Maden, A., Yücenur, G. N. / Journal of Yasar University, 2025, 20/79, 376-406

# SWARA-DEMATEL-ARAS Kullanarak Çevik Yazılım Geliştirme Yöntemlerinin Değerlendirilmesi: SCRUM, XP veya KANBAN?

# Evaluating Agile Software Development Methods using SWARA-DEMATEL-ARAS: SCRUM, XP, or KANBAN?

Ayça MADEN, İstanbul Beykent Üniversitesi, Türkiye, aycamaden@beykent.edu.tr

Orcid No: 0000-0002-8239-3084

G. Nilay YÜCENUR, İstanbul Beykent Üniversitesi, Türkiye, nilayyucenur@beykent.edu.tr

Orcid No: 0000-0002-2670-6277

*Öz: Agile Manifesto, yazılım mühendisliği alanında çeviklik, uyarlanabilirlik ve esneklik teşvik ederek devrim yaratmıştır. Müşteri memnuniyetini, iş birliğini ve zamanında yüksek kaliteli yazılım teslimini vurgulamaktadır. Agile Manifesto'nun ilkelerine bağlı kalarak ortaya çıkan çevik metodolojiler, uygulamalarında, süreçlerinde ve temel prensiplerinde farklılıklar göstermektedir. Çevik yazılım geliştirme yöntemlerinin değerlendirilmesi, yazılım projelerinin başarılı bir şekilde tamamlanması için proje paydaşları açısından kritik öneme sahiptir. Bu çalışma, çevik yazılım geliştirme metodolojilerini iki aşamalı bir yaklaşımla değerlendirmeyi amaçlamaktadır. İlk aşama, SWARA ve DEMATEL yöntemleri kullanılarak kritik başarı faktörlerinin ağırlıklarının belirlenmesini, ikinci aşama ise belirlenen kritik başarı faktörlerine dayanarak çevik yöntemlerin etkinliğini değerlendirmek ve sıralamak için ARAS yönteminin kullanılmasını içermektedir. Çalışmanın özgünlüğü, çevik yazılım geliştirme yöntemlerini değerlendirmek için SWARA, DEMATEL ve ARAS yöntemlerinin uygulanmasında yatmaktadır ki bu yöntemler genellikle endüstride kullanılmamaktadır.*

*Anahtar Sözcükler: Yazılım geliştirme, Çevik Yöntemler, SWARA, DEMATEL, ARAS*

*JEL Sınıflandırması: O31, O32, O33*

*Abstract: Agile Manifesto has revolutionized the field of software engineering by promoting agility, adaptability, and flexibility during the development process. It emphasizes customer satisfaction, collaboration, and delivering high-quality software in a timely manner. Agile methodologies have emerged, varying in adherence to the principles of Agile Manifesto. Despite shared features, these methodologies exhibit variations in their practices, processes, and fundamental principles. The evaluation of agile software development methods is crucial for project stakeholders to ensure the successful completion of software projects. This study aims to evaluate agile software development methodologies through a two-stage approach. The first stage involves determining the critical success factors weights using the SWARA and DEMATEL methods, and the second stage employs the ARAS method to evaluate and rank the effectiveness of the agile methods based on the identified CSFs. The originality of this study lies in its application of the SWARA, DEMATEL, and ARAS methods to evaluate agile software development methods, which are not typically employed in the industry.*

*Keywords: Software development, Agile Methods, SWARA, DEMATEL, ARAS*

*JEL Classification: O31, O32, O33*

## 1. Introduction

The Agile Manifesto has introduced transformative changes to the field of software engineering, leaving a lasting impact on the industry. The essence of agile software development lies in its commitment to agility, which signifies the ability to promptly and

**Makale Geçmişi / Article History**
Başvuru Tarihi / Date of Application : 7 Ağustos / August 2024
Kabul Tarihi / Acceptance Date : 25 Mart / March 2025

efficiently adapt to uncertain and unpredictable demands. The principles outlined in the Agile Manifesto underscore crucial aspects such as customer satisfaction, collaboration, flexibility, continuous improvement, and a commitment to delivering high-quality software in a timely and efficient manner. These objectives are pursued through teamwork, open communication, and a readiness to embrace change. The principles of the Manifesto provide a foundational framework for implementing agile software development practices (Fowler and Highsmith 2001). A diverse array of agile methodologies has emerged, each varying in its adherence to the principles of the Agile Manifesto. Notable methodologies include Extreme Programming (XP), SCRUM, Lean Software Development, and Feature-Driven Development (FDD), among others. These methodologies strive to address the core values and principles articulated in the Manifesto, albeit to different extents.

Agile methodologies stand out for their adaptability and flexibility in the software product development process (Fowler and Highsmith 2001). This adaptability stems from the recognition of the limitations of traditional software engineering approaches in effectively managing the dynamic and uncertain conditions prevalent in the software market (Barry et al. 2002; Zorzetti et al. 2022). These methodologies belong to a category of iterative and incremental approaches, sharing common traits but distinguished by their specific practices, processes, and underlying principles. Selecting the optimal agile methodology for a specific project can pose a challenge for project analysts, particularly in situations with multiple options and a lack of a structured, empirical approach (Sharma and Bawa 2017). As agile methodologies vary in their practices, processes, and underlying principles, this challenge becomes even more pronounced. Therefore, the evaluation of agile software development methods using Critical Success Factors (CSFs) is crucial for assessing their effectiveness in achieving desired objectives. Such evaluations help to understand the strengths and weaknesses of agile methodologies, along with their impact on business outcomes, providing valuable insights for selecting the most appropriate methodology for a given project.

This study aims to evaluate agile software development methods using a two-stage process. In the initial stage, we aim to determine the importance weights of CSFs crucial for the evaluation of these software development methods. The Step-wise Weight Assessment Ratio Analysis (SWARA) and The Decision-Making Trial and Evaluation Laboratory (DEMATEL) methods are employed to ascertain the relative significance of various CSFs contributing to the overall success of agile software development. Moving on to the second stage, we utilize the Additive Ratio Assessment (ARAS) method to evaluate and rank the effectiveness of these agile methods based on the identified CSFs. The adoption of this two-stage approach facilitates a thorough and objective evaluation of the performance of agile

software development methods. This systematic methodology aids in pinpointing both the strengths and areas for improvement within their processes and practices. Through the application of these evaluation methods, software development teams can optimize their project outcomes, ultimately contributing to the realization of more successful software projects and increased satisfaction among stakeholders.

This study makes a significant contribution by introducing an integrated SWARA-DEMATEL-ARAS methodology, addressing notable gaps in the literature on agile software development method selection. Unlike widely used approaches such as Analytic Hierarchy Process (AHP), the proposed framework addresses issues of consistency and redundancy while incorporating cause-and-effect relationships among the selection criteria—an aspect often overlooked in previous studies. By combining the strengths of SWARA for efficient weighting, DEMATEL for visualizing interdependencies, and ARAS for straightforward alternative evaluation, this study offers a novel and comprehensive decision-making model that has not been explored in existing research. To the best of our knowledge, no prior studies in the literature have employed an integrated SWARA-DEMATEL-ARAS methodology for evaluating agile software development methods. The research questions of this study can be summarized as follows:

- What are the key CSFs for agile methods, and how can SWARA and DEMATEL determine their importance?
- How does integrating SWARA, DEMATEL, and ARAS provide a robust framework for agile evaluation?
- How does ARAS rank SCRUM, XP, and KANBAN based on their strengths and weaknesses in CSFs?

The structure of this paper is as follows: Section 1 introduces the study, Section 2 provides a literature review, Section 3 outlines the research methodology, Section 4 presents a case study, Section 5 details the application, Section 6 discusses the findings, and Section 7 concludes the paper.

## 2. Literature Review

The realm of software development has garnered significant attention from managers, engineers, and researchers, particularly due to the high incidence of failures within the software industry. Failures in this context encompass a spectrum, ranging from delivering solutions that fail to meet requirements or pose challenges in maintenance to outright abandonment of software projects. A distinctive challenge that sets software development apart and contributes to these difficulties is the continuous evolution of technology and the

business environment throughout the project lifecycle (Zorzetti et al. 2022). This dynamic landscape necessitates a flexible approach capable of swiftly adapting to new requirements, leading to the emergence of innovative methodologies, such as agile software development.

The Agile software methodology comprises a collection of software engineering methods that emphasize iterative and incremental development, in accordance with the principles of the "Agile Philosophy" articulated in the "Agile Manifesto" (Fowler and Highsmith 2001). The Agile Manifesto serves as a declaration of values and principles guiding software development, prioritizing iterative and adaptive approaches, collaboration, and customer satisfaction over conventional, more rigid processes and extensive documentation (Fowler and Highsmith 2001). In contrast to traditional software development, which heavily emphasized adherence to detailed plans and comprehensive documentation, agile methodologies advocate for a more flexible and dynamic approach. This marks a notable departure from the plan-centric approach advocated by more traditional software development models, such as the waterfall model (Royce 1970; Hoda et al. 2017). With the increasing adoption of agile methodologies, even by large-scale organizations in the software industry, there is a discernible shift towards achieving on-time and within-budget delivery of software projects, while concurrently addressing customer requirements more frequently (Brhel et al. 2015).

Prominent agile methodologies widely adopted in practice include SCRUM, XP, KANBAN, Lean, FDD, and the Dynamic Systems Development Method (Abrahamson et al. 2002; Khan et al. 2021). These methodologies have garnered both acclaim and critique, with studies highlighting that the capacity to adapt to change plays a pivotal role in determining the success or failure of an agile approach (Boehm 2002; Dikert et al. 2016). Over the past two decades, the utilization of agile methodologies has proliferated, and numerous studies have identified various shortcomings associated with their implementation (Zorzetti et al. 2022). These challenges include difficulties in understanding problems and identifying suitable solutions, struggles in generating business value, and a perceived lack of emphasis on design and architectural considerations (Zorzetti et al. 2022). The literature reveals a diverse array of studies employing various agile methods. For instance, Weflen et al. (2022) introduced a novel approach to estimating the lead time of tasks in agile KANBAN project management, leveraging an influence diagram. This proposed method aims to offer a more accurate and comprehensive estimation of task lead times, ultimately contributing to the effective management of agile KANBAN projects. In another example, Zorzetti et al. (2022) conducted a detailed analysis of the integration of three different development methodologies - Agile Software Development, User-Centered Design, and Lean Startup - into a unified development process and explore how this integration influences overall development outcomes.

In the literature, numerous studies explore various topics related to agile software using Multiple Criteria Decision Making (MCDM) methods. For instance, Abusaeed et al. (2023) introduced a Fuzzy AHP-based approach for prioritizing cost overhead factors in agile software development. In a different study, (El Beggar 2024) suggested an Intuitionistic Fuzzy Expert Judgment approach for effort estimation in agile software development. Shameem et al. (2020) aimed to explore the main barriers and create a taxonomy based on prioritization for overcoming challenges in scaling agile development within the global software development context. They employed Fuzzy-AHP to prioritize the identified barriers, and a taxonomy of these barriers and their associated categories was developed. Additionally, Govil and Sharma (2022) used the Fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method to validate Agile software development as the optimal choice. Silva et al.(2016) proposed a multi-criteria method, SMARTER, which can assist in making decisions regarding the selection of the most suitable agile software development methodology MCDM methods have been widely applied across various domains, including software engineering and testing (Abdulwareth and Al-Shargabi 2021; Magabaleh et al. 2024), demonstrating their versatility in complex decision-making processes. Their application extends beyond agile practices, providing valuable insights for evaluating diverse software engineering practices. Despite this growing interest in agile practices and the use of MCDM methods, however, only a few studies have explored the application of MCDM in evaluating agile methodologies. For instance, Khan et al. (2021) sought to establish a taxonomy of elements with the potential to enhance the scaling process of agile methods in the Chinese global software development sector. This taxonomy, grounded in agile success factors, involved categorizing and prioritizing the identified elements through the application of the Fuzzy AHP methodology. Sharma and Bawa (2017) employed a multi-level hybrid approach for the selection of agile development methods, integrating AHP, Preference Ranking Organization Method for Enrichment of Evaluations (PROMETHEE), and Fuzzy Logic. An overview of the evaluation of agile software development methods using MCDM methods is provided in Table 1.

