using a formally defined coding system that clearly distinguished practices from influencing factors. Based on these analyses, the SLR identified 23 agile RCM practices and grouped them into five practice families: communication, requirements management, management, development/software engineering, and agile practices. Additionally, the analysis also identified eight contextual factors that influence RCM success. An explanatory conceptual model was developed to describe how practices can generate successful RCM processes by identifying six mechanisms: information alignment, requirement clarification, shared understanding formation, decision structuring, adaptive coordination, and technical validation, which can lead to process- and project-level outcomes. Therefore, the results of the SLR present a structured knowledge base that provides evidence for understanding RCM in AGSD and develops a foundation for future empirical verification.

INDEX TERMS Requirements change management (RCM), agile global software development (GSD), agile methodologies, systematic literature review, conceptual model.

I. INTRODUCTION

Global software development (GSD), a strategy that allows companies to develop software using resources from several geographic regions to achieve cost savings by employing a global talent base and faster development facilitated by time-zone advantages, has experienced considerable growth over the years. As GSD strategies continue to expand globally, issues related to effective communication, coordination of geographically distributed team members, and culturally and

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

organisationally different work environments have become increasingly relevant [1], [2].

However, the emergence of agile methodologies offers solutions for those working in environments with highly variable and uncertain conditions. The agile approach relies on an iterative development model and seeks consistent stakeholder feedback, enabling quick responses to changing requirements and uncertain environmental conditions [3]. Additionally, the application of agile frameworks, such as Scrum and Kanban, has been successful in addressing the complexity associated with constantly changing requirements [4], [5]. Nevertheless, although agile methodologies enable the integration of geographically distributed teams, they do not alleviate the difficulties inherent to continuous communication and interaction with knowledge stakeholders who provide feedback and iterate based on changing requirements [2], which can lead to misunderstandings and delays in project timelines.

Requirements change management (RCM) is one of the most difficult and sensitive aspects of agile global software development (AGSD). Although agility provides flexibility and the ability to rapidly respond to changes, the iterative and continuous nature of requirements modification can cause disruptions to a project's scope, timeline, and quality when not properly managed. Additionally, AGSD involves teams operating in geographically dispersed locations worldwide, leading to differences in time zones, organisational structures, and cultures that negatively affect communication, assessments, and decision-making related to requirement changes [1]. These challenges can lead to misunderstandings and delays in implementing necessary changes, ultimately impacting project success. Therefore, organisations should establish structured and transparent methods for managing requirement changes that are consistent with agile principles while also considering the realities of geographically distributed and culturally diverse environments. Prior research has shown that identifying RCM practices in AGSD environments is essential, as is acknowledging the effects of external factors, such as organisation structure, geographic distribution, cultural diversity, and coordination mechanisms, that influence RCM performance but are outside the direct control of AGSD development teams [5], [6]. However, previous research has often treated these items inconsistently, blurring the distinction between practices and factors, as well as between success factors and critical success factors, leading to conceptual ambiguity and overlapping results [6] [7], [8]. Although such inconsistency creates a clear classification problem, it also signifies an operational limitation that restricts the use of research findings in real-world applications.

When practices and contextual factors are grouped under the same analytical framework, practitioners struggle to distinguish what falls within their direct control for implementation from what they must anticipate and adapt to. Consequently, the set of recommendations is difficult to implement in specific GSD configurations, thereby limiting the usefulness of current research for practitioners managing changing requirements in globally distributed agile projects [6], [7].

Recent literature indicates that while agile practices are effective at the operational level, their effectiveness in managing requirements changes successfully is significantly influenced by multi-level external factors [9]. Therefore, the failure to distinguish between practices that teams can directly implement and factors that require organisational level strategies makes it difficult for researchers and practitioners to develop coherent frameworks for managing requirement changes in AGSD contexts. This ambiguity contributes to the limited accumulation of scientific knowledge and hinders the development of integrated conceptual frameworks for change management in globally distributed agile projects [10].

To bridge this gap, this study conducts a systematic literature review (SLR) of research related to RCM in globally agile development environments. This review employed established methodological guidelines to ensure scientific rigour and reproducibility [11], [12], [13]. The aim is to identify agile practices that support requirements and change management, and to pinpoint externalities that positively or negatively influence RCM outcomes. This contributes to a more structured and comprehensive understanding of how requirement changes are managed in globally agile environments.

In this review, we aim to make significant contributions to the RCM field within the context of agile global software development. We provide a contemporary and thorough systematic review that consolidates various viewpoints on RCM within agile, globally distributed development contexts. Furthermore, it clearly distinguishes between agile practices that can be directly implemented by development teams and external factors affecting RCM success that are not in their direct control. This distinction has been largely absent in previous reviews, which conflated these concepts, leading to theoretical overlap. The findings were also grouped into a structured statistical classification that highlights the relative role of different practices in supporting RCM. In addition, this study identifies key research gaps that have not been adequately addressed and seeks to consolidate current insights into the most important agile practices and external factors affecting RCM.

Building on this foundation, this study proposes a conceptual model that explains how different families of practices interact with each other and how external factors influence their effectiveness, either enabling or constraining them. The model also illustrates how these interactions collectively shape the overall requirements and change management environment in globally distributed agile projects. This contribution is important because it moves beyond simple classification toward a more integrative understanding, providing researchers and practitioners with a clearer basis for analysis, decision-making, and future model development.

Finally, the results provide a solid empirical foundation for future research, including the development of conceptual models, practice prioritisation, and the design of decision–support frameworks for practitioners working in globally distributed agile development projects.

A. PRIOR SYSTEMATIC LITERATURE REVIEWS (SLRS) ON AGILE PRACTICES AND FACTORS IN GLOBAL SOFTWARE DEVELOPMENT (GSD): A CRITICAL ASSESSMENT AND GAP ANALYSIS

Over the last decade, several SLRs have examined the use of agile methods, success factors, risks, and the context in which GSD occurs; these reviews collectively provide a foundation for further research on the topic, though they lack an analytical perspective. [14] completed an SLR of 28 studies to determine the global influences affecting agile adaptation. They indicated that Scrum and Extreme Programming (XP) were the dominant agile methodologies used in GSD. Furthermore, inadequate customer participation and insufficient documentation were the most significant barriers to successful implementation [14]. Although useful for describing the scale of agile implementation across various organisations worldwide, this study did not distinguish between practices and contextual circumstances and did not explain how the identified elements contribute to organisational performance. Santos et al. continued this descriptive tradition by providing an extensive list of 48 agile practices from 76 studies published over the last two decades, categorised based on the what, who, and how aspects of practice utilisation [15]. While the study was extremely broad in scope, the authors acknowledged its limitations: it lacked a conceptual or theoretical model to explain agile utilisation, did not consider contextual issues, and failed to establish a relationship between practices and results from an RCM perspective. Study [16] limited its focus to 12 challenges and 13 potential solutions for meeting quality requirements in large-scale distributed agile projects [16], while [17] synthesised the use of agile practices in GSD between 1999 and 2016, proposing the ADAPT framework as a descriptive categorisation of contextual factors [17]. Nevertheless, neither study developed explanatory mechanisms or theorised how agile practices create value through infrastructure operations.

Rashid et al. later conducted a review of the risks involved in using agile for GSD and indicated that communication was the largest risk factor, and that Scrum was the most effective mitigation approach [18]. However, the study combined three different types of analysis – risks, challenges, and factors – into a single category and did not provide any sequence of outcomes. Khan et al. made a more formal contribution by identifying 15 factors that influence the scalability of agile in GSD from a client-vendor perspective and suggesting the Agile Software Process Scaling and Management Model [19]. However, like the other models, the Agile Software Process Scaling and Management Model is merely a categorisation framework rather than an explanatory theory. Moreover, the identified success factors combine actionable practices with structural conditions and outcome descriptors, a limitation acknowledged by the authors. Finally, Conboy et al. identified 24 challenges and 23 best practices for large-scale AGSD and organised them into five categories [20]. Nevertheless, like all previous studies referenced above, this study does not propose a conceptual model and provides no explanation of how practices and challenges interact to produce quantifiable results.

Three structural limitations were consistently observed across these seven reviews. First, the studies did not explicitly and operationally distinguish between actionable agile practice behaviours, which are under the team's control, and the structural conditions and contextual influencing factors that shape practice effectiveness, despite this distinction being fundamental to generating actionable guidance. Second, no study theorises an explanatory mechanism layer that accounts for how and why practices generate value; all seven studies report what practices or factors exist rather than how they produce outcomes through intermediate processes. Third, no study identifies the repository and artefact infrastructure, the shared digital toolchains through which practices are operationalised in distributed settings, as an analytically significant layer in the practice-outcome chain. Consequently, the operational locus that connects practice adoption to the activation of explanatory mechanisms remains implicit. Therefore, the seven reviews provide knowledge that is descriptively rich but explanatorily limited. The studies can inform practitioners about what agile teams do and what conditions they face; however, they cannot explain why the same practice produces different outcomes across different AGSD environments. It is precisely this explanatory gap, i.e., the absence of a theoretically grounded model that connects practices, through their operational infrastructure and activating mechanisms, to outcomes moderated by contextual factors, that the present study is designed to address. Table 1 illustrates this gap across six analytically selected dimensions, demonstrating that the contribution of the present study is structurally distinct from the existing corpus of systematic review evidence in this domain.

What Distinguishes this Study from Previous Studies?

B. RESEARCH BACKGROUND

In contrast to previous studies that considered practices and influencing factors as different categories, the present study does not generalise from prior research. Therefore, the context of the findings should be provided by comparing these two variables with earlier findings. Several studies regarding global distributed software development have uncovered many success factors. As reported by [21], communication among team members is the single most influential success factor for distributed Extreme Programming (XP). Their research provides evidence that effective communication with distributed teams facilitates coordination, consequently enhancing the effectiveness of software development. In con-

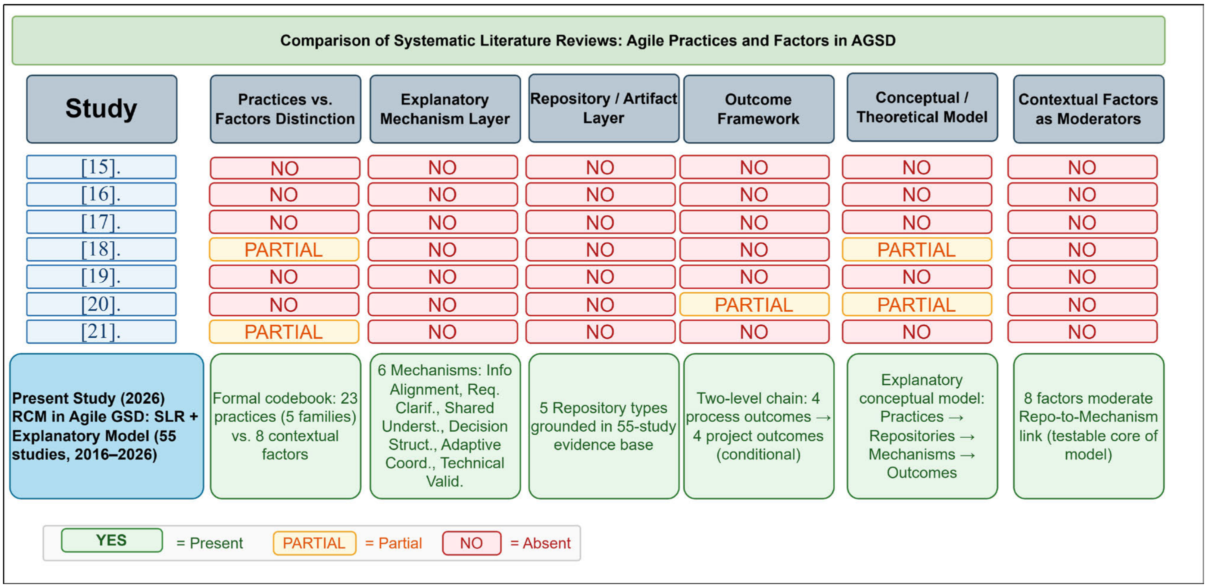

FIGURE 1. What distinguishes this study from previous studies.

trast, [22] cautioned that using heterogeneous communication tools to share project-based information among geographically separated team members may pose specific risks. These include project interruptions, security breaches, and difficulties in ensuring seamless information sharing. The difficulty of communicating across different platforms, particularly in distributed teams, remains a crucial issue in GSD. [3] emphasised the value of customer involvement and constant feedback to the agile development process, especially in distributed environments. Their results also indicate that strong relationships between companies and clients are strengthened when agile teams and customers coordinate and communicate closely, which the Agile Manifesto emphasises in the context of customer collaboration [23]. More recently, Paasivaara [2] conducted an extensive case study investigating how globally distributed software development organisations implement agile practices. This study emphasises the increasing significance of agile methodologies in such environments. Moreover, it proposes a framework to address the challenges of implementing agile methodologies in a GSD context by extending and embedding agile practices within geographically distributed teams. Likewise, [24] conducted an SLR, identified critical growth factors, barriers, and practices for agile development in GSD contexts. They emphasised management support, domain knowledge, training, technological infrastructure, and communication as significant success drivers in GSD organisations. In summary, these studies contribute to the rich literature on the determinants of agile development in a global context and confirm that practices and success factors should be further differentiated, as proposed in the present study.

C. THEORETICAL AND CONCEPTUAL BACKGROUND

A. Global software development (GSD) distributes software development activities across geographically distributed teams and organisations. It offers benefits such as access to global talent, cost efficiency, and accelerated delivery cycles, but also introduces significant coordination challenges. Research consistently delineates three analytically distinct dimensions of distance in GSD: temporal distance, resulting from time-zone separation, consequently hindering synchronous communication; geographical distance, indicating physical separation that obstructs tacit knowledge transfer; and socio-cultural distance, which includes variations in language, working styles, and organisational norms that create interpretative ambiguity in requirements and change requests [1], [2], [25]. Each dimension constrains the effectiveness of RCM practices differently, and treating GSD as a uniform context risks oversimplified conclusions regarding practice effectiveness.

B. Agile software development is an iterative and incremental approach characterised by flexibility, continuous stakeholder collaboration, and rapid response to change, as codified in the Agile Manifesto [26]. While agile frameworks such as Scrum and Kanban have proven effective in managing complex and evolving requirements, their deployment in globally distributed settings introduces additional challenges related to coordination, standardisation, and communication across geographically and culturally diverse teams [4], [5], [27].

