

Risk Assessment Model for Software Development Projects (A Case Study for Iranian Messaging Applications)

Ali Shalchi¹ , Darya Yousefi² , Farzam Nemat³ 

1. Department of Operations Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. E-mail: ali.shalchee@ut.ac.ir
2. Corresponding author, Department of Information Technology Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. E-mail: yousefi.darya@ut.ac.ir
3. Department of Production and Operations Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. E-mail: Farzam.nemat@ut.ac.ir

Article Info	ABSTRACT
Article type: Research Article Article history: Received 20 April 2026 Received in revised form 13 May 2026 Accepted 15 September 2026 Published online 1 October 2026 Keywords: software project management, risk management, messaging applications, best-worst method (BWM), VIKOR method	This study examines the risks associated with the development of messaging platforms in Iran, shaped by restrictive governmental policies and investments in national applications. To identify and prioritize software project risks, exploratory and confirmatory factor analysis were applied, followed by the Best–Worst Method (BWM). In addition, six domestic messaging platforms were evaluated using the VIKOR method to compare their relative risk levels. The analysis categorizes project risks into eight key domains: requirements, scheduling and planning, organizational and project management, personnel, user, environmental, technical, and design. Among these, technical risks emerged as the most critical, reflecting the significant challenges faced by developers in ensuring performance, security, and scalability under constrained conditions. By offering a structured framework for assessing and managing risks in software development projects, this research extends the existing literature on technology creation under restrictive environments. The findings provide practical insights for developers, policymakers, and other stakeholders aiming to balance innovation with regulatory constraints, ultimately supporting more resilient and sustainable growth of national digital platforms.

Cite this article: Shalchi, A. & et al, (2026)., Risk Assessment Model for Software Development Projects (A Case Study for Iranian Messaging Applications). *Journal of Engineering Management and Soft Computing*, 12 (4). 29-58.

DOI: <https://doi.org/10.22091/jemsc.2026.15723.1367>



© The Author(s) retains the copyright.

DOI: <https://doi.org/10.22091/jemsc.2026.15723.1367>

Publisher: University of Qom

1) Introduction

Messaging applications have become a central component of everyday digital communication in Iran, offering services that range from personal and organizational messaging to commerce, education, and public service delivery. The strategic importance of these platforms has increased in recent years due to limited access to international services, shifting user expectations, and national efforts to build resilient, secure, and scalable digital infrastructures. Despite significant investments, many domestic messaging platforms struggle with performance, reliability, and user adoption—issues that are closely tied to the risks inherent in software development projects.

Software development projects involve multiple stakeholders whose interests often conflict. In the context of Iranian messaging applications, four key stakeholder groups can be identified. First, software developers face the challenge of building secure and scalable platforms while operating under restricted access to global technologies and component libraries. Second, end users demand reliable services, privacy protection, and user-friendly experiences, yet they often have limited alternatives due to the restricted availability of international messaging apps. Third, policymakers and regulatory bodies impose security requirements, content filtering obligations, and data localization mandates that directly shape platform architecture and development timelines. Fourth, public and private investors fund these national platforms with expectations of return on investment and widespread user adoption.

The interactions among these stakeholder groups generate specific risk patterns. For instance, regulatory demands may conflict with user privacy expectations. Likewise, investor pressure for rapid deployment may clash with developers' need for thorough testing and quality assurance. Existing studies on software development projects emphasize that risk factors such as ambiguous requirements, poor project management, and technical vulnerabilities directly affect project success and product quality (Bugayenko et al., 2023; Chen & Deng, 2023). Research also demonstrates that platform-based communication systems are socio-technical entities influenced not only by internal engineering decisions but also by cultural, regulatory, and geopolitical contexts (Almeida, 2025; Fortuny-Sicart et al., 2024). Furthermore, the literature on communication technologies highlights the challenges of platform trust, privacy expectations, and evolving digital behaviors, particularly in environments shaped by political and regulatory pressures (Masur et al., 2025; Ullah & Xinlei, 2024). Although various models have been proposed for identifying and mitigating software project risks (Filippetto et al., 2021; Mamman et al., 2023), these models predominantly emerge from open and competitive markets. As a result, they insufficiently address contexts where developers must balance innovation with infrastructural constraints, restricted access to global technologies, and complex user dynamics (Adzgauskaitė et al., 2025; Tonellato et al., 2024). This gap leads to a fundamental question: How do the interactions among developers, users, regulators, and investors shape software development risks in Iran's regulated digital ecosystem.

Despite the extensive body of research on software project risk management, a critical gap remains in understanding how risks operate within nationally regulated digital ecosystems, such as Iran's domestic messaging platforms. Existing frameworks rarely integrate political, social, and cultural pressures with technical and organizational risks. This lack of contextualized risk assessment creates a fundamental puzzle: How do multi-layered constraints—technical, organizational,

environmental, and regulatory—simultaneously shape project outcomes in restricted technology environments? Furthermore, there is limited empirical evidence on how different domestic messaging platforms compare in terms of their overall software development risk exposure. Addressing this gap is essential because mismanaged risks in such critical platforms can lead to system unreliability, user distrust, and inefficient public investments.

To address this gap, the present study develops an integrated Software Development Project Risk (SDPR) model tailored to the Iranian digital ecosystem. Using a mixed-method approach—combining exploratory and confirmatory factor analysis with the Best–Worst Method and the VIKOR technique—the study identifies, validates, and prioritizes key risks while benchmarking their intensity across six major domestic messaging platforms. In doing so, the study responds to a set of guiding research questions that shape its analytical direction:

- RQ1: What are the risks and sub-risks of messaging software development projects?
- RQ2: What is the conceptual model of “Software Development Project Risk” for Iranian messaging applications?
- RQ3: What is the relative importance of each identified risk?
- RQ4: What is the overall risk level of each Iranian messaging application?
- RQ5: Which risks are most critical for improving the development of Iranian messaging platforms?

These questions collectively frame the study’s contribution by linking the identified gap to a systematic empirical investigation and a context-specific risk model.

This study is novel in two respects: substantive and methodological. From a substantive perspective, unlike most software risk models derived from open and competitive markets (Chen & Deng, 2023; Filippetto et al., 2021), this study presents a contextualized model for Iran's regulated digital ecosystem, incorporating institutional forces such as state regulation and national security requirements. Additionally, instead of examining a single platform, it evaluates six domestic messaging applications (Eitaa, Rubika, Bale, Soroush, iGap, and Gap) comparatively and introduces the concept of “forced resilience”—the long-term adaptation of domestic platforms to a restrictive technological environment. From a methodological perspective, this study employs a chain-based hybrid approach combining SEM, BWM, and VIKOR as complementary methods. The advantage over single-method studies is threefold: SEM validates constructs before weighting, BWM produces more reliable weights with lower inconsistency than AHP, and VIKOR provides ranking based on proximity to the ideal solution with minimal cumulative error. This analytical chain (validation → weighting → ranking) is applied for the first time in messaging application risk assessment.

The remainder of this paper is structured as follows. Section 2 reviews the literature on software project risks and messaging platform development. Section 3 presents the methodology, including instrument design, data collection, and analytical techniques. Section 4 reports the empirical results derived from factor analyses, risk prioritization, and platform comparison. Section 5 discusses the findings in relation to prior research. Section 6 concludes the paper with theoretical implications, managerial recommendations, and directions for future research.

2) Literature Review

One of the most critical aspects of IT projects is risk. Software project risk in the IT industry pertains to issues related to aspects such as budget and progress during the software development phase, as well as the impact of these issues on the software project. Software project risk can affect the realization of the project plan, and if the risk materializes, it may disrupt project progress, increase costs, or even render the software project unfeasible (Ming et al., 2024; Waqar et al., 2023). The risks of messaging software development projects were identified below.

2.1 Requirements Risk

Requirements risks (RR) encompass four main sub-risks. First, ambiguity in project requirements refers to the lack of clarity and mutual understanding in describing project objectives, leading to multiple interpretations (Chen & Deng, 2023). Second, misunderstandings of project requirements occur when ineffective communication and a lack of shared understanding lead to inaccurate interpretations, resulting in stakeholder dissatisfaction (Hsieh et al., 2018; Mamman et al., 2023; Wang et al., 2025). Third, frequent changes in requirements arise when requirements are continuously altered due to new stakeholder requests, unstable goals, or technical discoveries (Edison et al., 2021). Fourth, poor management during requirement changes becomes a significant risk when there is no structured process to evaluate and implement modifications effectively (Chen & Deng, 2023; Tuunanen et al., 2023).

2.2 Scheduling and Planning Risk

Scheduling and planning risks (SPR) include three sub-risks. Over-idealizations of the software development plan occur when stakeholders hold unrealistic expectations and teams pursue flawless solutions instead of focusing on actual needs and constraints (Ding et al., 2023). Frequent interruptions during development stem from technical issues, management decisions, resource shortages, or requirement changes (Bao et al., 2022; Sangaiah et al., 2018). Uncertainty in the deployment timeline arises when the delivery timeline is not properly determined or has changed multiple times (Adzgauskaitė et al., 2025; Ding et al., 2023).

2.3 Organizational and Project Management Risk

Organizational and project management risks (OPMR) consist of two sub-risks. Mismanagement of resources occurs when personnel, finances, equipment, and time are allocated inefficiently (Tonellato et al., 2024). Insufficient supervision and control happens when project managers fail to adequately supervise processes and progress, and performance is not properly monitored using key performance indicators (Chen & Deng, 2023; Vujović et al., 2020).

2.4 Personnel Risk

Personnel risks (PR) include two sub-risks. Insufficient skills and experience mean that team members lack required expertise, directly hindering task completion and causing delays (Carvalho & Ivanov, 2024). Employee turnover, driven by job dissatisfaction, better opportunities, or personal issues, creates additional risk (Ahmad, 2025; Suárez & Vizcaíno, 2024).

2.5 User Risks

User risks (UR) encompass four sub-risks. Ineffective user collaboration occurs when end users fail to actively participate in development, leading to unmet expectations and lower quality (Casola et al., 2024; Wu et al., 2019). Conflicting user requirements arise when different stakeholder groups have contradictions and inconsistencies (Yasin et al., 2024). Lack of oversight from senior management

results from ineffective communication and unclear role definitions (Tam et al., 2020). Frequent organizational and business environment changes directly affect project performance (Aghabegloo et al., 2024; Song et al., 2024).

2.6 Environmental Risk

Environmental risks (ER) include three sub-risks. High dependency on stakeholder communication creates instability when communication channels are weak (Leong et al., 2023; Wiesche, 2021). Poor oversight of contractors and consultants leads to disruptions (Barros-Justo et al., 2021; Beecham et al., 2021; Jingwen et al., 2025). Conflict between the perspectives of development team and software users arises when expectations and needs do not align (Almeida et al., 2022).

