Enhancing Agile Requirements Change Management: Integrating LLMs with Fuzzy Best-Worst Method for Decision Support

Bushra Aljohani, Abdulmajeed Aljuhani, Tawfeeq Alsanoosy
College of Computer Science and Engineering, Taibah University, Medina 41411, Saudi Arabia

***Abstract*—Agile Requirements Change Management (ARCM) in Global Software Development (GSD) posed significant challenges due to the dynamic nature of project requirements and the complexities of distributed team coordination. One approach used to mitigate these challenges and ensure efficient collaboration is the identification and prioritization of success factors. Traditional Multi-Criteria Decision-Making methods, such as the Best-Worst Method (BWM), had been employed successfully to prioritize success factors. However, these methods often failed to capture the inherent uncertainties of decision-making in a GSD. To address this limitation, this study integrated Large Language Models (LLMs) with the Fuzzy Best-Worst Method (FBWM) to enhance prioritization accuracy and decision support. We propose a model for comparing the prioritization outcomes of human expert assessments and LLM-generated decisions to evaluate the consistency and effectiveness of machine-generated decisions relative to those made by human experts. The findings indicate that the LLM-driven FBWM exhibit high reliability in mirroring expert judgments, demonstrating the potential of LLMs to support strategic decision-making in ARCM. This study contributed to the evolving landscape of AI-driven project management by providing empirical evidence of LLMs' utility in improving ARCM for GSD.**

Keywords*—Fuzzy Best-Worst Method; Large Language Models; Agile Requirements Change Management; Global Software Development

I. Introduction

In the context of Global Software Development (GSD), Agile Requirements Change Management (ARCM) depends on strong collaboration, yet prioritizing success factors poses notable challenges. The complexity of managing distributed teams, different time zones, and cultural differences further complicates the process [1], [2]. Moreover, the dynamic nature of requirements in GSD projects demands continuous reassessment of priorities and the ability to adapt quickly to changing conditions. Frequent changes in requirements require teams to constantly adjust their priorities in a fast-paced environment. The geographically dispersed nature of teams adds to the complexity, making communication and coordination crucial but often difficult to manage effectively [3]. Additionally, resource constraints, including limitations in both human and technological resources, further hinder the effective identification and prioritization of success criteria in such a dynamic environment [4]. As a result, effective decision-making in ARCM becomes critical to ensuring project success, requiring sophisticated tools and techniques to manage and prioritize requirements changes efficiently.

Furthermore, the inherent uncertainties associated with project requirements and changing environments demand robust decision-making frameworks. Traditional methods often fall short in addressing these complexities [5]. Thus, researchers have explored the effectiveness of Multi-Criteria Decision-Making (MCDM) approaches, such as the Analytic Hierarchy Process (AHP) [6], the Best-Worst Method (BWM) [7], and ELECTRE [8], to provide practical solutions for prioritizing success factors under these challenging conditions [9]. A prominent MCDM technique is the BWM [7], which involves identifying the most and least critical factors and comparing other factors relative to these extremes.

The emergence of Large Language Models (LLMs) has opened new avenues for enhancing ARCM in GSD. LLMs, such as OpenAI's GPT-4 [10], have demonstrated capabilities in understanding and generating human-like text, making them valuable as virtual experts across various domains. Incorporating LLMs into ARCM processes can assist in automating documentation, facilitating communication among distributed teams, and providing insights for decision-making, thereby addressing some of the inherent challenges in GSD.

Despite significant progress in ARCM and MCDM methods, such as BWM, challenges remained. Traditional methods relied on expert evaluations, which were time-consuming and prone to bias. Additionally, they struggled with uncertainty in dynamic environments. The integration of artificial intelligence (AI) and LLMs into ARCM was still in its early stages, with limited empirical validation of their effectiveness in decision support and prioritization.

To address the limitations, this study aims to enhance prioritization accuracy and decision support by integrating LLMs with the FBWM. Specifically, we aim to answer the following research questions:

- How can LLMs replicate human expert decision-making in prioritizing ARCM success factors?
- Does the integration of LLMs with FBWM improve the consistency and reliability of prioritization outcomes compared to traditional human-driven assessments?

Thus, in this paper, we extend the application of LLMs to ARCM within GSD by integrating LLMs with FBWM to enhance prioritization accuracy and decision support. We compared and validated the prioritization outcomes derived from human expert assessments with those generated by LLMs. The findings showed that the LLM-driven FBWM

demonstrated high reliability in mirroring expert judgments. The outcome of this research will offer practitioners a comprehensive taxonomy of success factors, prioritized effectively to improve decision-making processes and operational efficiency, ultimately enhancing software quality, accelerating delivery, and fostering better collaboration in GSD.

This paper is organized as follows: Section II presents existing studies on ARCM in GSD, the application of MCDM techniques and LLMs, and identifies the gaps that this research aims to address. Section III details the research methodology, including the design and implementation of the FBWM and LLM framework for ARCM. Section IV discusses the findings and their implications. Finally, Section VI summarizes the key contributions and concludes the paper.

II. Related Work

Several studies have addressed the adoption of MCDM techniques to enhance decision-making in software engineering and RCM practices [11], [12], [13], [14].

Akbar et al. [14] prioritized factors influencing RCM in GSD by using a questionnaire survey to gather feedback from practitioners. The authors applied the Fuzzy Analytical Hierarchy Process (FAHP) to address complex decision-making challenges. They offered a taxonomy-based prioritization of RCM success factors and introduced the FAHP method to help practitioners make informed decisions and enhance RCM processes in GSD environments.