C. Requirements Change Management (RCM) is a structured process for identifying, analysing, approving, implementing, and tracking changes to software requirements specifications throughout the project lifecycle, aiming to control scope, schedule, and cost [7]. According to [28], RCM also addresses requirements volatility, which refers to the inherent tendency of requirements to evolve continuously as stakeholder understanding, market conditions, and technical constraints change. This phenomenon is particularly amplified in GSD settings where distributed teams operate under differing information contexts and experience communication delays. Prior research indicates that structured RCM practices, supported by effective communication and stakeholder collaboration, are essential for managing volatility and preventing scope creep and schedule delays in globally distributed agile projects [6], [29], [30].

D. Agile software development practices, as defined in this study, represent deliberate, actionable behaviours, routines, and standard operating procedures (SOPs) that individuals or teams consciously adopt and execute to accomplish development tasks and manage requirement changes [15], [26]. Practices are, by definition, within the direct control of development teams and organisations: they can be intentionally implemented, adjusted, scaled, adopted, or abandoned based on the project context and team capability [2]. Examples include sprint-based iterative development, requirements backlog management, structured change-control processes, and stakeholder feedback loops. This definition aligns with established conceptualisations of practices in software engineering research as the operational and implementable components of development methodology [2], [15].

E. Factors influencing AGSD, as defined in this study, are contextual conditions imposed by the broader organisational, geographical, cultural, or technological environment that shape, enable, or constrain the effectiveness of agile practices in RCM. In terms of control, factors represent the context within which practices operate, rather than being directly enacted by development teams. Their impact may be either enabling or constraining, and their effect on RCM outcomes is moderated by the specific practices and mitigation strategies the team deploys, whether directly or indirectly. Examples include geographical and temporal distribution, cultural and linguistic differences, knowledge management maturity, and organisational governance structures [14], [31], [32].

D. DISTINGUISHING PRACTICES VS. INFLUENCING FACTORS

Rationale and Theoretical Grounding: A continual source of conceptual ambiguity in the current literature on RCM in AGSD is the amalgamation of categories. Previous research has often conflated practices and factors, amalgamating actionable team behaviours, environmental constraints, and successful outcomes within identical analytical frameworks. For instance, multiple studies within the reviewed corpus utilise the overarching construct of "success factors" to concurrently characterise team-level practices, such as sprint-based development, and external contextual conditions, including cultural diversity [7], [8], [14], [19]. Some works also enumerate practices and contextual influences within identical taxonomic categories without specifying operational distinctions [33], [34]. This merger has a specific effect on operations: practitioners may not be able to identify what they can directly control and implement, and what they need to strategically manage or adapt to. Separating practices from factors and learning about both aspects enhances practitioner guidance. Practitioners obtain a clearer understanding of the practices they can perform (practices) and the circumstances under which those practices will lead to success. The theoretical basis for this separation lies in two theories. Coordination theory [35] delineates common practices identified in this review, such as iterative development and stakeholder feedback, and explains how they help reduce coordination costs arising from task interdependencies in distributed work environments. Contingency theory [36] explains the underlying reason for the extensive variability in practice effectiveness across the reviewed studies. This theory asserts that there is no single universal practice that would guarantee success; rather, effectiveness is determined by the context in which it is being performed. The systematic method employed in the Data Coding Procedure helped differentiate between functional roles and surface terminology by applying a coding framework during the data extraction process, as illustrated in Appendices C and D. Appendix C demonstrates the coding index employed to ascertain the influencing external factors, identifying where the factors appear in each study, and providing an outline of the criteria used to determine factor presence. Appendix D enumerates the coding index for practices, outlining how each practice was identified and classified, and where each appears in the reviewed studies.

II. METHODOLOGY

This study employs the SLR methodology, conducted in accordance with the guidelines proposed by Kitchenham and Charters, as well as the Software Engineering Guidelines for Reporting Secondary Studies (SEGRESS) framework for secondary studies in software engineering [12], [13]. The SLR protocol is structured around three phases, namely, planning, conducting, and reporting, and is designed to ensure that all available evidence is gathered systematically in response to the defined research questions, that inclusion and exclusion decisions are reproducible, and that synthesis conclusions are grounded in the retrieved evidence. Figure 1 illustrates the research methodology used in this study to identify and examine agile practices and factors that affect RCM in AGSD settings. Research begins by identifying problems that can be studied and defining the scope of the issues regarding the application of RCM methods and their influencing factors in agile and globally distributed environments. This phase defines the research question(s) that guide the SLR. The next steps involved developing an SLR protocol, including a description of how the search was conducted (the search strategy), the databases used for the literature search, and the criteria for determining whether a source should be included or excluded. In the next phase, the developed protocol was applied to identify all primary studies found during the initial search. After this phase, a rigorous study selection process was undertaken, which involved both screening and quality assessment. The primary studies selected using the specified process were then subjected to structured data extraction to gather study characteristics, RCM-related practices, and factors affecting them. The collected data were analysed using coding and thematic analysis techniques to identify common patterns and themes. All these patterns and themes were subsequently grouped into categories based on their relation to an agile environment or to contextual factors.

A separate process for identifying research gaps was also completed by identifying gaps, inconsistencies, and missing conceptual views pertaining to previous studies to provide a base for future conceptual integration and research directions. This review was conducted in two phases: Phase 1 covered the period from 2016 to 2024, with the database search executed in June 2024. Phase 2 extended the coverage to include studies published between January 2025 and February 2026, with a supplementary search conducted in February 2026. Both phases used the same search string, databases, and inclusion and exclusion criteria. The results were merged before screening, and deduplication was applied across the combined set. The final corpus comprised 55 primary studies.

A. PLANNING PROCESS

The planning process is important for the validity and rigour of an SLR. The process includes formulating research questions, constructing the review protocol, and establishing clear inclusion and exclusion criteria. Such features are critical for guiding this review and facilitating a systematic and unbiased process. The following section provides a thorough explanation of each aspect involved in the SLR planning process.

B. RESEARCH QUESTIONS (RQS)

The formulation and identification of the research questions are the key phases on which the SLR protocol processes are built. Essentially, this represents the researcher's strong realisation of the urgency to deeply understand the previously published investigations. The research topic comprises three RQs, each associated with its motivation (Table 2).

C. REVIEW PROTOCOL DEVELOPMENT

Kitchenham [11], [12], [13] delineated a protocol for systematically conducting a review and mitigating the likelihood of bias in research. In the absence of such a procedure, the selection of sample studies may be based on the researcher's expectations. This might lead to relevant studies being excluded from the sample needed for a comprehensive investigation and a thorough understanding of the phenomenon. A protocol review is based on the specified steps: (i) defining research questions, (ii) formulating a search strategy, (iii) examining selection criteria and methods, (iv) analysing quality evaluation techniques, (v) executing data extraction strategies, and (vi) synthesising the extracted data. The phases of protocol evaluation used for this study are shown in Figure 1.

D. INCLUSION AND EXCLUSION CRITERIA

The inclusion and exclusion criteria were carefully determined to ensure that the review included only those studies that were relevant to the research questions. This process also ensures that the chosen primary studies address the exact SLR question. The literature search focused on a variety of research outputs, including journal articles and conference papers. Figure 1 illustrates the entire search procedure using a primary search string used across multiple databases. The inclusion and exclusion criteria ensured that only English-language studies were included. Studies addressing RCM in GSD that applied agile methodologies were selected. Furthermore, studies deemed irrelevant, vague, or fewer than four pages were not considered due to insufficient detail. The research was based on studies published between 2016 and 2024 to include the latest and most relevant findings available in the literature at the time of the study. The rationale for excluding studies pre-2016 stemmed from the accelerating evolution of agile techniques and the increasing role of globally distributed software development (GSD) paradigms. Indeed, agile techniques such as Scrum and DevOps have come a long way since 2016, adapting to globalisation and technological changes. Jabri's systematic review of the literature, however, shows that agile project management became quite popular from 2017, and interest surged at the beginning of 2021 due to the COVID-19 pandemic, which required the adoption of new techniques for distributed project management [37]. Natarajan [38] emphasises the transition from a plan-driven approach to agile thinking and states that this has led to increased productivity and flexibility in multinational teams by driving process improvements. This focus on studies from 2016 ensures that recent findings on the challenges and strategies of agile methodologies in GSD are included. Abrar [39] shows how agile-related concepts have gradually gained traction at scale and that organisations need to adapt agile principles for larger, globally distributed teams. Similarly, [40] demonstrates the rapid adoption of agile approaches in educational settings to upskill practitioners and foster the skills expected of a 21st-century global workforce.

Additionally, recent research emphasises the rise of tools and frameworks specifically designed for distributed software development. For instance, [41] proposes principles that balance architectural considerations with agile practices, which are crucial for the success of distributed projects. Moreover, the study highlights how organisations increasingly leverage agile methodologies to enhance collaboration and responsiveness across global teams, ensuring alignment with industry trends. This focus on recent studies ensures that the research remains relevant, actionable, and reflective of contemporary practice. Table 3 summarises the criteria applied.

In the context of this study's specific scope, it was challenging to find previous studies that concurrently addressed project RCM, GSD, and agile methodologies. Moreover, the areas of intersection between the fields are not significantly emphasised in the literature, making it difficult to find a substantial number of directly aligned studies. Therefore, the scope of the study was extended to include research on agile

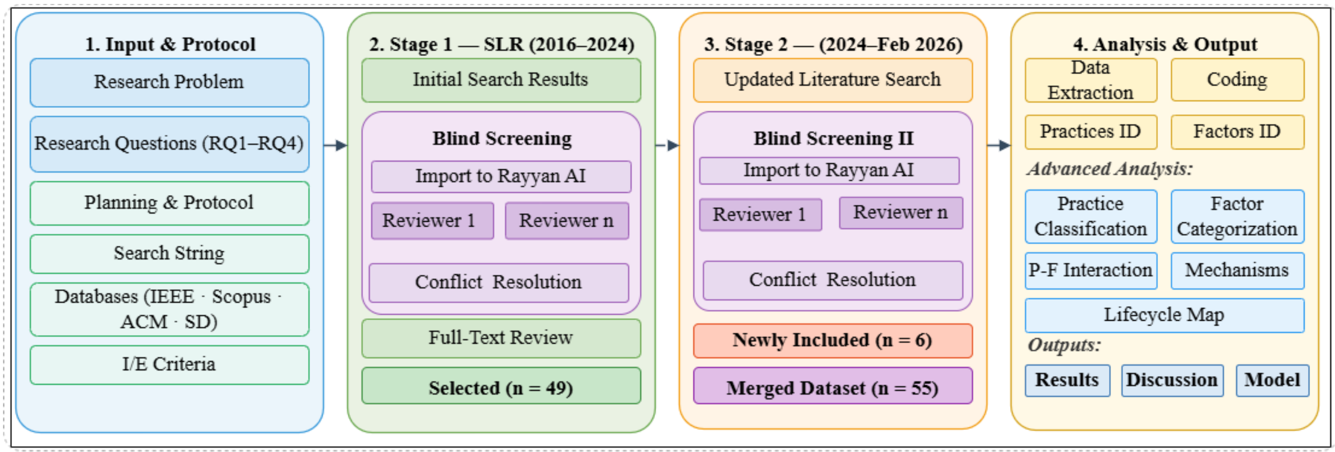

FIGURE 2. The research methodology.

TABLE 1. He research questions.

SN	Research Question (RQ)	Motivation
1	What RCM practices align with agile methodologies in GSD environments?	To identify specific practices related to RCM that are designed to work within the AGSD environment.
2	How do the practices used to manage requirement change systems, that are compatible with agile methodologies in GSD environments, contribute to project success?	To correlate the identified practices directly to project success, ensuring that the SLR not only identifies used practices, but also evaluates their effectiveness.
3	What are the factors influencing the effectiveness of RCM in AGSD environments?	To explore a wide range of factors that affect RCM as a whole in AGSD projects.

methodologies within GSD. This choice was made under the premise that agility in software development directly correlates with the ability to address changing requirements [42], [43]. As a result, the references on agile methodologies in GSD were considered relevant across all three domains of interest in this study, allowing the coverage to be as broad as possible. However, research that addressed only the need for changes in agile development rather than the global scope was excluded. The exclusion was based on the fact that practices that are well-suited to local contexts may not have the same efficacy in global environments, specifically considering geographical, temporal, and cultural challenges. Studies focusing on RCM in traditional GSD that did not consider agile methods were excluded because the iterative, flexible nature of agile processes is significantly different from that of conventional methodologies. Finally, studies lacking adequate discussion about RCM practices or components were excluded to keep the review relevant and focused. The rigour of this approach ensured that the selected studies provided significant insights relating to the intersection of RCM, agile methodologies, and GSD.

Execution Process

Once the protocol is established during the planning phase, the conduct phase begins with a formal review that entails identifying the search strategy, the study selection procedure, the selection of primary studies, and the quality assessment. The following subsections explain the SLR process.

E. SEARCH STRATEGY

A specific and comprehensive search strategy is critical to ensure acceptable results and to provide extensive coverage of published studies pertinent to the specified issues. Kitchenham and Charters [11], [12], [13] set out the search mechanism, which specifies the approach undertaken in the following stages of the primary research, including search string identification and the resources to be examined. The mechanics of the search techniques, both human and automated, are illustrated in Figure 2. Subsequently, the search method is described.

F. IDENTIFICATION OF THE SEARCH STRING

To formulate a proper search string, we applied the protocols recommended by Kitchenham and Charters, as well as the principles and methodologies proposed by Kitchenham which include: (a) identifying key terms from the research questions, (b) identifying synonyms, alternative spellings, and abbreviations for all key terms, (c) verifying the alignment of keywords with relevant research studies based on these steps, (d) using Boolean operators "OR" and "AND", where "AND" is used to connect key terms and "OR" to associate synonyms, alternative words, and abbreviations, and (e) integrating the key terms to create the final search string. The previous processes were finalised and executed to generate the keywords. Subsequently, the implementation of tests using the search string across the selected libraries was verified, yielding the overall search terms used in this

TABLE 2. He inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
Written in English	Written in languages other than English
Published between 2016–2026	Published outside the 2016–2026 range
Directly address RCM in GSD environments using agile methodologies	Address RCM, but do not consider global contexts or the GSD environment
Focus on agile methodologies within the GSD context, even if not directly addressing RCM	Focus solely on agile methodologies or RCM, but in a localised, single-organisation context (non-global context)
Discuss agile methodologies, and how they support managing changing requirements in global environments	Address traditional GSD without incorporating agile methodologies
Clearly discuss practices or factors affecting RCM	Do not contain a clear discussion of practices or factors affecting RCM
Journal articles, conference papers, books, and book chapters	Length is < 4 pages, as it is insufficient to provide a deep analysis

TABLE 3. A list of the databases searched.