Technical risks in software projects can arise from several forms of mismanagement and technological neglect. Frequent changes in physical location, organizational structure, processes, priorities, or market conditions can disrupt team focus, increase stress among technical staff, and negatively affect the overall quality of software products (Laporte et al., 2023; Roshdieh, 2024). In addition, weak software security practices—such as failing to implement updates and security patches, inadequate encryption, and insufficient security testing—create serious vulnerabilities that threaten system integrity and increase the likelihood of cyberattacks and data breaches (Al-Sharif & Pokharel, 2022; Gouthaman & Sankaranarayanan, 2021; Islam & Storer 2020; Metin et al., 2024; Kumar et al., 2026). Moreover, neglecting modern technological capabilities, such as Artificial Intelligence, Machine Learning, APIs, and Internet of Things, can further expose systems to security threats and reduce system efficiency and interoperability (Aslan et al., 2023; Duarte et al., 2021; Ebert & Louridas, 2023; Kern, 2024). Finally, failure to adopt modern technologies in software projects—often due to resistance to change, lack of knowledge about emerging technologies, or limited financial resources—can lead to technical challenges and hinder innovation and project success (AlSalem et al., 2023; Singal et al., 2022).

2.7 Design Risk

Design risks (DR) in software projects emerge when weaknesses occur in architecture planning, documentation, and overall design attention. Inadequate software architecture design may result from poor requirement analysis, limited technical knowledge and experience, or frequent changes during development, leading to an inefficient architecture that does not align with project or user needs (Li et al., 2022). Since software architecture forms the fundamental structure of a system and defines the relationships between its components, poor architectural decisions can negatively affect scalability, performance, stability, and maintainability, potentially causing significant costs or even project failure (Khan, 2023). Another major design risk is inadequate or incomplete design documentation, where the absence of properly recorded processes, design decisions, and technical details disrupts project continuity, maintenance, and future development (Razzaq & Ghayyur, 2023). Proper documentation acts as a critical reference for development progress and future projects, while its absence increases the likelihood of errors, reduces productivity, and complicates future modifications (de Paula Porto et al., 2021). Furthermore, improper attention during the design phase—such as ignoring stakeholder requirements, lacking specialized expertise, or starting development prematurely due to time pressure—can lead to flawed system designs that negatively affect

performance, scalability, security, and long-term sustainability (Hussain et al., 2024; Zhang et al., 2024).

2.8 The Conceptual Model and Main Hypothesis

The conceptual model of this research, which also includes its main hypothesis, is shown in Figure 1. According to Figure 1, the main hypothesis of the research is presented as follows:

H1: The lower-order latent variables of Requirements Risk (RR), Scheduling and Planning Risk (SPR), Project Organization and Management Risk (OPMR), Personnel Risk (PR), User Risk (UR), Environmental Risk (ER), Technical Risk (TR), and Design Risk (DR) explain the higher-order latent variable of Software Development Project Risk (SDPR).

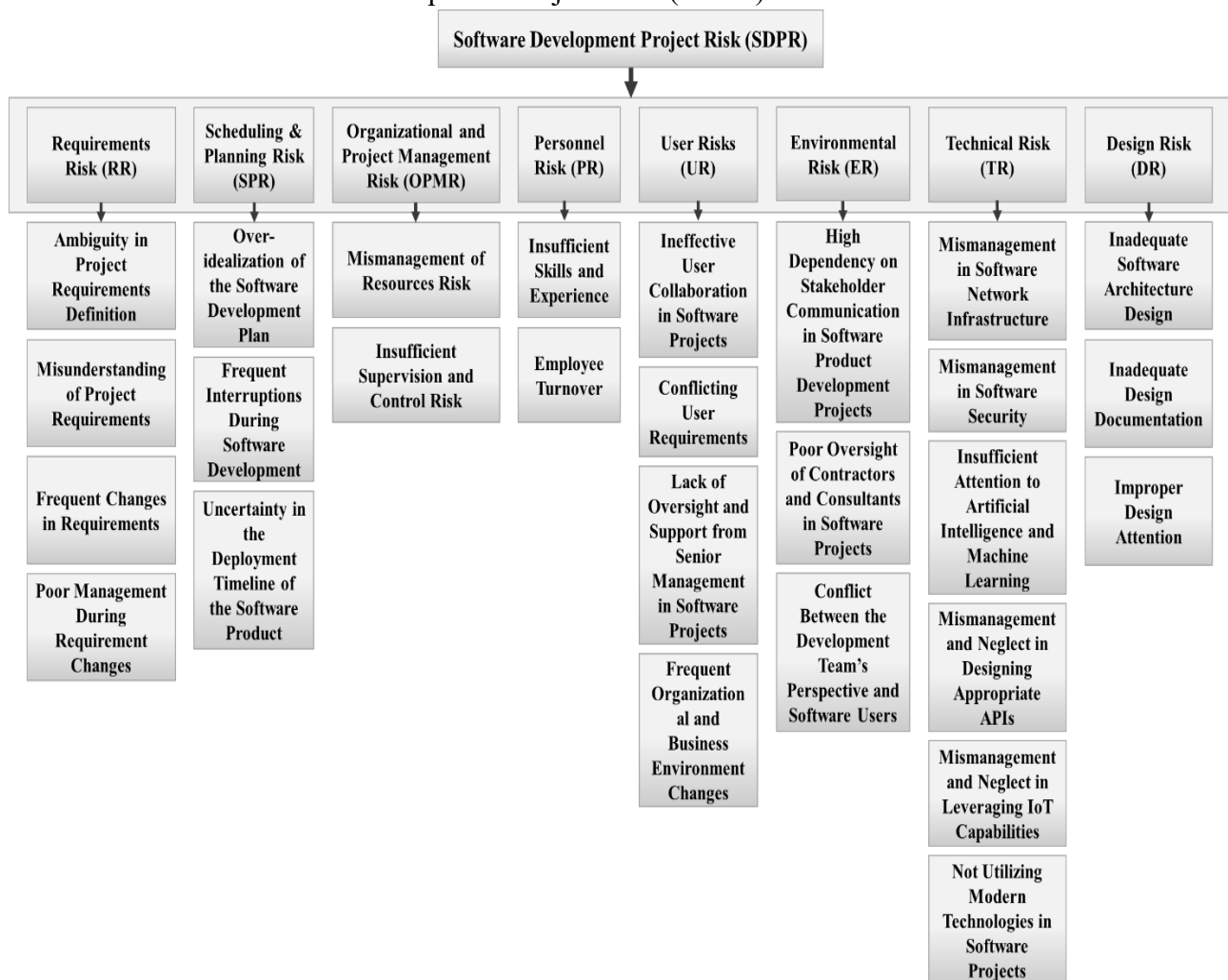


Figure 1. Proposed Research Model

2.9 Summary of Research Background

Table 1 presents a summary of key previous studies on software project risk management. The studies are organized by risk category, key references, research method, and main findings. This table provides readers with a quick overview of the existing literature and highlights the research gap addressed in the present study.

Table 1. Summary of Key Previous Studies on Software Project Risks

Risk Category	Key References	Research Method	Main Findings
Requirements Risks (RR)	Chen & Deng (2023); Mamman et al. (2023); Tuunanen et al. (2023)	Fuzzy logic, Machine learning, Design science	Ambiguity and frequent changes in requirements are major risk factors; poor change management exacerbates project failure
Scheduling & Planning Risks (SPR)	Ding et al. (2023); Bao et al. (2022); Adzgauskaitė et al. (2025)	Optimization models, Case study, Empirical survey	Over-idealization and frequent interruptions lead to timeline uncertainty, especially in remote work settings
Organizational & Project Management Risks (OPMR)	Tonellato et al. (2024); Vujović et al. (2020); Chen & Deng (2023)	Network analysis, Case study, Entropy-based model	Resource mismanagement and insufficient supervision are critical; organizational attention networks affect risk propagation
Personnel Risks (PR)	Carvalho & Ivanov (2024); Ahmad (2025); Suárez & Vizcaíno (2024)	Literature review, Case study, Empirical survey	Lack of skills and employee turnover directly reduce productivity and increase project uncertainty
User Risks (UR)	Casola et al. (2024); Yasin et al. (2024); Tam et al. (2020); Song et al. (2024)	Model-based methodology, Gamification, Empirical survey	Ineffective user collaboration, conflicting requirements, and lack of management support undermine project success
Environmental Risks (ER)	Beecham et al. (2021); Barros-Justo et al. (2021); Almeida et al. (2022)	Empirical study, Systematic review, Case study	Global software development faces environmental disruptions; contractor oversight and stakeholder communication are key challenges
Technical Risks (TR)	Al-Sharif & Pokharel (2022); Metin et al. (2024); Kern (2024); AlSalem et al. (2023); Singal et al. (2022)	Literature review, Risk assessment framework, Case study	Security mismanagement, neglecting modern technologies (AI, IoT, APIs), and poor network infrastructure create critical vulnerabilities
Design Risks (DR)	Khan (2023); Li et al. (2022); Razzaq & Ghayyur (2023); Hussain et al. (2024)	Systematic review, Mapping study, Case study	Inadequate architecture and poor documentation lead to erosion, scalability issues, and long-term maintenance challenges
Research Gap	Present study	SEM + BWM + VIKOR	No prior study has combined construct validation with MCDM ranking in a regulated digital ecosystem context

3) Methodology

This research follows the research onion model proposed by Saunders et al. (2019). According to this model, research design consists of six layers: research philosophy, research approach, methodological choice, research strategy, time horizon, and data collection techniques.

Research philosophy: This study adopts a positivist philosophy. It assumes that software project risks exist objectively and can be measured, quantified, and analyzed through empirical data and statistical methods.

Research approach: A deductive approach is employed. Based on existing theories and literature on software project risk management, hypotheses are formulated and then tested using empirical data collected from experts and users.

Methodological choice: This research uses a mixed-method quantitative design. It combines structural equation modeling (SEM) for construct validation, the Best-Worst Method (BWM) for risk weighting, and the VIKOR technique for the final ranking of messaging platforms.

Research strategy: The study adopts a survey strategy. Data are collected from two groups: software industry experts and users of domestic messaging applications in Tehran.

Time horizon: A cross-sectional time horizon is applied. Data are collected at a single point in time to capture the current state of software development risks in Iranian messaging platforms.

Data collection techniques: A questionnaire is used as the primary data collection tool. The questionnaire includes questions on user behavior and risk assessment using a five-point Likert scale.

In addition, this research is applied based on its objective and descriptive and non-experimental based on its data collection method. Regarding the type of data analysis, it is correlational. Figure 2 shows all the stages of the research process.