Table 1. Evaluation of agile software development methods using MCDM methods

| *Study* | *Methods and tools* |
|---|---|
| *(Srivastava et al. 2020)* | *Analytic Network Process (ANP)* |
| *(Sharma and Bawa 2017)* | *AHP, PROMETHEE, Fuzzy Logic* |
| *(Yegen and Gül 2023)* | *AHP, TOPSIS-sort-B, VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)-sort-B* |
| *(Pandey and Litoriya 2021)* | *AHP* |
| *(Pandey and Litoriya 2020)* | *Fuzzy AHP, TOPSIS* |
| *(Yel and Baysal 2023)* | *Fuzzy AHP, Fuzzy Weighted Aggregated Sum Product Assessment (WASPAS), Fuzzy Evaluation Based on Distance from Average Solution (EDAS), Neutrosophic Z numbers* |

Following a comprehensive review of the studies presented in Table 1, it is evident that researchers have employed a variety of methodologies for criterion weighting and alternative selection. These methodologies include AHP, TOPSIS, PROMETHEE, WASPAS, and EDAS, with AHP emerging as the most frequently utilized method. However, recognizing the limitations associated with the AHP technique, especially in the context of selecting agile software development methods, this study proposes the use of an integrated SWARA-DEMATEL-ARAS methodology. This recommendation arises from the challenges posed by AHP when dealing with more than seven criteria or alternatives in a given problem, which can lead to issues of consistency and redundancy within pairwise comparison matrices (Kaviani et al. 2020; Saaty and Ozdemir 2003). The SWARA is as an efficient approach for determining criteria weights, applicable across various fields such as economics, management, industry, manufacturing, design and architecture, and policy and environmental sustainability. Noteworthy for its simplicity, straightforwardness, and the involvement of a minimal number of comparisons compared to other weighting methods, SWARA has garnered attention in the literature (Ghenai et al. 2020). In comparison to tools like AHP, the computations in SWARA are more straightforward. In the evaluation of agile software development methods, cause-and-effect relationships among the evaluation criteria exist. For example, managerial factors may influence organizational factors. However, existing MCDM methods related to the assessment of agile software development methods have not adequately addressed these cause-and-effect relationships. While AHP overlooks these relationships and fails to consider the impact of interrelations on weight coefficient values, an integrated DEMATEL and SWARA methodology enables the identification of cause-and-effect relationships between criteria and provides a graphical representation of these relationships (Badi et al. 2021). The ARAS method has gained popularity due to its simple and straightforward process, making it easy to follow while yielding reasonably accurate and satisfactory results. The increasing prevalence of hybrid MCDM approaches provides decision-makers with enhanced confidence in their outcomes, especially when dealing with diverse and complex information or addressing challenging problems. It is noteworthy that the existing literature lacks studies employing an integrated SWARA-DEMATEL-ARAS methodology.

### *2.1. CSFs of software development*

CSFs denote key issues that, when identified and effectively addressed, can substantially enhance the likelihood of project success (Nasir and Sahibuddin 2011; Ahimbisibwe et al. 2015). While these factors may vary based on the nature and scope of the project, they

commonly represent crucial elements that must be adequately managed to ensure the project's success.

In project management, project managers commonly regard time, cost, and quality as the three essential elements that significantly influence a project's success (Schwaber and Beedle, n.d.; Misra et al. 2009). These elements are widely acknowledged as competitive criteria for software development project managers (Misra et al. 2009). Within the realm of software projects, analyses and evaluations of project outcomes frequently demonstrate a tendency to prioritize traditional measures—namely, time, cost, and performance—as the key indicators of success or failure (Yaghoobi 2018). These triple constraints are considered critical factors in determining the overall effectiveness and efficiency of a software project and often take center stage when evaluating project outcomes. Over the years, the number of CSFs considered pivotal for software projects has witnessed a significant increase (Ahimbisibwe et al. 2015). The foundational identification of these CSFs dates back to the late 1980s, with scholars such as Slevin and Pinto (1987), Jeffrey et al. (1988), and Pinto and Slevin (1988). Since then, ongoing research in the field has led to the continuous expansion of this list. While the original studies primarily concentrated on a limited number of CSFs, such as time, cost, and performance, subsequent research has broadened the spectrum to incorporate other crucial factors like team dynamics, stakeholder engagement, and risk management.

The existing literature encompasses numerous studies that delve into the success factors in software development. For instance, Yaghoobi (2018) utilized the Fuzzy AHP to prioritize the key success factors crucial for the successful execution of software development projects. Khan et al. (2019) focused on process improvement management in global software development, identifying, categorizing, and prioritizing key factors that could enhance software process improvement activities. Employing Fuzzy AHP, the authors developed a prioritization-based taxonomy of software process improvement success factors. Additionally, Stankovic et al. (2013) conducted an empirical study in the Southeastern Europe region to identify critical factors influencing the success of agile software projects. Meanwhile, Misra et al. (2009) conducted a statistical survey-based study to identify significant success factors associated with the adoption of agile software development practices, aiming to enhance the comprehension of agile software development.

Even though all agile development methods are rooted in an iterative and incremental approach, they exhibit variations in their practices, processes, and core principles, despite sharing numerous similarities (Sharma and Bawa 2017). The evaluation of agile software development methods based on CSFs is essential for ensuring the success of software projects. These factors provide a structured framework for assessing the effectiveness of agile

methods in achieving project goals and meeting customer needs. Through the identification and measurement of CSFs, such as customer satisfaction, team productivity, and software quality, stakeholders can gain valuable insights into the strengths and weaknesses of their agile approach, enabling them to make informed decisions for improvement. In line with this, this study identifies 20 CSFs for evaluating agile software development methods, derived from a thorough analysis of the existing literature. As presented in Table 2, these factors are categorized into five primary groups: technical, organizational, managerial, process, and customer factors.

Table 2. Critical success factors for evaluating agile software development methods

| **Main factors** | **Critical success factors** | **Studies** |
|---|---|---|
| Technical factors | Risk minimization | (Akbar et al. 2019; Stankovic et al. 2013; Abusaeed et al. 2023) |
| | Technical infrastructure | (Misra et al. 2009; Yaghoobi 2018; Akbar et al. 2019) |
| | Planning requirements | (Kausar et al. 2023; Misra et al. 2009; Yaghoobi 2018) |
| Organizational factors | Communication | (Misra et al. 2009; Yaghoobi 2018; Khan et al. 2019; Akbar et al. 2019; Chow and Cao 2008; Sulayman et al. 2012; Niazi 2015). |
| | Staff participation | (Dikert et al. 2016; Khan et al. 2019; Akbar et al. 2019; Sulayman et al. 2012; Niazi 2015) |
| | Qualified human resources (HR) requirements | (Dikert et al. 2016; Yaghoobi 2018; Khan et al. 2019; Stankovic et al. 2013; Niazi 2015) |
| | Effective leadership requirements | (Yaghoobi 2018; Khan et al. 2019; Akbar et al. 2019; Chow and Cao 2008; Niazi 2015) |
| | Education requirements | (Dikert et al. 2016; Akbar et al. 2019; Chow and Cao 2008; Niazi 2015) |
| | Team/role distribution | (Estrada-Esponda et al. 2024; Misra et al. 2009; Shameem et al. 2020) |
| Managerial factors | Correct allocation of resources | (Khan et al. 2019; Akbar et al. 2019) |
| | Cost advantage | (Yaghoobi 2018; Chow and Cao 2008; Sulayman et al. 2012; Unterkalmsteiner et al. 2012) |
| | Documentation | (Yaghoobi 2018; Chow and Cao 2008; Niazi 2015) |
| | Joint management structure | (Dikert et al. 2016; Khan et al. 2019; Estrada-Esponda et al. 2024) |
| Process factors | Process improvement possibility | (Estrada-Esponda et al. 2024; Khan et al. 2019; Niazi 2015) |
| | Pilot application possibility | (Khan et al. 2019; Govil and Sharma 2022; Ali et al. 2023) |
| | Flexibility | (Stankovic et al. 2013; Misra et al. 2009) |
| | Effective time management | (Estrada-Esponda et al. 2024; Stankovic et al. 2013; Unterkalmsteiner et al. 2012) |
| | Applicability to complex problems | (Shameem et al. 2020; Misra et al. 2009) |
| Customer factors | Customer satisfaction | (Misra et al. 2009; Chow and Cao 2008; Niazi 2015; Unterkalmsteiner et al. 2012) |
| | Customer collaboration | (Misra et al. 2009; Yaghoobi 2018; Chow and Cao 2008; Sulayman et al. 2012) |

## 3. Research methodology

This study is structured into two distinct stages to comprehensively evaluate agile software development methods. In the initial stage, the SWARA and DEMATEL methods were employed to determine the importance weights of CSFs that are integral to the evaluation of these software development methodologies. Subsequently, in the second stage, the ARAS method was applied to assess and rank these methodologies. The rationale behind the selection of these methods lies in their unique application to an area where they are not conventionally utilized within the sector. The SWARA method was chosen for its demonstrated effectiveness in previous applications, offering decision-makers enhanced opportunities to establish priorities owing to its user-friendly nature. The DEMATEL method was chosen for its distinctive advantages, particularly its proficiency in identifying cause-and-effect relationships and its capability for graphical representation. Lastly, the ARAS method was preferred due to its ease of use and suitability for application by users of varying proficiency levels. Fig. 1 shows the proposed research methodology with its application steps.

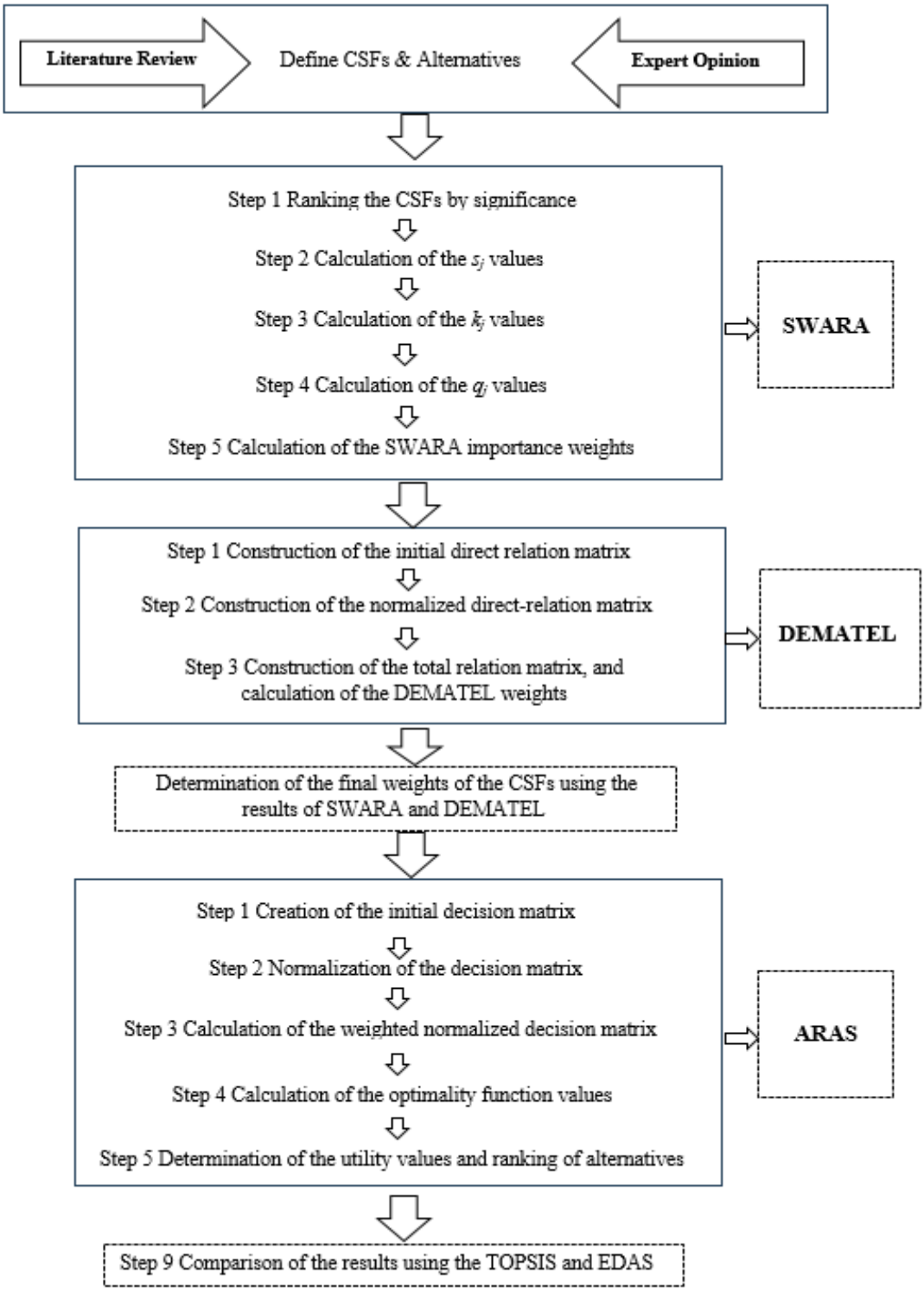


Figure 1. Proposed research methodology with SWARA-DEMATEL-ARAS

The successful completion of software projects is crucial for all project stakeholders, and this success hinges on addressing the CSFs specific to software development. In the context of this research study, 20 CSFs have been identified for the evaluation of agile software development methods in constructing a MCDM model. These CSFs are initially compiled from the existing literature and subsequently refined through valuable input provided by

experts. These success factors are considered as research criteria, and their explanations are listed below:

Technical factors:

- $SF_1$ Risk minimization: The imperative is to minimize risk, with a focus on the potential for detecting errors beforehand (Akbar et al. 2019; Stankovic et al. 2013).
- $SF_2$ Technical infrastructure: Ensuring that the technical infrastructure is sufficient for all process operations (Misra et al. 2009; Yaghoobi 2018; Akbar et al. 2019).
- $SF_3$ Planning requirements: The planning process, including determining project tasks, scheduling, prioritization, and assignment of responsibilities, should be easily achievable (Misra et al. 2009; Yaghoobi 2018).