Name	URL	Phase 1 Date	Phase 2 Date
IEEE Xplore®	https://ieeexplore.ieee.org	June, 2024	April, 2026
Science Direct©	https://www.sciencedirect.com	June, 2024	April, 2026
Scopus©	https://www.scopus.com	June, 2024	April, 2026
ACM	https://dl.acm.org	June, 2024	April, 2026

study. The following search string, was used to query all four databases. ScienceDirect© limits queries to a maximum of eight Boolean operators per field; therefore, the string was divided into subqueries linked by the "AND" operator and applied sequentially, while the remaining databases were queried using the complete string without modification.

> (“R*equirements change management” OR “requirements evolution management” OR “requirements change control”) OR “requirements volatility”) AND (“agile met.hodologies” OR “agile practices” OR “agile frameworks” OR “agile s.tware development” OR “Scrum” OR “Extreme Programming” OR “XP” OR “Kanban” OR “Lean software development”) AND (“global software development” OR “distributed software development” OR “off-shore software development” OR “geo- graphically distributed software development” OR “globally distributed software team” OR “global software teams”).*
>
> —Query string

G. RESEARCH RESOURCES

The selection of research resources is among the most critical factors determining the efficacy of an SLR. Consequently, researchers must identify public, inclusive research resources that encompass most research, as well as those that are relevant and appropriate for their research. The research resources identified in this study are illustrated in Figure 2. The search string may be refined for some research resources and reorganised according to the search engine's structure. Furthermore, the grammar differs across search engines. The SLR search process was completed in two steps. The first part was a query auto-search across four online databases (repositories), as shown in Table 4.

H. STUDY SELECTION PROCESS AND SELECTION OF PRIMARY STUDIES

This SLR evaluation was undertaken in two parts to ensure the breadth and currency of the evidence. Phase 1 comprised a search of four electronic databases: ACM Digital Library (n = 5), Scopus© (n = 436), IEEE Xplore® (n = 101), and ScienceDirect© (n = 69) from 2016 to 2024, yielding 611 records. Phase 2 increased temporal coverage by including studies published between January 2025 and February 2026, ascertained through searches of the ACM Digital Library (n = 1), Scopus© (n = 2), IEEE Xplore® (n = 3), and ScienceDirect© (n = 2) in February 2026, yielding eight additional records. The same search string, inclusion/exclusion criteria, and quality assessment systems were used in both phases. Before screening, the findings from both phases were integrated and deduplicated using Rayyan.ai (Rayyan, n.d.), as shown in the modified Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 flow diagram (Figure 4). Deduplication removed 36 duplicate records from Phase 1 and none from Phase 2, leaving 583 unique records for title and abstract screening. Records that addressed software development outside a global or geographically distributed setting, used only traditional (non-agile) development approaches, or did not include RCM or associated practices were excluded. This screening process led to the exclusion of 454 records during title and abstract screening. Consequently, 121 records from Phase 1 and 6 records from Phase 2 remained for full-text assessment.

Using Rayyan.ai, all research team members evaluated the candidate studies against pre-set criteria. They were not aware of others' decisions, which was necessary to ensure dependability and objectivity. The criteria were applied consistently during the initial screening phase, yielding 90% agreement [44], [45]. Disputes were resolved through organised discourse, in which the reviewers reviewed each instance until a consensus was reached. Before inclusion, three borderline studies on agile techniques in GSD settings without "RCM" in their title or abstract were reviewed in full, and structured team deliberations were held. The study substantively addressed recurring requirement change requests and distributed agile procedures suitable to manage them, and all three were retained by full consensus [12], [26], [44], [45].

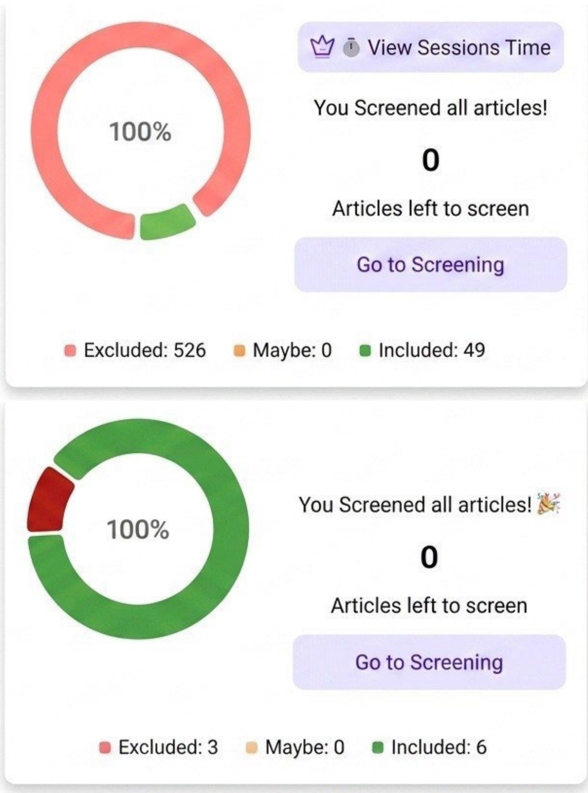

FIGURE 3. Results of screening by Rayyan.ai.

The full-text assessment stage rejected 72 records from Phase 1 and none from Phase 2 after evaluating the study's aims, research methodology, and conclusions. Studies that did not explicitly address RCM or provide concrete evidence for at least one research issue were excluded. Finally, all remaining studies were assessed for objective clarity, methodological transparency, application of the AGSD-RCM domain, and evidence quality (Table 4). No retained study had a quality below 40%; 24 were moderate (40–60%), while 25 were high-quality. The final synthesised corpus included 55 primary studies: 49 from Phase 1 and 6 from Phase 2. Two independent teams extracted the data. The principal investigator and a second researcher used structured extraction forms from [44] and [45] to extract evidence relevant to each study topic. The secondary team independently verified the completeness and accuracy of the data. MS Excel and Mendeley reference management software were used to store all extracted data.

The search was limited to research published from 2016 to 2026 due to three methodological factors. First, 2016 was a well-known turning point regarding the integration of agile approaches and globally distributed development. Agile adoption is evident from the use of scaling frameworks, such as SAFe and distributed Scrum, which have become popular in the industry. Second, limiting the search to this time frame ensures that the evidence base is internally consistent, since research conducted before 2016 was conducted under very different technological and organisational contexts, making it hard to compare it directly with current GSD practice. Third, the 2016–2026 period marked the growth of collaborative tool ecosystems that changed how distributed teams collectively handled requirement changes. It is recognised that the exclusion of foundational contributions made before 2016 is a limitation. However, important foundational works have been added to the theoretical background where they are immediately relevant, even though they are not included in the systematic corpus. Of the 55 studies retained for review, 7 were secondary studies comprising SLRs or mapping studies. The inclusion reasons are methodologically justified as the number of primary studies relevant to each of the three domains of RCM, agile methodologies, and GSD was small, and secondary studies can provide both synthesised theoretical insights and higher-order patterns that may be less apparent from individual primary studies. Secondary studies were not considered independent or supplemental sources of primary empirical data. They provided insight into the theoretical perspectives underlying conceptual patterns. Evidence inflation was further reduced by carefully considering overlapping evidence to prevent double-counting. This approach to integrating evidence follows widely accepted guidelines for SLRs conducted within fields with limited primary evidence, where an inclusive evidence gathering approach is necessary [44], [46].

I. STUDY QUALITY ASSESSMENT

The quality of the selected studies was evaluated using six distinct criteria (Table 4). Each study was assessed based on how well it satisfied these criteria. If a study met a criterion very well and clearly, it was awarded a score of 1. Studies that only partially fulfilled the criteria were given a score of 0.5, while those that did not meet the criterion at all were rated 0. After scoring each study on all six criteria, the average score was calculated.

The studies were then categorised into three quality groups based on their average score:

- Studies with an average score of $< 40\%$ were classified as "weak".
- Studies with an average score of 40–60% were classified as "average".
- Studies with an average score of $> 60\%$ were classified as "high-quality".

Upon applying these criteria, no study was found to score $< 40\%$. As such, no study was classified as weak. Of the 55 retained studies, 26 were classified as average quality (average score of 40–60%) and 29 were classified as high-quality (average score of $> 60\%$). This distribution reflects the effectiveness of the quality threshold that was applied

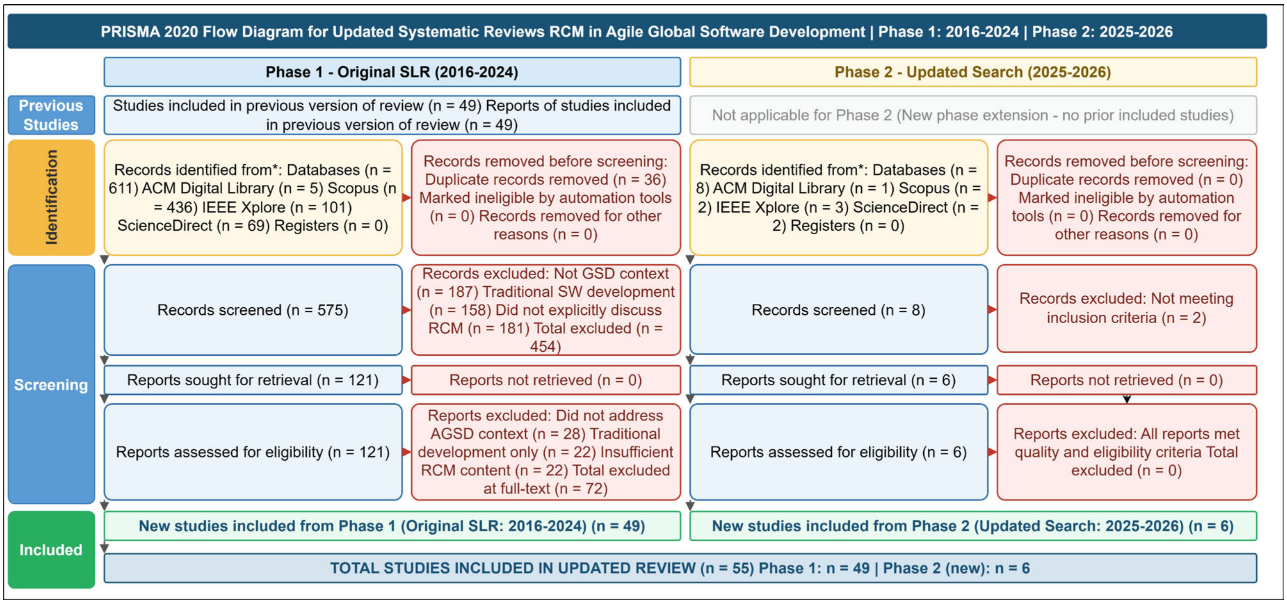

FIGURE 4. Diagram of the two phases in the PRISMA.

TABLE 4. He quality assessment criteria applied to the primary studies.

SN	Quality Assessment Criteria
QA1	Is the objective of the study clearly defined?
QA2	Does the study contribute valuable insights into the field of RCM in GSD?
QA3	Is the study's research methodology well-articulated and transparent?
QA4	Is the study's assessment and evaluation process explicitly detailed and reported?
QA5	Does the study provide a description of the agile practices relevant to RCM in GSD?
QA6	Does the study offer an analysis of the factors influencing agile RCM in GSD?

during the full-text screening. Figure 5 illustrates this distribution.

J. REPORTING PROCESS

Once the study methodology is finalised, the systematic review can be conducted by extracting relevant data in accordance with the defined research questions. The data were synthesised to provide a clear picture of the research, and a decision was reached on the scope of this research, both now and in the future.

1) DATA EXTRACTION PROCESS

Once the primary studies had been selected for the SLR, the data extraction process began to precisely record the information required to answer the established research questions. To ease the data extraction process, data design forms based on instructions in [11], [12], and [13] were employed from the selected studies based on their relevance to the study questions. Appendix A lists the key study references, such as the PS-ID, title, author, year of publication, and publication source. Finally, the data extraction form design was applied to 49 primary studies, and a brief description of all retrieved data items is included in Table 5, which served as the primary source for data synthesis. The extracted data was saved in Microsoft© Excel© files and in Mendeley© reference management software.

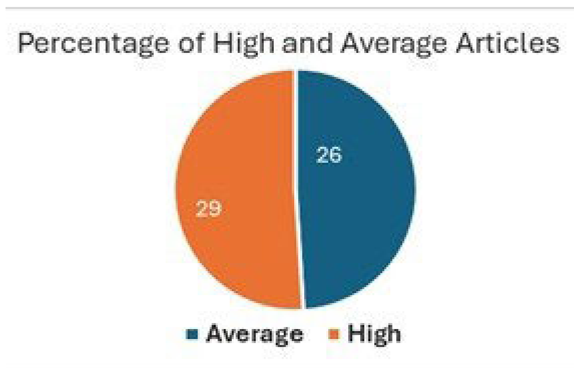

FIGURE 5. The quality of the primary studies.

Data Coding Process

To provide a consistent and systematic method for classifying all collected evidence, the authors developed a formalised coding system. The purpose of developing such a system was to ensure that systematic and transparent methods were employed, leading to a high level of clarity in the analysis. In terms of providing a structure for categorising items during the coding process, the authors' primary guidance source was the conceptual distinction between practices and factors identified in the theoretical and conceptual background (Sections D and E). The coding framework was developed using both deductive and inductive approaches. Initially, the categories were established based on prior research related to GSD and agile methods. As the data were extracted, subcategories and theme refinements were developed. All obtained data were grouped by function, allowing us to distinguish whether each item represented a contextually relevant aspect (factor) that affected how practices operated, or an actionable practice that could be performed directly by the development team. The framework provided that any actions taken by team members, including procedures and techniques, would be classified as practice, while any conditions, constraints, structural attributes, or environmental characteristics would be classified as factors. This ensured that items with ambiguous classifications were consistently categorised throughout the project. In case of potential ambiguities, the team discussed items that could have been classified as either a practice or a factor, using the same structured consensus approach used to select the studies. Appendices C & D contain the entire codebook, including practice definitions, classification criteria, family assignments, and the exemplary literature reviewed.

K. NARRATIVE AND THEMATIC DATA SYNTHESIS

This SLR utilised a narrative and thematic synthesis methodology to amalgamate evidence concerning RCM practices and the factors influencing AGSD. Due to the primarily qualitative and descriptive characteristics of the extracted data, encompassing reported practices, contextual factors, and observations, no statistical aggregation or quantitative synthesis was conducted.

