Identification of Agile Requirements Change Management Success Factors in Global Software Development Based on the Best-Worst Method

Abdulmajeed Aljuhani
College of Computer Science and Engineering, Taibah University, Medina 41411, Saudi Arabia

***Abstract*—To create products that are both cost effective and high quality, a majority of software development companies are following the principles of global software development, or GSD. One of the most significant and challenging stages of the agile software development process is requirements change management (RCM); however, the execution of agile software development activities is hindered by the geographical distance between the GSD teams, especially when it comes to agile requirements change management (ARCM). The literature claims that, in a particular context, ARCM can profit from applying Multi-Criteria Decision-Making (MCDM) techniques. Within the area of ARCM, an optimal framework can be offered constitutionally, thus presenting an effective decision-making process that ought to encourage higher consumer satisfaction with software projects created in such a way. A methodology for applying the MCDM method in the ARCM context is presented in this paper. In particular, we propose a model for investigating the prioritization of ARCM success factors in the GSD context based on a decision-making method; namely, the Best-Worst Method (BWM). The BWM's ability to solve intricate decision-making problems with multiple criteria and alternatives is demonstrated by the proposed model's findings.**

Keywords*—*Best-Worst Method (BWM); Agile Requirements Change Management (ARCM); success factors; Global Software Development (GSD)

I. Introduction

Software development has recently become more globalized, with teams working across borders and cultures to build sophisticated software systems. As software development projects expand in size and complexity, project success is increasingly dependent on managing requirements changes during the development process. The process of recognizing, evaluating, and properly handling changes to requirements during the life cycle of software development is known as requirements change management (RCM). These changes may result from a variety of triggers, such as changing stakeholder requirements, developing technology, changing business demands, and legal changes. Effective requirements change management is essential to the software development process, as it may significantly influence a project's budget, schedule, and quality [1]. In particular, RCM in the context of global software development (GSD) is a crucial aspect of the software development process, which may assist in guaranteeing that the project's budget, schedule, and factors hindering quality are minimized while efficiently addressing the changing demands and expectations of stakeholders [2].

The RCM process has become even more difficult as a result of global software development. Coordinating and communicating effectively becomes essential for successful RCM when development teams are dispersed across several nations, cultures, and time zones. In addition, time zone variations, technological constraints, and language and cultural limitations may all make the RCM process more difficult and raise the possibility of mistakes or incorrect interpretation of requirements. The obstacles associated with RCM in GSD emphasize the need for creative and inclusive frameworks in RCM that make use of a wider variety of technologies and encompass all necessary RCM activities. To properly handle the dynamic and constantly changing complexities that are inherent to GSD, these frameworks should aim to bridge current gaps, especially in the integration of technology and the breadth of activities.

The traditional software development process has undergone important changes as a result of the growing trend of global software development. The emphasis on software development has shifted to remote and heterogeneous environments, such as the agile methodology, which has created significant difficulties for agile development processes when trying to handle needs changes. It becomes imperative to implement agile requirements change management (ARCM) in order to reduce risks and adapt to evolving customer needs. The discovery, evaluation, assessment, and execution of proposed requirement changes are made easier through ARCM. Agile development has grown in popularity, being a method that may be more flexible and adaptable to changing customer requirements, which works especially well in GSD contexts. Notwithstanding agile development, requirements change management persists as a multifaceted and demanding activity. Thus, the implementation of an efficient and well-organized process that can adapt quickly to changing customer requirements is essential for the successful delivery of products in the ARCM context.

Within the agile software development process, emphasis should be placed on practitioner activities, complex documentation development, development tools and processes, contract negotiation and collaboration with customers, and various changes when following a certain plan. Nevertheless, in an agile development environment where the global software development paradigm is being used, change management is a challenging task [3]. According to the literature, implementing global software development has increased the complexity of the change management process, which is already a challenging process in single-site environments.

Thus, an appropriate requirements change management process is significant to ensure successful software development activities. In order to handle the change in requirements, RCM is a collaborative process that requires coordination and communication between software developers and customers. However, ineffective RCM activity execution can result in excessive project costs, unstable requirements, and poor quality. Moreover, RCM typically receives minimal attention, contributing to the poor success rate of GSD initiatives. The change management process is made more difficult by the fact that RCM is a collaborative process, and GSD practitioners operate across regional borders, thus creating a communication and coordination gap.

The purpose of this article is to prioritize the factors that affect the ARCM in GSD projects through examining the integration of an MCDM—namely, the BWM—into ARCM activity in the field of global software development. The BWM framework is composed of a number of ARCM success factors that operate as alternatives, which are compared to one another based on a number of criteria that impact the software project. These criteria are used to evaluate each ARCM success factor, and the final prioritization is determined by adding together the weights of all the factors.