Organizational factors:

- $SF_4$ Communication: Focusing on enhancing communication among project stakeholders to achieve high-quality targets and expedite market delivery (Misra et al. 2009; Yaghoobi 2018; Khan et al. 2019; Akbar et al. 2019; Chow and Cao 2008; Sulayman et al. 2012; Niazi 2015).
- $SF_5$ Staff participation: Ensuring a high level of staff participation in all processes (Dikert et al. 2016; Khan et al. 2019; Akbar et al. 2019; Sulayman et al. 2012; Niazi 2015).
- $SF_6$ Qualified HR requirements: Ensuring that the process can be managed with the minimum required level of qualified HR (Dikert et al. 2016; Yaghoobi 2018; Khan et al. 2019; Stankovic et al. 2013; Niazi 2015).
- $SF_7$ Effective leadership requirements: Minimizing the need for effective leadership with teams that can self-organize and manage (Yaghoobi 2018; Khan et al. 2019; Akbar et al. 2019; Chow and Cao 2008; Niazi 2015).
- $SF_8$ Education requirements: Conducting training activities at the level of need to eliminate deficiencies, rather than constant team training (Dikert et al. 2016; Akbar et al. 2019; Chow and Cao 2008; Niazi 2015).
- $SF_9$ Team/role distribution: Ensuring that the entire team actively participates and takes responsibility in the processes, rather than adhering to traditional role distribution in teams (Misra et al. 2009).

Managerial factors:

- $SF_{10}$ Correct allocation of resources: Eliminating waste in all processes through the correct allocation of resources (Khan et al. 2019; Akbar et al. 2019).

- $SF_{11}$ Cost advantage: Providing cost advantages by reducing cycle time and increasing quality (Yaghoobi 2018; Chow and Cao 2008; Sulayman et al. 2012; Unterkalmsteiner et al. 2012).
- $SF_{12}$ Documentation: Providing plan-oriented documentation that details all needs and process steps (Yaghoobi 2018; Chow and Cao 2008; Niazi 2015).
- $SF_{13}$ Joint management structure: Establishing a joint management structure involving all stakeholders with the customer at the focal point (Dikert et al. 2016; Khan et al. 2019).

Process factors:

- $SF_{14}$ Process improvement potential: The potential for process improvement is crucial for ensuring quality and customer satisfaction (Khan et al. 2019; Niazi 2015).
- $SF_{15}$ Pilot application potential: The ability to conduct pilot applications is important for testing new software features and making necessary corrections (Khan et al. 2019).
- $SF_{16}$ Flexibility: Processes and coding should remain as simple, understandable, and flexible as possible (Misra et al. 2009).
- $SF_{17}$ Effective time management: Effective time management is crucial for project planning and achieving customer satisfaction (Stankovic et al. 2013; Unterkalmsteiner et al. 2012).
- $SF_{18}$ Applicability to complex problems: The software development process should be able to directly and quickly address complex problems (Misra et al. 2009)..

Customer factors:

- $SF_{19}$ Customer satisfaction: Ensuring high customer satisfaction from the initial stages of the project (Misra et al. 2009; Chow and Cao 2008; Niazi 2015; Unterkalmsteiner et al. 2012).
- $SF_{20}$ Customer collaboration: Encouraging collaboration and feedback from the customer through their prior use of the software (Misra et al. 2009; Yaghoobi 2018; Chow and Cao 2008; Sulayman et al. 2012).

After determining the 20 CSFs as decision criteria, three agile software development methods were identified as decision alternatives. The methods are listed as follows:

- $A_1$ SCRUM: Developed by Ken Schwaber, SCRUM is an agile process that shifts the focus from describing technical, design, and implementation steps in software development to how team members should function in a system development environment that is constantly changing (Schwaber 1997). The key concept in SCRUM is the acknowledgment that environmental and technical variables, such as needs, time, resources, and technology, can change throughout the process, necessitating flexibility to

respond to these changes. SCRUM involves a self-organizing and guiding team (Hron and Obwegeser 2022; Cano et al. 2021).

- $A_2$ Extreme Programming (XP): Defined by Kent Beck, XP has gained significant popularity in recent years. XP is primarily a software development method emphasizing simplicity, communication, feedback, and encouragement. In specific terms, it incorporates practices such as planning games, refactoring, simple design, testing, binary coding, frequent releases, metaphor, co-ownership, continuous integration, customer engagement, coding standards, and an open business space (Zorzetti et al. 2022).
- $A_3$ KANBAN: This method is a highly adaptable, visual, and cost-oriented technique derived from production operations. While initially utilized by Toyota in the 1950s, its application to the software development process was first introduced by David J. Anderson in 2004 (Ahmad et al. 2018). The fundamental idea behind the KANBAN method is rooted in lean thinking. The three main philosophies of the method include visualizing the workflow, limiting the work in progress, and setting cycle times. The KANBAN method visualizes tasks that are in progress and those that are possible to be done on a board known as the " KANBAN Board." This approach allows the entire team and the customer to make informed decisions by observing the latest status, priorities, and potential bottlenecks in ongoing tasks (Weflen et al. 2022).

In Fig. 2, the proposed research model is shown with its 20 CSFs and 3 agile software development method alternatives.

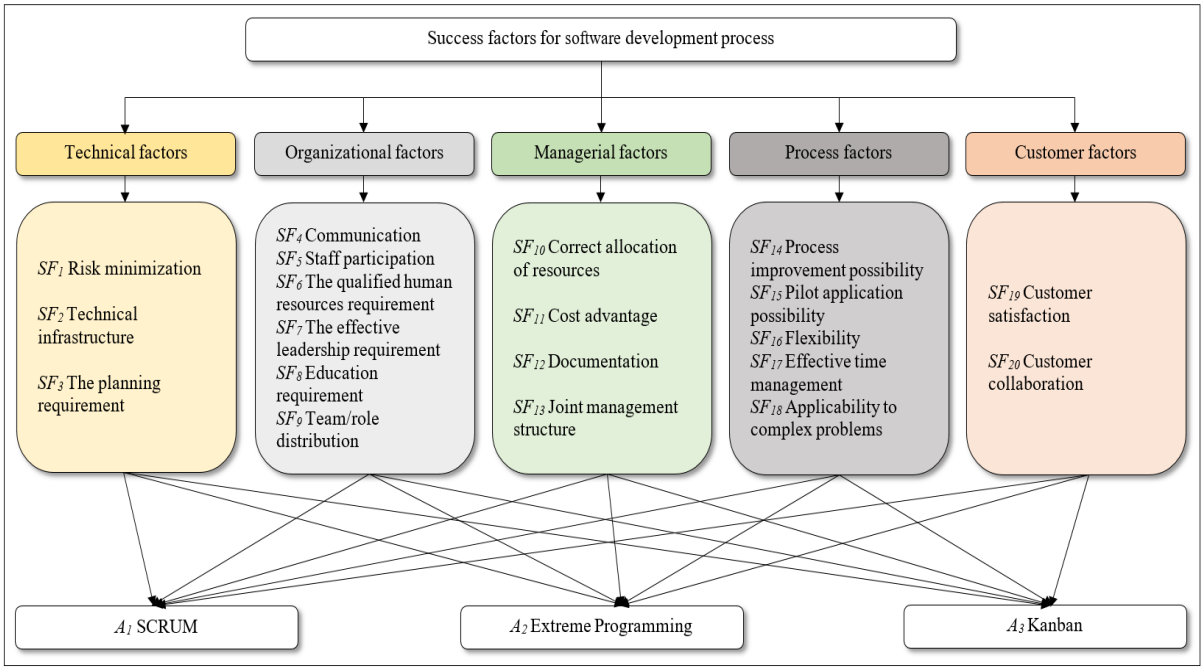


Figure 2. The proposed research model for evaluating agile software development methods

The first method used in this study is the SWARA method, developed by Keršulienė et al. in 2010. It is an effective tool for incorporating expert opinions into the decision-making process through a straightforward relative comparison (Keršulienė et al. 2010). This method places a significant emphasis on the insights provided by experts in calculating criteria weights and determining their respective importance levels. The process of determining the relative importance weights of criteria using the SWARA method involves the following steps (Keršulienė et al. 2010; Cui et al. 2021):

***Step 1:*** *Ranking the criteria by significance*: In this step, the criteria are ranked in descending order of importance based on expert opinions. The $p_j^k$ values are then assigned to the criteria with 0.05 intervals to comply with this order ($0 \leq p_j^k \leq 1$). The value $p_j^k$ is set to *1.00* for the most crucial criterion and *0.00* for the least significant one. In cases where there are multiple decision-makers, an overall ranking is achieved by calculating the geometric mean of these processes. $P_j$ values are then computed for each criterion using Eq. (1).

$$P_j = \frac{\sum_k^l p_j^k}{l} \tag{1}$$

In this equation, $l$ represents the number of experts ($k = 1, \ldots, l$).

***Step 2:*** *Calculation of $s_j$ values:* In this step, the relative importance of each criterion is determined. For this, criterion $j$ is compared with criterion $(j+1)$, and it is determined how important criterion $j$ is than criterion $(j+1)$. This comparison yields the $s_j$ value, representing the comparative significance with the mean value.

***Step 3:*** *Calculation of $k_j$ values:* In this step, the $k_j$ value for each criterion is calculated using Eq. (2). The initial assumption is that the $k_j$ value is 1.00 for criterion $j$, and for criteria ranked below criterion $j$, the $k_j$ value is obtained by adding the $s_j$ value calculated in the previous step.

$$k_j = \begin{cases} 1, & j = 1 \\ s_j + 1, & j > 1 \end{cases} \tag{2}$$

***Step 4:*** *Calculation of the $q_j$ value:* The adjusted weight values ($q_j$) for all criteria are calculated using Eq. (3).

$$q_j = \begin{cases} 1, & j = 1 \\ \frac{q_{j-1}}{k_j}, & j > 1 \end{cases} \tag{3}$$

***Step 5:*** *Calculation of $W_j^S$ values:* The relative importance weights $W_j^S$ of the evaluation criteria are then calculated using Eq. (4), where $W_j^S$ represents the relative weight of criterion $j$.

$$W_j^S = \frac{q_j}{\sum_{j=1}^n q_j} \tag{4}$$

The DEMATEL technique, developed at the Battelle Geneva Institute in 1971, is widely employed to address complex causal issues within intricate systems. Its primary objective is to unveil relationships among different factors, identifying both direct and indirect dependencies among criteria (Si et al. 2018; Rekik and El Alimi 2023). The DEMATEL model, grounded in graph theory, necessitates the creation of an influential network relation map. This visual representation of relationships simplifies the understanding of significant factors and their causal implications within the complex structure of a problem. The outlined steps below delineate the essential procedures for executing the DEMATEL approach:

***Step 1:*** Generate a direct comparison relation matrix for k experts and n criteria, utilizing a 10-point scale ranging from 0 (indicating no influence) to 10 (representing very high influence).

$$Z = [Z_{ij}^{k}] \tag{5}$$

Then, calculate the aggregated matrix for k experts by employing the arithmetic mean.

***Step 2:*** Normalize the aggregated matrix using the provided equations:

$$L = \frac{1}{\max\ 1 \le i \le n\ \sum_{j=1}^{n} Z_{ij}} \tag{6}$$

$$X = L \times Z \tag{7}$$

***Step 3:*** Calculate the total-relation matrix ($T$) as follows:

$$T = X \times (I - X)^{-1} \tag{8}$$

where $I$ is the identity matrix.