The extracted data were organised into structured summary tables for systematic and clear comparison. Thematic categories were used to group RCM practices and their influencing factors. There was a clear difference between internal, actionable practices and external contextual influences. Occurrence frequencies were computed to demonstrate the relative significance of practices and factors across the analysed studies. The synthesis results are illustrated using narrative interpretation, accompanied by summary and comparison tables (Tables 6–7). In accordance with established protocols for systematic reviews in software engineering, no meta-analysis was performed due to the lack of comparable quantitative outcome measures.

III. RESULTS

This section presents the findings of the review regarding the research areas outlined in Table 6. The research questions were structured in the order RQ1–>RQ2–>RQ3, but the results are presented in a different order: RQ1, RQ3, and then RQ2, to better show the study's logic. It is not possible to meaningfully understand how agile practices help RCM succeed (RQ2) without also taking into account the circumstances in which those practices operate (RQ3). By placing RQ3 before RQ2, the reader can first understand what teams do with regard to agile practices (RQ1) and the environment in which those practices are adopted (RQ3). Then they can read the higher-order explanatory account of how these two elements work together to produce the outcomes described in RQ2. This ordering aligns with a contingency-based epistemological perspective [36], which posits that outcomes are not merely the direct linear result of practices, but rather arise from the interplay between manageable team behaviours and the contextual factors that facilitate or hinder them. The proposed conceptual model formalises this interaction and represents the principal analytical contribution of this review.

A. RQ1: WHAT RCM PRACTICES ALIGN WITH AGILE METHODOLOGIES IN GSD ENVIRONMENTS?

Considering RQ1, the purpose of this review was to investigate and analyse practices for controlling changing requirements in globally distributed agile software development projects. An analysis of 55 primary studies revealed practices that not only conform to agile methodologies but also specifically address the unique challenges of software development across distributed teams and global organisations. This study provided details on methods for managing dynamic requirements, thereby allowing agile teams to seamlessly respond to change and to keep track of stakeholders and the project's aims. The practices are derived from the core concepts of agile methodologies but are further optimised to address the complexities of GSD, which includes communication across time zones, managing distributed teams, and continuing work across geographic boundaries. Table 7 lists some examples of these practices and their role in smoothly managing changing requirements in a global agile environment. By focusing on practices that are implementable and trainable, this review highlights the actionable ingredients that can be highly significant for the success of projects and RCM processes. When addressing the results of the first research question, reports will focus only on practices reported in over 10% of the studies.

The most frequently cited practices among the four agile methodologies, cited in 29 of 55 studies (53%), are sprint-based iterative development and effective communication. The most frequently reported practices implement agility as a continuous adaptation mechanism within the constraints of distributed environments. These practices comprise the main organisational framework for RCM in AGSD. Effective communication facilitates the creation and maintenance of shared knowledge among geographically distributed site teams. Moreover, sprint-based iterative development helps establish a temporal cadence that defines when and how to absorb and act on requirement changes. These practices have been reported as frequently practised concurrently. Their prevalence reflects the fact that the major coordination challenges in GSD environments arise from the difficulty of aligning and providing adequate feedback to team members when they are physically dispersed rather than colocated. The second tier of relatively well-known practices, despite less frequent use, addresses the decision-making and governance aspects of RCM. For example, stakeholder collaboration and continuous feedback (24/55 studies, 44%), prioritisation of needs (15/55 studies, 27%), and agile planning and adaptation (15/55 studies, 27%) are collectively related to a decision

TABLE 5. The data extraction items.

Data Extraction Item	Description
Study Code	The unique identifier that was assigned to each study to differentiate it from the others in this SLR.
Application Year	When the study was published or the year that the research was conducted.
Study Title	The study's title, which provides an initial understanding of its focus and objectives.
Study Aim	The study's main purpose or goal, explaining what it intends to achieve or investigate.
Study Focus	The study's central topic or area of interest, such as specific issues, concepts, or methodologies, within the field of software development.
Study Constraints and Limitations	The study's challenges, restrictions, or limitations, including methodological issues, scope limitations, or contextual challenges, that may impact its results.
Study Design/Research Method	The study's methodology, such as whether it was a qualitative, quantitative, mixed methods study, or if it was based on case studies, surveys, or experiments.
Study Result	The study's key findings or outcomes, summarising the conclusions drawn.
Practices Covered by the Study	The specific agile or software development practices discussed or implemented in the study, particularly in the context of RCM.
Influential Factors Discussed by the Study	The various factors, whether organisational, technological or cultural, that influence or impact RCM processes in terms of AGSD.

TABLE 6. The frequency distribution of the agile practices identified in the primary studies (N = 55).

Practice	Frequency	%
Sprint, Continuous Iterative, Incremental Development	29	53
Effective Communication	29	53
Stakeholder Collaboration and Continuous Feedback	24	44
Agile Planning and Adaptation	15	27
Prioritisation of Requirements	15	27
Prototyping and User Story Validation	9	16
Forming Self-Organising Teams	8	15
Requirements Refinement and Continuous Gathering	9	16
Continuous Validation	9	16
Project Management practices	11	20
Change Control and Governance	9	16
Change Impact Analysis and Reporting	6	11
Agile Roles and Accountability	10	18
Cross-Functional Collaboration via Distributed Tools	8	15
Product Backlog Management	7	13
Software Engineering Practices (Reuse, Modelling)	5	9
Synchronisation and Version Control	5	9
Test-Driven Development (TDD)	5	9
Change Identification	5	9
Documentation of Changes and Rationale	5	9
Coding Standards/Pair Programming	3	5
Requirements Change Management (RCM)	6	11
Standardised Project Process	4	7

discipline that ensures stakeholder needs are balanced with project objectives as the needs evolve. Moreover, project management practices (11/55 studies, 20%), change control and governance (9/55 studies, 16%), and agile roles and accountability (10/55 studies, 18%) provide additional forms of formalised governance to assist distributed teams in coordinating actions related to changes across various geographically remote sites. Recent research indicates growing interest in developing more structured methods for managing requirements evolution.

B. RQ3: WHAT ARE THE FACTORS INFLUENCING THE EFFECTIVENESS OF RCM IN AGSD ENVIRONMENTS?

Before moving to the explanatory synthesis in the following section, we need to provide the context in which RCM

TABLE 7. He frequency distribution of the factors identified in the primary studies (N = 55).

Factor	Freq	%
Geographical and Temporal Distribution	28	50.90%
Cultural and Linguistic Differences	26	47.30%
Knowledge Management	23	41.80%
Organisational and Management Issues	23	41.80%
Team Dynamics and Human Resource Management	21	38.20%
Technology and Infrastructure Challenges	21	38.20%
Project Scope, Budget, and Timeline Constraints	18	32.70%
Process and Methodology Issues	15	27.30%

practices are implemented. The eight influential factors identified in the 55 primary studies are not merely barriers that must be removed; rather, they are structural conditions that delineate the problem space within which any practice must operate. It is analytically necessary to understand these factors before interpreting practice effectiveness. A practice that seems effective based on the literature may practically be very sensitive to one or more of these factors, and that sensitivity is precisely what the conceptual model in RQ2 is meant to show. Table 8 summarises the eight factors and indicates how frequently they are cited in the corpus. Three observations organise the subsequent interpretation.

The two most cited factors, i.e., cultural and linguistic differences (22/55, 45%) and geographical and temporal distribution (22/55, 45%), are not independent. They occur at the same time in the same studies and complement each other's effects: temporal distance makes real-time communication more challenging, which limits how cultural misunderstandings can be resolved, while cultural diversity makes it harder to understand artefacts and documents produced at different times. They collectively embody the fundamental structural challenge of AGSD that any efficacious RCM practice must be designed to address.

Second, knowledge management (19/55, 39%) is different from the other factors. It is partly actionable, unlike the others. Organisations can invest in knowledge repositories, documentation standards, and structured onboarding to lessen their effects. This position is knowledge management at the intersection of an influential factor and a practice, and its prominence in the evidence base indicates that the distinction between contextual conditions and actionable behaviours is not always clear-cut, underscoring the conceptual contribution of this study's explicit coding framework.

Third, the other five factors, i.e., team dynamics and HR management (17/55, 35%), organisational and management issues (17/55, 35%), technology and infrastructure challenges (15/55, 31%), project scope, budget, and timeline constraints (15/55, 31%), and process and methodology issues (9/55, 18%) show how well RCM works in terms of people and resources. Their lower frequency does not mean they are less important; it may simply reflect that they are less common in the field because communication and cultural factors have received more research attention than organisational governance and resource constraints.

None of these factors is determinative alone. The influence on RCM outcomes is variable, contingent on the implementation of practices and repository-based mechanisms used to manage them. The model in Section 4.3 is meant to explain and operationalise this interaction between contextual factors and practice-mechanism chains. When interpreting the explanatory synthesis that follows, the reader should consider the eight factors which comprise the moderation layer of the proposed model.

C. RQ2: HOW DO THE PRACTICES USED TO MANAGE REQUIREMENT CHANGE SYSTEMS, THAT ARE COMPATIBLE WITH AGILE METHODOLOGIES IN GSD ENVIRONMENTS, CONTRIBUTE TO PROJECT SUCCESS?

After determining the existing practices (RQ1) and the contextual factors influencing their operational environment (RQ3), this section progresses to the primary explanatory contribution of the review: a theoretically informed explanation of how practices create value within AGSD constraints. The analysis does not regard success as a direct consequence of practice adoption. Instead, it presents an explanatory conceptual model derived from combining 55 primary studies. In this model, practices lead to outcomes through a four-stage chain, which is affected at a key point by the contextual factors identified in RQ3.

Answering RQ2 rigorously requires moving beyond cataloguing prevalent practices toward explaining how they generate value under the structural constraints of globally distributed agile development. A preliminary examination of the evidence base reveals that explicit treatment of RCM success is relatively narrower than the overall practice evidence: only 13 of the 55 reviewed studies explicitly articulate RCM success factors, while 19 discuss project level success outcomes. This gap confirms that the available evidence does not support a direct, practice-to-success causal chain. Accordingly, the answer to RQ2 is structured through an explanatory conceptual model developed inductively from the synthesis of the 55 reviewed studies. The model proposes that practices contribute to project success through a four-stage layered chain: practices are first organised into functional families, then enacted through specific shared repositories and artefacts, which in turn activate explanatory mechanisms, finally producing observable outcomes at the process and project levels. Each stage of this chain is described below.

Stage 1 — Practice Families. The 23 identified practices are first organised into five coherent practice families that reflect their primary functional role in the RCM process. (1) Communication practices encompass effective communication, stakeholder collaboration, continuous feedback, and synchronisation and version control practices whose central function is maintaining shared information across distributed boundaries. (2) Requirements practices include prioritisation of requirements, continuous gathering and refinement of requirements, backlog management, and change identification practices that manage the content and evolution of the requirements corpus. (3) Management practices cover project management, change control and governance, agile roles and accountability, and standardised project processes and practices that structure authority, responsibility, and decision making. (4) Development/software engineering practices comprise TDD, coding standards, pair programming, prototyping and user story validation, and software engineering reuse and modelling practices that ensure technical accuracy and implementation quality. (5) Agile practices include sprint-based iterative development, agile planning and adaptation, continuous validation, and forming self-organising teams' practices that embed the adaptive rhythm through which changing requirements are absorbed. This family based organisation is not merely taxonomic; it provides the conceptual bridge between a flat list of practices and an explanatory account of how they collectively produce value.

Stage 2 — Repository and Artifact Layer. A distinctive finding of this synthesis is that practices are not enacted in isolation. The evidence across the 55 reviewed studies consistently shows that practices operate through, and leave traces within, shared repositories and artefacts that serve as the operational infrastructure of distributed agile work. This repository and artefact layer, identified inductively from the data rather than being derived from prior frameworks, represents the physical and digital loci through which practice families are operationalised in globally distributed toolchains. Collaboration platforms and shared communication tools are referenced in 29 of the 55 studies; documentation and traceability artefacts in 24; version control and code repositories in 20; and backlog and requirements repositories in 13. Each practice family maps primarily to a specific repository type: Communication practices are anchored in collaboration repositories that enable cross-site alignment. Requirements management and agile practices operate through backlog and requirements repositories. Management practices are materialised using governance and change record repositories. Development/software engineering practices are enacted through code repositories and continuous integration/continuous delivery pipelines. Continuity across team boundaries and time zones is maintained through knowledge repositories that preserve decision rationale and traceability links. This layer is analytically important for two reasons. First, it explicitly identifies where practices produce observable and measurable evidence, an aspect that has consistently remained implicit in the reviewed literature but has not been formally theorised. Second, it establishes a foundation for future empirical validation, since repository traces, which include backlog evolution patterns, commit logs, change-log completeness, trace link density, continuous integration validation outcomes, are, in principle, measurable through quantitative methods such as structural equation modelling or repository mining.

Stage 3 - Mechanisms Layer. Practices enacted through repositories do not directly produce outcomes. They generate value by activating recurring explanatory processes, referred to mechanisms, which alignmentPunderstanding of requirements and change-specific decisions. (2) Requirement clarificationIterative, arteunderstanding formationRhelps develop shared mental frameworks for requirements, change intent, and project priorities across team and organisational boundaries.

Gordered and auditable decision pathways for evaluating, approving, and prioritiscoordinationIterative practices enable teams to continuously recalibrate plans, roles, and processes in response to emerging change-related information, maintaining development continuity without rigidity.

(6) Technical validationDexplore in detailFigure 6 illustrates an updated model that adds an intermediate layer of digital repositories and tools between practices and interpretive mechanisms. In this model, practices are not activated directly; rather, they operate through digital structures that support coordination, information storage, and communication within agile, distributed teams. Each practice category is associated with a specific type of repository, while knowledge repositories operate as a cross-cutting layer across all categories. Primary and secondary relationships are indicated by different arrows, whereas eight mediating factors influence the transition to the mechanisms layer, determining the strength of the practices' impact on the final outcomes.

Contextual Moderation of the Repository-to- Mechanism Link. The eight influential factors identified in RQ3 enter the model at a specific, theoretically motivated point: between the repository and artefact layer (Stage and the explanatory mechanism layer (Stage 3). This placement reflects the position that contextual factors do not prevent practices from being enacted, nor do they directly determine project outcomes. Rather, they alter the efficiency and completeness with which shared repositories and artefacts activate the explanatory mechanisms. For instance, a high temporal distance does not eliminate the use of collaboration repositories, but it constrains their synchronous operation, thereby weakening the activation of information alignment and the formation of shared understanding. Similarly, high cultural distance increases the cognitive effort required to transform governance repository entries into shared decision structures, slowing mechanism activation without disabling it completely.