The structure of the remainder of this paper is as follows: The works related to this study are described in Section II. The BWM approach is introduced in Section III. The proposed ARCM success factors and criteria are explained in Section IV. The BWM structure used to prioritize the ARCM factors in GSD is presented in Section V. The findings and discussion are given in Section VI. Finally, Section VII provides a conclusion and recommendations for potential further research.

II. Related Work

A model called the Requirements Change Management Readiness Model (RCMRM) has been proposed by Akbar et al. [4], in order to evaluate how prepared GSD enterprises are to handle requirements changes over the course of the software development life cycle. Critical obstacles in the RCM process for GSD include language and cultural hurdles, the absence of in-person meetings, time zone variations, and delays in responses from abroad sites. According to the authors, low-quality software products may result from improper requirements change management. The RCM procedure may not be effectively implemented in GSD due to the physical distance between teams and infrequent information sharing. Through two rounds of case studies, the RCMRM was validated, and the authors believe that an organization of any size may handle its RCM process in the GSD context using the RCMRM, based on the findings of the case study.

An exploratory study on quality requirements change management in the context of software development and maintenance was presented by Ahmad et al. [5]. In addition to outlining the difficulties and success factors that software suppliers, sellers, merchants, and retailers encounter when trying to manage software quality requirements, the research also emphasizes the significance of quality change management in software development and maintenance. The study stated 14 primary obstacles to software quality requirement change management, including inadequate requirements, poor project management, miscellaneous cultural issues, poor communication and coordination, lack of technology, and lack of understanding of requirements. The authors find that managing software quality requirements changes efficiently is essential in software development and maintenance, and that ineffective management of these changes can lead to subpar software or even project failure.

A model-driven strategy was proposed by Gull et al. [6] to solve the problem of incompatible requirements in international software development. Models are used in a model-driven approach to explain the requirements, design, and implementation of the system. In addition, to make system requirements more comprehensible and straightforward for developers from different cultures and backgrounds, the models offer a formal and accurate representation. Blockchain technology is used by the authors to handle consistency problems that arise during development. Thus, handling inconsistent requirements in GSD may be resolved with the help of the suggested blockchain-oriented model-driven framework. Moreover, the framework offers a systematic approach for gathering requirements and managing them, which can aid in minimizing the cultural and communication barriers that arise throughout the development process.

A framework for semantic-based component requirements management in GSD environments, spanning from the mapping and linking of requirements to specification, was presented by Ali et al. [7]. The suggested approach seeks to address problems with conflicting stakeholder perspectives and difficulties in collaboration throughout the software development life cycle. Aspect-based sentiment analysis is employed in the suggested framework to perform a semantic analysis of the requirements from the various viewpoints of the stakeholders. This lessens the ambiguity and incompleteness of requirements. However, in order to confirm and validate the requirements, decision tree-based categorization has been applied to traceability requirements. In order to ensure accurate requirements management, the framework prioritizes missing requirements depending on historical information.

Koulecar and Ghimire [8] proposed a comprehensive and robust model for managing changing requirements within the GSD paradigm. The model incorporates unique stages and expands upon current requirements change management frameworks and models found in the literature. The authors stated that, due to the extra complexity imposed by GSD projects, traditional techniques for requirements change management may not be suitable. Changing requirements is a common difficulty in the agile software development context. The authors suggested the presented ARCM model as a solution to this problem, as it can be easily adapted to various GSD environments and used in conjunction with agile software development techniques.

The communication and coordination challenges that arise during RCM in GSD have been investigated by Qureshi et al. [9]. The authors proposed a conceptual model that delineates the variety of communication and coordination obstacles that arise during RCM in GSD, along with the various factors that impact them. They also illustrated several approaches and tools that may be applied to overcome the obstacles and enhance collaboration and communication. The four main categories of obstacles that occur during RCM in GSD—namely, communication, coordination, cultural, and technological challenges—are highlighted in the suggested conceptual model.

An improved AZ-Model for RCM in the GSD environment was presented by Mughal et al. [10]. The improved model is intended to address issues including inadequate traceability and monitoring of RCM activities, as well as a lack of collaboration and communication among project stakeholders. It aids in the creation of a high-quality product while accomplishing corporate goals and customer satisfaction. The empirical and simulation findings show that the required changes are efficiently and successfully managed by the enhanced AZ-Model, in accordance with the time and cost limitations associated to GSD.

In order to evaluate and enhance the RCM process maturity level of GSD enterprises, Akbar et al. [11] have suggested a new RCM maturity model called the software requirement change management and implementation maturity model (SR-CMIMM). The model is composed of five maturity levels, each of which has a number of critical success elements and challenges to overcome in order to reach the goal. Additionally, the model offers best practices to help GSD organizations to improve their RCM process and reach the appropriate degree of maturity. Furthermore, the study's empirical findings demonstrated that the model is effective for professionals in the industry and offers them insightful knowledge that will help them with decision making and managing software development projects.