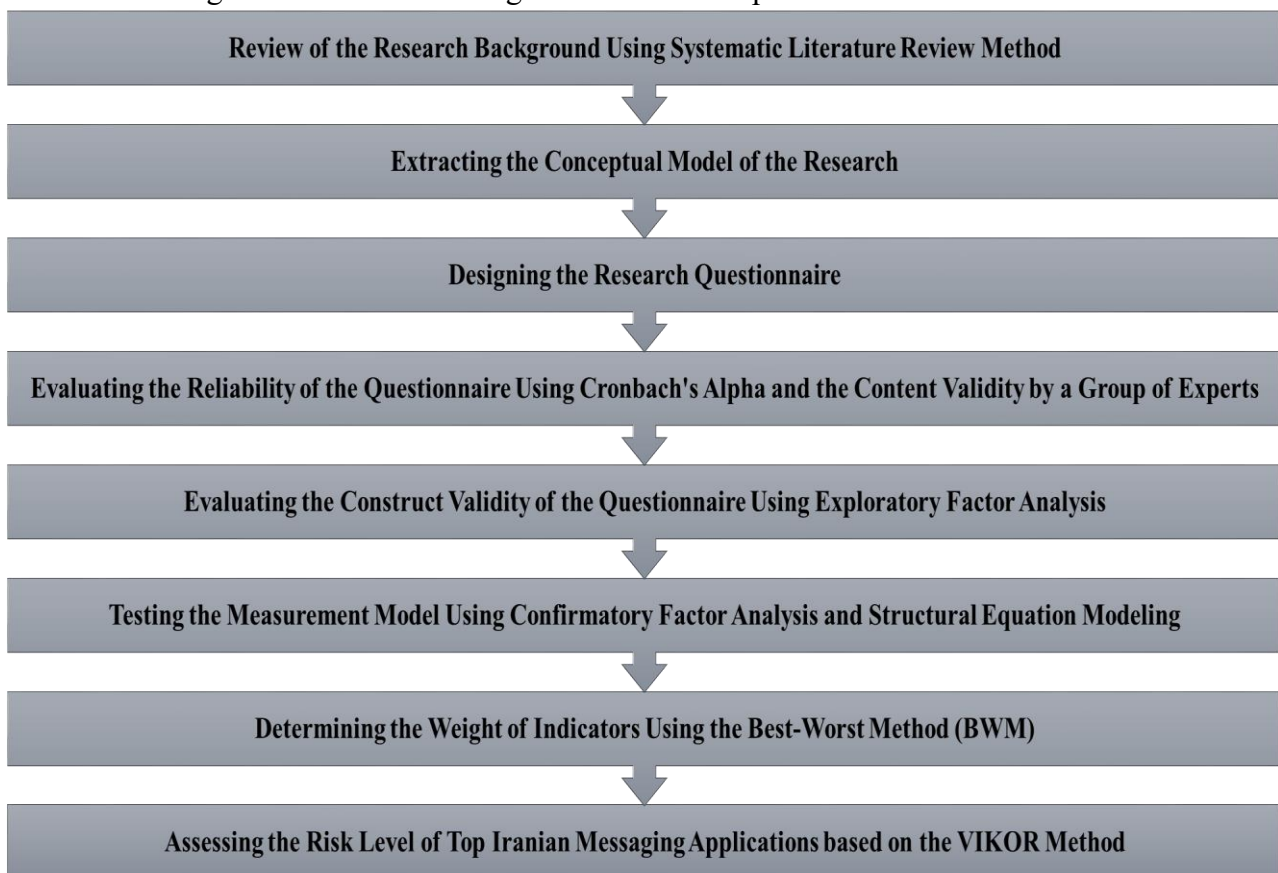


Figure 2. Research Steps

The complementarity of the methods constitutes the first rationale. SEM is employed for construct validation and confirming the validity of the measurement model, whereas BWM is utilized for precise weighting of risks with minimal inconsistency, and VIKOR is applied for the final ranking of alternatives based on their proximity to the ideal solution. Each method offers distinct strengths. Second, regarding increased accuracy and reliability, applying SEM prior to BWM ensures that only constructs with confirmed validity enter the weighting and ranking process, thereby preventing invalid variables from progressing to subsequent decision-making stages. Third, concerning the reduction of inconsistency in expert judgments, the BWM demonstrates a lower inconsistency rate compared to traditional methods such as AHP, and its integration with VIKOR enables ranking with

minimal cumulative error. Fourth, regarding compatibility with the multidimensional nature of risk, software project risk is inherently multidimensional, relying on both expert judgment and empirical data. The selected hybrid approach possesses the capability to simultaneously model these diverse dimensions. Collectively, these clarifications render the research design logic more transparent to the reader and demonstrate that the selection of methods is not merely additive but rather grounded in complementary mechanisms and the pursuit of enhanced precision.

In the first step of this study, the literature on evaluating the risks of software development projects for messaging applications was reviewed, and 44 studies were ultimately selected from a total of 3692 identified. Table 2 shows the databases of the selected studies.

Table 2. Databases of Selected Papers

Database	Frequency
MDPI	7
Elsevier	15
Springer	3
Wiley	6
Emerald	1
ACM Digital Library	2
IEEE	3
INFORMS PubsOnline	1
World Scientific	1
WIT Press	1
Sage Publications	1
Open Science	1
FSJA	1
PLOS One	1

To identify the most relevant articles for the study, a search was conducted for research published between 2018 and 2025, yielding 3692 studies published in English. By reviewing the retrieved articles and applying the relevant criteria, unsuitable studies were excluded, and appropriate articles were selected. The exclusion of irrelevant studies was carried out in three stages. In the first stage, articles with titles unrelated to the research topic were set aside. Subsequently, the abstracts of the remaining studies were reviewed, and articles whose abstracts indicated a lack of relevance to the research topic were also excluded. Finally, the content of the remaining articles was analyzed to identify high-quality studies, while the rest were discarded. The article selection process is illustrated in Figure 3. To identify the most relevant articles for the study, a search was conducted for research published between 2018 and 2025, yielding 3692 studies published in English. By reviewing the retrieved articles and applying the relevant criteria, unsuitable studies were excluded, and appropriate articles were selected. The exclusion of irrelevant studies was carried out in three stages. In the first stage, articles with titles unrelated to the research topic were set aside. Subsequently, the abstracts of the remaining studies were reviewed, and articles whose abstracts indicated a lack of relevance to the research topic were also excluded. Finally, the content of the remaining articles was analyzed to identify high-quality studies, while the rest were discarded. The article selection process is illustrated in Figure 3.

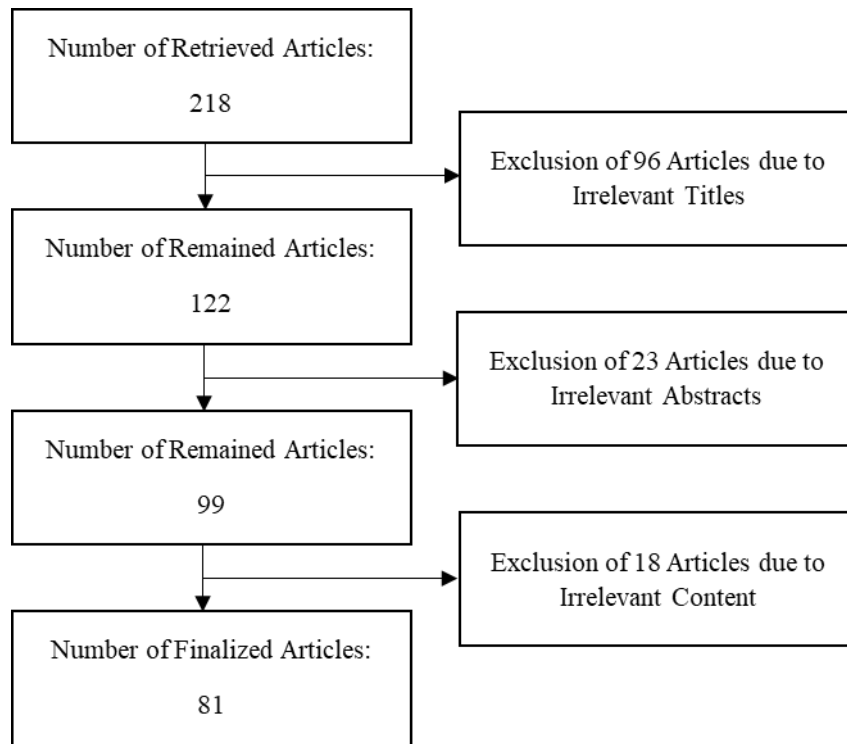


Figure 3. Summary of the Paper Selection Process

In this study, the risks associated with domestic messaging software development projects were categorized into eight groups: requirement risks, scheduling and planning risks, organizational and project management risks, personnel risks, user risks, environmental risks, technical risks, and design risks. Table 3 presents some references related to each of these risks.

Table 3. Identified Risks and Sub-Risks

Risks	Sub-Risks	References
Requirements Risks (RR)	Ambiguity in Project Requirements Definition (RR1)	Chen & Deng (2023)
	Misunderstanding of Project Requirements (RR2)	Wang et al. (2025)
	Frequent Changes in Requirements (RR3)	Edison et al. (2021)
	Poor Management During Requirement Changes (RR4)	Tuunanen et al. (2023)
Scheduling and Planning Risk (SPR)	Over-idealization of the Software Development Plan (SPR1)	Ding et al. (2023)
	Frequent Interruptions During Software Development (SPR2)	Bao et al. (2022)
	Uncertainty in the Deployment Timeline of the Software Product (SPR3)	Adzgauskaitė et al. (2025)
Organizational and Project Management Risk (OPMR)	Mismanagement of Resources Risk (OPMR1)	Tonellato et al. (2024)
	Insufficient Supervision and Control Risk (OPMR2)	Vujović et al. (2020)
Personnel Risk (PR)	Insufficient Skills and Experience (PR1)	Carvalho & Ivanov (2024)
	Employee Turnover (PR2)	Suárez & Vizcaíno (2024)
User Risks (UR)	Ineffective User Collaboration in Software Projects (UR1)	Casola et al. (2024)
	Conflicting User Requirements (UR2)	Yasin et al. (2024)

Table 3. Identified Risks and Sub-Risks

Risks	Sub-Risks	References
	Lack of Oversight and Support from Senior Management in Software Projects (UR3)	Tam et al. (2020)
	Frequent Organizational and Business Environment Changes (UR4)	Song et al. (2024)
Environmental Risk (ER)	High Dependency on Stakeholder Communication in Software Product Development Projects (ER1)	Wiesche (2021)
	Poor Oversight of Contractors and Consultants in Software Projects (ER2)	Jingwen et al. (2025)
	Conflict Between the Development Team's Perspective and Software Users (ER3)	Almeida et al. (2022)
Technical Risk (TR)	Mismanagement in Software Network Infrastructure (TR1)	Laporte et al. (2023)
	Mismanagement in Software Security (TR2)	Al-Sharif & Pokharel (2022)
	Insufficient Attention to Artificial Intelligence and Machine Learning (TR3)	Metin et al. (2024)
	Mismanagement and Neglect in Designing Appropriate APIs (TR4)	Kern (2024)
	Mismanagement and Neglect in Leveraging IoT Capabilities (TR5)	AlSalem et al. (2023)

3.1. Data Gathering Tool

The data required for this study were collected through a questionnaire. The questionnaire was designed in two sections:

- The first section included three questions about user behavior: the type of messaging application used, the duration of use, and the frequency of use.
- The second section comprised 27 questions that assessed the impact of each identified risk using a five-point Likert scale.