Then, calculate the sum of rows $R_i$ and columns $C_i$ from the total matrix using the following equations:

$$R_i = \sum_{j=1}^{n} T_{ij} \quad \forall\, i, \tag{9}$$

$$C_i = \sum_{j=1}^{n} T_{ij} \quad \forall\, j, \tag{10}$$

After this step, determine the DEMATEL weights by:

$$W_j = \sqrt{(R_i + C_i)^2 + (R_i - C_i)^2} \tag{11}$$

Then, calculate the final weights as:

$$W_j^D = \frac{W_j}{\sum_{k=1}^{n} W_j}, \tag{12}$$

The equation for computing the final weights for ranking the CSFs related to agile software development methods is as follows:

$$W_j^F = \frac{W_j^S \times W_j^D}{\sum_{k=1}^{n} W_j^S \times W_j^D}, \tag{13}$$

The ARAS method, introduced to the literature by Zavadskas and Turskis (2010), is based on ranking alternatives for the decision-making process according to the utility function within the scope of the selected evaluation criteria (Sivalingam et al. 2022). Based on the results obtained from this method, the proportional similarity of each alternative concerning the ideal alternative is taken into account while determining the performance ranking of the alternatives. The process of evaluating and ranking alternatives using the ARAS method includes the following steps (Sivalingam et al. 2022; Ecer 2021):

***Step 1:*** *Creating the initial decision matrix:* In the first step of the method, the decision matrix is created according to Eq. (14), which includes optimal values and is organized as follows:

$$X = [x_{ij}]_{m\times n} = \begin{bmatrix} x_{01} & x_{02} & \cdots & x_{0n} \\ x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (14)$$

In the decision matrix, consisting of *m* alternatives and *n* evaluation criteria, $x_{ij}$ represents the value of alternative *i* according to criterion *j*, and $x_{0j}$ represents the optimal value of the evaluation criteria. If the optimal value for the evaluation criteria is unknown, the optimal values according to the benefit and cost characteristics of the criteria are obtained using Eq. (15) and Eq. (16), respectively.

$x_{0j} = max_i x_{ij}, i = 0, 1, \dots, m$ and $j = 1, \dots,$ n  for benefit-based criteria  (15)

$x_{0j} = min_i x_{ij}, i = 0, 1, \dots, m$ and $j = 1, \dots,$ n  for cost-based criteria  (16)

***Step 2:*** *Normalizing decision matrix:* To standardize each evaluation criterion in the decision matrix, the evaluation criteria must be normalized based on their benefit and cost characteristics. Eq. (17) is used to normalize benefit-based criteria, and Eq. (18) is used to normalize cost-based criteria.

$$\bar{x}_{ij} = \frac{x_{ij}}{\sum_{i=0}^{m} x_{ij}} \quad (17)$$

$$\bar{x}_{ij} = \frac{1/x_{ij}}{\sum_{i=0}^{m} 1/x_{ij}} \quad (18)$$

***Step 3:*** *Calculation of the weighted normalized decision matrix:* The SWARA method's importance weights of the evaluation criteria are included in the ARAS method, and a weighted normalized decision matrix is obtained with Eq. (19).

$$x_{ij} = \bar{x}_{ij} \times w_j \quad (19)$$

***Step 4:*** *Calculation of the optimality function values*: The optimality values ($S_i$) of the decision alternatives are calculated using Eq. (20). The decision alternative with the highest $S_i$ value is considered the best alternative.

$$S_i = \sum_{j=1}^{n} x_{ij} \quad (20)$$

***Step 5:*** *Determining the utility values and ranking of decision alternatives:* For each alternative, the utility degree ($K_i$) is calculated using Eq. (21).

$$K_i = \frac{S_i}{S_0} \quad (21)$$

## 4. Case Study

This study focuses on a company operating in the software development sector, aiming to evaluate agile software development methodologies in line with the company's specific needs. The company has been a prominent player in the industry, which has seen rapid growth in recent years driven by the increasing demand for faster, more efficient, and adaptable systems. To maintain its competitive edge, the company faces the challenge of selecting the most suitable agile methodology to navigate the ever-evolving technological landscape and meet the dynamic needs of users. In this study, insights were gathered from four experts, each with over five years of experience in agile methodologies. One expert, an agile professional, specializes in integrating SCRUM, XP, and KANBAN methodologies into large-scale software projects. Another expert, an agile program manager, focuses on performance evaluation and assessing the organizational impacts of agile practices. The third expert, an agile coach, has extensive experience in enhancing team productivity and leading agile transformations. The fourth expert, a senior agile consultant, is renowned for optimizing software development processes across various industries. The expert opinions were used to evaluate the ranking of criteria and the interrelationships among them, with the necessary data collected to support this evaluation.

## 5. Application

The analysis phase of the study was carried out in two stages. In the initial stage, we calculated the importance weights of the criteria by analyzing the rankings provided by experts using the SWARA and DEMATEL methods. During this stage, four experts ranked the 20 CSFs for software development as decision criteria, ranging from the most important to the least important. The order determined by the experts was used to establish the $p_j^k$ values for all criteria. Then, $P_j$ values were calculated for all criteria using Eq. (1). Table 3 presents experts' rankings, $p_j^k$, and $P_j$ values.

Table 3. Experts' rankings, $p_j^k$ and $P_j$ values

| *Criteria* | *Experts* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $P_j$ |
| $SF_1$ Risk minimization | 6 | 1 | 6 | 2 | 0.75 | 1.00 | 0.75 | 0.95 | 0.863 |
| $SF_2$ Technical infrastructure | 7 | 12 | 5 | 11 | 0.70 | 0.45 | 0.80 | 0.50 | 0.613 |
| $SF_3$ The planning requirement | 14 | 13 | 7 | 12 | 0.35 | 0.40 | 0.70 | 0.45 | 0.475 |
| $SF_4$ Communication | 19 | 17 | 12 | 16 | 0.10 | 0.20 | 0.45 | 0.25 | 0.250 |
| $SF_5$ Staff participation | 15 | 20 | 11 | 17 | 0.30 | 0.05 | 0.50 | 0.20 | 0.263 |
| $SF_6$ The qualified HR requirement | 5 | 11 | 4 | 13 | 0.80 | 0.50 | 0.85 | 0.40 | 0.638 |
| $SF_7$ The effective leadership requirement | 4 | 9 | 16 | 10 | 0.85 | 0.60 | 0.25 | 0.55 | 0.563 |
| $SF_8$ Education requirement | 20 | 18 | 20 | 15 | 0.05 | 0.15 | 0.05 | 0.30 | 0.138 |
| $SF_9$ Team/role distribution | 17 | 16 | 19 | 19 | 0.20 | 0.25 | 0.10 | 0.10 | 0.163 |
| $SF_{10}$ Global competitive advantage | 3 | 7 | 10 | 14 | 0.90 | 0.70 | 0.55 | 0.35 | 0.625 |
| $SF_{11}$ Cost advantage | 11 | 8 | 3 | 4 | 0.50 | 0.65 | 0.90 | 0.85 | 0.725 |
| $SF_{12}$ Documentation | 18 | 10 | 13 | 8 | 0.15 | 0.55 | 0.40 | 0.65 | 0.438 |
| $SF_{13}$ Joint management structure | 13 | 14 | 14 | 20 | 0.40 | 0.35 | 0.35 | 0.05 | 0.288 |
| $SF_{14}$ Process improvement possibility | 8 | 2 | 9 | 1 | 0.65 | 0.95 | 0.60 | 1.00 | 0.800 |
| $SF_{15}$ Pilot application possibility | 9 | 3 | 15 | 9 | 0.60 | 0.90 | 0.30 | 0.60 | 0.600 |
| $SF_{16}$ Flexibility | 1 | 6 | 2 | 6 | 1.00 | 0.75 | 0.95 | 0.75 | 0.863 |
| $SF_{17}$ Effective time management | 10 | 5 | 8 | 7 | 0.55 | 0.80 | 0.65 | 0.70 | 0.675 |
| $SF_{18}$ Applicability to complex problems | 2 | 4 | 1 | 3 | 0.95 | 0.85 | 1.00 | 0.90 | 0.925 |
| $SF_{19}$ Customer satisfaction | 12 | 15 | 18 | 5 | 0.45 | 0.30 | 0.15 | 0.80 | 0.425 |
| $SF_{20}$ Customer collaboration | 16 | 19 | 17 | 18 | 0.25 | 0.10 | 0.20 | 0.15 | 0.175 |

After calculating the $P_j$ values, the $s_j$, $k_j$, $q_j$, and $w_j^s$ values were calculated for all criteria, which were then reordered from the largest to the smallest according to these values, using Equations 2-4. The calculated values are shown in Table 4.

Table 4. Calculated $s_j$, $k_j$, $q_j$ and $w_j$ values

| | $P_j$ | $s_j$ | $k_j$ | $q_j$ | $w_j^s$ | | $P_j$ | $s_j$ | $k_j$ | $q_j$ | $w_j^s$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $SF_{18}$ | 0.925 | - | 1.000 | 1.000 | 0.072 | $SF_7$ | 0.563 | 0.037 | 1.037 | 0.703 | 0.050 |
| $SF_1$ | 0.863 | 0.062 | 1.062 | 0.942 | 0.067 | $SF_3$ | 0.475 | 0.088 | 1.088 | 0.646 | 0.046 |
| $SF_{16}$ | 0.863 | 0.000 | 1.000 | 0.942 | 0.067 | $SF_{12}$ | 0.438 | 0.037 | 1.037 | 0.623 | 0.045 |
| $SF_{14}$ | 0.800 | 0.063 | 1.063 | 0.886 | 0.063 | $SF_{19}$ | 0.425 | 0.013 | 1.013 | 0.615 | 0.044 |
| $SF_{11}$ | 0.725 | 0.075 | 1.075 | 0.824 | 0.059 | $SF_{13}$ | 0.288 | 0.137 | 1.137 | 0.541 | 0.039 |
| $SF_{17}$ | 0.675 | 0.050 | 1.050 | 0.785 | 0.056 | $SF_5$ | 0.263 | 0.025 | 1.025 | 0.528 | 0.038 |
| $SF_6$ | 0.638 | 0.037 | 1.037 | 0.757 | 0.054 | $SF_4$ | 0.250 | 0.013 | 1.013 | 0.521 | 0.037 |
| $SF_{10}$ | 0.625 | 0.013 | 1.013 | 0.747 | 0.054 | $SF_{20}$ | 0.175 | 0.075 | 1.075 | 0.484 | 0.035 |
| $SF_2$ | 0.613 | 0.012 | 1.012 | 0.738 | 0.053 | $SF_9$ | 0.163 | 0.012 | 1.012 | 0.479 | 0.034 |
| $SF_{15}$ | 0.600 | 0.013 | 1.013 | 0.729 | 0.052 | $SF_8$ | 0.138 | 0.025 | 1.025 | 0.467 | 0.033 |

After determining the SWARA weights, the calculation process for DEMATEL weights was initiated. Firstly, a direct comparison relation matrix was generated using a 10-point scale, ranging from 0 to 10 as provided in Table 5. The initial direct relation matrix and total relation matrix were generated using Equations 6-8, as presented in Tables 6 and 7, respectively.

Table 5. Comparison matrix

| | $SF_1$ | $SF_2$ | $SF_3$ | $SF_4$ | $SF_5$ | $SF_6$ | $SF_7$ | $SF_8$ | $SF_9$ | $SF_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 0 | 4 | 6 | 3 | 2 | 2 | 3 | 2 | 2 | 3 |
| $SF_2$ | 4 | 0 | 5 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| $SF_3$ | 5 | 4 | 0 | 3 | 4 | 3 | 2 | 4 | 4 | 3 |
| $SF_4$ | 3 | 2 | 3 | 0 | 5 | 4 | 5 | 5 | 6 | 3 |
| $SF_5$ | 2 | 3 | 3 | 6 | 0 | 5 | 4 | 5 | 4 | 3 |
| $SF_6$ | 2 | 2 | 2 | 3 | 4 | 0 | 5 | 6 | 5 | 3 |
| $SF_7$ | 3 | 2 | 3 | 4 | 5 | 5 | 0 | 3 | 5 | 3 |
| $SF_8$ | 2 | 2 | 3 | 4 | 4 | 5 | 5 | 0 | 4 | 3 |
| $SF_9$ | 2 | 2 | 4 | 4 | 5 | 5 | 6 | 4 | 0 | 2 |
| $SF_{10}$ | 4 | 3 | 3 | 2 | 3 | 5 | 3 | 4 | 3 | 0 |
| $SF_{11}$ | 3 | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 6 |
| $SF_{12}$ | 2 | 2 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 3 |
| $SF_{13}$ | 4 | 3 | 3 | 6 | 6 | 5 | 6 | 5 | 6 | 4 |
| $SF_{14}$ | 6 | 5 | 5 | 3 | 3 | 2 | 4 | 5 | 5 | 5 |
| $SF_{15}$ | 6 | 5 | 6 | 3 | 3 | 3 | 4 | 3 | 4 | 4 |
| $SF_{16}$ | 4 | 4 | 5 | 3 | 3 | 5 | 4 | 4 | 4 | 5 |
| $SF_{17}$ | 6 | 4 | 6 | 3 | 3 | 3 | 5 | 4 | 4 | 6 |
| $SF_{18}$ | 6 | 5 | 3 | 3 | 3 | 2 | 2 | 4 | 5 | 5 |
| $SF_{19}$ | 2 | 4 | 5 | 3 | 3 | 3 | 2 | 3 | 3 | 7 |
| $SF_{20}$ | 3 | 3 | 4 | 4 | 2 | 2 | 3 | 4 | 4 | 6 |