Kausar et al. [12] presented a valuable contribution to the field of RCM and GSD through conducting a systematic literature review that provides a detailed analysis of various primary research relevant to RCM problems in GSD. The highlighted obstacles can assist individuals and organizations in overcoming these concerns, enhancing the efficiency and effectiveness of the RCM process in GSD.

Michalski and Zaleski [13] presented a comprehensive framework to assess the factors that lead to the success of IT service projects. They determined a number of factors, such as organizational and people management, project management procedures, quality of work environment, and stakeholder and risk management, that affect project performance. The findings offer valuable insights to IT professionals and project managers who are involved in managing IT service projects.

Through the use of online surveys, literature studies, and expert perspectives, Akbar et al. [14] performed an empirical investigation that examines the difficulties associated with change management activities in the GSD context. The study's objective was to give researchers and practitioners a knowledge foundation that will be useful in the development of an RCM maturity model, thus facilitating the assessment and improvement of change management techniques in GSD contexts. The study's findings indicated that the RCM process in GSD contexts may be adversely affected by 31 challenging factors. These factors include factors related to communication, coordination, culture, and time zones.

In their model, Tam et al. [15] identified five people-factors that affect the performance of ongoing agile software development projects. Among these, "team capability" and "customer involvement" are the key factors that make ongoing agile software development projects successful. Their main objective was to investigate and evaluate the factors that influence the performance of ongoing agile software development projects, and the goal of the study was to determine the key success factors (CSFs) that apply to agile projects and how personal characteristics, social norms, team capability, and customer participation affect these factors. In addition, a combination of qualitative and quantitative research methodologies was used in this study.

Albuquerque et al. [16] examined several agile approaches used in the process to offer insight into ARCM. This systematic mapping study contributes significantly to the area of ARCM through identifying essential elements, noteworthy obstacles, and research gaps in ARCM. In order to mitigate risks and guarantee the success of agile projects, stakeholders engaged in the ARCM process can benefit greatly from the analysis provided by the authors.

Kamal et al. [17] proposed a model that identifies the critical success factors for the ARCM process in GSD environments. Through efficient management of change requirements, the researchers intended to assist businesses in delivering software development projects successfully in a globalized environment. The findings of the study showed that, in the context of GSD, efficient communication and requirement traceability are the two most important success factors for the ARCM process.

Javed et al. [18] emphasized the crucial socio-cultural distance problems in GSD and proposed effective mitigation techniques to deal with these obstacles. Preventing miscommunications and guaranteeing the success of GSD initiatives may be achieved through recognizing the socio-cultural variations among team members, communicating effectively, building trust, and being culturally aware. Prioritizing the importance of several mitigation strategies is performed using the Analytical Hierarchy Process (AHP) method, in which the mitigation strategies are viewed as factors and the corresponding techniques as sub-factors. Using pairwise comparisons, the mitigation strategies are contrasted. Then, the practitioners can apply the most relevant and suitable strategy for socio-cultural distance challenges based on rank order. Furthermore, Akbar et al. [19] investigated the adoption of AHP in prioritizing RCM challenges in the GSD context. Four primary categories—organizational management, team, technology, and process—were used in the study to map out and identify 25 challenging factors.

Kamal et al. [20] identified the success factors of ARCM and prioritized them for successful implementation in GSD projects. In order to rank the identified success factors, the authors employed a mixed-method strategy that incorporates questionnaire surveys, case studies, interviews, action research, grounded theory, and the AHP. Moreover, 21 ARCM success factors were found in the study, which were divided into six categories—process, people, project, technology, quality, and communication—the most important of which being process. Using the AHP, the authors determined the following five success factors as the top five: dynamic decision-making, management and leadership support, ongoing coordination and communication, comprehensively managed changes, and stakeholder collaboration.

Akbar et al. [21] investigated the success factors, challenges, and practices of adopting DevOps techniques. The goal of the study was to identify and rank these factors in order to suggest practical and efficient methods for enterprises to implement DevOps successfully. This study included a systematic literature review (SLR) as part of its research approach, along with a quantitative analysis of the factors found using the Fuzzy Analytic Hierarchy Process (FAHP). The investigation outlined 25 practices, 17 obstacles, and 23 crucial success factors for the implementation of DevOps. It was discovered that factors such as "automation" and "team collaboration" are essential for a successful DevOps deployment. Similar to this, challenges like "lack of skillset" and "cultural resistance to change" were noted as major obstacles that might prevent the implementation of DevOps.