Low technological infrastructure maturity limits the reliability of code repositories and continuous integration pipelines, thereby degrading technical validation even when the practice is formally adopted. This moderating relationship constitutes the testable core of the proposed model and serves as the primary agenda for future empirical validation through structural equation modelling/partial least squares, or repository-mining methods. Formally: the same practice, enacted through the same repository type, will activate mechanisms with varying strength depending on the factor profile of the AGSD environment, and this variation explains the heterogeneity of RCM outcomes reported across the reviewed studies. Figure 7 shows the proposed conceptual framework for RCM in AGSD. The framework is based on the results of RQ1 and RQ3, and on the explanatory analysis of RQ2. It integrates the identified practice families, the repository and artefact layers that enable these practices, and the underlying explanatory mechanisms that turn practice enactment into observable outcomes. It also includes important contextual factors that serve as moderating conditions, affecting how well this transformation process works. The model is shown as a structured, multi-layered picture of how RCM practices create value in distributed agile settings. This sets the stage for future empirical testing and analytical improvement.

systems that **Stage 4 — Process and Project Outcomes.** The activation of mechanisms through repository-anchored practices produces outcomes at two distinct levels. At the process level, the mechanisms generate four observable improvements: coordination efficiency, reflecting reduced delays and misalignments in change management across distributed teams; decision quality, reflecting more consistent, justified, and traceable change approval decisions; requirement clarity, reflecting reduced ambiguity and rework arising from misunderstood or under-specified changes; and change consistency, reflecting the reliable propagation of approved changes across distributed codebases and team artefacts. At the project level, sustained process-level improvements culminate in downstream outcomes that the reviewed studies associate with RCM success: reduced scope creep, faster adaptation to legitimate requirement changes, improved overall requirement quality, and enhanced RCM effectiveness. Crucially, this model treats project-level outcomes as conditional and context-sensitive rather than as direct causal effects of any single practice. The same practice may produce different project outcomes depending on the contextual factors, such as temporal distance, cultural diversity, organisational governance maturity, and technological infrastructure within which it is deployed. This conditionality consists of a contingency-theoretic interpretation of the findings [36] and explains the heterogeneity of success outcomes reported across the reviewed studies.

Synthesis. The layered model proposed in response to RQ2 advances beyond the descriptive practice catalogues that characterise prior research in this domain. By introducing the

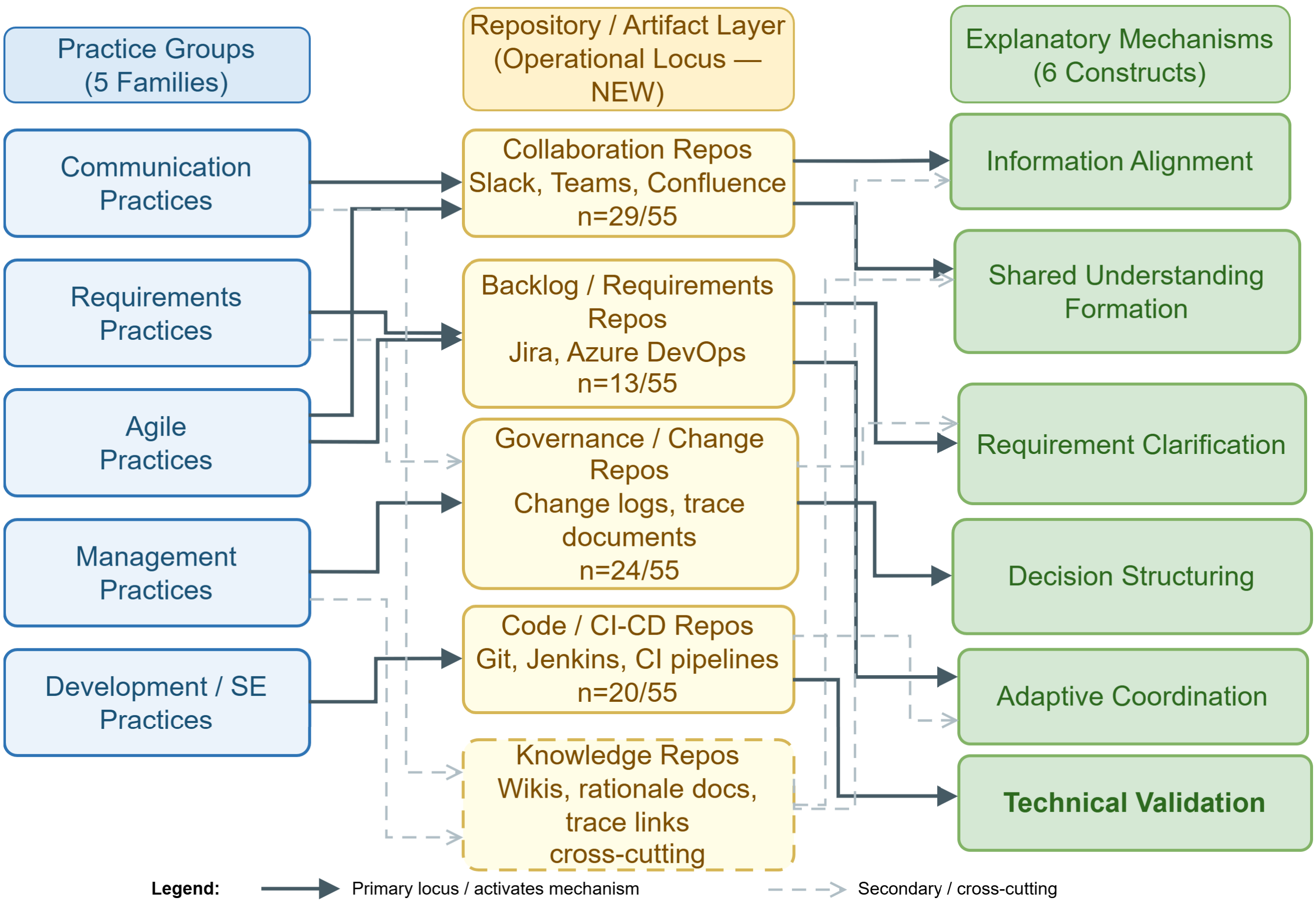

FIGURE 6. **Practice-to-outcome pathways through repositories and explanatory mechanisms.**

repository and artefact layer as an explicit operational stage between practices and mechanisms, the model explains not only which practices are used but also where they are enacted and how their effects propagate through the distributed team infrastructures. The model is not presented as a validated causal theory; it is an explanatory synthesis grounded in the patterns observed across 55 primary studies, intended to serve as a structured foundation for future empirical validation through methods such as partial least squares structural equation modelling, repository mining, and longitudinal field study. Figure 7 presents the complete conceptual model.

IV. DISCUSSION

By synthesising evidence from 55 primary studies, the authors move beyond merely describing practices and factors to explain why RCM succeeds or fails in globally distributed agile settings. The conceptual model proposed here, arranged in a five-stage sequence from practice family to shared repository and artefacts, to explanatory mechanisms, to process-level outcomes, and finally to project level outcomes, with contextual factors moderating the link between the repository and mechanism layers, represents the primary analytical contribution of this review. The following discussion establishes where this model fits within the larger body of knowledge, interprets the findings revealed by the model, and identifies directions for future research emerging from the recommended framework.

Communication and iterative practices are dominant. Communication, as related to providing information, iterative development (in terms of sprints), and collaboration among stakeholders, were the top three practices reported in 55 studies reviewed as effective at addressing the high cost of coordinated effort, such as getting people around the world to accomplish tasks. According to coordination theory [35], these practices are important because, in situations where task dependencies cannot be addressed using face-to-face communication or synchronous interactions, practices that systematically reduce informational asymmetry, such as ensuring everyone knows what others know, and maintaining a common understanding among team members, are necessary rather than optional. While the literature may indicate that such practices dominate the current situation, their prevalence is not simply a temporary phenomenon. Rather, it indicates a basic structural requirement of AGSD. At the same time, the reviewed studies indicate relatively less emphasis on specific practices, such as documentation and governance. Such an imbalance is not accidental; in fact, it reflects a longstanding philosophical divide within agile

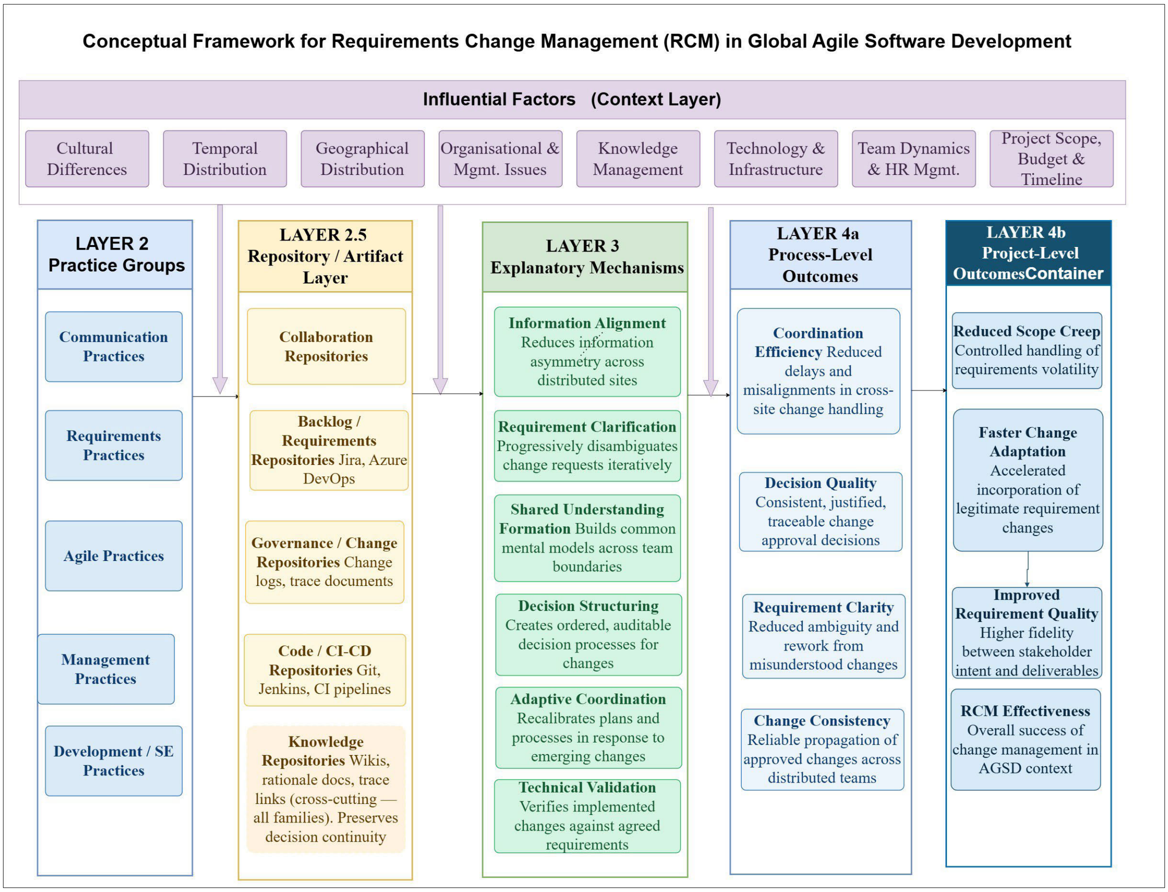

FIGURE 7. The proposed conceptual model.

methodologies. On the one hand, in an agile environment, there is a need to minimise documentation, as excessive documentation can reduce agility. On the other hand, there is a growing recognition of the need for coordination in globally distributed teams. Such teams comprise geographically distributed stakeholders separated by cultural and temporal differences. Hence, it cannot be assumed that they will understand one another; instead, they must actively build and maintain that understanding through effective communication. The proposed model directly addresses this issue by establishing documentation and traceability artefacts as a separate layer in the repository structure spanning all five practice families. Therefore, while documentation was historically viewed as a secondary issue relative to other concerns (i.e., speed/agility), with this proposed model, documentation becomes a structural enabler of activating mechanisms that are particularly important for forming and maintaining shared understanding and structuring decisions. **Influencing factors**. Another limitation of prior research in this area is that it treated contextual factors as determinative barriers. For instance, cultural differences may hinder communication, temporal distance may hinder synchronous coordination, and so on. The current synthesis supports neither of these views. Instead, the proposed model treats the eight identified contextual factors as moderators of the repository-to-mechanism link rather than as direct antecedents of outcomes. There are several practical and theoretical implications to making this distinction. From a practical perspective, this distinction implies that the same practice performed in the same repository context will activate mechanisms with varying degrees of intensity based on the characteristics of the AGSD environment, but it will not completely prevent those mechanisms from being activated. For example, organisations with extensive temporal distances can partially mitigate this distance through disciplined asynchronous coordination processes/protocols, strong backlog governance, and artefact-based visibility.

Similarly, organisations with highly diverse cultures can partially mitigate these differences through more explicitly defined communication protocols and shared vocabulary. Thus, the factors constrain or moderate effectiveness, but they do not determine outcomes directly. Theoretically, this position aligns with contingency theory [36], [47], which suggests that organisational effectiveness is a function of the fit between structural conditions and response strategies rather than structural conditions alone. The proposed moderation relationship is thus a testable core of the proposed framework. Future empirical work should not merely exam-ine whether cultural distance impacts outcomes; rather, the emphasis should be on how it attenuates specific repository-to-mechanism pathways under specified conditions.

Synthesising fragmented catalogues into explanatory synthesis. The most significant limitation of previous work in this domain is the lack of integrated explanatory frameworks.

From fragmented catalogues to explanatory synthesis. The most significant limitation of prior research in this domain, as confirmed by the present synthesis, is the absence of integrated explanatory perspectives. Existing studies focus on individual practices or discrete factors in isolation, thereby producing catalogues that cannot properly explain interaction effects, contextual variation, or the mechanisms through which practices generate value. The proposed model addresses this limitation by addressing three structural factors that prior work has not addressed simultaneously. First, the model separates practices from factors operationally rather than terminologically. Second, it introduces the repository and artefact layer as an explicit, evidence-grounded operational stage that prior frameworks have implicitly considered. Third, it theorises contextual factors as moderators of a specific link in the practice outcome chain rather than as diffuse background conditions. Together, these contributions transform the findings from a descriptive inventory into an explanatory architecture that, in principle, can generate specific, falsifiable hypotheses. For instance, the framework suggests that high temporal distance attenuates the effect of collaboration repository intensity on information alignment, or that governance repository completeness partially compensates for cultural distance in its effect on decision structuring.