In addition, Aljuhani [9] investigated the use of MCDM techniques, specifically BWM, within the context of ARCM. The author proposed a model for prioritizing ARCM success factors in the context of GSD using BWM. The BWM was used to rank success factors based on criteria such as integration, communication, and human resources. The model aimed to address complex decision-making problems involving multiple criteria and alternatives. The results demonstrated that BWM could be applied effectively to optimize decisions and outcomes in ARCM processes, providing a structured and efficient approach to managing competing factors in GSD projects.

Kamal et al. [15] identified and prioritized the success factors for ARCM in the context of GSD by applying AHP to the identified factors. The authors listed 21 success factors through a systematic mapping study and survey. The results of the AHP analysis revealed that the highest priority success factors were the allocation of resources at overseas sites (including communication, coordination, and control), a geographically distributed change control board (CCB), RCM process improvement expertise, and continuous top management support.

Additionally, Batool and Inayat [16] conducted an empirical investigation into RCM practices within Pakistani agile-based software development. The authors identified 30 RCM practices through a survey of 140 agile practitioners, employing PROMETHEE [17] as an MCDM method to rank these practices based on perceived importance. The findings highlighted that proper training for employees, maintaining version control, conducting review meetings, and using traceability tools (e.g., Jira) were the most critical practices. The study provided insights into the role of RCM in agile environments, emphasizing its dependence on project characteristics such as methodology, domain, and application type.

Several researchers have investigated the factors that affect Requirements Engineering (RE) or RCM in GSD or proposed frameworks to address problems in GSD [15], [18], [19], [20], [21], [22], [3]. For example, Koulecar and Ghimire [3] proposed the ARCM-GSD model, an extension of existing RCM frameworks, designed to better address requirements changes in GSD environments. The model introduced new phases such as traceability, categorization, prioritization, and effort estimation while also integrating agile methodologies into the RCM process. The results demonstrated that the model could be considered an effective framework for globally distributed agile teams dealing with requirements changes.

Furthermore, Khan et al. [23] investigated how communication during RCM in GSD is negatively affected by three types of distance: geographical, sociocultural, and temporal. The authors proposed a framework to explain these effects and validated it through a quantitative pilot study conducted in three GSD organizations. The findings revealed that increased physical distance, cultural differences, and time zone variations significantly hinder communication, highlighting the need for strategies to overcome these challenges.

Despite the promising contributions of these studies, several limitations can be identified. Aljuhani [9] applied the BWM to provide a systematic model for ARCM; however, the application of BWM relies on precise and deterministic values, which may not always capture the uncertainty inherent in real-world decision-making. As a result, this paper aims to address this limitation by integrating LLMs with FBWM to improve the accuracy and reliability of the prioritization process. Kamal et al. [15], while successfully identifying a broad set of success factors through the AHP model, faced challenges related to the consistency of pairwise comparisons and the subjectivity involved in weight assignments, which can undermine the robustness of their model in complex and evolving GSD environments. Additionally, Batool and Inayat's empirical investigation using PROMETHEE to rank RCM practices in agile contexts is insightful; however, its findings may be constrained by the localized context of Pakistani agile development and a static ranking framework that may not adapt well to the dynamic nature of agile projects.

To the best of our knowledge, this is the first study to integrate FBWM and LLMs in the context of ARCM for GSD. This addresses a critical gap in the existing literature, as the combination of these techniques has the potential to significantly enhance decision-making processes in GSD environments. While FBWM provides a structured approach to prioritizing requirements, LLMs are capable of handling complex, context-dependent issues. Their integration could offer a more robust and dynamic decision-support mechanism. Therefore, this research represents the first attempt to explore the integration of LLMs with FBWM, offering a novel approach to improving decision-making in GSD.

III. Methodology

To address the uncertainties inherent in decision-making, the Fuzzy Best-Worst Method (FBWM) [24] was introduced

as an extension of the traditional BWM [7]. By incorporating fuzzy logic, FBWM enhances the flexibility and reliability of the original method, making it particularly useful in scenarios where qualitative judgments dominate. Unlike techniques such as AHP, FBWM uses a simplified comparison structure with fewer pairwise comparisons, enabling steadier and more consistent judgments. FBWM leverages triangular fuzzy numbers (TFNs) to express the relative importance of criteria, thereby capturing the ambiguity of decision-makers' preferences. As described in Table I, this method introduces linguistic terms (e.g. "Equally Important," "Very Important"), which are transformed into TFNs for mathematical modeling. Two vectors—fuzzy Best-to-Others and fuzzy Others-to-Worst—are critical components of the method. These vectors reflect the decision-makers' assessments of the best criterion's dominance over others and the relative inferiority of other criteria compared to the worst criterion.

TABLE I. MEMBERSHIP FUNCTION [24]

Linguistic Terms	Membership Function
Equally Important (EI)	(1, 1, 1)
Weakly Important (WI)	(2/3, 1, 3/2)
Fairly Important (FI)	(3/2, 2, 5/2)
Very Important (VI)	(5/2, 3, 7/2)
Absolutely Important (AI)	(7/2, 4, 9/2)

The FBWM framework assumes that decision-makers can reliably identify the best and worst criteria, but it also accommodates the uncertainty and imprecision inherent in their judgments. To determine the criteria weights, a constrained nonlinear optimization problem is solved, minimizing the maximum deviation between fuzzy pairwise comparisons and the calculated weights. This approach ensures the consistency and reliability of the derived fuzzy weights.