| | $SF_{11}$ | $SF_{12}$ | $SF_{13}$ | $SF_{14}$ | $SF_{15}$ | $SF_{16}$ | $SF_{17}$ | $SF_{18}$ | $SF_{19}$ | $SF_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 2 | 2 | 3 | 5 | 6 | 6 | 6 | 6 | 4 | 4 |
| $SF_2$ | 2 | 3 | 2 | 6 | 5 | 6 | 6 | 7 | 4 | 5 |
| $SF_3$ | 3 | 4 | 3 | 5 | 4 | 5 | 6 | 5 | 3 | 3 |
| $SF_4$ | 2 | 2 | 5 | 3 | 2 | 2 | 3 | 2 | 4 | 4 |
| $SF_5$ | 2 | 2 | 6 | 3 | 3 | 2 | 2 | 3 | 4 | 3 |
| $SF_6$ | 3 | 2 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 3 |
| $SF_7$ | 2 | 2 | 4 | 3 | 3 | 2 | 2 | 2 | 3 | 3 |
| $SF_8$ | 2 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 3 | 3 |
| $SF_9$ | 2 | 3 | 5 | 3 | 3 | 2 | 4 | 3 | 2 | 2 |
| $SF_{10}$ | 3 | 3 | 4 | 4 | 3 | 4 | 4 | 3 | 4 | 3 |
| $SF_{11}$ | 0 | 3 | 2 | 2 | 2 | 2 | 3 | 2 | 5 | 5 |
| $SF_{12}$ | 3 | 0 | 4 | 4 | 2 | 2 | 4 | 3 | 3 | 3 |
| $SF_{13}$ | 4 | 3 | 0 | 5 | 4 | 3 | 3 | 3 | 4 | 4 |
| $SF_{14}$ | 3 | 2 | 2 | 0 | 6 | 6 | 6 | 5 | 5 | 5 |
| $SF_{15}$ | 5 | 3 | 3 | 6 | 0 | 5 | 6 | 5 | 3 | 3 |
| $SF_{16}$ | 4 | 3 | 3 | 6 | 5 | 0 | 5 | 6 | 4 | 4 |
| $SF_{17}$ | 6 | 4 | 5 | 6 | 4 | 6 | 0 | 6 | 5 | 5 |
| $SF_{18}$ | 5 | 3 | 3 | 6 | 5 | 7 | 6 | 0 | 4 | 4 |
| $SF_{19}$ | 6 | 3 | 3 | 5 | 4 | 4 | 5 | 5 | 0 | 7 |
| $SF_{20}$ | 5 | 4 | 2 | 4 | 3 | 5 | 4 | 4 | 7 | 0 |

Table 6. Initial direct relation matrix

| | $SF_1$ | $SF_2$ | $SF_3$ | $SF_4$ | $SF_5$ | $SF_6$ | $SF_7$ | $SF_8$ | $SF_9$ | $SF_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 0 | 0.044 | 0.066 | 0.033 | 0.022 | 0.022 | 0.033 | 0.022 | 0.022 | 0.033 |
| $SF_2$ | 0.044 | 0 | 0.055 | 0.022 | 0.033 | 0.022 | 0.033 | 0.022 | 0.033 | 0.022 |
| $SF_3$ | 0.055 | 0.044 | 0 | 0.033 | 0.044 | 0.033 | 0.022 | 0.044 | 0.044 | 0.033 |
| $SF_4$ | 0.033 | 0.022 | 0.033 | 0 | 0.055 | 0.044 | 0.055 | 0.055 | 0.066 | 0.033 |
| $SF_5$ | 0.022 | 0.033 | 0.033 | 0.066 | 0 | 0.055 | 0.044 | 0.055 | 0.044 | 0.033 |
| $SF_6$ | 0.022 | 0.022 | 0.022 | 0.033 | 0.044 | 0 | 0.055 | 0.066 | 0.055 | 0.033 |
| $SF_7$ | 0.033 | 0.022 | 0.033 | 0.044 | 0.055 | 0.055 | 0 | 0.033 | 0.055 | 0.033 |
| $SF_8$ | 0.022 | 0.022 | 0.033 | 0.044 | 0.044 | 0.055 | 0.055 | 0 | 0.044 | 0.033 |
| $SF_9$ | 0.022 | 0.022 | 0.044 | 0.044 | 0.055 | 0.055 | 0.066 | 0.044 | 0 | 0.022 |
| $SF_{10}$ | 0.044 | 0.033 | 0.033 | 0.022 | 0.033 | 0.055 | 0.033 | 0.044 | 0.033 | 0 |
| $SF_{11}$ | 0.033 | 0.022 | 0.033 | 0.022 | 0.022 | 0.022 | 0.022 | 0.022 | 0.022 | 0.066 |
| $SF_{12}$ | 0.022 | 0.022 | 0.044 | 0.033 | 0.033 | 0.022 | 0.022 | 0.022 | 0.022 | 0.033 |
| $SF_{13}$ | 0.044 | 0.033 | 0.033 | 0.066 | 0.066 | 0.055 | 0.066 | 0.055 | 0.066 | 0.044 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_{14}$ | 0.066 | 0.055 | 0.055 | 0.033 | 0.033 | 0.022 | 0.044 | 0.055 | 0.055 | 0.055 |
| $SF_{15}$ | 0.066 | 0.055 | 0.066 | 0.033 | 0.033 | 0.033 | 0.044 | 0.033 | 0.044 | 0.044 |
| $SF_{16}$ | 0.044 | 0.044 | 0.055 | 0.033 | 0.033 | 0.055 | 0.044 | 0.044 | 0.044 | 0.055 |
| $SF_{17}$ | 0.066 | 0.044 | 0.066 | 0.033 | 0.033 | 0.033 | 0.055 | 0.044 | 0.044 | 0.066 |
| $SF_{18}$ | 0.066 | 0.055 | 0.033 | 0.033 | 0.033 | 0.022 | 0,022 | 0.044 | 0.055 | 0.055 |
| $SF_{19}$ | 0.022 | 0.044 | 0.055 | 0.033 | 0.033 | 0.033 | 0,022 | 0.033 | 0.033 | 0.077 |
| $SF_{20}$ | 0.033 | 0.033 | 0.044 | 0.044 | 0.022 | 0.022 | 0,033 | 0.044 | 0.044 | 0.066 |

| | $SF_{11}$ | $SF_{12}$ | $SF_{13}$ | $SF_{14}$ | $SF_{15}$ | $SF_{16}$ | $SF_{17}$ | $SF_{18}$ | $SF_{19}$ | $SF_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 0.022 | 0.022 | 0.033 | 0.055 | 0.066 | 0.066 | 0.066 | 0.066 | 0.044 | 0.044 |
| $SF_2$ | 0.022 | 0.033 | 0.022 | 0.066 | 0.055 | 0.066 | 0.066 | 0.077 | 0.044 | 0.055 |
| $SF_3$ | 0.033 | 0.044 | 0.033 | 0.055 | 0.044 | 0.055 | 0.066 | 0.055 | 0.033 | 0.033 |
| $SF_4$ | 0.022 | 0.022 | 0.055 | 0.033 | 0.022 | 0.022 | 0.033 | 0.022 | 0.044 | 0.044 |
| $SF_5$ | 0.022 | 0.022 | 0.066 | 0.033 | 0.033 | 0.022 | 0.022 | 0.033 | 0.044 | 0.033 |
| $SF_6$ | 0.033 | 0.022 | 0.033 | 0.022 | 0.033 | 0.022 | 0.022 | 0.033 | 0.033 | 0.033 |
| $SF_7$ | 0.022 | 0.022 | 0.044 | 0.033 | 0.033 | 0.022 | 0.022 | 0.022 | 0.033 | 0.033 |
| $SF_8$ | 0.022 | 0.022 | 0.033 | 0.022 | 0.022 | 0.033 | 0.022 | 0.022 | 0.033 | 0.033 |
| $SF_9$ | 0.022 | 0.033 | 0.055 | 0.033 | 0.033 | 0.022 | 0.044 | 0.033 | 0.022 | 0.022 |
| $SF_{10}$ | 0.033 | 0.033 | 0.044 | 0.044 | 0.033 | 0.044 | 0.044 | 0.033 | 0.044 | 0.033 |
| $SF_{11}$ | 0 | 0.033 | 0.022 | 0.022 | 0.022 | 0.022 | 0.033 | 0.022 | 0.055 | 0.055 |
| $SF_{12}$ | 0.033 | 0 | 0.044 | 0.044 | 0.022 | 0.022 | 0.044 | 0.033 | 0.033 | 0.033 |
| $SF_{13}$ | 0.044 | 0.033 | 0 | 0.055 | 0.044 | 0.033 | 0.033 | 0.033 | 0.044 | 0.044 |
| $SF_{14}$ | 0.033 | 0.022 | 0.022 | 0 | 0.066 | 0.066 | 0.066 | 0.055 | 0.055 | 0.055 |
| $SF_{15}$ | 0.055 | 0.033 | 0.033 | 0.066 | 0 | 0.055 | 0.066 | 0.055 | 0.033 | 0.033 |
| $SF_{16}$ | 0.044 | 0.033 | 0.033 | 0.066 | 0.055 | 0 | 0,055 | 0.066 | 0.044 | 0.044 |
| $SF_{17}$ | 0.066 | 0.044 | 0.055 | 0.066 | 0.044 | 0,066 | 0 | 0.066 | 0.055 | 0.055 |
| $SF_{18}$ | 0.055 | 0.033 | 0.033 | 0.066 | 0.055 | 0,077 | 0.066 | 0 | 0.044 | 0.044 |
| $SF_{19}$ | 0.066 | 0.033 | 0.033 | 0.055 | 0.044 | 0,044 | 0.055 | 0.055 | 0 | 0.077 |
| $SF_{20}$ | 0.055 | 0.044 | 0.022 | 0.044 | 0.033 | 0,055 | 0.044 | 0.044 | 0.077 | 0 |

Table 7. Total relation matrix

| | $SF_1$ | $SF_2$ | $SF_3$ | $SF_4$ | $SF_5$ | $SF_6$ | $SF_7$ | $SF_8$ | $SF_9$ | $SF_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 0.145 | 0.170 | 0.217 | 0.159 | 0.152 | 0.151 | 0.171 | 0.164 | 0.172 | 0.187 |
| $SF_2$ | 0.188 | 0.129 | 0.209 | 0.150 | 0.163 | 0.152 | 0.172 | 0.166 | 0.184 | 0.179 |
| $SF_3$ | 0.194 | 0.168 | 0.154 | 0.160 | 0.173 | 0.162 | 0.162 | 0.185 | 0.192 | 0.186 |
| $SF_4$ | 0.151 | 0.128 | 0.163 | 0.113 | 0.169 | 0.158 | 0.176 | 0.178 | 0.194 | 0.164 |
| $SF_5$ | 0.142 | 0.139 | 0.163 | 0.176 | 0.117 | 0.168 | 0.167 | 0.178 | 0.175 | 0.164 |
| $SF_6$ | 0.127 | 0.116 | 0.137 | 0.132 | 0.145 | 0.103 | 0.162 | 0.173 | 0.169 | 0.148 |
| $SF_7$ | 0.141 | 0.119 | 0.151 | 0.145 | 0.158 | 0.158 | 0.113 | 0.147 | 0.172 | 0.152 |
| $SF_8$ | 0.124 | 0.113 | 0.144 | 0.139 | 0.142 | 0.152 | 0.159 | 0.108 | 0.156 | 0.145 |
| $SF_9$ | 0.139 | 0.125 | 0.169 | 0.152 | 0.166 | 0.165 | 0.183 | 0.164 | 0.129 | 0.149 |
| $SF_{10}$ | 0.166 | 0.142 | 0.167 | 0.135 | 0.148 | 0.168 | 0.156 | 0.169 | 0.165 | 0.136 |
| $SF_{11}$ | 0.131 | 0.110 | 0.140 | 0.112 | 0.114 | 0.114 | 0.120 | 0.122 | 0.127 | 0.173 |
| $SF_{12}$ | 0.125 | 0.113 | 0.154 | 0.126 | 0.129 | 0.117 | 0.124 | 0.127 | 0.132 | 0.145 |
| $SF_{13}$ | 0.190 | 0.163 | 0.194 | 0.201 | 0.205 | 0.194 | 0.214 | 0.206 | 0.224 | 0.205 |
| $SF_{14}$ | 0.224 | 0.195 | 0.228 | 0.176 | 0.180 | 0.170 | 0.201 | 0.213 | 0.222 | 0.227 |
| $SF_{15}$ | 0.218 | 0.190 | 0.231 | 0.170 | 0.175 | 0.173 | 0.194 | 0.187 | 0.205 | 0.210 |
| $SF_{16}$ | 0.198 | 0.180 | 0.220 | 0.172 | 0.176 | 0.195 | 0.196 | 0.199 | 0.207 | 0.221 |
| $SF_{17}$ | 0.236 | 0.196 | 0.250 | 0.188 | 0.193 | 0.191 | 0.223 | 0.216 | 0.226 | 0.252 |
| $SF_{18}$ | 0.220 | 0.192 | 0.204 | 0.172 | 0.176 | 0.166 | 0.177 | 0.199 | 0.217 | 0.224 |
| $SF_{19}$ | 0.170 | 0.173 | 0.212 | 0.164 | 0.168 | 0.167 | 0.167 | 0.181 | 0.188 | 0.235 |
| $SF_{20}$ | 0.170 | 0.155 | 0.193 | 0.167 | 0.151 | 0.150 | 0.169 | 0.182 | 0.189 | 0.216 |