III. The Best-Worst Method

The Best-Worst Method, also known as the BWM, was introduced by Rezaei [22] as a decision-making approach that distinguishes the best (generally positive or significant) or worst (least significant) criteria. In contrast to techniques such as the AHP and Analytic Network Process (ANP), the BWM has an easier-to-use fundamental scale, fewer comparisons, and steadier judgments. As a result, it has gained the trust of researchers in various disciplines and is widely recognized as a reliable and attractive approach. The BWM helps decision-makers to determine the weights for criteria through making pairwise comparisons based on each of the two criteria (best and worst) and other criteria. Afterward, a minimax problem is solved to establish the criteria weights. Although prioritization in BWM has been shown to be sensible, it can be improved to account for the uncertainty of decision-makers. Two vectors of comparison, the best-to-other criteria and other criteria-to-worst, are equally important in BWM, and the decision-maker's confidence in the best-to-others and others-to-worst judgments is treated as equally important. Furthermore, the BWM assumes that decision-makers must be completely convinced of the best and worst criteria, along with the corresponding pairwise comparisons. To obtain their judgments, decision-makers utilize the AHP fundamental scale introduced by Saaty [23], as shown in Table I. As a result, the BWM is an efficient and trustworthy technique that can aid decision-makers in making better decisions through identifying the most critical criteria.

TABLE I. Fundamental Scale [23]

Value	Level of Importance
1	Equal importance
2	Weak or slight
3	Moderate importance
4	Moderate plus
5	Strong importance
6	Strong plus
7	Very strong
8	Very, very strong
9	Extreme importance

Pairwise comparisons are employed in the BWM in a manner similar to that in the AHP and ANP; however, the BWM has recently gained in popularity as it is a more successful approach in some aspects. In comparison to the AHP, the BWM necessitates less pairwise comparisons. Moreover, decision-makers find the BWM to be less complicated when comparing pairs, as they simply need to complete the up part of the pairwise comparison and do not need to use the 1–9 scale's reciprocal, which makes measurements simpler.

The use of the BWM in software development has been studied by a number of researchers. For example, Aljuhani [24] investigated the adoption of the BWM in order to select the appropriate software requirements elicitation technique. In addition, the authors in [25] employed the BWM in the context of cloud computing environments in order to rank several service providers, resources, and tasks. To prioritize many activities and manage resource allocation in cloud computing, Alhubaishy and Aljuhani [26] studied the use of the BWM. Furthermore, Aljuhani and Alhubaishy [27] adopted the BWM in the development of Mobile-D in order to identify nine insertion places where its implementation might help to resolve divergent viewpoints within the team.

A. Steps of BWM

As stated by Rezaei [22], there are five primary steps in the BWM, which are as follows:

Step 1. The first step of the BWM involves specifying the decision criteria $\{c_1, c_2, ..., c_n\}$ for the proposed solutions or alternatives.

Step 2. The second step of the BWM involves the decision-makers specifying the best and worst criteria without making any comparisons. In this step, the decision-makers are required to identify the most significant (best) and least significant (worst) criteria.

Step 3. The third step of the BWM involves making pairwise comparisons for the other criteria with respect to the best criterion. In this step, a series of judgments are made by the decision-makers based on the proposed fundamental scale shown in Table I. The outcome vector $A_B = (a_{B1}, a_{B2}, ..., a_{Bn})$ is determined, where a_{Bj} reflects the comparison of criterion j concerning the best criterion B.

Step 4. This step involves making pairwise comparisons of the other criteria in relation to the worst criterion. Similar to the third step, a series of judgments are made by the decision-makers in this step, based on the proposed fundamental scale shown in Table I. The outcome vector $A_W = (a_{1W}, a_{2W}, ..., a_{nW})$ is determined, where a_{1W} reflects the comparison of criterion j concerning the worst criterion W. The worst criterion serves as the reference point, and the decision-makers need to compare the other criteria with it.

Step 5. The fifth and final step of the BWM involves determining the optimal weights for the criteria. In this step, the optimal weights $w*_1, w*_2, ..., w*_n$ are determined based on the criteria. These weights must satisfy the constraints $w_B/w_j = a_{Bj}$ and $w_j/w_w = a_{jw}$ for each pair w_B/w_j and w_j/w_w, where w_B is the weight of the best criterion, w_j is the weight of criterion j, and w_w is the weight of the worst criterion [22].

The optimal weights are obtained by solving a minimax problem, where the maximum absolute differences between $\left|\frac{w_B}{w_j} - a_{Bj}\right|$ and $\left|\frac{w_j}{w_w} - a_{jw}\right|$ should be reduced in order to meet these conditions for every criterion.

This gives rise to the following problem:

$$\min \max_j \left\{ \left|\frac{w_B}{w_j} - a_{Bj}\right|, \left|\frac{w_j}{w_w} - a_{jw}\right| \right\}$$
$$\text{s.t.}$$
$$\sum_j w_j = 1$$
$$w_j \geq 0, \text{ for all j} \tag{1}$$

Problem 1 can be transformed to the following problem as a result:

$$\min \xi$$
$$\text{s.t.}$$
$$\left|\frac{w_B}{w_j} - a_{Bj}\right| \leq \xi, \text{ for all j}$$
$$\left|\frac{w_j}{w_w} - a_{jw}\right| \leq \xi, \text{ for all j}$$
$$\sum_j w_j = 1$$
$$w_j \geq 0, \text{ for all j} \tag{2}$$

Solving problem 2, we derive the ideal weights and ξ^*.