3.2. Validity and Reliability of the Questionnaire

In this study, content validity was assessed by presenting the questionnaire to a panel of experts. Initially, the panel consisted of two members with at least 10 years of experience in software project risk management. The panel was subsequently expanded to six members using the snowball sampling method. The experts confirmed the content validity of the questionnaire based on the comprehensiveness of the identified risks. To evaluate the reliability of the questionnaire, the Cronbach's alpha method was employed. The Cronbach's alpha coefficient was calculated to be 0.86. Since this value exceeded 0.70, the questionnaire was considered to have sufficient reliability.

3.3. Statistical Population and Sample Size

The statistical population in this study consisted of two expert groups. The first group consisted of professionals in the software industry who were familiar with software project risks. The second group consisted of users residing in Tehran who used these applications in their daily lives. The rationale for selecting these two groups of experts was their valuable knowledge and experience in the relevant field, which was instrumental in effectively weighing the criteria.

Expert opinions were utilized during the stages of identifying and weighting the criteria. Due to the limited number of experts, the snowball sampling method was employed to determine the

sample size. To assess the risks of domestic messaging application development projects, user feedback was collected. Cochran's formula for an infinite population was then applied to determine the sample size. The sample was randomly distributed among users in Tehran. Table 4 presents the statistical population, including the number of experts and users.

Table 4. The Population and Sample Size in each of the Messaging Applications

Messaging Apps	Eitaa	Bale	Gap	Soroush	iGap	Rubica	Total
Experts	12	4	2	2	5	15	40
Expert Percentage	%30	%10	%5	%5	%12.5	%37.5	%100
User	109	38	10	15	31	148	351
User Percentage	%31	%11	%3	%4	%9	%42	%100

3.4 Structural Equation Modeling

Structural Equation Modeling (SEM) is a comprehensive statistical approach for testing hypotheses regarding relationships among observed and latent variables. A key advantage of SEM is its ability to estimate a complete model that incorporates both measurement and structural components. In this study, the measurement model was evaluated using SEM with the software program LISREL 8.83. Seven fit indices were employed to assess model adequacy, with commonly accepted thresholds for acceptable fit as follows:

- χ^2/df ratio: values less than 3 indicate an acceptable fit
- Root Mean Squared Error of Approximation (RMSEA): values below 0.08 indicate acceptable fit
- Goodness of Fit Index (GFI): values greater than 0.9 indicate good fit
- Adjusted Goodness of Fit Index (AGFI): values greater than 0.8 indicate acceptable fit (Foldnes et al., 2025; Tavakoli et al., 2026).

3.5 Best Worst Method

The Best Worst Method (BWM) consists of five steps for determining the weights. Steps 2 to 4 are conducted using an expert questionnaire, the content of which, along with the characteristics of the respondents, is described after the explanation of all five steps.

3.5.1 Step 1: Determining a Set of Evaluation Criteria

In the first step, a set of evaluation criteria $\{C_1, C_2, \dots, C_n\}$ is considered that should be used to evaluate the alternatives. In our study, these are the LPI indicators.

3.5.2 Step 2: Determining the Best (Most Important) and Worst (Least Important) Criteria

In this step, the evaluator (expert, decision-maker) has to identify the best criterion (e.g. the most important criterion to evaluate the alternatives) and the worst criterion (e.g. the least important criterion to evaluate the alternative) in general.

3.5.3 Step 3: Determining the Preference of the Best Criterion over the Other Criteria

The evaluator then has to indicate the preference of the most important criterion over the other criteria, using a number between 1 and 9, where 1 indicates equal importance and 9 means that the best criterion is a lot more important compared to the criterion in question, resulting in a Best-to-Others vector, A_B :

$$A_B = (a_{B1}, a_{B2}, \dots, a_{Bn}) \quad (1)$$

3.5.4 Step 4: Determining the Preference of the Criteria over the Worst Criterion

The evaluator has to indicate the preference of all the other criteria over the criterion selected as being the least important, using a number between 1 and 9, where 1 indicates equal importance and 9

indicates that the criterion in question is a lot more important than the least important criterion, resulting in an Others-to-Worst vector, A_w :

$$A_w = (a_{1w}, a_{2w}, \dots, a_{nw}) \quad (2)$$

3.5.5 Step 5: Finding the Optimal Weights

In this step, the optimal weights ($w_1, w_2, \dots, w_n$) are identified. Two different models have been proposed for BWM, the former of which (Rezaei, 2015) could lead to multiple optimal solutions, while the latter (Rezaei, 2016) aims at finding unique weights. This linear model was used in this study to determine unique weights.

The set of optimal weights for the linear model is the one where the maximum absolute difference for the following set $\{|w_B - a_{Bj}w_j|, |w_j - a_{jw}w_w|\}$ is minimized. The sum of the weights has to be equal to one and none of the weights can be negative, leading to model (3) to find the optimal solution.

$$\begin{aligned} & \min \max \{|w_B - a_{Bj}w_j|, |w_j - a_{jw}w_w|\} \\ & \text{s.t.} \\ & \sum_{j=1}^n w_j = 1 \\ & w_j \geq 1, \text{ for all } j \end{aligned} \quad (3)$$

This problem can be solved by transferring it to a linear programming problem (4).

$$\begin{aligned} & \min \xi^L \\ & \text{s.t.} \\ & |w_B - a_{Bj}w_j| \leq \xi^L, \text{ for all } j \\ & |w_j - a_{jw}w_w| \leq \xi^L, \text{ for all } j \\ & \sum_{j=1}^n w_j = 1 \\ & w_j \geq 1, \text{ for all } j \end{aligned} \quad (4)$$

Solving this linear programming problem will lead to a single solution in which the optimal weights ($w_1, w_2, \dots, w_n$) and ξ^{L*} are obtained. ξ^{L*} is a direct indicator of the consistency of the comparisons that are made in the method. The value for ξ^{L*} shows the reliability of the outcomes based on the consistency of the comparisons. A value close to zero indicates a high consistency and therefore a high reliability. Complete consistency is reached when $a_{Bj} \times a_{jw} = a_{Bw}$ for all j (Johny et al., 2026).

3.6. VIKOR Method

The VIKOR ranking algorithm has the following steps (Opricovic & Tzeng, 2007; Wan et al., 2026):

3.6.1 Step 1: Determining the Best Value of f_j^* and the Worst Value of f_j^- in the Following Functions:

$$\begin{aligned} f_j^* &= \max_j f_{ij} \\ f_j^- &= \min_i f_{ij} \end{aligned} \quad (5)$$

3.6.2 Step 2: Calculating the Values of S_j and R_j Through the Following Relations:

$$S_i = \sum_{j=1}^n W_j \left(\frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \right)$$

$$R_i = \max_j \left(W_j * \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \right) \quad (6)$$

3.6.3 Step 3: Calculating the Value of Q_j Using the Following Equation:

$$Q_j = V \left(* \frac{(S_{ij} - S^*)}{(S^- - S^*)} \right) + (1 - V) * \left(\frac{(R_{ij} - R^*)}{(R^- - R^*)} \right) \quad (7)$$

in which:

$$R^- = \max_i R_i, S^- = \max_i S_i, R^* = \min_i R_i, S^* = \min_i S_i$$

Additionally, V is introduced as the weight for the group maximum utility strategy, while $(1 - V)$ represents the weight of individual regret.

3.6.4 Step 4: Ranking Q Values in Descending Order:

In this study, to create the decision matrix—one of the inputs for the VIKOR method—a questionnaire was distributed to a group of six experts, each with a minimum of 10 years of experience in software project risk management. Each expert assigned scores to the sub-indicators of risk for Iranian messaging applications, including Rubika, Bale, Eitaa, iGap, Gap, and Soroush.

3.7 Integration of Methods: The Analytical Chain

This study employs a sequential analytical chain consisting of three complementary methods: SEM, BWM, and VIKOR. Each method serves a specific purpose, and the output of one method serves as the validated input for the next. This integration ensures both construct validity and decision-making reliability.

Step 1: Construct Validation using SEM. Before any weighting or ranking takes place, confirmatory factor analysis (CFA) within SEM is applied to the questionnaire data. This step tests whether the observed variables accurately represent the eight latent risk constructs (RR, SPR, OPMR, PR, UR, ER, TR, DR). Only constructs that meet acceptable fit indices ($CFI > 0.90$, $RMSEA < 0.08$) are retained. This prevents invalid or unreliable variables from entering subsequent stages.

Step 2: Risk Weighting using BWM. Once the measurement model is validated, the Best-Worst Method (BWM) is applied to the validated constructs. A separate expert questionnaire is used to compare the importance of each risk category. BWM produces optimal weights with minimal inconsistency compared to traditional methods like AHP. The output is a set of reliable weights ($w_1, w_2, \dots, w_n$) for each risk category and sub-risk.

Step 3: Platform Ranking using VIKOR. Finally, the weights obtained from BWM are fed into the VIKOR method. A decision matrix is constructed based on expert evaluations of six messaging platforms against each sub-risk. VIKOR calculates the Q -index for each platform, representing its distance from the ideal solution. Platforms with lower Q values are considered less risky.

The Analytical Chain: The logical flow is as follows: SEM validates what to measure. Then, BWM determines how important each risk is. Finally, VIKOR ranks platforms based on validated and weighted criteria. This chain approach minimizes cumulative error and ensures that each decision stage builds on valid inputs. Figure 4 illustrates the sequential integration of the three methods.