FBWM retains the core strengths of the traditional BWM while addressing its limitations in handling subjective uncertainty. The use of fuzzy logic makes FBWM a robust and attractive approach across various disciplines, providing decision-makers with a structured and trustworthy method for identifying the most critical criteria in MCDM problems. As a result, FBWM has gained recognition as an advanced and practical tool for tackling complex decision-making scenarios.

This section outlines the research methodology, as depicted in Fig. 1, which consists of seven main phases: data collection, model selection, expert input, applying FBWM, weight calculation, and consistency check.

A. Data Collection

One important step to start with is data collection regarding criteria and success factors that need to be identified in order to apply FBWM. These factors have been categorized based on a literature review, expert opinions, and empirical studies.

Building upon the foundational work of Aljuhani [9], this research utilizes an identified hierarchy of critical success factors (CSFs), as illustrated in Fig. 2. These factors, originally proposed in [25], [26], and [15], categorize the CSFs under six main criteria:

- Integration (C1)
- Communication (C2)
- Project administration (C3)
- Human resources (C4)
- Technology factors (C5)
- Time (C6)

Similarly, for alternatives, nine success factors have been utilized, as shown in Fig. 2, which are:

- 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)

B. Model Selection

In this research we utilize the openAI model, which is ChatGPT-4 due to its reasoning ability and cost effectiveness.

- LLM Model: The GPT-4 was set to the following settings:
  - Model: gpt-4
  - Temperature: 0.8
  - Verbose: False
- LLM Interaction: LangChain library was used to manage the conversation and enable role based prompting.[1]
- Computational Environment: The experiments were carried out on Google Colab, a cloud-based platform that offers access to high-performance computing resources and a Python-based environment.

C. Expert Input

In this phase, we obtained opinions from both human and virtual experts, where human experts were provided with a structured questionnaire to evaluate the CSFs. On the other hand, the virtual expert (e.g. LLM) was utilized based on the role-based prompting technique to ensure guided and context-aware responses.

We utilized a prompt engineering technique to allow the LLM to mimic a domain expert role, guiding its responses and ensuring high-quality outputs. The task has been decomposed into four main tasks, which are: label=•

- Level 1 label=–
  - Best and Worst Criteria Selection.

[1]ConversationBufferMemory

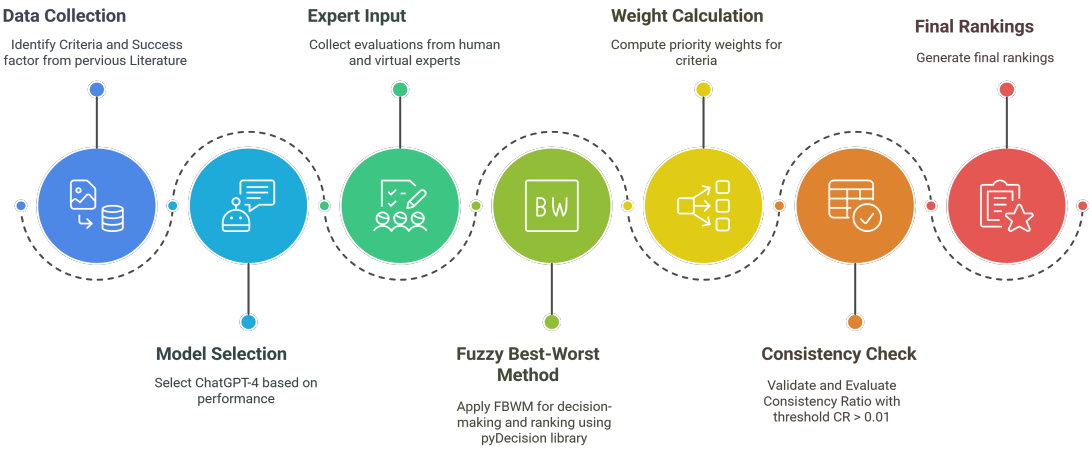

Fig. 1. Overview of the research methodology.

- Best-to-other and Worst-to-others criteria pairwise comparison.

- Level 2 label=–
  - Success factors Best and Worst Selection.
  - Success factors Best-to-other and Worst-to-others pairwise comparison.

Below, a prompt template example of task 1 which is best and worst criteria selection:

```
response = conversation.predict(input="""
You are a Software Engineer with deep knowledge of
    development processes and the technical
    challenges associated with Agile transformations
     in global teams.
Your focus is on code quality, maintainability, and
    team efficiency.
Your task is to analyze the given six criteria and
    select one as the best and one as the worst
    based on your role
The six criteria are:
1. Integration
2. Communication
3. Project Administration
4. Human Resources
5. Technology Factors
6. Time

Provide a structured response with:
1. The **best** criterion and why it is the most
    important.
2. The **worst** criterion and why it is the least
    important.
""")

print(response)
```