Additionally, the following problem is solved to determine the consistency ratio:

$$\text{Consistency Ratio} = \frac{\xi^*}{\text{Consistency Index}}$$

As stated in [22], the consistency index is contingent upon the number of criteria incorporated in the decision-making problem. However, as the comparisons would be deemed inconsistent otherwise, the consistency ratio value should be less than 0.10. The BWM steps are diagrammatically presented in Fig. 1.

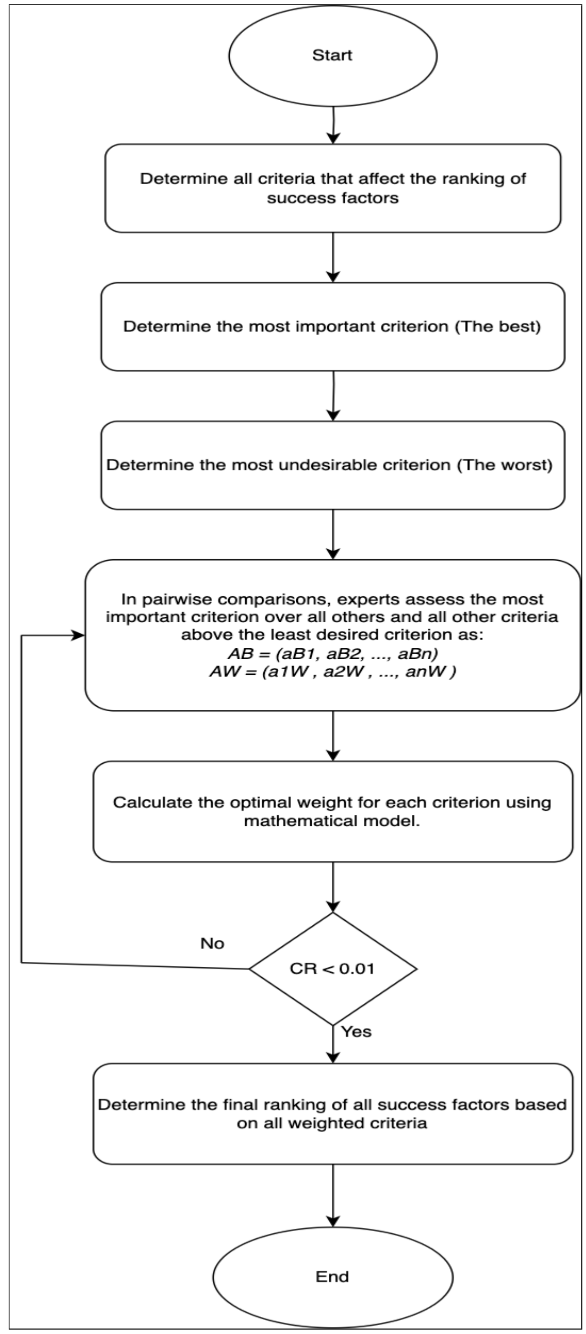

Fig. 1. BWM steps to rank ARCM success factors.

IV. Proposed Criteria for Ranking Success Factors

Determining the efficacy and efficiency of a decision-making process requires careful consideration of the determination criteria that should be specified for the proposed alternatives or solutions. The criteria must be relevant to the problem at hand and should significantly affect the suggested solutions. For the purpose of choosing the best set of criteria to fit the proposed decision problem, the decision-makers must be clear about their aims and objectives. As it is the cornerstone of the whole decision-making process, the effective identification of the appropriate decision criteria is therefore crucial. Thus, this study utilizes six criteria—derived from [19], [17], and [20]—to assist decision-makers in ranking the success factors. The studied criteria are as follows:

- Integration (C1)
- Communication (C2)
- Project Administration (C3)
- Human Resources (C4)
- Technology Factors (C5)
- Time (C6)

V. BWM Structure for Ranking ARCM Success Factors

The BWM framework for ranking the success factors (SF) has three distinct levels, in the same manner as the ANP and

AHP. The first level outlines the purpose of using the BWM which, in this case, is ranking the success factors. The selection criteria, which are explained in the preceding part, are covered in the second level. The third stage consists of the alternatives, which are the different success factors (SF) that are compared to determine the overall ranking and weights of all SFs based on various criteria. According to the literature, there are several SFs that can affect the RCM process in GSD; nevertheless, in this article, nine SFs that have an impact on the RCM process are chosen and evaluated in the BWM model [19], [17], as follows:

- Allocation resources at GSD sites (SF1)
- Requirements traceability (SF2)
- Communication, coordination, and control (SF3)
- Geographical distributed change control block (SF4)
- Effective share of information (SF5)
- Skilled human resources (SF6)
- RCM process awareness (SF7)
- Roles and responsibilities (SF8)
- Guarantee a quick response between geographically dispersed GSD teams (SF9)

The BWM structure for ranking ARCM success factors is visually represented in Fig. 2.