| | $SF_{11}$ | $SF_{12}$ | $SF_{13}$ | $SF_{14}$ | $SF_{15}$ | $SF_{16}$ | $SF_{17}$ | $SF_{18}$ | $SF_{19}$ | $SF_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_1$ | 0,154 | 0.130 | 0.161 | 0.217 | 0.205 | 0.217 | 0.225 | 0.217 | 0.191 | 0.189 |
| $SF_2$ | 0,156 | 0.142 | 0.152 | 0.228 | 0.197 | 0.219 | 0.227 | 0.229 | 0.193 | 0.201 |
| $SF_3$ | 0,163 | 0.150 | 0.162 | 0.214 | 0.184 | 0.205 | 0.223 | 0.206 | 0.181 | 0.179 |
| $SF_4$ | 0,133 | 0.114 | 0.167 | 0.169 | 0.142 | 0.149 | 0.167 | 0.150 | 0.170 | 0.168 |
| $SF_5$ | 0,134 | 0.114 | 0.177 | 0.169 | 0.152 | 0.150 | 0.157 | 0.161 | 0.170 | 0.159 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $SF_6$ | 0,130 | 0.104 | 0.133 | 0.142 | 0.137 | 0.135 | 0.141 | 0.145 | 0.145 | 0.143 |
| $SF_7$ | 0,123 | 0.106 | 0.146 | 0.157 | 0.141 | 0.138 | 0.145 | 0.139 | 0.148 | 0.146 |
| $SF_8$ | 0,117 | 0.101 | 0.130 | 0.139 | 0.125 | 0.141 | 0.138 | 0.132 | 0.142 | 0.140 |
| $SF_9$ | 0,130 | 0.122 | 0.164 | 0.166 | 0.149 | 0.146 | 0.173 | 0.157 | 0.146 | 0.144 |
| $SF_{10}$ | 0,147 | 0.127 | 0.157 | 0.184 | 0.156 | 0.175 | 0.182 | 0.166 | 0.173 | 0.161 |
| $SF_{11}$ | 0,094 | 0.109 | 0.114 | 0.136 | 0.120 | 0.129 | 0.145 | 0.129 | 0.159 | 0.157 |
| $SF_{12}$ | 0,128 | 0.079 | 0.138 | 0.160 | 0.124 | 0.132 | 0.159 | 0.142 | 0.141 | 0.139 |
| $SF_{13}$ | 0,179 | 0.146 | 0.141 | 0.221 | 0.190 | 0.190 | 0.199 | 0.191 | 0.200 | 0.198 |
| $SF_{14}$ | 0,181 | 0.145 | 0.169 | 0.185 | 0.223 | 0.236 | 0.245 | 0.227 | 0.221 | 0.219 |
| $SF_{15}$ | 0,194 | 0.150 | 0.173 | 0.239 | 0.155 | 0.219 | 0.238 | 0.220 | 0.194 | 0.192 |
| $SF_{16}$ | 0,186 | 0.150 | 0.174 | 0.239 | 0.207 | 0.167 | 0.228 | 0.229 | 0.205 | 0.203 |
| $SF_{17}$ | 0,222 | 0.175 | 0.210 | 0.260 | 0.215 | 0.249 | 0.197 | 0.249 | 0.235 | 0.232 |
| $SF_{18}$ | 0,197 | 0.152 | 0.175 | 0.242 | 0.209 | 0.242 | 0.241 | 0.171 | 0.208 | 0.205 |
| $SF_{19}$ | 0,200 | 0.146 | 0.167 | 0.221 | 0.188 | 0.201 | 0.220 | 0.211 | 0.157 | 0.226 |
| $SF_{20}$ | 0,182 | 0.149 | 0.150 | 0.201 | 0.169 | 0.201 | 0.200 | 0.191 | 0.219 | 0.146 |

Subsequently, DEMATEL weights $W_j^D$ were determined using Equations 9-12. The DEMATEL results and final weights of the CSFs are provided in Table 8.

Table 8. DEMATEL results and final weights of the CSFs

| | $R$ | $C$ | $R+C$ | $R-C$ | $W_j$ | $W_j^D$ |
|---|---|---|---|---|---|---|
| $SF_1$ | 3.595 | 3.400 | 6.995 | 0.196 | 6.998 | 0.051 |
| $SF_2$ | 3.635 | 3.016 | 6.651 | 0.619 | 6.680 | 0.049 |
| $SF_3$ | 3.603 | 3.700 | 7.303 | -0.097 | 7.304 | 0.053 |
| $SF_4$ | 3.123 | 3.109 | 6.232 | 0.013 | 6.232 | 0.046 |
| $SF_5$ | 3.131 | 3.202 | 6.334 | -0.071 | 6.334 | 0.046 |
| $SF_6$ | 2.766 | 3.173 | 5.940 | -0.407 | 5.954 | 0.043 |
| $SF_7$ | 2.847 | 3.404 | 6.252 | -0.557 | 6.276 | 0.046 |
| $SF_8$ | 2.688 | 3.466 | 6.154 | -0.778 | 6.203 | 0.045 |
| $SF_9$ | 3.039 | 3.646 | 6.684 | -0.607 | 6.712 | 0.049 |
| $SF_{10}$ | 3.178 | 3.716 | 6.894 | -0.538 | 6.915 | 0.050 |
| $SF_{11}$ | 2.554 | 3.150 | 5.704 | -0.596 | 5.735 | 0.042 |
| $SF_{12}$ | 2.633 | 2.611 | 5.244 | 0.022 | 5.244 | 0.038 |
| $SF_{13}$ | 3.852 | 3.159 | 7.011 | 0.693 | 7.045 | 0.051 |
| $SF_{14}$ | 4.087 | 3.888 | 7.975 | 0.199 | 7.977 | 0.058 |
| $SF_{15}$ | 3.930 | 3.388 | 7.318 | 0.542 | 7.338 | 0.054 |
| $SF_{16}$ | 3.954 | 3.643 | 7.597 | 0.311 | 7.603 | 0.056 |
| $SF_{17}$ | 4.413 | 3.850 | 8.263 | 0.563 | 8.282 | 0.060 |
| $SF_{18}$ | 3.991 | 3.663 | 7.654 | 0.328 | 7.661 | 0.056 |
| $SF_{19}$ | 3.764 | 3.598 | 7.362 | 0.166 | 7.364 | 0.054 |
| $SF_{20}$ | 3.547 | 3.548 | 7.095 | -0.001 | 7.095 | 0.052 |

Cause and effect diagram according to the DEMATEL results is provided in Figure 3.

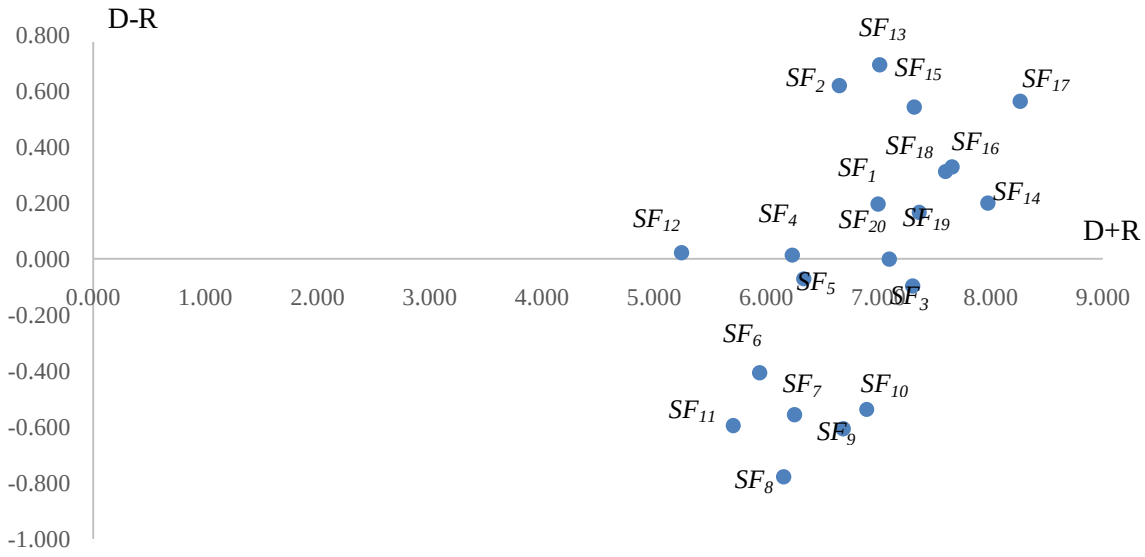


Figure 3. Cause and effect diagram

The significance of CSFs can be evaluated through the ($R$+$C$) values. As shown in Table 8, effective time management stands out as the most crucial factor, boasting the highest ($R$+$C$) value of 8.263, while documentation ranks as the least important factor with the smallest value. A deeper exploration into the cause-and-effect relationships of the CSFs, as revealed by the DEMATEL results, indicates that risk minimization, technical infrastructure, communication, documentation, joint management structure, pilot application possibility, flexibility, effective time management, applicability to complex problems, and customer satisfaction emerge as net causes, as indicated by positive ($R$-$C$) values. In contrast, the planning requirement, staff participation, the qualified HR requirement, the effective leadership requirement, education requirement, team/role distribution, global competitive advantage, cost advantage, process improvement possibility, and customer collaboration are net receivers, signified by their negative ($R$-$C$) values. Finally, leveraging the acquired SWARA $W_j^S$ and DEMATEL $W_j^D$ weights, the final weights for the CSFs $W_j^F$ were computed using Eq. (13), as outlined in Table 9.

Table 9. Final weights of the CSFs

| | $W_j^D$ | $W_j^S$ | $W_j^F$ | | $W_j^D$ | $W_j^S$ | $W_j^F$ |
|---|---|---|---|---|---|---|---|
| $SF_1$ | 0.051 | 0.067 | 0.068 | $SF_{11}$ | 0.042 | 0.059 | 0.049 |
| $SF_2$ | 0.049 | 0.053 | 0.051 | $SF_{12}$ | 0.038 | 0.045 | 0.034 |
| $SF_3$ | 0.053 | 0.046 | 0.049 | $SF_{13}$ | 0.051 | 0.039 | 0.040 |
| $SF_4$ | 0.046 | 0.037 | 0.033 | $SF_{14}$ | 0.058 | 0.063 | 0.073 |
| $SF_5$ | 0.046 | 0.038 | 0.035 | $SF_{15}$ | 0.054 | 0.052 | 0.055 |
| $SF_6$ | 0.043 | 0.054 | 0.047 | $SF_{16}$ | 0.056 | 0.067 | 0.074 |
| $SF_7$ | 0.046 | 0.050 | 0.045 | $SF_{17}$ | 0.060 | 0.056 | 0.067 |
| $SF_8$ | 0.045 | 0.033 | 0.030 | $SF_{18}$ | 0.056 | 0.072 | 0.080 |
| $SF_9$ | 0.049 | 0.034 | 0.033 | $SF_{19}$ | 0.054 | 0.044 | 0.047 |
| $SF_{10}$ | 0.050 | 0.054 | 0.054 | $SF_{20}$ | 0.052 | 0.035 | 0.036 |

In Table 9, the most important CSFs among the 20 considered in the proposed MCDM model for evaluating agile software development methods, based on criterion importance weights obtained through the SWARA and DEMATEL methods, is the "applicability factor to complex problems" with a weight of 8%. Following closely are "flexibility" and "process improvement possibility" with weights of 7.4% and 7.3%, respectively. The least important factor is "education requirements" with a weight of 3%. In the second stage of the analysis phase of this study, the evaluation and ranking of the alternatives according to decision criteria were carried out using the ARAS method. Firstly, the initial decision matrix was created using Eq. (14), and it is represented in Table 10.