Listing 1: LLM prompt for selecting Best and Worst Criteria

```
Based on my perspective as a Software Engineer with
    extensive experience in development processes
    and the technicalities involved in Agile
    transformations for global teams, I would rank
    the given criteria as follows:

1. **Best Criterion: Technology Factors**

Technology Factors are the most significant for the
    following reasons:

   - Essential Tools: Technology acts as the
     backbone for any software development process
     . The appropriate use of technology can
     simplify complex processes, increase
     efficiency, and reduce the risk of errors.

   - Facilitates Other Factors: A robust
     technological infrastructure facilitates
     better communication, integration, and even
     project administration as it allows for more
     efficient tracking and management of tasks.

   - Code Quality and Maintainability: Technology
     influences code quality and maintainability.
     The right technology can make the process of
     writing, testing, and maintaining code much
     easier and efficient.

2. **Worst Criterion: Human Resources**

While all aspects are crucial for the success of a
    project, as per my role and emphasis on
    technical aspects of software development, Human
     Resources is relatively less important for the
    following reasons:

   - Indirect Influence: Human resources have an
     indirect influence on code quality,
     maintainability, and team efficiency. While
     crucial, this factor does not directly
     address these technical aspects.

   - Dependency: The effectiveness of human
     resources largely depends on other factors
     like communication, technology, and project
     administration. Without these in place, even
     the most skilled human resource may not be
     able to guarantee success.

Please note, however, that this evaluation does not
    downplay the significance of human resources. It
     merely emphasizes the relative importance of
    technology factors from a software engineer's␣
    perspective.␣The␣importance␣of␣these␣criteria␣
    can␣shift␣based␣on␣the␣specific␣context␣and␣
    requirements␣of␣a␣project.
```

Listing 2: LLM response of selecting Best and Worst Criteria

We then repeated the same process with other virtual

experts. It is worth noting that GPT-4 excels in managing diverse expert roles and in its ability to justify its answers by providing a clear rationale behind its selections. This capability ensures that GPT-4 not only meets the task requirements accurately but also explains the basis of its choices, making it an invaluable asset for scenarios where detailed explanations are essential for validating the decision-making process.

D. Steps of FBWM in the Context of Agile Requirements Change Management

In this research, we employ FBWM to prioritize CSFs for effective ARCM in GSD. As stated by Guo et al. [24], the adapted FBWM procedure involves the following steps:

1) Establish Decision Criteria: The first step involves identifying a set of CSFs that influence the effectiveness of ARCM in GSD which are essential for evaluating alternatives.
$C = \{c_1, c_2, ..., c_n\}$
This step has already been done in the data collection phase.
2) Determine Best and Worst Criteria:
Domain experts such as project managers or team leads are consulted to select the most crucial (Best) criterion c_B and least crucial (Worst) criterion c_W from the identified set without pairwise comparisons. based on their experience and understanding of ARCM in GSD. For instance, Human Resources (C4) might be identified as the best criterion, while Integration (C1) might be the worst.
3) Fuzzy Pairwise Comparisons with the Best Criterion: In this step ,each CSF is compared with the best criterion c_B using linguistic terms (e.g. "Equally Important," "Fairly Important," "Very Important"). These linguistic assessments are then transformed into triangular fuzzy numbers (TFNs) using membership function I. This step aims to capture the inherent uncertainty and subjectivity associated with expert judgments. The fuzzy Best-to-Others vector is formulated as in Eq. (1):

$$\tilde{A}_B = (\tilde{a}_{B1}, \tilde{a}_{B2}, \ldots, \tilde{a}_{Bn}) \tag{1}$$

In the context of ARCM in GSD, (e.g. "Communication C2") compared to the best criterion (e.g. "Human Resources C4").
- Linguistic assessment: "Very Important"
- Transformed to TFN: (5/2, 3, 7/2)

4) Fuzzy Pairwise Comparisons with the Worst Criterion: Similarly, compare all criteria to the worst criterion c_W using linguistic terms and transformed into TFNs. The fuzzy Others-to-Worst vector is formulated as in Eq. (2):

$$\tilde{A}_W = (\tilde{a}_{1W}, \tilde{a}_{2W}, ..., \tilde{a}_{nW}) \tag{2}$$

For instance, (e.g. "Communication C2") compared to Worst criterion (e.g. " Integration C1")
- Linguistic assessment: "Fairly Important"
- Transformed to TFN: (3/2, 2, 5/2)

5) Determine Fuzzy Weights: This step ensures that the weights assigned to each criterion reflect their relative importance in the context of ARCM in GSD. Where weights of each CSF are determined $(\tilde{w}_1^*, \tilde{w}_2^*, \ldots, \tilde{w}_n^*)$ by solving an optimization problem, as shown in Eq. (3),(4):

$$\begin{aligned}
&\min \quad \tilde{\xi} \\
&\text{s.t.} \quad \begin{cases} \left| \dfrac{\tilde{w}_B}{\tilde{w}_j} - \tilde{a}_{Bj} \right| \leq \tilde{\xi}, \\ \left| \dfrac{\tilde{w}_j}{\tilde{w}_W} - \tilde{a}_{jW} \right| \leq \tilde{\xi}, \end{cases} \\
&\sum_{j=1}^{n} R(\tilde{w}_j) = 1, \\
&l_j^w \leq m_j^w \leq u_j^w, \\
&l_j^w \geq 0, \\
&j = 1, 2, \ldots, n.
\end{aligned} \tag{3}$$

$$\text{where} \quad \tilde{\xi} = (l^{\xi}, m^{\xi}, u^{\xi}). \tag{4}$$

This step has been carried out by utilizing pyDecision library[2].
6) Defuzzification (Converting to Crisp Values): The final step, fuzzy weights $\tilde{w}_i$ can be converted to crisp values which help in prioritizing success factors, guiding project managers on which aspects to focus on for improving change management in GSD. This is done using the Graded Mean Integration Representation (GMIR) method, as formulated in Eq. (5):

$$R(\tilde{a}) = \frac{l + 4m + u}{6} \tag{5}$$

where l, m, u are the lower, middle, and upper values of the TFN.