A. BWM Model Evaluation Based on Experts' Opinions

The aim of this study is to investigate how the BWM can be used to prioritize the ARCM success criteria in the context of GSD. The case study methodology is used to address two research questions: 1) How can the BWM be useful in ranking the ARCM success factors within the GSD domain, and 2) how does the adoption of the BWM affect the communication and productivity of team members during the development process? These questions provide the basis for the proposed units of analysis in this study. In addition to the BWM expert judgments in ranking the ARCM success factors, two units of analysis that are suitable for application are prioritizing and evaluating. To emphasize the capabilities and advantages of the BWM, criteria that influence the prioritization of the ARCM success factors were identified as a first step in the assessment process. The data-gathering technique was a questionnaire issued to domain experts, who served as the data source. Moreover, to assign a weight to each criterion in the model, the experts were asked to assess the proposed criteria. As indicated in Table II, the experts employed the BWM procedures to identify the best criterion and then performed a pairwise comparison to evaluate the criterion's weight relative to all other criteria. Table II presents a pairwise comparison wherein the C4 criterion is 8, 2, 3, 3, and 4 times more significant than the corresponding C1, C2, C3, C5, and C6 criteria, respectively.

After that, a comparison among the selected criteria is made with respect to the worst criterion, which is the integration (C1) criterion in this case. As shown in Table III, five criteria were given preference over C1; for instance, C4 had an extreme significance over the C1 criterion.

TABLE II. Pairwise Comparison of Human Resources (C4) Criterion with Respect to other Criteria

Best to Others	C1	C2	C3	C4	C5	C6
C4	8	2	3	1	3	4

TABLE III. Pairwise Comparison of Criteria with Respect to C1 Criterion

Others to the Worst	C1
C2	6
C3	5
C4	8
C5	4
C6	5

VI. Results and Discussion

The judgments on the adoption of the BWM were computed using the Solver Linear BWM. According to the combined outcome derived from domain experts, C4 was deemed the most significant attribute when it came to determining the order of importance for the success factors. Meanwhile, C1 was shown to be the least significant criterion. Among the proposed criteria, C2 was ranked second, followed by C3, C5, and C6, respectively. Table IV displays the aggregate weights of all the criteria. Moreover, the consistency ratio of the criteria aggregated weights was 0.071, which is less than 0.01, indicating that the result of this judgment was consistent (as stated previously in the BWM steps).

TABLE IV. The Aggregate Weights of All the Criteria

Ranking	Criteria	Weights (%)
1	C4	35.71%
2	C2	21.42%
3	C5	14.29%
4	C3	14.27%
5	C6	10.71%
6	C1	3.57%

Furthermore, based on the BWM, SF3 was evaluated as the most important alternative. The findings also exhibit that SF6 was ranked in the second position, followed by SF5. Meanwhile, SF4 was ranked as the least significant factor. Moreover, SF2 was ranked in the fourth position, followed by SF8 and SF1, respectively. Furthermore, SF7 was ranked in the seventh position, followed by SF9. Table V illustrates the final weights for each success factor.

TABLE V. The Importance of ARCM Success Factors

Ranking	Criteria	Weights (%)
1	SF3	26.66%
2	SF6	15.39%
3	SF5	15.31%
4	SF2	10.34%
5	SF8	10.12%
6	SF1	7.67%
7	SF7	6.14%
8	SF9	5.11%
9	SF4	3.23%

Several benefits were addressed by the domain experts with respect to the presented framework. The development team found it easier to tackle complicated and unstructured problems due to the BWM's power. Furthermore, each member