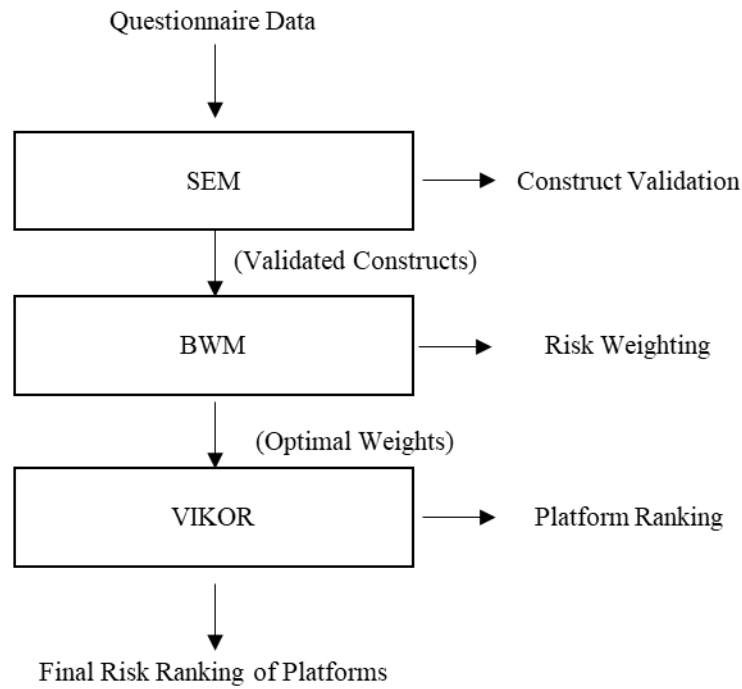


Figure 4. Sequential Integration of SEM, BWM, and VIKOR in the Analytical Chain

4) Findings and Discussion

The data collected from the questionnaires were analyzed using exploratory factor analysis (for construct validity) and confirmatory factor analysis (SEM) techniques (to evaluate the measurement model). SPSS 26 was used for exploratory factor analysis, and LISREL 8.83 was employed for confirmatory factor analysis (SEM). Next, the identified risks and sub-risks were weighted and prioritized using the BWM method. Finally, the risk level of each selected internal messenger was evaluated and ranked using the VIKOR method. The results of these methods are presented below.

4.1. Exploratory Factor Analysis Results

After conducting exploratory factor analysis on the initial questionnaire with 27 variables, the Kaiser-Meyer-Olkin (KMO) test yielded a value of 0.880, which is greater than the recommended threshold of 0.6, indicating the adequacy of the sample size (Table 5). Additionally, Bartlett's test of sphericity was significant ($p < 0.05$), confirming the construct validity of the questionnaire.

Table 5. Rotated Matrix of Components

Questions	Component							
	TT	RR	DR	UR	SPR	OPMR	ER	PR
RR1	0.195	0.783	0.117	0.197	0.099	0.077	0.191	0.13
RR2	0.266	0.749	0.103	0.148	0.284	0.227	0.12	0.079
RR3	0.052	0.663	0.29	0.231	0.072	0.036	0.27	0.226
RR4	0.217	0.802	0.216	0.189	0.026	0.091	0.06	0.116
SPR1	0.382	0.122	0.075	0.117	0.792	0.048	0.109	0.072

Table 5. Rotated Matrix of Components

Questions	Component							
	TT	RR	DR	UR	SPR	OPMR	ER	PR
SPR2	0.209	0.12	0.129	0.176	0.826	0.162	0.117	0.148
SPR3	0.125	0.125	0.237	0.219	0.753	0.256	0.159	0.093
OPMR1	0.213	0.171	0.035	0.198	0.186	0.847	0.117	0.08
OPMR2	0.187	0.112	0.167	0.223	0.18	0.847	0.135	0.079
PR1	0.122	0.134	0.111	0.093	0.199	0.173	0.099	0.811
PR2	0.194	0.27	0.074	0.342	0.025	-0.067	-0.056	0.664
UR1	0.176	0.275	0.24	0.671	0.224	0.21	0.094	0.17
UR2	0.179	0.06	0.073	0.775	-0.009	-0.009	0.21	0.277
UR3	0.101	0.282	0.063	0.716	0.25	0.285	0.017	-0.002
UR4	-0.033	0.208	0.163	0.828	0.195	0.182	0.135	0.082
ER1	0.324	0.183	0.331	0.296	0.137	0.068	0.662	-0.095
ER2	0.168	0.277	0.46	0.101	0.302	0.287	0.604	0.122
ER3	0.093	0.324	0.249	0.191	0.202	0.166	0.717	0.17
TR1	0.655	0.355	0.204	0.166	0.183	0.268	-0.093	0.12
TR2	0.708	0.133	0.009	0.126	0.216	0.003	0.011	0.362
TR3	0.764	0.079	0.158	0.101	0.173	0.12	0.19	-0.071
TR4	0.749	0.255	0.243	0.051	0.175	0.233	0.108	0.088
TR5	0.615	0.127	0.415	-0.008	0.127	0.022	0.126	0.119
TR6	0.579	0.15	0.055	0.155	0.106	0.228	0.312	0.478
DR1	0.237	0.231	0.81	0.136	0.12	0.136	0.123	0.151
DR2	0.311	0.109	0.788	0.2	0.099	-0.03	0.233	-0.109
DR3	0.103	0.231	0.854	0.119	0.167	0.132	0.167	0.16

As shown in Table 4, the items of the first questionnaire collectively represent 8 risks, and the total variance explained was 79.326%. This indicates the structural validity of the questions in this domain. Moreover, the rotated component matrix (Table 5) shows the association of each item with its corresponding factor, confirming the categorization of factors into 8 distinct groups.

4.2. Confirmatory Factor Analysis Results

In this research, confirmatory factor analysis was employed to examine the relationships between latent and observed variables. Figures 5 and 6 present the results of the t-value statistics obtained using LISREL 8.83. As shown in Figure 5, the t-values for each scale were greater than 1.96 at the 5% significance level, confirming and supporting the research hypothesis. The ratio of chi-square to degrees of freedom was 2.7629, below the allowable threshold of 3, and the average squared error was 0.067, below the acceptable limit of 1.0, indicating minimal need for modifications. The P-value was less than 0.05. The Goodness of Fit Index (GFI) for this model was 0.93, exceeding the recommended 0.90, and the Adjusted Goodness of Fit Index (AGFI) was 0.82, above the threshold of 0.80. Figure 6 illustrates the extent to which each variable explains SDRP.

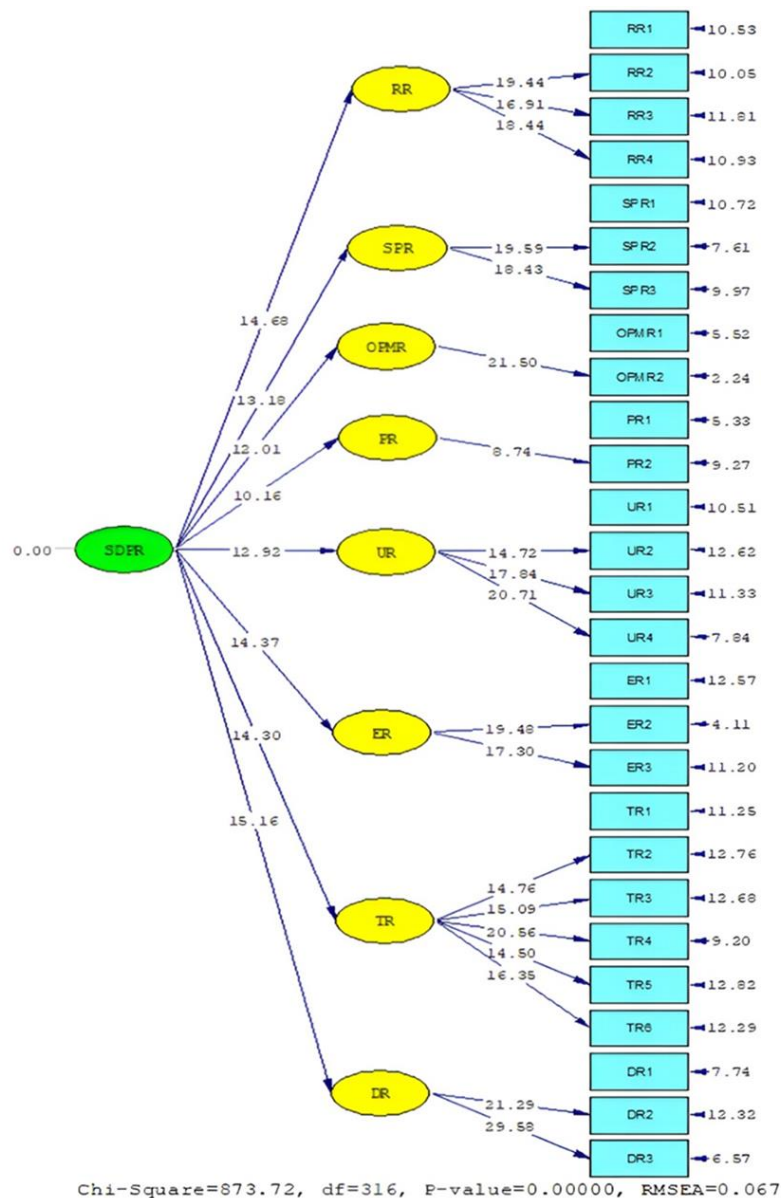


Figure 5. T-Value Model for Software Development Project Risks (SDRP)