E. Evaluation

This section covers the metrics used to evaluate the model which are consistency ratio evaluation and correlation check.

1) Consistency Check: The Consistency Ratio (CR) ensures the reliability of fuzzy pairwise comparisons in FBWM, crucial for ranking ARCM success factors in GSD. A comparison is fully consistent if (6):

$$\tilde{a}_{Bj} \cdot \tilde{a}_{jW} \approx \tilde{a}_{BW} \tag{6}$$

where $\tilde{a}_{BW}$ is the fuzzy preference relative to the best and worst criteria.The CR is then calculated as shown in equation (7):

$$CR = \frac{\tilde{\xi}^*}{\text{Consistency Index}} \tag{7}$$

where low CR values indicate better consistency. If CR is high, pairwise comparisons need to be revised to ensure the reliable prioritization of success factors in GSD.

In this study, we set a strict threshold of 0.01 for weight evaluations, ensuring high decision consistency, reduced subjective bias, and enhanced model precision. This threshold, implemented using the pyDecision library, required weights to deviate no more than 0.01 from a fully consistent pairwise comparison, ensuring optimal consistency.

[2]pyDecision

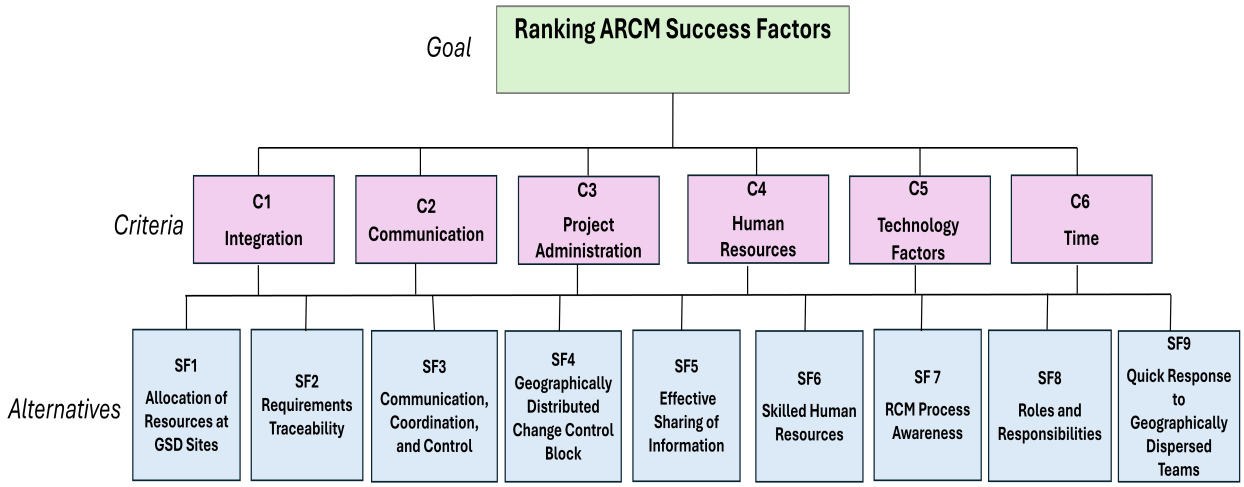

Fig. 2. A list of criteria and success factors adopted

2) Correlation Check: To evaluate the similarity between rankings generated by LLMs and human experts, we employed four key ranking similarity measures: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Spearman's Rank Correlation (ρ), and Kendall's Tau Correlation (τ).

These metrics are crucial for evaluating model performance by measuring ranking differences and correlations [27], [28], [29]. They have been widely used in studies to analyze ranking consistency across different evaluation models, especially in machine learning and software effort estimation. Using these metrics, we can effectively assess how well LLM-generated rankings match those assigned by human experts [30], [31].

1) MAE (8): Quantifies the average absolute difference between the rankings assigned by human experts and the LLM-generated rankings. A lower MAE indicates a closer alignment between the two ranking sets, as shown in Eq. (8).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\text{Human\_Rank}_i - \text{LLM\_Rank}_i| \quad (8)$$

Where:
- N is the total number of items (e.g. criteria or CFSs) being ranked.
- i represents the index that identifies each item in N.
- Human_Rank_i is the rank assigned to the i-th item by the human experts.
- LLM_Rank_i is the rank assigned to the i-th item by the LLM.

2) RMSE (9): Penalizes larger ranking discrepancies more heavily, providing a measure of deviation between LLM and human rankings, as shown in Eq. (9):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{Human\_Rank}_i - \text{LLM\_Rank}_i)^2} \quad (9)$$

Where:
- N is the total number of items (e.g. criteria or CSFs) being ranked.
- i represents the index that identifies each item in N.
- Human_Rank_i is the rank assigned to the i-th item by the human experts.
- LLM_Rank_i is the rank assigned to the i-th item by the Large Language Model (LLM).

3) Spearman's Rank Correlation Coefficient (ρ): Evaluates the monotonic relationship between LLM and human rankings. A value close to 1 indicates high correlation, as shown in Eq. (10):

$$\rho = 1 - \frac{6\sum d_i^2}{N(N^2-1)} \quad (10)$$

where N is the total number of ranked items and d_i is the difference between the ranks of the same element in the two lists.

4) Kendall's Tau (τ): Measures the ordinal association between the two ranking sets, assessing the strength of agreement, as shown in Eq. (11).

$$\tau = \frac{(C-D)}{\frac{1}{2}N(N-1)} \quad (11)$$

where N is the total number of ranked items, C represents the number of concordant pairs and D represents the number of discordant pairs.