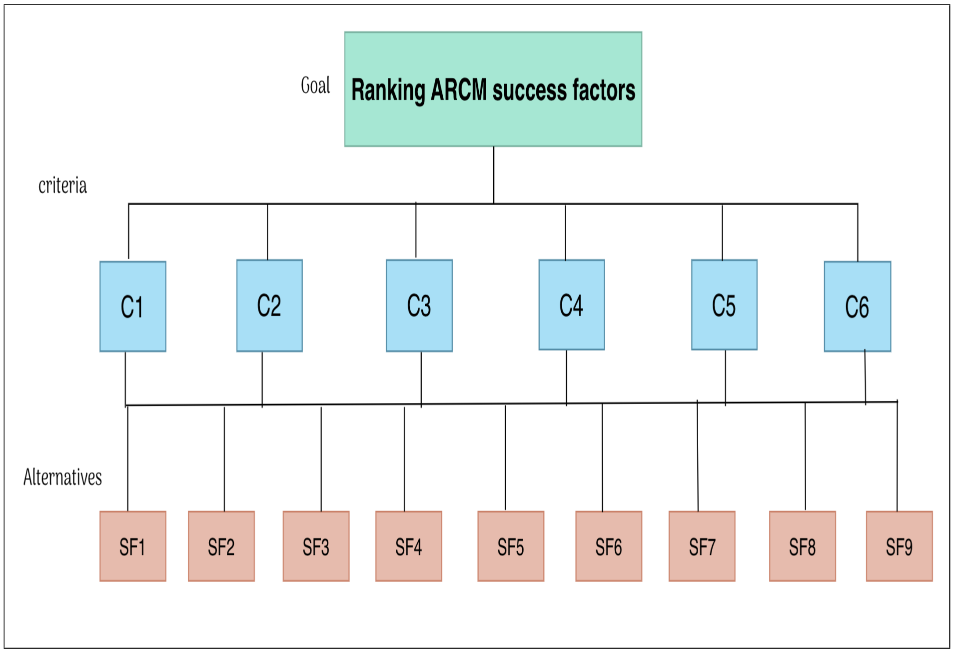

Fig. 2. BWM Structure for ranking ARCM success factors.

was able to contribute to the decision-making process through drawing on their individual experiences, according to the way in which the technique is structured. This guarantees a high degree of contentment among the development teams, which may show in the quality of the project. Considering a number of factors that influence the decision-making process, the BWM facilitates decision-making. Furthermore, the BWM helps managers or team members grasp the most important variables and criteria to take into consideration while prioritizing the success factors. For each paired comparison, the BWM yielded extremely consistent results for the consistency ratio value. The consistency ratio in this study was 0.071 for the criteria overall weights and 0.040 for the success factor overall weights, both below the maximum acceptable consistency ratio of 0.10. As indicated by Tables IV and V, these results validate the feasibility of the framework through demonstrating how the BWM may be utilized to prioritize the ARCM success factors. For spontaneous decision crises not addressed by an existing model, the BWM can be used. It should be noted that this includes the expense of incorporating the BWM into the ARCM success factors.

VII. Conclusion and Future Work

The growing tendency toward GSD prompted us to investigate the factors that may have a beneficial effect on the activities in the ARCM process. Six critical criteria and nine success factors were identified in this study, considering their effects on the success of ARCM activities within the context of GSD. Further, in order to integrate the agile development process within the framework of GSD and execute RCM activities, the BWM method was adopted to rank the investigated factors according to their significance. The execution of agile software development activities is hindered by the spread of GSD teams, especially when it comes to requirements change management. The avoidance of these issues may be achieved by giving priority to the success factors. The findings showed that, at the criteria level, C4 (Human resources) was ranked as the most significant criterion (weight = 35.71%). Moreover, SF3 (Communication, coordination, and control), SF6 (Skilled human resources), SF5 (Effective sharing of information), and SF2 (Requirements traceability) were deemed to be the most important ARCM process success factors within the context of GSD. Industry practitioners may benefit from this study's results through adoption of the high-priority success elements for the effective execution of ARCM activities in the context of GSD.

Furthermore, the results of the study demonstrated the effectiveness of the BWM in resolving complex problems in less time than comparable methods such as the AHP and ANP. The AHP requires $n(n-1)/2$ comparisons (where n is the number of variables in the model), while the introduced technique requires only $2n-3$ comparisons. Another benefit of using the BWM in this study is that the structure and flexibility of ARCM success factor prioritization were enhanced through the adoption of a defined decision-making method.

To improve the accuracy of its outputs in the future, the BWM can be combined with other methods; for example, it may be used with a fuzzy set to improve the ways in which subjective judgments and item roughness are handled when assessing the model's elements. Another potential work in the future would be to develop an automated BWM tool that complies with the prioritization of ARCM success factors and its standards.

References

[1] M. A. Akbar, A. A. Khan, S. Mahmood, and S. Rafi, "A vision of devops requirements change management standardization," in *2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*. IEEE, 2022, pp. 587–592.

[2] F. Aizaz, S. U. R. Khan, J. A. Khan, A. Akhunzada *et al.*, "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. 109 166–109 195, 2021.

[3] R. Jain and U. Suman, "Effectiveness of agile practices in global software development," *International Journal of Grid and Distributed Computing*, vol. 9, no. 10, pp. 231–248, 2016.

[4] M. A. Akbar, S. Mahmood, Z. Huang, A. A. Khan, and M. Shameem, "Readiness model for requirements change management in global software development," *Journal of Software: Evolution and Process*, vol. 32, no. 10, p. e2264, 2020.