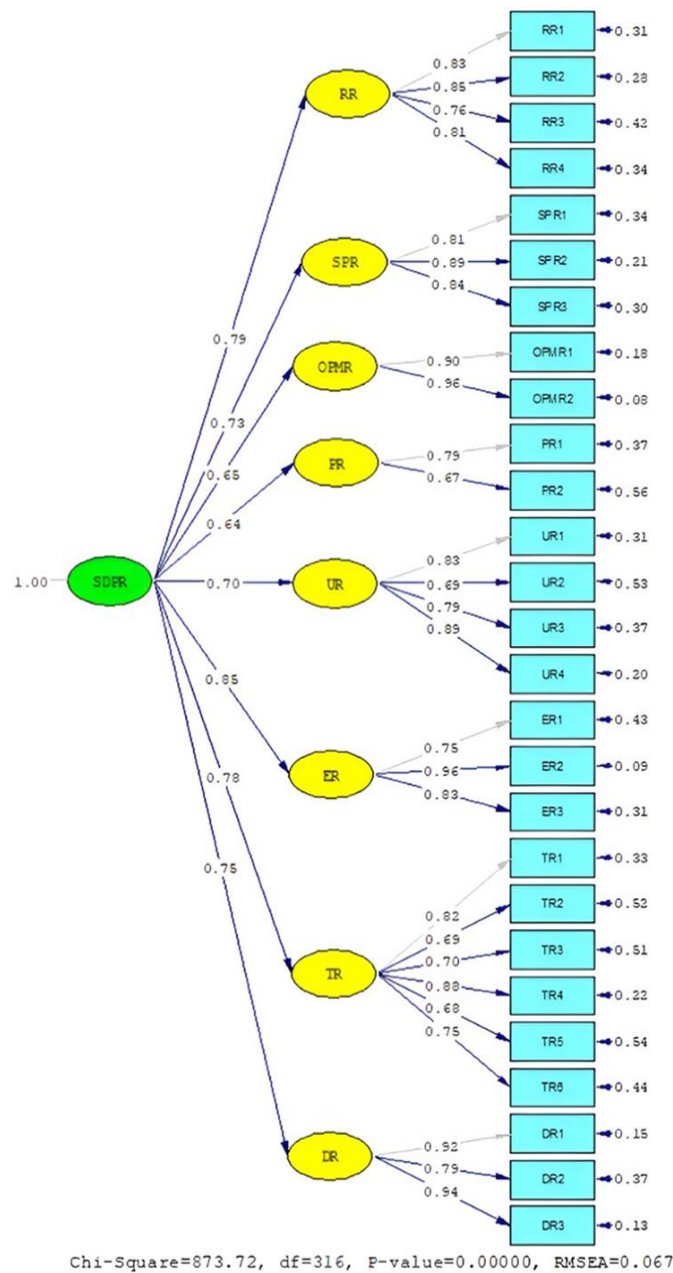


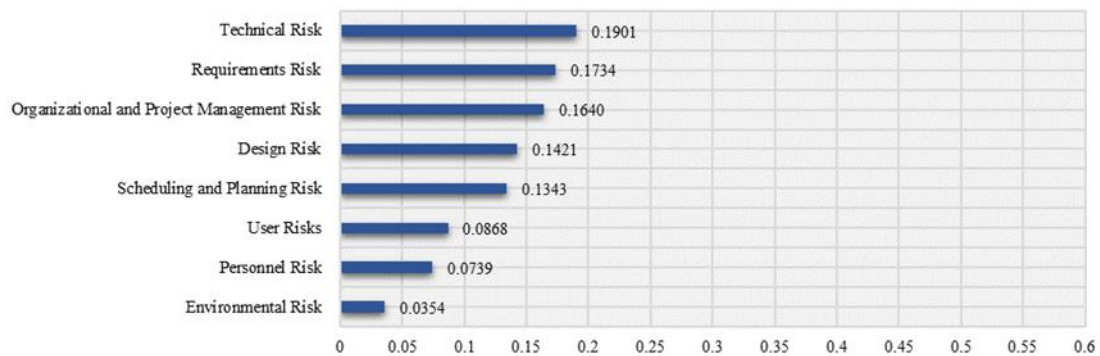
Figure 6. Standardized Solutions Model for Software Development Project Risks (SDRP)

4.3 Best-Worst Method Results

First, the best and worst criteria were determined through expert surveys. Next, experts specified the priority of the best criterion relative to the other criteria, as well as the priority of all criteria relative to the worst criterion, using a scale from 1 to 9. In the final step, the optimal weights were calculated and are presented in Table 6. Figures 7 and 8 illustrate the final weights of risks and sub-risks. Table 7 and Figure 9 present the KSI rate for risks, while Table 8 shows the same measure for sub-risks.

Table 6. Weights of Risks and Sub-Risks

Risk	Weight			Sub-Risk	Sub-risks Weight (local)			Final Weight
	Max	Min	Mean		Max	Min	Mean	
RR	0.2916	0.0899	0.1735	RR1	0.4629	0.2817	0.3196	0.0555
				RR2	0.5100	0.1878	0.3973	0.0689
				RR3	0.0810	0.0392	0.0641	0.0111
				RR4	0.4751	0.1200	0.2188	0.0380
SPR	0.2935	0.0622	0.1343	SPR1	0.3250	0.0769	0.1762	0.0237
				SPR2	0.7625	0.1000	0.4875	0.0654
				SPR3	0.7179	0.1000	0.3362	0.0452
OPMR	0.1867	0.1048	0.1640	OPMR1	0.6500	0.4700	0.5850	0.0960
				OPMR2	0.5300	0.3500	0.4150	0.0680
PR	0.1048	0.0339	0.0740	PR1	0.6500	0.3300	0.5183	0.0385
				PR2	0.6700	0.3500	0.4816	0.0355
UR	0.1244	0.0514	0.0867	UR1	0.2978	0.0862	0.2145	0.0186
				UR2	0.5982	0.0567	0.3974	0.0344
				UR3	0.6034	0.0598	0.2823	0.0245
				UR4	0.1724	0.0618	0.1056	.0092
ER	0.0691	0.0252	0.0353	ER1	0.5681	0.0666	0.2581	0.0093
				ER2	0.3500	0.0909	0.1470	0.0053
				ER3	0.7121	0.3409	0.5947	0.0207
TR	0.3013	0.0576	0.1901	TR1	0.2954	0.1530	0.1907	0.0362
				TR2	0.3814	0.1818	0.2721	0.0519
				TR3	0.1212	0.0678	0.0899	0.0170
				TR4	0.4046	0.1818	0.2993	0.0569
				TR5	0.0402	0.0359	0.0385	0.0073
				TR6	0.1818	0.0809	0.1092	0.0208
DR	0.2824	0.0864	0.1421	DR1	0.6545	0.3409	0.5570	0.0792
				DR2	0.5681	0.2444	0.3425	0.0487
				DR3	0.1428	0.0833	0.1004	0.0142

**Figure 7.** Weights of Risks

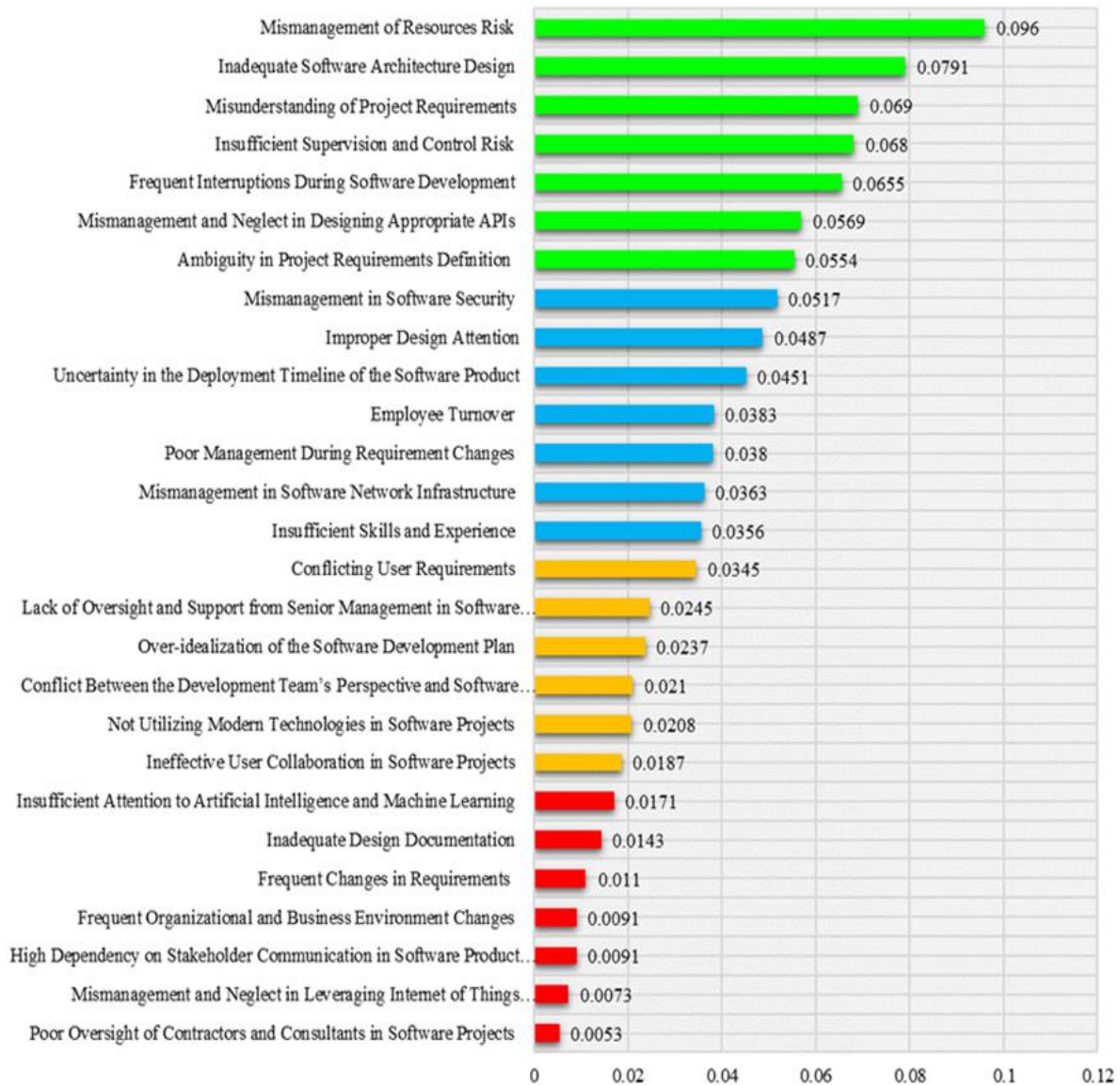


Figure 8. Final Weights of All Sub-Risks

Table 7. KSI Rate for Risks

Risks	KSI					
	E ¹ 1	E2	E3	E4	E5	E6
RR	0.0663	0.0886	0.0542	0.072	0.0564	0.0524
SPR						
OPMR						
PR						
UR						
ER						
TR						
DR						

¹ Expert

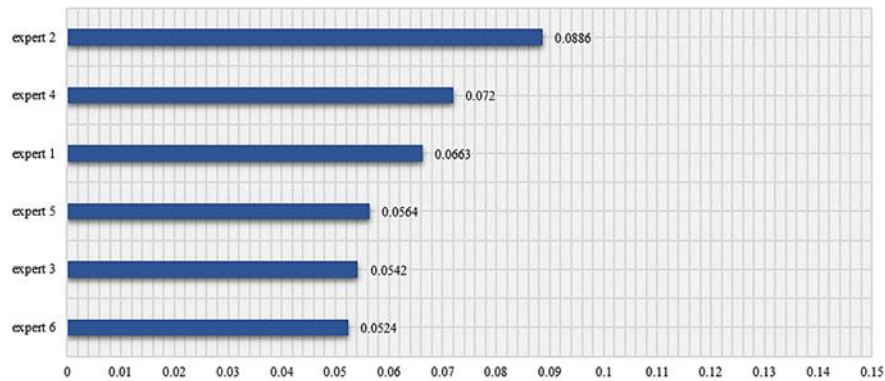


Figure 9. KSI Rate for Risks

Table 8. KSI Rate for Sub-Risks

Risks	Sub-Risks	KSI					
		E1	E2	E3	E4	E5	E6
RR	RR1	0.1216	0.09	0.1176	0.0883	0.0983	0.0925
	RR2						
	RR3						
	RR4						
SPR	SPR1	0.0697	0.0625	0.1025	0.075	0.075	0.109
	SPR2						
	SPR3						
OPMR	OPMR1	0* ¹	0*	0*	0*	0*	0*
	OPMR2						
PR	PR1	0*	0*	0*	0*	0*	0*
	PR2						
UR	UR1	0.0866	0.1196	0.0992	0.0862	0.1007	0.0916
	UR2						
	UR3						
	UR4						
ER	ER1	0.0757	0.06	0.1166	0.1136	0.0888	0.0697
	ER2						
	ER3						
TR	TR1	0.0747	0.0886	0.0809	0.0681	0.0947	0.0935
	TR2						
	TR3						
	TR4						
	TR5						
	TR6						
DR	DR1	0.109	0.1136	0.0888	0.1071	0.0833	0.0833
	DR2						
	DR3						

4.4 VIKOR Method Results

First, to create the decision matrix, a questionnaire was distributed to six experts to obtain the scores of the sub-risk indices for the internal messaging platforms Rubika, Bale, Eitaa, iGap, Gap, and Soroush. The results were then aggregated across all sub-risks and platforms using the arithmetic mean. Subsequently, using the weights and the cumulative decision matrix derived from the average opinions of six experts (Table 9), the VIKOR ranking method was applied.