Implications for practice. For organisations managing requirement changes in globally distributed agile environments, the model offers more precise guidance than conventional practice checklists. The recommendation is not merely to adopt a set of practices but to ensure that practices are enacted through the appropriate repository infrastructure and that the mechanisms they are intended to activate are monitored as observable process-level outcomes. Specifically, collaboration repository discipline, which includes response times, participation patterns, and artefact completeness, should be monitored as a proxy for the activation of information alignment and shared understanding formation. Similarly, backlog repository health comprises aspects such as item clarity, priority stability, and change velocity, which may act as proxies for requirement clarification and adaptive coordination. Governance repository completeness may act as a proxy for decision structuring. Also, the reliability of the continuous integration/continuous delivery pipeline could serve as a proxy for technical validation. Organisations operating under high contextual factor load, including large temporal distance, high cultural diversity, and complex governance, should invest disproportionately in the repository infrastructure rather than merely adding more practices. The model predicts that weakened mechanism activation under adverse factor conditions is primarily a repository-quality problem rather than a practice-adoption problem.

TABLE 8. Primary study evaluation.

PS-CODE	QA1	QA2	QA3	QA4	QA5	QA6	Reff
PS-1	1	0.5	1	0	0	0	[49]
PS-2	1	0.5	1	0	0	0	[50]
PS-3	1	0.5	1	1	0	1	[51]
PS-4	1	1	1	0	0	0	[52]
PS-5	1	1	1	1	0	0	[21]
PS-6	1	1	1	1	0	0	[53]
PS-7	1	1	1	1	1	1	[49]
PS-8	1	1	1	1	1	1	[54]
PS-9	1	0.5	1	0	0	1	[55]
PS-10	1	1	1	0	1	1	[20]
PS-11	1	0.5	1	0	0	0	[19]
PS-12	1	0	1	0	0	0	[18]
PS-13	1	1	1	1	0	1	[56]
PS-14	1	0.5	1	1	0	0	[57]
PS-15	1	1	1	0	0	0	[35]
PS-16	1	0.5	1	1	0	1	[58]
PS-17	1	1	1	0	0	1	[59]
PS-18	1	0	1	0	0	0	[17]
PS-19	1	0.5	1	1	0	0	[60]
PS-20	1	0.5	1	0	0	0	[61]
PS-21	1	1	1	1	0	0	[62]
PS-22	1	0.5	1	0	0	0	[63]
PS-23	1	0.5	1	0	0	0	[64]
PS-24	1	0.5	1	0	0	0	[65]
PS-25	1	0.5	1	0	0	1	[8]
PS-26	0	0	1	1	0	1	[66]
PS-27	1	1	1	0	0	1	[67]
PS-28	1	1	0	1	0	1	[68]
PS-29	1	1	1	1	0	0	[69]
PS-30	1	1	0.5	1	0	0	[34]
PS-31	1	1	1	1	0	0	[70]
PS-32	1	1	1	1	0	0	[71]
PS-33	1	1	0	0	0	0	[72]
PS-34	1	1	0	1	0	0	[9]
PS-35	1	1	1	0	0	0	[73]
PS-36	1	1	1	1	0	1	[74]
PS-37	1	1	1	0	0	0	[75]
PS-38	1	1	1	1	0	0	[76]
PS-39	1	0	1	1	0	0	[77]
PS-40	1	0.5	1	1	0	0	[78]
PS-41	1	1	0.5	0	0	0	[79]
PS-42	1	1	1	1	0	1	[80]
PS-43	1	1	1	1	0	0	[81]
PS-44	1	1	0.5	1	0	0.5	[82]
PS-45	1	1	1	1	0	0	[16]
PS-46	1	0	1	1	0	1	[15]
PS-47	1	1	1	1	0	0	[83]
PS-48	1	1	1	1	0.5	0	[84]
PS-49	1	0.5	0.5	0.5	1	1	[85]
PS-50	1	1	1	0.5	0.5	1	[86]
PS-51	1	1	1	1	0.5	1	[87]
PS-52	1	1	1	1	0.5	1	[88]
PS-53	1	1	1	0.5	0.5	1	[89]
PS-54	1	1	0.5	0.5	0.5	1	[90]
PS-55	1	1	1	1	0.5	1	[91]

Implications for research. The proposed model establishes a structured agenda for future empirical work. Three research directions follow directly from its architecture. First, the moderation relationships between contextual factors and the repository-to-mechanism link should be tested using structural equation modelling/partial least squares with data from AGSD organisations, yielding factor-specific moderation coefficients to guide context-sensitive practice recommendations. Second, repository mining studies should operationalise the model's repository layer through measurable

TABLE 9. Coding framework for influencing factors.

ID	Factor	Definition	Classification Rule	Freq	Supporting Studies (PS IDs)
F1	Geographical and Temporal Distribution	Spatial and temporal dispersion of teams across locations and time zones	Classified as a factor if it reflects **physical or temporal separation influencing coordination**	28	PS-6, PS-9, PS-11, PS-12, PS-13, PS-14, PS-16, PS-17, PS-20, PS-21, PS-22, PS-23, PS-29, PS-30, PS-31, PS-32, PS-33, PS-35, PS-40, PS-44, PS-48, PS-49, PS-50, PS-51, PS-52, PS-53, PS-54, PS-55
F2	Cultural and Linguistic Differences	Differences in language, norms, and communication styles across teams	Contextual influence affecting communication and understanding	26	PS-1, PS-6, PS-9, PS-11, PS-12, PS-13, PS-14, PS-16, PS-19, PS-20, PS-21, PS-22, PS-23, PS-30, PS-31, PS-32, PS-33, PS-35, PS-40, PS-44, PS-45, PS-49, PS-51, PS-52, PS-53, PS-54
F3	Knowledge Management	Processes related to knowledge sharing, transfer, and retention	Factor if it influences how knowledge is created or shared	23	PS-9, PS-10, PS-12, PS-14, PS-17, PS-19, PS-20, PS-21, PS-22, PS-24, PS-28, PS-31, PS-32, PS-34, PS-36, PS-38, PS-42, PS-46, PS-47, PS-51, PS-52, PS-54, PS-55
F4	Organizational and Management Issues	Organizational structure, policies, and management practices	External structural influence beyond team-level control	23	PS-1, PS-6, PS-9, PS-10, PS-12, PS-13, PS-14, PS-19, PS-20, PS-25, PS-26, PS-28, PS-33, PS-34, PS-38, PS-42, PS-44, PS-50, PS-51, PS-52, PS-53, PS-54, PS-55
F5	Team Dynamics and Human Resource Management	Team cohesion, skills, motivation, and collaboration behavior	Factor if it reflects human or team-related conditions	21	PS-1, PS-5, PS-7, PS-9, PS-10, PS-12, PS-13, PS-16, PS-19, PS-23, PS-27, PS-29, PS-32, PS-33, PS-34, PS-38, PS-44, PS-51, PS-52, PS-54, PS-55
F6	Technology and Infrastructure Challenges	Availability and adequacy of tools, platforms, and infrastructure	Enables or constrains execution of practices	21	PS-5, PS-7, PS-9, PS-10, PS-12, PS-20, PS-28, PS-29, PS-30, PS-34, PS-35, PS-36, PS-40, PS-42, PS-49, PS-50, PS-51, PS-52, PS-53, PS-54, PS-55
F7	Project Scope, Budget, and Timeline Constraints	Limitations related to project resources and constraints	External project-level condition affecting outcomes	18	PS-5, PS-9, PS-10, PS-13, PS-16, PS-19, PS-23, PS-24, PS-25, PS-27, PS-32, PS-38, PS-44, PS-48, PS-49, PS-51, PS-53, PS-54
F8	Process and Methodology Issues	Limitations or misalignment in development processes or methodologies	Factor if it reflects process-level constraints	15	PS-1, PS-5, PS-10, PS-12, PS-18, PS-20, PS-25, PS-40, PS-42, PS-50, PS-51, PS-52, PS-53, PS-54, PS-55

digital traces, including backlog evolution patterns, commit frequencies, change-log completeness, trace link density, and continuous integration validation outcomes, enabling a quasi-experimental examination of how repository-level variables mediate between practice adoption and mechanism activation. Third, longitudinal field studies should examine whether the factor-moderation effect attenuates over time as organisations develop compensatory routines and repository disciplines, since the current cross-sectional evidence base cannot assess temporal dynamics. The proposed model is not presented as a validated causal theory; it is an explanatory synthesis grounded in 55 studies, designed to serve as a falsifiable foundation for the next generation of empirical work in this domain.

V. THREATS TO THE VALIDITY OF THE STUDY

As with any SLR, this investigation is vulnerable to specific validity threats that must be acknowledged and critically evaluated to ensure a balanced interpretation of the findings. One key limitation is the potential for selection bias, which might have been introduced through the identification and incorporation of primary studies. Although rigorous inclusion

TABLE 10. Codebook for agile RCM practices.

ID	Practice Name	Definition	Classification Rule	Frq	Study IDs
P1	Sprint, Continuous Iterative, Incremental Development	Iterative delivery approach based on short cycles enabling continuous change handling	Classified as Practice if it represents **structured iterative execution**	29	PS-1, PS-3, PS-4, PS-6, PS-10, PS-11, PS-12, PS-13, PS-14, PS-18, PS-19, PS-21, PS-24, PS-26, PS-28, PS-29, PS-32, PS-34, PS-35, PS-36, PS-37, PS-38, PS-39, PS-42, PS-44, PS-49, PS-52, PS-53, PS-54
P2	Effective Communication	Continuous exchange of information among distributed stakeholders	Must represent **intentional communication activity**	29	PS-1, PS-11, PS-12, PS-15, PS-16, PS-18, PS-19, PS-21, PS-23, PS-25, PS-27, PS-30, PS-31, PS-32, PS-33, PS-34, PS-36, PS-37, PS-38, PS-39, PS-40, PS-47, PS-49, PS-50, PS-51, PS-52, PS-53, PS-54, PS-55
P3	Stakeholder Collaboration and Continuous Feedback	Ongoing engagement with stakeholders to refine requirements	Must involve **interaction with stakeholders**	24	PS-5, PS-6, PS-10, PS-13, PS-14, PS-15, PS-16, PS-19, PS-24, PS-26, PS-29, PS-34, PS-36, PS-37, PS-38, PS-39, PS-40, PS-42, PS-43, PS-47, PS-49, PS-50, PS-52, PS-54
P4	Agile Planning and Adaptation	Adaptive planning to accommodate requirement changes	Must involve **planning or re-planning activity**	15	PS-2, PS-3, PS-14, PS-17, PS-20, PS-22, PS-31, PS-33, PS-37, PS-39, PS-41, PS-42, PS-49, PS-53, PS-54
P5	Prioritization of Requirements	Ranking requirements based on value and urgency	Must involve **decision-making on requirement importance**	15	PS-11, PS-13, PS-14, PS-15, PS-16, PS-18, PS-19, PS-29, PS-33, PS-37, PS-42, PS-47, PS-51, PS-52, PS-53
P6	Prototyping and User Story Validation	Use of prototypes to validate requirements	Must involve **early validation activity**	9	PS-2, PS-4, PS-12, PS-15, PS-19, PS-29, PS-37, PS-43, PS-48
P7	Self-Organizing Teams	Teams autonomously manage their work and decisions	Must reflect **team structure/behavior**	8	PS-14, PS-15, PS-16, PS-19, PS-33, PS-34, PS-36, PS-41
P8	Requirements Refinement and Continuous Gathering	Continuous elicitation and refinement of requirements	Must involve **ongoing requirement updates**	9	PS-19, PS-20, PS-26, PS-28, PS-33, PS-34, PS-35, PS-36, PS-55
P9	Continuous Validation	Ongoing validation of requirements during development	Must involve **repeated validation cycles**	9	PS-3, PS-17, PS-18, PS-20, PS-26, PS-36, PS-37, PS-53, PS-55
P10	Project Management Practices	Practices used to manage project execution and coordination	Must involve **project-level control activity**	11	PS-16, PS-33, PS-39, PS-41, PS-42, PS-44, PS-48, PS-50, PS-51, PS-54, PS-55
P11	Change Control and Governance	Formal processes for managing changes	Must involve **structured governance mechanism**	9	PS-5, PS-8, PS-15, PS-25, PS-33, PS-38, PS-50, PS-51, PS-55
P12	Change Impact Analysis and Reporting	Assessing impact of requirement changes	Must involve **evaluation/analysis activity**	6	PS-8, PS-25, PS-37, PS-40, PS-47, PS-48

TABLE 10. ***(Continued.)*** **Codebook for agile RCM practices.**

P13	Agile Roles & Accountability	Definition of roles and responsibilities in agile teams	Must define **responsibility structure**	10	PS-1, PS-12, PS-25, PS-27, PS-30, PS-50, PS-51, PS-52, PS-54, PS-55
P14	Cross-Functional Collaboration via Tools	Use of tools to support distributed collaboration	Must involve **tool-mediated collaboration**	8	PS-5, PS-19, PS-22, PS-24, PS-41, PS-51, PS-54, PS-55
P15	Product Backlog Management	Managing and updating backlog items	Must involve **backlog handling activity**	7	PS-5, PS-12, PS-40, PS-45, PS-47, PS-52, PS-53
P16	Software Engineering Practices	Engineering activities like reuse and modeling	Must involve **technical implementation**	5	PS-17, PS-29, PS-37, PS-41, PS-50
P17	Synchronization and Version Control	Managing code synchronization across teams	Must involve **code/version coordination**	5	PS-27, PS-37, PS-40, PS-44, PS-51
P18	Test-Driven Development (TDD)	Writing tests before code implementation	Must involve **test-first development**	5	PS-6, PS-12, PS-34, PS-36, PS-53
P19	Change Identification	Identifying requirement changes	Must involve **recognition of change events**	5	PS-37, PS-38, PS-47, PS-54, PS-55
P20	Documentation of Changes and Rationale	Recording changes and their justification	Must involve **formal documentation activity**	5	PS-8, PS-13, PS-37, PS-50, PS-51
P21	Coding Standards / Pair Programming	Standard coding practices and collaboration in coding	Must involve **coding-level practices**	3	PS-4, PS-21, PS-53
P22	Requirements Change Management	Overall handling of requirement changes	Must represent **holistic RCM activity**	6	PS-23, PS-38, PS-50, PS-51, PS-54, PS-55
P23	Standardized Project Process	Standard procedures applied across projects	Must involve **organizational standardization**	4	PS-5, PS-41, PS-54, PS-55

and exclusion criteria were applied, some relevant studies may have been unintentionally omitted due to limi- tations in selected search terms or databases. In addition, the exclusive use of English-language studies may have led to the exclusion of data published in other languages and introduced a language bias. Another threat relates to the potential for researcher bias through categorisation practices, factor analyses, and syntheses. The procedures used for classifying practices and influences were conducted systematically; however, relying upon subjective judgment can result in minor interpretation discrepancies arising due to reader bias. To address this issue, several rounds of cross-validation were conducted within the research group. Furthermore, a limited time horizon for the included studies (2016–2026) may limit the scope of our analysis, as previous seminal works may not be included, and newer features in the rapidly changing agile and GSD fields may not be fully reflected in the literature. Finally, the diversity of GSD environments is extraordinary; hence, the outcomes of this study are not generalisable. Moreover, best practices devised for one field may not be identically applicable to other fields or domains.