[5] J. Ahmad, T. M. Ghazal, A. W. Khan, M. A. Khan, M. Inairat, N. Sahawneh, and F. Khan, "Quality requirement change management's challenges: an exploratory study using slr," *IEEE Access*, vol. 10, pp. 127 575–127 588, 2022.

[6] N. Gull, M. Rashid, F. Azam, Y. Rasheed, and M. Waseem Anwar, "A block-chain oriented model driven framework for handling inconsistent requirements in global software development," in *Proceedings of the 2021 10th International Conference on Software and Computer Applications*, 2021, pp. 105–111.

[7] S. Ali, Y. Hafeez, M. Humayun, N. Jhanjhi, and D.-N. Le, "Towards aspect based requirements mining for trace retrieval of component-based software management process in globally distributed environment," *Information Technology and Management*, vol. 23, no. 3, pp. 151–165, 2022.

[8] N. Koulecar and B. Ghimire, "Agile requirement change management model for global software development," *arXiv preprint arXiv:2402.14595*, 2024.

[9] S. Qureshi, S. U. R. Khan, Y. Javed, S. Saleem, A. Iqbal *et al.*, "A conceptual model to address the communication and coordination challenges during requirements change management in global software development," *IEEE Access*, vol. 9, pp. 102 290–102 303, 2021.

[10] S. Mughal, M. S. Mazhar, and A. Abbas, "Enhanced az-model for requirement change management in the context of global software development," *Authorea Preprints*, 2023.

[11] M. A. Akbar, A. A. Khan, S. Mahmood, and A. Mishra, "Srcmimm: the software requirements change management and implementation maturity model in the domain of global software development industry," *Information Technology and Management*, vol. 24, no. 3, pp. 195–219, 2023.

[12] M. Kausar, A. W. Muhammad, R. Jabbar, and M. Ishtiaq, "Key challenges of requirement change management in the context of global software development: systematic literature review," *Pakistan Journal of Engineering and Applied Sciences*, 2022.

[13] R. Michalski and S. Zaleski, "Success factors in management of it service projects: Regression, confirmatory factor analysis, and structural equation models," *Information*, vol. 15, no. 2, p. 105, 2024.

[14] M. A. Akbar, W. Naveed, A. A. Alsanad, L. Alsuwaidan, A. Alsanad, A. Gumaei, M. Shafiq, and M. T. Riaz, "Requirements change management challenges of global software development: An empirical investigation," *ieee access*, vol. 8, pp. 203 070–203 085, 2020.

[15] C. Tam, E. J. da Costa Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, 2020.

[16] D. Albuquerque, E. Guimaraes, M. Perkusich, A. Costa, E. Dantas, F. Ramos, and H. Almeida, "Defining agile requirements change management: a mapping study," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 1421–1424.

[17] T. Kamal, Q. Zhang, and M. A. Akbar, "Toward successful agile requirements change management process in global software development: a client–vendor analysis," *IET Software*, vol. 14, no. 3, pp. 265–274, 2020.

[18] I. Javed, U. I. Janjua, T. M. Madni, A. Akhunzada *et al.*, "The impact of mitigation strategies for socio-cultural distance issues in gsd: An empirical study," *IEEE Access*, 2023.

[19] M. A. Akbar, A. A. Khan, A. W. Khan, and S. Mahmood, "Requirement change management challenges in gsd: An analytical hierarchy process approach," *Journal of Software: Evolution and Process*, vol. 32, no. 7, p. e2246, 2020.

[20] T. Kamal, Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei, and A. Alsanad, "Identification and prioritization of agile requirements change management success factors in the domain of global software development," *IEEE Access*, vol. 8, pp. 44 714–44 726, 2020.

[21] M. A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A. A.-A. Alsanad, and A. Gumaei, "Identification and prioritization of devops success factors using fuzzy-ahp approach," *Soft computing*, pp. 1–25, 2023.

[22] J. Rezaei, "Best-worst multi-criteria decision-making method: Some properties and a linear model," *Omega*, vol. 64, pp. 126–130, 2016.

[23] T. L. Saaty, "How to make a decision: the analytic hierarchy process," *European journal of operational research*, vol. 48, no. 1, pp. 9–26, 1990.

[24] A. Aljuhani, "Multi-criteria decision-making approach for selection of requirements elicitation techniques based on the best-worst method," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021.

[25] A. Aljuhani and A. Alhubaishy, "Dynamic cloud resource allocation: A broker-based multi-criteria approach for optimal task assignment," *Applied Sciences*, vol. 14, no. 1, p. 302, 2023.

[26] A. Alhubaishy and A. Aljuhani, "The best-worst method for resource allocation and task scheduling in cloud computing," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2020, pp. 1–6.

[27] A. Aljuhani and A. Alhubaishy, "Incorporating a decision support approach within the agile mobile application development process," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2020, pp. 1–6.