IV. Results

This section presents the findings derived from the evaluation of both human experts and LLMs to enhance prioritization accuracy and decision support in ARCM. First, we present the ranking analysis and comparison of the SF rankings between human experts and virtual experts, followed by the results of the consistency ratio. Then, we present the similarity assessment results of the metrics used to understand the differences between the results of humans and LLMs. The results provide insights into whether LLMs can serve as viable decision-support tools for software development teams managing requirement changes in global projects.

TABLE II. HUMAN VS. LLM RANKED CRITERIA

Human Results			LLM Results		
Rank	**Criteria**	**Weight %**	**Rank**	**Criteria**	**Weight %**
1	C4 human resources	18.85	1	C2 communication	18.74
2	C2 communication	18.01	2	C1 integration	17.69
3	C1 integration	16.79	3	C5 technology factors	17.39
4	C3 project administration	16.01	4	C3 project administration	15.67
5	C5 technology factors	15.95	5	C6 time	15.30
6	C6 time	14.39	6	C4 human resources	15.21

TABLE III. HUMAN VS. LLM RANKED SUCCESS FACTORS

Human Results			LLM Results		
Rank	**Success Factors**	**Weight %**	**Rank**	**Success Factors**	**Weight %**
1	SF2 requirements traceability	14.93	1	SF9 quick response in GSD teams	12.25
2	SF4 geographically distributed change	13.71	2	SF5 effective share of information	12.04
3	SF1 allocation of resources	13.54	3	SF2 requirements traceability	11.97
4	SF5 effective sharing of information	12.25	4	SF3 communication & coordination	11.30
5	SF9 quick response in GSD teams	11.00	5	SF4 geographical distributed change	11.07
6	SF3 communication & coordination	10.12	6	SF8 roles & responsibilities	11.13
7	SF7 RCM process awareness	8.55	7	SF1 allocation of resources	10.57
8	SF8 roles & responsibilities	8.26	8	SF7 RCM process awareness	9.97
9	SF6 skilled human resources	7.64	9	SF6 skilled human resources	9.70

A. Ranking Analysis

Our experimental setup was strategically designed to incorporate prompt engineering techniques and persona development to ensure each virtual expert provided unique and insightful criteria rankings. This approach has shown great results with LLM-specific domain tasks [32], which, in our case, involve ranking ARCM success factors in the GSD context.

Table II demonstrates the aggregated results from human experts and virtual experts on criteria. The findings from human experts indicate that human resources (18.85%) and communication (18.01%) emerged as the most critical criteria, highlighting their significant influence on overall project success. On the other hand, virtual experts assigned the highest importance to communication (18.74%) and integration (17.69%), shifting the focus toward systematic collaboration and seamless interoperability.

Table III compares human and LLM rankings of SF and highlights notable similarities and differences in prioritization.

Human experts identified traceability (14.91%) and allocation of resources (13.54%) as key contributors to achieving project objectives, emphasizing the importance of effective resource management and maintaining a clear link between requirements and their implementation. Conversely, virtual experts assigned the highest priority to quick response in GSD teams (12.25%), effective sharing of information (12.04%), and requirements traceability (11.97%), highlighting a stronger preference for responsiveness, knowledge distribution, and maintaining requirement clarity. While both rankings acknowledge requirements traceability as crucial, the virtual experts place greater emphasis on responsiveness and knowledge sharing, whereas human experts lean towards resource and change management as pivotal for project success.

Overall, as shown in Fig. 3, both recognize the importance of strategic criteria in ARCM for GSD but prioritize different aspects. Human experts focus on context-specific elements and the human aspect, while the LLM emphasizes systematic aspects and effective information sharing. Integrating these perspectives can enhance the framework for managing ARCM in global projects.

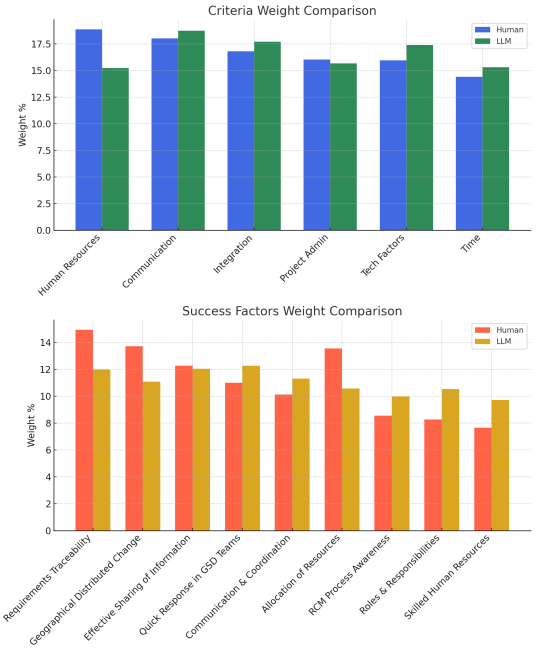

Fig. 3. Side-by-side comparisons for both criteria and SF between the Human and LLM.

B. Consistency Evaluation

Table IV presents the CR for both groups, which remained significantly below the threshold of 0.01. The results demonstrated consistent performance, with both human participants and LLMs achieving consistency ratios below 0.01, indicating reliable decision-making processes in ranking ARCM within the context of GSD. Particularly, the LLM demonstrated

consistency ratios under 0.06, emphasizing its precision in handling complex decision-making scenarios.