VI. REPORTING BIAS CONSIDERATIONS

The study did not undertake a rigorous evaluation of reporting bias, including publication bias or selective outcome reporting. Because the synthesis primarily comprises qualitative and narrative methods and lacks equivalent quantitative effect measures, statistical methods commonly used in intervention evaluations, such as funnel plots, were inapplicable. The review was limited to peer-reviewed studies indexed in prominent academic databases, excluding grey literature such as technical reports, industrial white papers, and unpublished theses and dissertations. Consequently, there is a risk of underrepresentation of pertinent studies, especially those yielding negative, inconclusive, or practice-oriented outcomes. This limitation is acknowledged and needs to be considered when interpreting the results of this review.

VII. CONCLUSION

This study aimed to identify the agile practices and influential contextual factors associated with RCM in AGSD and to explain how these practices contribute to successful RCM outcomes. Through a two-phase SLR covering the period from 2016 to 2026 and synthesising evidence from 55 primary studies, the review addresses three research questions and advances from descriptive findings toward an explanatory theoretical contribution. With respect to RQ1, 23 agile practices were identified and grouped into five coherent practice families: communication, requirements, agile, management, and development/software engineering practices. Hence, the study provided a structured, replicable taxonomy that addresses the terminological fragmentation found in previous works. Regarding RQ3, eight significant contextual factors were identified and viewed not merely as determinative barriers to implementation success, but also as moderating factors whose influence on RCM outcomes depends on the practices and repository-based mechanisms used to manage them. With respect to RQ2, the study made its first major theoretical contribution by developing an explanatory conceptual model proposing a four-stage chain through which practices generate value. The chain begins with practice enactment, through shared repositories and artefacts representing the operational basis of distributed agile work, through six explanatory mechanisms, namely, information alignment, requirement clarification, shared understanding formation, decision structuring, adaptive coordination, and technical validation, to process-level and project-level outcomes. Most importantly, the eight contextual factors were operationalised within the model as moderators of the repository-to-mechanism relationship rather than as direct precursors of outcomes. This represents a new theoretical position in this field and aligns with contingency theory.

Therefore, the primary contribution of this study is not merely listing the practices and factors but in explaining how they interrelate. The study established the operational repository and artefact layer as an explicit evidence-based operational stage placed between practices and mechanisms. Thus, the model reveals what many earlier models did not emphasise: practices do not activate mechanisms independently, but rather through the collaborative software infrastructure (digital platforms such as backlog systems, governance repositories, code repositories, knowledge bases) that distributed agile teams utilise to communicate decision-making about changes and to coordinate and record evidence. Hence, the model provides both empirical actionability, as the variables at the repository level are observable via digital traces, and theoretical relevance, as the mechanism activation can be operationalised using digital trace data in subsequent studies.

For practitioners, the model presents a shift in perspective, from merely adopting practices towards evaluating the quality of repository infrastructure and monitoring mechanisms. For organisations managing requirements changes among stakeholders who have temporal distance, cultural differences, or complex governance structures, investment should be directed toward increasing the reliability and discipline of their shared repository infrastructure. The proposed model indicates that diminished mechanism activation under unfavourable contextual conditions is mainly a problem related to poor repository quality and operations, rather than to the adoption of proper practices.

The model establishes a structure for further empirical investigation by defining a falsifiable research agenda. Future research should test the relationships between the moderating effects of contextual factors and the repository-to-mechanism link using structural equation modelling/partial least squares, investigate the repository layer using digital trace analysis, and explore the temporal development of the moderating effects of contextual factors with longitudinal field studies. It marks a step forward from descriptively and cross-sectionally oriented, fragmented research that has characterised this domain to date toward empirically oriented, theoretically justified, and practically actionable inquiry. There are some limits to this study. The evidence base for this study includes only English-language peer-reviewed studies published from 2016 to 2026; it does not include grey literature, and it uses both narrative and frequency-based analytical techniques that cannot establish causality. As an explanatory synthesis and not a confirmed causal theory, the model's propositions will require testing in future empirical studies. These limitations, however, do not diminish the contribution. Rather, they define the boundary conditions within which the model holds and the research agenda that lies beyond this study.

APPENDIX A
DATA AVAILABILITY AND TRANSPARENCY STATEMENT

All primary studies, extracted data, and supporting materials used in this review are available at the following repository: https://drive.google.com/drive/folders/1V7Bwi_AiMC_qi AVICoz6Cg4aj8kUgGT9

APPENDIX B
PRIMARY STUDY EVALUATION

Table 8 summarizes the quality assessment of the 55 primary studies using six evaluation criteria (QA1–QA6), with aggregated quality ratings and reference traceability to the main manuscript.

APPENDIX C
CODING FRAMEWORK FOR INFLUENCING FACTORS IN AGSD RCM

This appendix presents in Table 9 the codebook for the eight contextual factors identified in the review. For each factor, it provides a definition, illustrative example, and associated primary studies (PS-codes).

APPENDIX D
CODEBOOK FOR AGILE RCM PRACTICES (AGSD CONTEXT)

This appendix presents in Table 10 the codebook for the 23 agile RCM practices, organized into five practice families. Each entry includes the practice definition and associated primary studies (PS-codes).

REFERENCES

[1] D. Damian, J. Chisan, P. Allen, and B. Corrie, "Awareness meets requirements management: Awareness needs in global software development," in *Proc. Int. Workshop Global Softw. Develop., Int. Conf. Softw. Eng.*, 2003, pp. 7–11.

[2] M. Paasivaara, "Adopting SAFe to scale agile in a globally distributed organization," in *Proc. IEEE 12th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2017, pp. 36–40.

[3] R. Hoda, N. Salleh, and J. Grundy, "The rise and evolution of agile software development," *IEEE Softw.*, vol. 35, no. 5, pp. 58–63, Sep. 2018.

[4] M. O. Ahmad, D. Dennehy, K. Conboy, and M. Oivo, "Kanban in software engineering: A systematic mapping study," *J. Syst. Softw.*, vol. 137, pp. 96–113, Mar. 2018.

[5] D. Kostin and D. Strode, "Effective communication in globally distributed scrum: A model and practical guidance," *Australas. J. Inf. Syst.*, vol. 27, p. 2023, Dec. 2023.

[6] W. Hussain, J. Buchan, and T. Clear, "Managing requirements in globally distributed COTS customization," in *Proc. IEEE Int. Conf. Global Softw. Engineeering Workshops*, Aug. 2014, pp. 33–38.

[7] T. Kamal, Q. Zhang, and M. A. Akbar, "Toward successful agile requirements change management process in global software development: A client–vendor analysis," *IET Softw.*, vol. 14, no. 3, pp. 265–274, Jun. 2020, doi: 10.1049/iet-sen.2019.0128.

[8] M. Shameem, A. A. Khan, M. G. Hasan, and M. A. Akbar, "Analytic hierarchy process based prioritisation and taxonomy of success factors for scaling agile methods in global software development," *IET Softw.*, vol. 14, no. 4, pp. 389–401, Aug. 2020.

[9] A. Windle, A. Marshall, L. de la Perrelle, S. Champion, P. D. S. Ross, G. Harvey, and C. Davy, "Factors that influence the implementation of innovation in aged care: A scoping review," *JBI Evidence Implement.*, vol. 22, no. 1, pp. 61–80, Feb. 2024.

[10] M. Adzgauskaite, C. Tam, and R. Martins, "What helps agile remote teams to be successful in developing software? Empirical evidence," *Inf. Softw. Technol.*, vol. 177, Jan. 2025, Art. no. 107593.

[11] N. R. Haddaway, M. J. Page, C. C. Pritchard, and L. A. McGuinness, "PRISMA2020 : An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and open synthesis," *Campbell Systematic Rev.*, vol. 18, no. 2, Jun. 2022.

[12] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, and B Kitchenham, "Systematic literature reviews in software engineering-a systematic literature review," *Inf. Softw. Technol.*, vol. 1, no. 1, pp. 7–15, 2009.

[13] B. Kitchenham, L. Madeyski, and D. Budgen, "SEGRESS: Software engineering guidelines for REporting secondary studies," *IEEE Trans. Softw. Eng.*, vol. 49, no. 3, pp. 1273–1298, Mar. 2023.

[14] A. Altaf, U. Fatima, W. H. Butt, M. W. Anwar, and M. Hamdani, "A systematic literature review on factors impacting agile adaptation in global software development," in *Proc. 7th Int. Conf. Comput. Commun. Manage.*, Jul. 2019, pp. 158–163.

[15] R. Camara, A. Alves, I. Monte, and M. Marinho, "Agile global software development: A systematic literature review," in *Proc. ACM Int. Conf. Ser.*, 2020, pp. 31–40.

[16] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects—A systematic literature review," in *Requirements Engineering: Foundation for Software Quality*, vol. 10153, 2017, pp. 219–234.

[17] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Inf. Softw. Technol.*, vol. 96, pp. 161–180, Apr. 2018.

[18] Z. Podari, A. F. Arbain, N. Ibrahim, D. N. A. Jawawi, W. M. N. W. Kadir, and A. M. Fahmi, "Systematic literature review on global software development risks in agile methodology," in *Proc. 8th Int. Conf. Inf. Technol. Multimedia (ICIMU)*, Aug. 2020, pp. 231–236.

[19] M. Shameem, C. Kumar, B. Chandra, and A. A. Khan, "Systematic review of success factors for scaling agile methods in global software development environment: A client-vendor perspective," in *Proc. 24th Asia–Pacific Softw. Eng. Conf. Workshops (APSECW)*, Dec. 2017, pp. 17–24.

[20] I. Z. Mustaqim, T. Raharjo, and N. W. Trisnawaty, "Toward unveiling the execution of large-scale agile project in global software development: A systematic literature review," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Oct. 2023, pp. 1–8.

[21] L. Layman, L. Williams, D. Damian, and H. Bures, "Essential communication practices for extreme programming in a global software development team," *Inf. Softw. Technol.*, vol. 48, no. 9, pp. 781–794, Sep. 2006.

[22] J. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," *IEEE Trans. Softw. Eng.*, vol. 29, no. 6, pp. 481–494, 2003.

[23] R. Hoda, J. Noble, and S. Marshall, "Agile undercover: When customers don't collaborate," *LNBIP*, vol. 48, pp. 73–87, 2010.

[24] M. Kalenda, P. Hyna, and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," *J. Softw., Evol. Process*, vol. 30, no. 10, p. 1954, Oct. 2018.

[25] 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.

[26] K. Beck. (2001). [Online]. Available: https://ai-learn.it/wp-content/uploads/2019/03/03_ManifestoofAgileSoftwareDevelopment-1.pdf

[27] R. D. S. Santos, P. Ralph, A. Arshad, and K.-J. Stol, "Distributed scrum: A case meta-analysis," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–37, Apr. 2024.

[28] 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.

[29] S. Anwer, L. Wen, Z. Wang, and S. Mahmood, "Comparative analysis of requirement change management challenges between in-house and global software development: Findings of literature and industry survey," *IEEE Access*, vol. 7, pp. 116585–116611, 2019.

[30] D. E. Damian and D. Zowghi, "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization," in *Proc. IEEE Joint Int. Conf. Requirements Eng. (ICRE)*, 2002, pp. 319–328.

[31] R. Martin. (2003). [Online]. Available: https://dl.acm.org/doi/abs/10.5555/515230

[32] A. Martini, L. Pareto, and J. Bosch, "Communication factors for speed and reuse in large-scale agile software development," in *Proc. 17th Int. Softw. Product Line Conf.*, Aug. 2013, pp. 42–51.

[33] S. Beecham, T. Clear, R. Lal, and J. Noll, "Do scaling agile frameworks address global software development risks? An empirical study," *J. Syst. Softw.*, vol. 171, Jan. 2021, Art. no. 110823.

[34] R. D. Estrada-Esponda, M. López-Benítez, G. Matturro, and J. C. Osorio-Gómez, "Selection of software agile practices using analytic hierarchy process," *Heliyon*, vol. 10, no. 1, Jan. 2024, Art. no. e22948.

[35] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Comput. Surv.*, vol. 26, no. 1, pp. 87–119, 1994.

[36] P. R. Lawrence and J. W. Lorsch, "Differentiation and integration in complex organizations," *Administ. Sci. Quart.*, vol. 12, no. 1, pp. 1–47, Jun. 1967.

[37] A. K. Al Jabri, M. K. Islam, and S. B. Daud, "Review of agile project management research through systematic literature network analysis approach," *Multidisciplinary Rev.*, vol. 7, no. 2, Dec. 2023, Art. no. 2024032.

[38] T. Natarajan and S. Pichai. (2024). *Transition From Waterfall to Agile Methodology-An Action Research Study*. Accessed: 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10488855/

[39] M. F. Abrar, M. S. Khan, S. Ali, U. Ali, M. F. Majeed, A. Ali, B. Amin, and N. Rasheed, "Motivators for large-scale agile adoption from management perspective: A systematic literature review," *IEEE Access*, vol. 7, pp. 22660–22674, 2019.

[40] A. L?pez-Alcarria, A. Olivares-Vicente, and F. Poza-Vilches, "A systematic review of the use of agile methodologies in education to foster sustainability competencies," *Sustainability*, vol. 11, no. 10, p. 2915, 2019.