Table 10. Initial decision matrix

| *Criteria* | $SF_1$ | $SF_2$ | $SF_3$ | $SF_4$ | $SF_5$ | $SF_6$ | $SF_7$ | $SF_8$ | $SF_9$ | $SF_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| *Criteria type* | *max* | *Max* | *min* | *max* | *max* | *min* | *min* | *min* | *min* | *max* |
| $W_j^F$ *values* | *0.068* | *0.051* | *0.049* | *0.033* | *0.035* | *0.047* | *0.045* | *0.030* | *0.033* | *0.054* |
| $A_0$ *Optimal* | 10 | 9 | 4 | 10 | 10 | 6 | 3 | 4 | 2 | 10 |
| $A_1$ *SCRUM* | 10 | 9 | 6 | 9 | 10 | 6 | 3 | 4 | 8 | 8 |
| $A_2$ *XP* | 7 | 8 | 8 | 9 | 10 | 10 | 5 | 5 | 8 | 9 |
| $A_3$ *KANBAN* | 8 | 6 | 4 | 10 | 6 | 8 | 10 | 7 | 2 | 10 |
| *Criteria* | $SF_{11}$ | $SF_{12}$ | $SF_{13}$ | $SF_{14}$ | $SF_{15}$ | $SF_{16}$ | $SF_{17}$ | $SF_{18}$ | $SF_{19}$ | $SF_{20}$ |
| *Criteria type* | *max* | *Max* | *max* | *max* | *max* | *max* | *max* | *max* | *max* | *max* |
| $W_j^F$ *values* | *0.049* | *0.034* | *0.040* | *0.073* | *0.055* | *0.074* | *0.067* | *0.080* | *0.047* | *0.036* |
| $A_0$ *Optimal* | 9 | 10 | 10 | 10 | 9 | 9 | 7 | 10 | 10 | 10 |
| $A_1$ *SCRUM* | 9 | 10 | 8 | 9 | 7 | 4 | 6 | 10 | 10 | 9 |
| $A_2$ *XP* | 7 | 9 | 7 | 7 | 8 | 5 | 4 | 9 | 7 | 8 |
| $A_3$ *KANBAN* | 6 | 5 | 10 | 10 | 9 | 9 | 7 | 5 | 8 | 10 |

After creating the initial decision matrix, the normalization process was performed using Eq. (17) and Eq. (18) based on whether the criteria were benefit-based or cost-based. The weighted decision matrix was obtained by multiplying the normalized decision matrix with the criteria importance weights determined by the SWARA and DEMATEL methods, as shown in Table 11.

Table 11. Weighted decision matrix

| | $SF_1$ | $SF_2$ | $SF_3$ | $SF_4$ | $SF_5$ | $SF_6$ | $SF_7$ | $SF_8$ | $SF_9$ | $SF_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $A_0$ | 0.0194 | 0.0144 | 0.0154 | 0.0088 | 0.0097 | 0.0139 | 0.0157 | 0.0088 | 0.0132 | 0.0146 |
| $A_1$ | 0.0194 | 0.0144 | 0.0102 | 0.0079 | 0.0097 | 0.0139 | 0.0157 | 0.0088 | 0.0033 | 0.0117 |
| $A_2$ | 0.0136 | 0.0128 | 0.0077 | 0.0079 | 0.0097 | 0.0083 | 0.0094 | 0.0070 | 0.0033 | 0.0132 |
| $A_3$ | 0.0155 | 0.0096 | 0.0154 | 0.0088 | 0.0058 | 0.0104 | 0.0047 | 0.0050 | 0.0132 | 0.0146 |
| | $SF_{11}$ | $SF_{12}$ | $SF_{13}$ | $SF_{14}$ | $SF_{15}$ | $SF_{16}$ | $SF_{17}$ | $SF_{18}$ | $SF_{19}$ | $SF_{20}$ |
| $A_0$ | 0.0142 | 0.0101 | 0.0114 | 0.0202 | 0.0151 | 0.0246 | 0.0196 | 0.0235 | 0.0134 | 0.0097 |
| $A_1$ | 0.0142 | 0.0101 | 0.0091 | 0.0182 | 0.0117 | 0.0109 | 0.0168 | 0.0235 | 0.0134 | 0.0088 |
| $A_2$ | 0.0111 | 0.0091 | 0.0080 | 0.0142 | 0.0134 | 0.0137 | 0.0112 | 0.0212 | 0.0094 | 0.0078 |
| $A_3$ | 0.0095 | 0.0050 | 0.0114 | 0.0202 | 0.0151 | 0.0246 | 0.0196 | 0.0118 | 0.0107 | 0.0097 |

The final stages of the ARAS method involved the calculation of $S_i$ and $K_i$ values. These values, along with the ranking of the alternative agile software development methods, are shown in Table 12.

Table 12. Calculated $S_i$ and $K_i$ values and the ranking of the alternatives

| Criteria | $S_i$ | $K_i$ | *PRIORITY* |
|---|---|---|---|
| $A_0$ Optimal | 0.2957 | 1.0000 | |
| $A_1$ SCRUM | 0.2518 | 0.8514 | **1** |
| $A_2$ XP | 0.2118 | 0.7161 | 3 |
| $A_3$ KANBAN | 0.2407 | 0.8140 | 2 |

For each decision alternative, the optimality function $S_i$ and $K_i$ values, which express the utility degrees, were calculated, as shown in Table 12. A performance ranking was then created for agile software development methods according to the ARAS method. Based on the ranking values in the last column of Table 12, SCRUM is identified as the most suitable method to achieve the 20 CSFs, followed by KANBAN and XP. In addition, important cause-and-effect relationships were identified in the study. Through an in-depth analysis, it was observed that factors such as risk minimization, technical infrastructure, communication, documentation, joint management structure, pilot application possibility, flexibility, effective time management, applicability to complex problems, and customer satisfaction emerged as net causes, as reflected by their positive (R-C) values. On the other hand, criteria such as planning requirements, staff participation, the need for qualified HR, effective leadership, education, team/role distribution, global competitive advantage, cost advantage, process improvement potential, and customer collaboration were identified as net receivers, indicated by their negative (R-C) values.

### *5.1. Comparison of the results using TOPSIS and EDAS*

To enhance the reliability of the findings derived from implementing the ARAS method in this study, both the TOPSIS and EDAS methodologies were employed. The calculations obtained by applying the ARAS, TOPSIS, and EDAS methods are presented in Table 13.

Table 13. Comparison of alternatives' rankings using three different MCDM methods

| | *ARAS METHOD* | | *TOPSIS METHOD* | | *EDAS METHOD* | |
|---|---|---|---|---|---|---|
| Alternatives | *$K_i$ values* | *PRIORITY* | *C values* | *PRIORITY* | *AS values* | *PRIORITY* |
| $A_1$ SCRUM | 0.8514 | 1 | 0.5667 | 1 | 0.7530 | 1 |
| $A_2$ XP | 0.7161 | 3 | 0.4033 | 3 | 0.1201 | 3 |
| $A_3$ KANBAN | 0.8140 | 2 | 0.5180 | 2 | 0.5000 | 3 |

The consistent rankings obtained from the ARAS, TOPSIS, and EDAS methods for alternative selection in agile software development methods enhance the reliability and confidence of decision-makers. This alignment in results provides a robust foundation for

decision-making. If discrepancies had occurred in the ranking orders, decision-makers would have been prompted to conduct a more thorough analysis, scrutinizing the sources of variations to ensure a comprehensive and informed decision.

## 6. Discussion

This study utilized a comprehensive MCDM approach by integrating the SWARA-DEMATEL-ARAS methods to evaluate agile software development methodologies. The unique contribution of this research lies in the seamless integration of these methods, enabling a more nuanced and reliable evaluation process for ranking alternatives based on a well-defined set of criteria. Initially, the SWARA-DEMATEL approach was employed to establish a robust framework of criteria and their interrelationships, which formed the foundation for the subsequent application of the ARAS method. The DEMATEL method played a pivotal role in uncovering the cause-and-effect relationships among the criteria for agile software methodology selection. Through this analysis, cause-and-effect relationships among the CSFs were identified, with factors like risk minimization and flexibility emerging as net causes, while factors such as planning requirements and staff participation were recognized as net receivers.

Compared to previous studies, which often relied on a single MCDM technique like AHP or ANP, this research adopts a combined approach that offers a more comprehensive and reliable evaluation. The comparison of the ARAS, EDAS, and TOPSIS methods ensures a robust analysis, with each method complementing the others to address potential inconsistencies and enhance the credibility of the rankings. The results reveal a strong consistency across the methods, with SCRUM consistently emerging as the most suitable agile methodology, followed by KANBAN and XP. This finding not only aligns with general trends in the literature but also provides added confidence due to the methodological rigor of this multi-method strategy. By combining these techniques, the study minimizes bias and ensures a well-rounded evaluation, offering a holistic perspective on agile methodology selection. Moreover, this approach contributes to the existing body of knowledge by confirming the dominance of SCRUM while showcasing the advantages of integrating multiple MCDM methods to support decision-making in complex scenarios. The consistency in rankings across methods solidifies the results, providing valuable insights for both practitioners and researchers seeking reliable frameworks for evaluating agile methodologies.

## 7. Conclusion

Agile software development methods play a crucial role in software development by providing a flexible and iterative approach. This approach enables teams to respond swiftly to

changing requirements and deliver high-quality software products with increased efficiency and effectiveness. Evaluating these methods is essential for successful project completion and for identifying the most suitable approach to achieving CSFs.

This study adopts a two-stage evaluation process to assess agile software development methods. In the first stage, the relative importance weights of CSFs are determined using the SWARA and DEMATEL methods. The second stage involves ranking the evaluated methods based on their performance in relation to the identified success factors, utilizing the ARAS method. This approach provides a comprehensive assessment of each method's effectiveness in achieving success in software development, helping decision-makers select and implement the most suitable agile methodologies. The study focuses on 20 CSFs essential for evaluating agile software development methods and examines three popular alternatives: SCRUM, XP, and KANBAN. Each method has its own unique strengths and can be tailored to meet the specific needs of a software development project. According to the ranking results, SCRUM emerges as the most suitable method for fulfilling the 20 CSFs. The empirical findings suggest that SCRUM outperforms KANBAN and XP in addressing these success factors. While SCRUM leads the ranking, both KANBAN and XP also demonstrate effectiveness in achieving the identified success factors.

The increasing adoption of hybrid MCDM approaches, such as the integrated SWARA-DEMATEL-ARAS methodology, significantly enhances decision-makers' confidence, particularly when dealing with complex information or challenging problems. This integrated approach provides a robust and reliable evaluation framework, ensuring a comprehensive assessment of agile software development methods. From a managerial perspective, the methodology offers simplicity, efficiency, and an understanding of cause-and-effect relationships, enabling managers to adopt a holistic approach when making strategic decisions related to agile software development. By considering multiple criteria, the method supports well-rounded decision-making, ensuring that all essential aspects are accounted for. Building on these strengths, this study provides practical insights for software development companies and policymakers by offering a structured decision-making framework for selecting agile methodologies. By identifying critical success factors (CSFs) through SWARA and DEMATEL, managers can prioritize key aspects like collaboration and adaptability, enhancing project efficiency. The ARAS method further supports data-driven ranking of SCRUM, XP, and KANBAN, helping organizations mitigate risks and improve stakeholder satisfaction. Additionally, the study highlights the necessity of tailored agile adoption strategies, industry-specific guidelines, and expert consultation to refine development processes. Companies can leverage these insights to develop customized agile frameworks

aligned with organizational goals and industry demands, ensuring more effective agile management across various industries. The findings also emphasize the importance of agile training and the role of experienced professionals in optimizing software development practices, further reinforcing the value of structured, data-driven decision-making in agile environments.

The user-friendly nature of the ARAS method makes it accessible to users at various organizational levels, promoting widespread understanding and acceptance of the results. This, in turn, enhances managerial confidence and trust in the project's success, leading to increased satisfaction among stakeholders, including customers, developers, and managers. Practically, organizations can leverage this integrated methodology to gain more nuanced and insightful assessment of agile methods, thereby improving decision-making in agile project selection and management. The SWARA method, particularly beneficial when handling a large number of criteria or alternatives, facilitates the efficient prioritization of key factors. Additionally, the DEMATEL method’s graphical representation of cause-and-effect relationships helps decision-makers visualize intricate connections within the system, enhancing the practical utility of the methodology. For policymakers, the integrated SWARA-DEMATEL-ARAS approach offers a comprehensive and systematic method for evaluating the effectiveness of agile software development practices across various sectors. It ensures that decisions regarding the adoption of agile methodologies are well-informed, balanced, and consider all relevant criteria, supporting more effective policy design in technology and innovation sectors. This approach also assists in the development of industry standards and best practices, ensuring that agile methodologies are applied in ways that maximize value for organizations and stakeholders. Moreover, it overcomes the limitations of the commonly used AHP method, providing a more adaptable and comprehensive framework for evaluating agile methods.