Human expert evaluations yielded CR values of 0.0855 for criteria selection and 0.0592 for CSFs. In comparison, the LLM produced values of 0.0660 and 0.0684, respectively. These findings suggest that the LLM performs similarly to human experts in maintaining ranking stability and making coherent decisions in complex MCDM scenarios. The LLM's lower CR for criteria selection indicates that it effectively captures ranking relationships while minimizing subjective inconsistencies. Overall, the results highlight the LLM's potential as a decision-support tool for ARCM in GSD.

TABLE IV. CONSISTENCY RATIO OF HUMAN AND LLM ON CRITERIA AND SUCCESS FACTORS

Metric	Human	LLM
Level 1 (criteria)	0.0855	0.0660
Level 1 (success factors)	0.0592	0.0684

C. LLM and Human Ranking Similarity Assessment

Table V presents the values of evaluation metrics used to compare the rankings of LLMs and human experts. The results highlight the nuanced differences in their performance across the criteria and SF.

The MAE indicates a minimal average difference for criteria (0.47), suggesting that the LLM closely aligns with human judgments, while a higher MAE for SF (1.52) reflects greater divergence in this area. Similarly, the RMSE, which emphasizes larger discrepancies, remains low for criteria (0.52) but rises to 1.68 for SF, underscoring the LLM's reliable performance in criteria ranking and its comparatively larger deviations in success factor prioritization. Spearman's (ρ) demonstrates a perfect match (1.00) in the ranking order of criteria and an almost perfect correlation (0.98) for SF, highlighting the LLM's strong ability to preserve ranking order even when exact values differ. Kendall's Tau (τ) further confirms this consistency, showing full agreement (1.00) in criteria ranking pairs and very strong agreement (0.94) for SF.

TABLE V. COMPARISON OF LLM AND HUMAN RANKINGS USING VARIOUS METRICS

Metric	Criteria	Success Factor
Mean Absolute Error (MAE)	0.47	1.53
Root Mean Squared Error (RMSE)	0.53	1.69
Spearman's Rank Correlation (ρ)	1.00	0.98
Kendall's Tau Correlation (τ)	1.00	0.94

Overall, the results indicate that while LLMs can effectively replicate human rankings for criteria with near-perfect accuracy, their performance in ranking CSFs, although still robust, demonstrates slight variations due to differences in weight assignment and prioritization.

V. DISCUSSION

The differences between LLMs and human decision-making come down to a few key factors. LLMs are trained on vast amounts of data, which helps them generate responses based on patterns they have learned. However, they lack real-time learning and experience-based adaptation, hindering their ability to adjust to new situations. Humans, on the other hand, are always learning from their experiences, which helps them adapt to unexpected circumstances [33], [34].

LLMs exhibit a capability for maintaining logical consistency in structured tasks; however, they may struggle with understanding context because they rely on statistical correlations rather than true comprehension. Humans have intuition and contextual awareness, which help them navigate ambiguous situations and make decisions based on the specifics of each scenario [35], [36].

Another difference is that LLMs can reflect biases from their training data, which can lead to outputs that fail to align with human ethical standards. Humans, while also prone to bias, use moral reasoning and ethical considerations to make decisions that reflect societal norms and values [37], [38], [39].

These differences show that LLMs and human decision-making complement each other. A hybrid approach can be utilized where LLMs provide consistency and efficiency in data-driven tasks, and humans bring depth in ethical reasoning and contextual understanding. By using this hybrid approach, we can combine computational precision with human insight to improve decision-making processes [40], [41].

VI. CONCLUSION

This study explored the integration of LLMs with FBWM to enhance decision-making in ARCM within GSD. The findings reveal that LLMs can effectively replicate expert decision-making, producing consistent and reliable prioritization of CSFs. The results highlight the significance of CSFs, such as communication and human resources, in shaping ARCM success. By leveraging LLMs, this research can assist practitioners and decision-makers in enhancing decision-making processes and operational efficiency, ultimately improving software quality, accelerating delivery, and fostering better collaboration in GSD. The study underscores the potential of AI-driven methodologies in optimizing software development practices and lays the foundation for future research in integrating advanced AI models with decision-support frameworks. However, the study has some limitations, including scalability to larger datasets and resource constraints, such as limited access to computational tools, which hinder the broader applicability of the proposed model. Future work should focus on integrating domain-specific models and testing the scalability of FBWM with larger datasets to validate its robustness. Additionally, exploring lightweight computational tools can enhance accessibility for resource-constrained organizations.

REFERENCES

[1] M. Neumann, K. Schmid, and L. Baumann, "What you use is what you get: Unforced errors in studying cultural aspects in agile software development," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, 2024, pp. 405–410.

[2] T. Alsanoosy, M. Spichkova, and J. Harland, "Cultural influence on requirements engineering activities: Australian practitioners' view," 2019.

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

[4] J. Ferdous, F. Bensebaa, A. S. Milani, K. Hewage, P. Bhowmik, and N. Pelletier, "Development of a generic decision tree for the integration of multi-criteria decision-making (mcdm) and multi-objective optimization (moo) methods under uncertainty to facilitate sustainability assessment: a methodical review," *Sustainability*, vol. 16, no. 7, p. 2684, 2024.

[5] F. Marle and L.-A. Vidal, "Limits of traditional project management approaches when facing complexity," in *Managing Complex, High Risk Projects: A Guide to Basic and Advanced Project Management*. Springer, 2016, ch. 2.

[6] 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.

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

[8] K. Govindan and M. B. Jepsen, "Electre: A comprehensive literature review on methodologies and applications," *European Journal of Operational Research (EJOR)*, vol. 250, pp. 1–29, 4 2016.