¹ In pairwise comparisons, when only two indicators are to be compared, the inconsistency rate in a comparison will be zero

Table 9. The Cumulative Decision Matrix

Sub-Risks	Messaging Apps					
	Rubika	Gap	iGap	Eitaa	Bale	Soroush
RR1	3.7975	4.2500	4.2500	3.8843	4.0000	4.2941
RR2	3.9939	4.8333	4.3889	3.8843	4.1905	4.2941
RR3	3.6135	3.7500	4.3056	3.5785	3.5000	3.8235
RR4	3.9816	4.5000	4.0833	3.6198	3.8571	4.3529
SPR1	3.3742	4.2500	4.1944	3.6364	3.4762	3.9412
SPR2	3.6380	4.3333	4.2778	3.7190	3.6190	3.7059
SPR3	3.6748	4.4167	4.1667	3.5124	3.1905	3.8235
OPMR1	3.8834	4.6667	3.9722	3.4132	3.4048	4.3529
OPMR2	3.8405	4.3333	4.0833	3.4545	3.4524	4.0588
PR1	3.4785	3.5833	4.2778	3.6694	3.6905	4.0000
PR2	3.4724	3.4167	4.0278	3.3471	3.2857	4.1176
UR1	3.7485	4.5833	4.3611	3.5289	3.7619	3.9412
UR2	3.3865	3.8333	4.0000	3.3884	3.7143	3.7059
UR3	3.6564	4.4167	4.0833	3.4463	3.9048	3.6471
UR4	3.5399	3.7500	4.1111	3.3884	3.7143	3.9412
ER1	3.3006	3.8333	3.9722	3.2314	3.4762	3.6471
ER2	3.6442	3.8333	4.2222	3.5041	3.5952	3.6471
ER3	3.6074	4.1667	4.1944	3.7190	3.7619	4.0588
TR1	3.7423	4.5000	4.2778	3.7603	3.7143	4.2941
TR2	3.6135	4.7500	4.4167	4.0579	4.1429	4.2941
TR3	3.5215	4.2500	4.3056	3.6942	3.3095	4.1765
TR4	3.6196	4.5000	4.2500	3.7190	3.7857	4.2353
TR5	3.4049	3.2500	4.1111	3.5207	3.4524	4.1765
TR6	3.4969	4.3333	4.2778	3.6281	3.7857	4.1176
DR1	3.4785	4.0833	4.2778	3.5372	3.6905	3.2353
DR2	3.1534	3.5000	4.1389	3.3802	3.3095	3.2353
DR3	3.6074	3.8333	4.3056	3.4876	3.7619	3.4706

After constructing the decision matrix, the ideal values for each sub-criterion were determined. In this study, since all the sub-criteria are non-beneficial, lower values are considered desirable. In other words, lower values improve the performance of the messaging platforms. Consequently, the minimum value is chosen as the desirable value, and the maximum value is selected as the undesirable value. In the next step, the S_i and R_i values associated with each messaging platform were calculated using Equation No. 6, as shown in Table 10.

Table 10. The Values of S_i and R_i for Each Messenger

Messaging Application	S_i	R_i
Rubika	0.1643	0.0363
Gap	0.8437	0.0959
iGap	0.7905	0.0791
Eitaa	0.1221	0.0229
Bale	0.2113	0.0345
Soroush	0.5432	0.0721

Then, the values for S^* , R^* , S^- , and R^- were calculated according to Equation No. 7, and the results are presented in Table 11.

Table 11. The Values of S^* , R^* , S^- , and R^-

S^*	0.1221
R^*	0.0229
S^-	0.8437
R^-	0.0959

In the next step, using the values of S^* , R^* , S^- , and R^- the VIKOR index (Q) for each messenger was calculated using Equation No. 7, as shown in Table 12.

Table 12. The VIKOR Index Value for Each Messaging Application

Q_i	Messaging Application	Value	Rank (Lowest Risk)
Q_1	Rubika	0.1209	2
Q_2	Gap	1	6
Q_3	iGap	0.8480	5
Q_4	Eitaa	0	1
Q_5	Bale	0.1412	3
Q_6	Soroush	0.6286	4

As previously mentioned, the messenger with the lowest Q value was considered the least risky, while the one with the highest Q value was considered the riskiest. More specifically, the evaluated messengers' levels of risk, from lowest to highest, were as follows: Eitaa, Rubika, Baleh, Soroush, iGap, and Gap.

5) Conclusion

The purpose of this study was to identify, prioritize, and compare the risks associated with software development projects in Iranian messaging applications. The findings reveal a multidimensional risk landscape in which technical, requirements-related, and organizational risks dominate, while environmental risks exhibit the lowest criticality. This section synthesizes and interprets these findings in light of the existing literature and clarifies how this study advances scholarly understanding of software development risks in regulated digital ecosystems.

A central finding is the predominance of technical risks, particularly those related to security, infrastructure reliability, and the integration of modern technologies. This aligns with prior studies indicating that technical vulnerabilities are the most immediate and consequential threats to digital communication services (Almeida, 2025; Al-Sharif & Pokharel, 2022; Gouthaman & Sankaranarayanan, 2021). However, the present study extends this line of research by demonstrating that technical risks assume an even greater significance in restrictive environments where developers lack full access to global technological resources and face institutional dependencies. Unlike studies conducted in open or competitive markets (Metin et al., 2024; Singal et al., 2022), our results show that technological stagnation and fragmented infrastructure can amplify risk severity, creating project conditions not fully captured by traditional risk models.

Requirements risks emerged as the second-most critical category, largely due to ambiguity, misinterpretation, and constant changes in user and stakeholder demands. This finding is consistent with previous work highlighting the destabilizing effects of unclear or evolving requirements (Chen & Deng, 2023; Mamman et al., 2023). However, our study advances the literature by showing how requirements risks interact with contextual pressures such as limited user trust, political expectations, and mandated functionality in national platforms. These contextual tensions make requirement management more complex than in settings described in classical software engineering literature (Edison et al., 2021; Filippetto et al., 2021).

Organizational and project management risks were the third-most influential category, reinforcing earlier claims that inadequate supervision and resource mismanagement can derail IT projects (Tonellato et al., 2024; Vujović et al., 2020). Yet, the present findings indicate that these

risks are intensified in Iran's national platform ecosystem, where organizations must manage multiple external stakeholders, comply with state-driven milestones, and rely heavily on subcontractors and consultants. This environment compounds managerial uncertainty in ways not emphasized in prior research on corporate or commercial software environments.

Environmental risks ranked lowest in this study, which contrasts with global software development literature emphasizing environmental disruptions as a major barrier (Barros-Justo et al., 2021; Beecham et al., 2021). This divergence suggests that, in Iran's messaging ecosystem, risk is perceived as more internally generated (technical, organizational, and requirements-based) than externally imposed. This difference highlights a context-specific resilience stemming from the platforms' long-term exposure to constrained technological environments.

One of the most significant contributions of this study is the empirical evidence revealing fundamental differences in risk prioritization patterns between open and regulated digital ecosystems. Previous studies conducted primarily in open and competitive markets of Europe, North America, and East Asia have typically identified environmental risks (ER) and user risks (UR) as the most critical risk categories (Beecham et al., 2021; Chen & Deng, 2023; Filippetto et al., 2021). In contrast, our findings from the Iranian messaging ecosystem demonstrate that technical risks (TR) with a weight of 0.1901 and requirements risks (RR) with a weight of 0.1735 rank first and second, respectively. This significant shift in priorities can be interpreted as a consequence of limited access to cutting-edge technologies (due to software and hardware sanctions), dependency on domestic infrastructure with lower technological maturity, and institutional pressure for rapid compliance with internal security and regulatory requirements (Yalcintas & Alizadeh, 2020). Furthermore, in Iran's regulated environment, additional layers of government regulation, cybersecurity council mandates, state financing of national platforms, and restricted access to international services emerge as influential forces that directly affect the intensity of organizational and project management risks (OPMR) (Tonellato et al., 2024). On the other hand, our findings revealed that environmental risks (ER) received the lowest weight (0.0353) among all eight risk categories. This finding stands in stark contrast to a substantial body of global literature emphasizing the disruptive role of environmental changes (Barros-Justo et al., 2021; Song et al., 2024). This apparent contradiction can be interpreted as evidence of what may be termed "forced resilience" developed by domestic platforms as a result of prolonged exposure to a restrictive yet relatively stable technological environment (in terms of macro-policies). In summary, these findings demonstrate that software development risk models are inherently context-dependent, and the simplistic generalization of findings from open markets to regulated environments can lead to the misallocation of resources and exacerbation of actual project risks.

A novel contribution of this study lies in the benchmarking of risk intensity across six national messaging platforms using VIKOR. Previous research on platform evaluation typically focuses on usability, adoption, or security features (Almeida, 2025; Masur et al., 2025), whereas our analysis offers a comparative lens rooted in structural project risks. The finding that Eitaa exhibits the lowest overall risk, while Gap shows the highest, introduces an evidence-based differentiation that has been largely absent in the literature. This benchmarking provides new empirical insight into how platform-

specific development strategies, technical architecture decisions, and managerial capacity shape risk outcomes.