[41] F. G. Rocha, S. Misra, and M. S. Soares, "Guidelines for future agile methodologies and architecture reconciliation for software-intensive systems," *Electronics*, vol. 12, no. 7, p. 1582, Mar. 2023.

[42] J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp. 120–122, 2001.

[43] D. Truong and T. Jitbaipoon, ''How can agile methodologies be used to enhance the success of information technology projects?'' *Int. J. Inf. Technol. Project Manage. (IJITPM)*, vol. 7, pp. 1–16, Apr. 2016.

[44] D. Budgen, B. Kitchenham, S. Charters, M. Turner, P. Brereton, and S. Linkman, ''Preliminary results of a study of the completeness and clarity of structured abstracts,'' in *Proc. Electron. Workshops Comput.*, 2007.

[45] B. Kitchenham, D. Budgen, P. Brereton, M. Turner, S. Charters, and S. Linkman, ''Large-scale software engineering questions—Expert opinion or empirical evidence?'' *IET Softw.*, vol. 1, no. 5, pp. 161–171, 2007.

[46] D. Badampudi, C. Wohlin, and K. Petersen, ''Experiences from using snowballing and database searches in systematic literature studies,'' in *Proc. 19th Int. Conf. Eval. Assessment Softw. Eng. Assoc. Comput. Machinery*, 2015, pp. 1–10.

[47] T. Garavan and F. O'Brien. (2024). [Online]. Available: https://www.elgaronline.com/edcollchap/book/9781035308767/ch06.xml

[48] S. Matthiesen and P. Bjørn, ''When distribution of tasks and skills are fun-damentally problematic: A failure story from global software outsourcing,'' in *Proc. ACM Hum.-Comput. Interact*, vol. 1, 2017, pp. 1–16.

[49] L. A. M. Zaina, H. Sharp, and L. Barroca, ''UX information in the daily work of an agile team: A distributed cognition analysis,'' *Int. J. Hum. Comput. Stud.*, vol. 147, Mar. 2021, Art. no. 102574.

[50] N. Rashid and S. U. Khan, ''Using agile methods for the development of green and sustainable software: Success factors for GSD vendors,'' *J. Softw., Evol. Process*, vol. 30, no. 8, 2018, Art. no. e1927.

[51] I. Nurdiani, J. Börstler, S. Fricker, K. Petersen, and P. Chatzipetrou, ''Understanding the order of agile practice introduction: Comparing agile maturity models and practitioners' experience,'' *J. Syst. Softw.*, vol. 156, pp. 1–20, Oct. 2019.

[52] M. Marinho, R. Camara, and S. Sampaio, ''Toward unveiling how SAFe framework supports agile in global software development,'' *IEEE Access*, vol. 9, pp. 109671–109692, 2021.

[53] A. Alsahli, H. Khan, and S. Alyahya, ''Toward an agile approach to managing the effect of requirements on software architecture during global software development,'' *Sci Program*, vol. 2016, no. 1, p. 9, 2016.

[54] M. Shameem, R. R. Kumar, M. Nadeem, and A. A. Khan, ''Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process,'' *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106122.

[55] R. Sinha, M. Shameem, and C. Kumar, ''SWOT: Strength, weaknesses, opportunities, and threats for scaling agile methods in global software development,'' in *Proc. 13th Innov. Softw. Eng. Conf. (Formerly Known India Softw. Eng. Conf.)*, Feb. 2020, pp. 1–10.

[56] G. D. García, C. J. Pardo Calvache, and F. J. Á. Rodríguez, ''Society 5.0 and soft skills in agile global software development,'' *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 17, no. 2, pp. 197–207, May 2022.

[57] F. Ijaz, W. Aslam, A. A. AlSanad, Z. Aslam, I. Ullah, and F. Ullah, ''A critique on task allocation processes in distributed agile software development,'' *Sci. Program.*, vol. 2022, pp. 1–19, May 2022.

[58] Y. Hafeez, S. Asghar, B. Arif, and S. Ali, ''Role of situational method engineering to improve visual information systems in agile distributed environment,'' *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 8881–8908, Mar. 2021.

[59] M. Shameem, R. Kumar, C. Kumar, B. Chandra, and A. Khan, ''Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process,'' *J. Softw., Evol. Process*, vol. 11, Nov. 2018, Art. no. e1979.

[60] M. Marinho, J. Noll, I. Richardson, and S. Beecham, ''Plan-driven approaches are alive and kicking in agile global software development,'' in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Sep. 2019, pp. 1–11.

[61] N. Rashid and S. U. Khan, ''Agile practices for global software development vendors in the development of green and sustainable software,'' *J. Softw., Evol. Process*, vol. Process, vol. 30, no. 10, p. 1964, 2018.

[62] G. Borrego, A. L. Morán, R. R. P. Cinco, O. M. Rodríguez-Elias, and E. García-Canseco, ''Review of approaches to manage architectural knowledge in agile global software development,'' *IET Softw.*, vol. 11, no. 3, pp. 77–88, Jun. 2017.

[63] H. Noor, B. Hayat, Z. Amjad, M. Hanif, S. Tabussum, R. Mansha, and K. Mubasher, ''Identifying communication issues contributing to the formation of chaotic situation: An AGSD view,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 550–557, 2021.

[64] Ö. Uludag, M. Kleehaus, C. Caprano, and F. Matthes, ''Identifying and structuring challenges in large-scale agile development based on a structured literature review,'' in *Proc. IEEE 22nd Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2018, pp. 191–197.

[65] N. Rashid, S. U. Khan, H. U. Khan, and M. Ilyas, ''Green-agile maturity model: An evaluation framework for global software development vendors,'' *IEEE Access*, vol. 9, pp. 71868–71886, 2021.

[66] M. D. Kahya and Ç. Seneler, ''Geographical distance challenges in distributed agile software development: Case study of a global company,'' in *Proc. Comput. Sci. Eng. UBMK*, Sep. 2018, pp. 78–83.

[67] M. Shameem, M. Nadeem, and A. T. Zamani, ''Genetic algorithm based probabilistic model for agile project success in global software development,'' *Appl. Soft Comput.*, vol. 135, Mar. 2023, Art. no. 109998.

[68] B. Shahzad, W. N. Awan, Fazal-E-Amin, A. Abro, M. Shoaib, and S. Alyahya, ''Framework for effective utilization of distributed scrum in software projects,'' *Comput. Syst. Sci. Eng.*, vol. 44, no. 1, pp. 407–422, 2023.

[69] M. Kausar, M. Ishtiaq, and S. Hussain, ''Distributed agile patterns-using agile practices to solve offshore development issues,'' *IEEE Access*, vol. 10, pp. 8840–8854, 2022.

[70] K. Bundhun and R. K. Sungkur, ''Developing a framework to overcome communication challenges in agile distributed teams – case study of a mauritian-based IT service delivery centre,'' *Global Transitions Proc.*, vol. 2, no. 2, pp. 315–322, Nov. 2021.

[71] C. P. Godoy, A. F. Cruz, E. P. Silva, L. M. Santos, R. S. Zerbini, and C. A. L. Pahins, ''Blueprint model: A new approach to scrum agile methodology,'' in *Proc. ACM/IEEE 14th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2019, pp. 95–99.

[72] A. Yagüe, J. Garbajosa, and J. Díaz, ''An exploratory study in communication in agile global software development,'' *Comput Stand Interface*, vol. 48, pp. 184–197, Nov. 2016.

[73] V. N. Vithana, D. Asirvatham, and M. G. M. Johar, ''An empirical study on using agile methods in global software development,'' in *Proc. 18th Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Sep. 2018, pp. 150–156.

[74] K. Batool and I. Inayat, ''An empirical investigation on requirements change management practices in Pakistani agile based industry,'' in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2019, pp. 7–12.

[75] F. Aizaz, S. U. R. Khan, J. A. Khan, and A. Akhunzada, ''An empirical investigation of factors causing scope creep in agile global software development context: A conceptual model for project managers,'' *IEEE Access*, vol. 9, pp. 109166–109195, 2021.

[76] J. Mihajlović-Milićević, M. Radenković, A. Labus, D. Stojanović, and Z. Bogdanović, ''An approach to agile management of virtual student teams in smart environment development,'' *Interact. Learn. Environ.*, vol. 32, no. 3, pp. 921–941, Mar. 2024.

[77] P. Chilito, D. Viveros, C. Pardo, and F. J. Pino, ''Scrum+: An agile guide for the global software development (GSD) multi-model project management,'' in *Proc. IEEE Colomb. Conf. Commun. Comput. COLCOM*, May 2018, pp. 1–6.

[78] K. B. Awar, M. S. I. Sameem, and Y. Hafeez, ''A model for applying agile practices in distributed environment: A case of local software industry,'' in *Proc. Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, Mar. 2017, pp. 228–232.

[79] A. A. Khan, M. Shameem, M. Nadeem, and M. A. Akbar, ''Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping,'' *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107090.

[80] R. Camara and M. Marinho, ''Agile tailoring in distributed large-scale environments using agile frameworks: A systematic literature review,'' *CLEI Electron. J.*, vol. 27, no. 1, p. 8, May 2024.

[81] J. Noll, A. Razzak, I. Richardson, and S. Beecham, ''Agile practices for the global teaming model,'' in *Proc. IEEE 11th Int. Conf. Global Softw. Eng. Workshops (ICGSEW)*, Aug. 2016, pp. 13–18.

[82] Z. Shehzadi, F. Azam, M. W. Anwar, and I. Qasim, ''A novel framework for change requirement management (CRM),'' in *Proc. 9th Int. Conf. Inf. Commun. Manage- ment. Assoc. Comput. Machinery*, vol. 2019, pp. 22–26.

[83] R. Nadeem, R. M. A. Latif, K. Hussain, N. Z. Jhanjhi, and M. Humayun, ''A flexible framework for requirement management (FFRM) from software architecture toward distributed agile framework,'' *Open Comput. Sci.*, vol. 12, no. 1, pp. 364–377, Nov. 2022.

[84] H. Lamsellak and M. Ghaouth Belkasmi, ''Global software development agile planning model: Challenges and current trends,'' *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 1774–1784, Dec. 2023.

[85] A. Aljuhani, "Identification of agile requirements change management success factors in global software development based on the best-worst method," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 7, 2024.
[86] B. Aljohani, A. Aljuhani, and T. Alsanoosy, "Enhancing agile requirements change management: Integrating LLMs with fuzzy best-worst method for decision support," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 3, 2025.
[87] S. Ohri and I. B. Ohri, "Stakeholder hierarchy and Conway's law: Impact on backlog management strategies in agile teams post-COVID," in *Proc. IEEE 5th Int. Conf. ICT Bus. Ind. Government (ICTBIG)*, Dec. 2025, pp. 1–7.
[88] K. Model, C. Mombrey, and G. Herzwurm, "Paving the way to a software-supported requirements prioritization in distributed scrum projects," in *Proc. IEEE/ACM Int. Workshop Softw.-Intensive Bus. (IWSiB)*, May 2022, pp. 51–58.
[89] A.-H. Al-Herani, D. Zakarneh, and A. Qusef, "Navigating challenges and opportunities in project management for global software development," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Apr. 2025, pp. 32–38.
[90] M. A. Akbar, A. A. Khan, S. Mahmood, and S. Rafi, "A vision of DevOps requirements change management standardization," in *Proc. IEEE 22nd Int. Conf. Softw. Quality, Rel., Secur. Companion (QRS-C)*, Dec. 2022, pp. 587–592.

NEMER WAHBEH received the bachelor's degree in software engineering from Philadelphia University, in 2003, and the master's degree in software engineering from the University of Jordan, in 2009. He is currently pursuing the Ph.D. degree in software engineering with Universiti Putra Malaysia (UPM).

He is also the Project Manager of Jordan Education Association. He also works as a part-time Researcher with Amman Arab University. His doctoral research focuses on requirements change management in global agile software development, emphasizing trust and collaboration among distributed teams. He has authored six peer-reviewed publications and secured research funding for his work. He participates in software engineering research groups and serves as an united nations arbitrator specializing in digital transformation for smart and e-governments.

NOVIA ADMODISASTRO received the master's degree in software engineering from the University of Malaya and the Ph.D. degree in computing from Lancaster University, U.K. She is currently an Associate Professor with the Department of Software Engineering and Information Systems, Universiti Putra Malaysia (UPM). With over 16 years of academic experience, she has supervised and co-supervised more than 45 postgraduate students and published over 100 journals and conference papers in software engineering, cloud engineering, human–computer interaction, and cross-disciplinary studies in education and medical technologies. Her works have received multiple best paper and presentation awards, including at iCiCSE2018, AilC2019, ICOSICA2020, and AilC2024. She actively collaborates with ministries and industries in research, training, and consultation; and has secured over RM1 million in research funding as a principal investigator and a co-researcher. She is actively involved in research innovation, competitions, and student mentoring, through which she has received numerous awards. As the Founder of Hive Digital Education, a university startup, she is passionate about advancing sustainable and inclusive digital education for diverse learning communities. She also serves as a curriculum accreditation panel member for a German body under European Qualifications Framework (EQF) and works as an external academic advisor for several Malaysian public and private universities. In addition, she is a Committee Member of t_2h_6e Malaysian Standards (NSC 07/TC 11) and a registered Professional Technologist with Malaysia Board of Technologists (MBOT).

SAADAH HASSAN received the Ph.D. degree from the University of Ulster. She is currently an Associate Professor with the Department of Software Engineering and Information Systems, Universiti Putra Malaysia (UPM). Her expertise lies in requirements engineering and software engineering. She has an extensive publication record, reflecting her active engagement in research and contributions to the academic community.

SALFARINA ABDULLAH works as a Senior Lecturer with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). Her research interests include knowledge management systems, e-service quality, and crisis informatics. She has authored several publications addressing these topics, contributing to advancements in her field.

AHMAD BAABAAD received the Ph.D. degree in software engineering from Universiti Putra Malaysia (UPM). His research interests include software architecture, particularly architectural erosion and degradation in open-source software. He has published multiple articles exploring metrics and empirical analyses related to architectural decay, providing valuable insights into maintaining software architecture integrity.

NORAZLINA KHAMIS received the M.Sc. degree in real-time software engineering and the Ph.D. degree in computer science from Universiti Malaysia Sabah (UMS). She is currently a Senior Lecturer with the Faculty of Computing and Informatics, UMS. Her research interests include software engineering, object-oriented programming, and educational assessment. She has contributed to various research projects, including IoMT-based air quality monitoring and sentiment analysis of noisy Malay text. Previously, she worked as an Associate Professor with the University of Malaya, demonstrating her extensive experience in academia and research.