This study has several limitations. The evaluation is based on a limited set of CSFs, which may not fully capture the diverse needs of different agile projects. Moreover, the subjective nature of expert opinions in the SWARA, DEMATEL, and ARAS methods could introduce bias, affecting the reliability of the results. Additionally, the case study approach used in this research limits the generalizability of the findings. Future research could validate this methodology across different industries to enhance its applicability and robustness. For future studies, it would be valuable to integrate additional MCDM methodologies, such as VIKOR, PROMETHEE, and ELECTRE, to provide a comparative analysis. This could offer a more comprehensive understanding of the advantages and limitations of various MCDM methods in selecting agile software development approaches. Furthermore, integrating agile software

development evaluations with Fuzzy Cognitive Map could enhance the comprehensiveness of assessments by capturing the interrelationships between various factors. Future research could also explore a broader range of agile methodologies, providing a more nuanced understanding of their performance and suitability across different contexts. Incorporating advanced machine learning techniques, such as predictive modeling and clustering, could improve the prediction and analysis of agile software development outcomes by identifying patterns and trends within agile projects. Additionally, research could investigate the scalability and adaptability of agile methodologies in large-scale or complex projects, where unique challenges arise that traditional agile methods may not fully address.

## REFERENCES


Abdulwareth, A. J., & Al-Shargabi, A. A. (2021). Toward a multi-criteria framework for selecting software testing tools. *IEEE Access, 9*, 158872–158891. https://doi.org/10.1109/ACCESS.2021.3128071

Abrahamson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. *VTT Publications, 112*. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

Abusaeed, S., Khan, S. U. R., & Mashkoor, A. (2023). A fuzzy AHP-based approach for prioritization of cost overhead factors in agile software development. *Applied Soft Computing, 133*, 109977. https://doi.org/10.1016/j.asoc.2022.109977

Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management, 28*(1), 7–33. https://doi.org/10.1108/JEIM-08-2013-0060

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software, 137*, 96–113. https://doi.org/10.1016/J.JSS.2017.11.045

Akbar, M. A., Sang, J., Nasrullah, Khan, A. A., Mahmood, S., Qadri, S. F., Hu, H., & Xiang, H. (2019). Success factors influencing requirements change management process in global software development. *Journal of Computer Languages, 51*, 112–130. https://doi.org/10.1016/J.COLA.2018.12.005

Ali, F., Usman, M., Abrar, M. F., Rahman, S. U., Khan, I., & Niazi, B. (2023). Practices of de-motivators in adopting agile software development methods at large scale development teams from management perspective. *IEEE Access, 11*, 130368–130390. https://doi.org/10.1109/ACCESS.2023.3331759

Badi, I., Pamucar, D., Gigović, L., & Tatomirović, S. (2021). Optimal site selection for sitting a solar park using a novel GIS-SWA'TEL model: A case study in Libya. *International Journal of Green Energy, 18*(4), 336–350. https://doi.org/10.1080/15435075.2020.1854264

Barry, E. J., Mukhopadhyay, T., & Slaughter, S. A. (2002). Software project duration and effort: An empirical study. *Information Technology and Management, 3*(1–2), 113–136. http://link.springer.com/article/10.1023/A%3A1013168927238

Beggar, O. E. (2024). IFEJM: New intuitionistic fuzzy expert judgment method for effort estimation in agile software development. *Arabian Journal for Science and Engineering, 49*(3), 2887–2908. https://doi.org/10.1007/s13369-023-07711-1

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64–69. https://doi.org/10.1109/2.976920

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology, 61*, 163–181. https://doi.org/10.1016/J.INFSOF.2015.01.004

Cano, E. L., García-Camús, J. M., Garzás, J., Moguerza, J. M., & Sánchez, N. N. (2021). A scrum-based framework for new product development in the non-software industry. *Journal of Engineering and Technology Management, 61*, 101634. https://doi.org/10.1016/J.JENGTECMAN.2021.101634

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961–971. https://doi.org/10.1016/J.JSS.2007.08.020

Cui, Y., Liu, W., Rani, P., & Alrasheedi, M. (2021). Internet of Things (IoT) adoption barriers for the circular economy using Pythagorean fuzzy SWARA-CoCoSo decision-making approach in the manufacturing sector. *Technological Forecasting and Social Change, 171*, 120951. https://doi.org/10.1016/J.TECHFORE.2021.120951

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software, 119*, 87–108. https://doi.org/10.1016/J.JSS.2016.06.013

Ecer, F. (2021). A consolidated MCDM framework for performance assessment of battery electric vehicles based on ranking strategies. *Renewable and Sustainable Energy Reviews, 143*, 110916. https://doi.org/10.1016/J.RSER.2021.110916

Estrada-Esponda, R. D., López-Benítez, M., Matturro, G., & Osorio-Gómez, J. C. (2024). Selection of software agile practices using analytic hierarchy process. *Heliyon, 10*(1). https://doi.org/10.1016/j.heliyon.2023.e22948

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. http://www.agilemanifesto.org/

Ghenai, C., Albawab, M., & Bettayeb, M. (2020). Sustainability indicators for renewable energy systems using multi-criteria decision-making model and extended SWARA/ARAS hybrid method. *Renewable Energy, 146*, 580–597. https://doi.org/10.1016/j.renene.2019.06.157

Govil, N., & Sharma, A. (2022). Validation of agile methodology as ideal software development process using fuzzy-TOPSIS method. *Advances in Engineering Software, 168*, 103125. https://doi.org/10.1016/j.advengsoft.2022.103125

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology, 85*, 60–70. https://doi.org/10.1016/J.INFSOF.2017.01.007

Hron, M., & Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *Journal of Systems and Software, 183*, 111110. https://doi.org/10.1016/J.JSS.2021.111110

Kausar, M., Mazhar, N., Ishtiaq, M., & Alabrah, A. (2023). Decision making of agile patterns in offshore software development outsourcing: A fuzzy logic-based analysis. *Axioms, 12*(3), 1–19. https://doi.org/10.3390/axioms12030307

Kaviani, M. A., Yazdi, A. K., Ocampo, L., & Kusi-Sarpong, S. (2020). An integrated grey-based multi-criteria decision-making approach for supplier evaluation and selection in the oil and gas industry. *Kybernetes, 49*(2), 406–441. https://doi.org/10.1108/K-05-2018-0265

Keršulienė, V., Zavadskas, E. K., & Turskis, Z. (2010). Selection of rational dispute resolution method by applying new step-wise weight assessment ratio analysis (SWARA). *Journal of Business Economics and Management, 11*(2), 243–258. https://doi.org/10.3846/jbem.2010.12

Khan, A. A., Shameem, M., Kumar, R. R., Hussain, S., & Yan, X. (2019). Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development. *Applied Soft Computing, 83*, 105648. https://doi.org/10.1016/J.ASOC.2019.105648

Khan, A. A., Shameem, M., Nadeem, M., & Akbar, M. A. (2021). Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping. *Applied Soft Computing, 102*, 107090. https://doi.org/10.1016/J.ASOC.2021.107090

Magabaleh, A. A., Ghraibeh, L. L., Audeh, A. Y., Albahri, A. S., Deveci, M., & Antucheviciene, J. (2024). Systematic review of software engineering uses of multi-criteria decision-making methods: Trends, bibliographic analysis, challenges, recommendations, and future directions. *Applied Soft Computing, 163*, 111859. https://doi.org/10.1016/j.asoc.2024.111859

Misra, S. C., Kumar, V., & Kumar, U. (2009a). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869–1890. https://doi.org/10.1016/J.JSS.2009.05.052

Misra, S. C., Kumar, V., & Kumar, U. (2009b). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869–1890. https://doi.org/10.1016/j.jss.2009.05.052

Nasir, M. H. N., & Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. *Scientific Research and Essays, 6*(10), 2174–2186. https://doi.org/10.5897/sre10.1171

Niazi, M. (2015). A comparative study of software process improvement implementation success factors. *Journal of Software: Evolution and Process, 27*, 700–722. https://doi.org/10.1002/smr.1704

Pandey, P., & Litoriya, R. (2020a). Fuzzy AHP based identification model for efficient application development. *Journal of Intelligent & Fuzzy Systems, 38*(3). https://doi.org/10.3233/JIFS-190508

Pandey, P., & Litoriya, R. (2020b). Software process selection system based on multicriteria decision making. *Journal of Software: Evolution and Process, 33*(2). https://doi.org/10.1002/smr.2305

Pinto, J. K., & Slevin, D. P. (1988). Critical success factors across the project life cycle: Definitions and measurement techniques. *Project Management Journal, 19*(3), 67–75. https://www.pmi.org/learning/library/critical-success-factors-project-life-cycle-2131

Pinto, J. K., & Prescott, J. E. (1988). Variations in critical success factors over the stages in the project life cycle. *Journal of Management, 14*(1), 5–18. https://doi.org/10.1177/014920638801400102

Rekik, S., & El Alimi, S. (2023). Unlocking renewable energy potential: A case study of solar and wind site selection in the Kasserine region, Central-Western Tunisia. *Energy Science and Engineering*. https://doi.org/10.1002/ese3.1650

Royce, W. W. (1970). Managing the development of large software systems. In *IEEE WISCON* (pp. 1–9). https://doi.org/10.7551/mitpress/12274.003.0035

Saaty, T. L., & Ozdemir, M. S. (2003). Why the magic number seven plus or minus two. *Mathematical and Computer Modelling, 38*(3–4), 233–244. https://doi.org/10.1016/S0895-7177(03)90083-5

Schwaber, K. (1997). SCRUM development process. In *Business Object Design and Implementation* (pp. 117–134). https://doi.org/10.1007/978-1-4471-0947-1_11

Schwaber, K., & Beedle, M. (n.d.). *Agile software development with Scrum*. Prentice Hall PTR.

Shameem, M., Kumar, R. R., Nadeem, M., & Khan, A. A. (2020). Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Applied Soft Computing Journal, 90*, 106122. https://doi.org/10.1016/j.asoc.2020.106122

Sharma, A., & Bawa, R. K. (2017). A multilevel hybrid approach for selection of agile development method using AHP, PROMETHEE and fuzzy logic. *Structural Integrity and Life, 17*(1), 49–54.

Si, S. L., You, X. Y., Liu, H. C., & Zhang, P. (2018). DEMATEL technique: A systematic review of the state-of-the-art literature on methodologies and applications. *Mathematical Problems in Engineering, 2018*(1). https://doi.org/10.1155/2018/3696457

Silva, V. B. S., Schramm, F., & Damasceno, A. C. (n.d.). A multicriteria approach for selection of agile methodologies in software development projects. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE.

Sivalingam, V., Kumar, P. G., Prabakaran, R., Sun, J., Velraj, R., & Kim, S. C. (2022). An automotive radiator with multi-walled carbon-based nanofluids: A study on heat transfer optimization using MCDM techniques. *Case Studies in Thermal Engineering, 29*, 101724. https://doi.org/10.1016/J.CSITE.2021.101724

Slevin, D. P., & Pinto, J. K. (1987). Balancing strategy and tactics in project implementation. *Sloan Management Review, 29*(1), 33–41.

Srivastava, A., Mehrotra, D., Kapur, P. K., & Aggarwal, A. G. (2020). Analytical evaluation of agile success factors influencing quality in software industry. *International Journal of System Assurance Engineering and Management, 11*(s2), 247–257. https://doi.org/10.1007/s13198-020-00966-z

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D. B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software, 86*(6), 1663–1678. https://doi.org/10.1016/J.JSS.2013.02.027

Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium web companies: A qualitative study. *Information and Software Technology, 54*(5), 479–500. https://doi.org/10.1016/J.INFSOF.2011.12.007

Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and measurement of software process improvement: A systematic literature review. *IEEE Transactions on Software Engineering, 38*(2), 398–424. https://doi.org/10.1109/TSE.2011.26

Weflen, E., MacKenzie, C. A., & Rivero, I. V. (2022). An influence diagram approach to automating lead time estimation in agile kanban project management. *Expert Systems with Applications, 187*, 115866. https://doi.org/10.1016/J.ESWA.2021.115866

Yaghoobi, T. (2018). Prioritizing key success factors of software projects using fuzzy AHP. *Journal of Software: Evolution and Process, 30*(1), 1–11. https://doi.org/10.1002/smr.1891

Yegen, N., & Gül, S. (2023). Çevik proje yönetiminde Scrum takımlarının başarı sınıflandırmasına yönelik bir ÇKKV modeli: AHS bütünleşik TOPSIS-Sort-B. *Journal of Polytechnic, 0900*. https://doi.org/10.2339/politeknik.1172615

Yel, İ., & Baysal, M. E. (2023). An application on the use of fuzzy multi criteria decision making methods for software project development process selection. *Journal of the Faculty of Engineering and Architecture of Gazi University, 38*(4), 2325–2337. https://doi.org/10.17341/gazimmfd.1132638

Zorzetti, M., Signoretti, I., Salerno, L., Marczak, S., & Bastos, R. (2022). Improving agile software development using user-centered design and lean startup. *Information and Software Technology, 141*, 106718. https://doi.org/10.1016/J.INFSOF.2021.106718