[9] A. Aljuhani, "Identification of agile requirements change management success factors in global software development based on the best-worst method," p. 2024. [Online]. Available: www.ijacsa.thesai.org

[10] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[11] A. Kumar and K. Kaur, "Mcdm-based framework to solve decision making problems in software engineering," in *2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. IEEE, 2022, pp. 1–5.

[12] A. Kumar, M. Nadeem, and M. Shameem, "Multicriteria decision-making–based framework for implementing devops practices: A fuzzy best–worst approach," *Journal of Software: Evolution and Process*, p. e2631, 2024.

[13] 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.

[14] M. A. Akbar, M. Shameem, A. A. Khan, M. Nadeem, A. Alsanad, and A. Gumaei, "A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development," *Journal of Software: Evolution and Process*, vol. 33, no. 2, p. e2292, 2021.

[15] 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.

[16] K. Batool and I. Inayat, "An empirical investigation on requirements change management practices in pakistani agile based industry," in *Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019*. Institute of Electrical and Electronics Engineers Inc., 12 2019, pp. 7–12.

[17] J. Figueira, S. Greco, and M. Ehrgott, Eds., *Multiple Criteria Decision Analysis: State of the Art Surveys*, ser. International Series in Operations Research & Management Science. New York: Springer, 2005, vol. 78. [Online]. Available: https://link.springer.com/book/10.1007/b100605

[18] S. Siddique, M. Naveed, A. Ali, I. Keshta, M. I. Satti, A. Irshad *et al.*, "An effective framework to improve the managerial activities in global software development," *Nonlinear Engineering*, vol. 12, no. 1, p. 20220312, 2023.

[19] T. Alsanoosy, M. Spichova, and J. Harland, "A framework for identifying cultural influences on requirements engineering activities," 2020.

[20] T. Alsanoosy, M. Spichkova, and J. Harland, "Identification of cultural influences on requirements engineering activities," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, 2020, pp. 290–291.

[21] M. Azeem Akbar, W. Naveed, A. Alsanad, L. Alsuwaidan, A. Alsanad, Gumaei *et al.*, "Requirements change management challenges of global software development: An empirical investigation," *IEEE Access*, vol. 8, 11 2020.

[22] I. Keshta, M. Niazi, and M. Alshayeb, "Towards implementation of requirements management specific practices (sp1.3 and sp1.4) for saudi arabian small and medium sized software development organizations," *IEEE Access*, vol. PP, pp. 1–1, 10 2017.

[23] A. A. Khan, J. Keung, S. Hussain, and K. E. Bennin, "Effects of geographical, socio-cultural and temporal distances on communication in global software development during requirements change management: A pilot study," in *ENASE 2015 - Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering*. SciTePress, 2015, pp. 159–168.

[24] S. Guo and H. Zhao, "Fuzzy best-worst multi-criteria decision-making method and its applications," *Knowledge-Based Systems*, vol. 121, pp. 23–31, 4 2017.

[25] 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, 02 2020.

[26] 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.

[27] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005. [Online]. Available: http://www.jstor.org/stable/24869236

[28] M. G. KENDALL, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1-2, pp. 81–93, 06 1938. [Online]. Available: https://doi.org/10.1093/biomet/30.1-2.81

[29] T. O. Hodson, "Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, 2022. [Online]. Available: https://gmd.copernicus.org/articles/15/5481/2022/

[30] S. K. Sehra, Y. S. Brar, and N. Kaur, "Applying fuzzy-ahp for software effort estimation in data scarcity," *International Journal of Engineering Trends and Technology (IJETT)*. [Online]. Available: http://www.ijettjournal.org

[31] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, p. 72, 1 1904.

[32] I. Svoboda and D. V. Lande, "Enhancing multi-criteria decision analysis with ai: Integrating analytic hierarchy process and gpt-4 for automated decision support," *Preprint*, February 2024.

[33] M. Steyvers, H. Tejeda, A. Kumar, C. Belem, S. Karny, X. Hu *et al.*, "What large language models know and what people think they know," *Nature Machine Intelligence*, vol. 7, pp. 221–231, February 2025.

[34] C. R. Jones, S. Trott, and B. Bergen, "Comparing humans and large language models on an experimental protocol inventory for theory of mind evaluation (epitome)," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 803–819, 06 2024.

[35] V. Lai, C. Chen, Q. V. Liao, A. Smith-Renner, and C. Tan, "Towards a science of human-ai decision making: A survey of empirical studies," 12 2021.

[36] D. Alsagheer, R. Karanjai, N. Diallo, W. Shi, Y. Lu, S. Beydoun, and Q. Zhang, "Comparing rationality between large language models and humans: Insights and open questions," 3 2024.

[37] A. Passerini, A. Gema, P. Minervini, B. Sayin, and K. Tentori, "Fostering effective hybrid human-llm reasoning and decision making," *Frontiers in Artificial Intelligence*, vol. 7, 2024.

[38] E. Eigner and T. Händler, "Determinants of llm-assisted decision-making," *arXiv preprint*, vol. arXiv:2402.17385, 2024.

[39] M. Lamparth, A. Corso, J. Ganz, O. Mastro, J. Schneider, Trinkunas *et al.*, "Human vs. machine: Behavioral differences between expert humans and language models in wargame simulations," *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, vol. 7, pp. 807–817, 10 2024.

[40] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, Dhariwal *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[41] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, and other, "A survey on llm-as-a-judge," 11 2024.