This study developed a comprehensive risk assessment model for software development projects in Iranian messaging applications by integrating SEM, the Best–Worst Method, and the VIKOR technique. The findings highlight eight major categories of risk and reveal substantial differences in risk levels among national platforms. This section outlines the theoretical implications, managerial and policy implications, and suggestions for future research.

5.1 Comparison of Findings with Previous Studies

The findings of this study confirm, extend, and, in some cases, contradict previous research on software project risks.

Technical Risks (TR): Our finding that technical risks are the most critical category (weight: 0.1901) aligns with Gouthaman and Sankaranarayanan (2021) and Al-Sharif and Pokharel (2022), who identified security and infrastructure vulnerabilities as major threats. However, our study extends these findings by showing that technical risks become even more significant in restrictive environments where access to global technologies is limited. This contrasts with studies in open markets (Metin et al., 2024; Singal et al., 2022), where technical risks ranked lower than environmental or user-related risks.

Requirements Risks (RR): The second-highest priority of requirements risks (weight: 0.1735) is consistent with Chen and Deng (2023) and Mamman et al. (2023), who highlighted ambiguity and frequent changes as destabilizing factors. Our study advances this literature by demonstrating how requirements risks interact with contextual pressures such as political expectations and mandated functionality in national platforms.

Environmental Risks (ER): Our finding that environmental risks received the lowest weight (0.0353) contradicts global studies such as Beecham et al. (2021) and Barros-Justo et al. (2021), which emphasize environmental disruptions as major barriers. This divergence suggests that prolonged exposure to a restrictive yet stable technological environment has led to "forced resilience" in domestic platforms.

Organizational and Project Management Risks (OPMR): The third-highest priority of OPMR (weight: 0.1640) reinforces Tonellato et al. (2024) and Vujović et al. (2020), but our study shows these risks are intensified in regulated ecosystems where organizations must manage multiple external stakeholders and comply with state-driven milestones.

User Risks (UR): The relatively low priority of user risks (weight: 0.0867) contrasts with studies in open markets (Casola et al., 2024; Tam et al., 2020). This may reflect limited user alternatives in Iran's restricted messaging ecosystem, reducing the perceived criticality of user-related risks.

In summary, while our findings align with previous studies on the importance of technical and requirements risks, they diverge on the role of environmental and user risks, highlighting the context-dependent nature of software project risk assessment.

5.2 Theoretical Implications

The study makes several contributions to the theoretical understanding of software development risks in regulated ecosystems. First, it introduces a higher-order Software Development Project Risk (SDPR) model that synthesizes requirements, technical, organizational, personnel, user,

environmental, and design risks into a unified framework. Unlike prior models based primarily on technical or managerial dimensions (e.g., Chen & Deng, 2023; Filippetto et al., 2021), this model incorporates the unique socio-political and infrastructural constraints of national digital platforms.

Second, the study demonstrates that risk prioritization patterns differ significantly between open-market digital ecosystems and nationally regulated ones. In contrast to global findings where environmental or user-related risks often dominate, this study shows that technical and requirements risks are more salient in Iran's constrained context. This highlights the need for theory to better account for political regulation, infrastructural limitations, and institutional dependencies in software risk models.

Third, by employing a multi-method approach—SEM for validation, BWM for prioritization, and VIKOR for comparative evaluation—the study advances methodological rigor in software risk research. It provides a replicable analytical framework that future scholars can apply to other digital ecosystems facing regulatory or infrastructural constraints.

5.3. Managerial or Policy Implications

For software developers and project managers, the results underscore the importance of targeting technical and requirement-related risks as top priorities. Messaging platforms should invest in robust infrastructure, modern security mechanisms, and continuous architectural updates to mitigate vulnerabilities. Given the elevated importance of requirement-related risks, managers must ensure clearer requirement documentation, better communication with stakeholders, and formalized change-management procedures.

For platform operators and executives, the study provides an evidence-based benchmark showing which platforms face the highest and lowest risk exposure. Platforms such as Gap, iGap, and Soroush—identified as having higher risk scores—should allocate resources toward strengthening their technical architecture, improving documentation, and adopting more agile project management practices.

From a policy perspective, the findings highlight the need for more transparent, stable, and coordinated regulatory frameworks. National authorities overseeing digital services should provide consistent development standards, facilitate access to modern technologies, and reduce institutional fragmentation. By doing so, they can help domestic platforms mitigate risks arising from infrastructural and regulatory uncertainty.

5.4. Research Limitations

Despite its efforts to present a comprehensive model and methodologically rigorous approach, this study faces several limitations that should be considered when interpreting the findings. First, the statistical population is limited to users residing in Tehran province. Although Tehran, as Iran's primary metropolis, encompasses a relatively diverse range of users, risk perception and usage patterns in other provinces and cities may differ, and generalizing the results to the entire country requires caution. Second, the research employs a cross-sectional design, with data collected within a specific time frame; consequently, the proposed model cannot capture the dynamic and evolutionary changes in risk profiles over time. Third, the findings are specifically produced within the context of Iran's regulated digital ecosystem. Although this contextual specificity is a strength in terms of localization, generalizing the results to other countries with different levels of regulation,

infrastructure development, or digital culture should be undertaken with great caution. Fourth, risk weighting using the BWM is based on expert judgments, and although this method exhibits a low inconsistency rate, its output is inherently dependent on the knowledge, experience, and composition of the expert panel. Future research could employ hybrid methods, big data, or machine learning approaches to reduce reliance on expert judgment.

5.5 Ideas for Future Research

Several avenues merit further exploration. First, future studies could incorporate longitudinal data to evaluate how risk profiles evolve as platforms mature or as regulatory conditions change. Second, comparative studies between domestic and global messaging platforms could reveal how risk exposure differs in open versus restricted markets, expanding on the context-specific insights provided here. Third, qualitative methods such as in-depth interviews or case studies could complement the quantitative findings by offering richer understanding of managerial behavior, decision-making under constraints, and internal development practices.

Additionally, future research could adapt the SDPR model to other types of critical software such as payment systems, health applications, or government service platforms. Such extensions would test the generalizability of the model and contribute to broader theories of digital resilience in emerging economies.

References

- Adzgauskaitė, M., Tam, C., & Martins, R. (2025). What helps Agile remote teams to be successful in developing software? Empirical evidence. *Information and Software Technology*, 177(1), Article 107593. <https://doi.org/10.1016/j.infsof.2024.107593>
- Aghabegloo, M., Rezaie, K., Torabi, S. A., & Yazdani, M. (2024). Integrating business impact analysis and risk assessment for physical asset criticality analysis: A framework for sustainable operations in process industries. *Expert Systems with Applications*, 241(1), Article 122737. <https://doi.org/10.1016/j.eswa.2023.122737>
- Ahmad, M. O. (2025). Effective handling of large scale agile secure solution development teams. *Journal of Decision Systems*, 34(1), Article 2458875. <https://doi.org/10.1080/12460125.2025.2458875>
- Almeida, F. (2025). Implementation of a chatbot in a unified communication channel. *Journal of Systems and Information Technology*, 27(1), 94–115. <https://doi.org/10.3390/fi14020063>
- AlSalem, T. S., Almaiah, M. A., & Lutfi, A. (2023). Cybersecurity risk analysis in the IoT: A systematic review. *Electronics*, 12(18), Article 3958. <https://doi.org/10.3390/electronics12183958>
- Al-Sharif, R., & Pokharel, S. (2022). Smart city dimensions and associated risks: Review of literature. *Sustainable Cities and Society*, 77(19), Article 103542. <https://doi.org/10.1016/j.scs.2021.103542>
- Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), Article 1333. <https://doi.org/10.3390/electronics12061333>
- Bao, L., Li, T., Xia, X., Zhu, K., Li, H., & Yang, X. (2022). How does working from home affect developer productivity? A case study of Baidu during the COVID-19 pandemic. *Science China Information Sciences*, 65(4), Article 142102. <https://doi.org/10.1007/s11432-020-3278-4>
- Barros-Justo, J. L., Benitti, F. B., & Moller, J. S. (2021). Risks and risk mitigation in global software development: An update. *Journal of Software: Evolution and Process*, 33(11), Article e2370. <https://doi.org/10.1002/smr.2370>
- Beecham, S., Clear, T., Lal, R., & Noll, J. (2021). Do scaling agile frameworks address global software development risks? An empirical study. *Journal of Systems and Software*, 171(1), Article 110823. <https://doi.org/10.1016/j.jss.2020.110823>

- Bugayenko, Y., Bakare, A., Cheverda, A., Farina, M., Kruglov, A., Plaksin, Y., & Succi, G. (2023). Prioritizing tasks in software development: A systematic literature review. *PLOS ONE*, 18(4), Article e0283838. <https://doi.org/10.1371/journal.pone.0283838>
- Carvalho, I., & Ivanov, S. (2024). ChatGPT for tourism: Applications, benefits and risks. *Tourism Review*, 79(2), 290–303. <https://doi.org/10.1108/TR-02-2023-0088>
- Casola, V., De Benedictis, A., Mazzocca, C., & Orbinato, V. (2024). Secure software development and testing: A model-based methodology. *Computers & Security*, 137(2), Article 103639. <https://doi.org/10.1016/j.cose.2023.103639>
- Chen, X., & Deng, Y. (2023). A new belief entropy and its application in software risk analysis. *International Journal of Computers Communications & Control*, 18(2), Article 5299. <https://doi.org/10.15837/ijccc.2023.2.5299>
- de Paula Porto, D., de Jesus, G. M., Ferrari, F. C., & Fabbri, S. C. P. F. (2021). Initiatives and challenges of using gamification in software engineering: A systematic mapping. *Journal of Systems and Software*, 173(9), Article 110870. <https://doi.org/10.1016/j.jss.2020.110870>
- Ding, H., Zhuang, C., & Liu, J. (2023). Extensions of the resource-constrained project scheduling problem. *Automation in Construction*, 153, Article 104958. <https://doi.org/10.1016/j.autcon.2023.104958>
- Duarte, B. B., de Almeida Falbo, R., Guizzardi, G., Guizzardi, R., & Souza, V. E. S. (2021). An ontological analysis of software system anomalies and their associated risks. *Data & Knowledge Engineering*, 134(1), Article 101892. <https://doi.org/10.1016/j.datak.2021.101892>
- Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, 40(4), 30–38. <https://doi.org/10.1109/MS.2023.3265877>
- Edison, H., Wang, X., & Conboy, K. (2021). Comparing methods for large-scale agile software development: A systematic literature review. *IEEE Transactions on Software Engineering*, 48(8), 2709–2731. <https://doi.org/10.1109/TSE.2021.3069039>
- Filippetto, A. S., Lima, R., & Barbosa, J. L. V. (2021). A risk prediction model for software project management based on similarity analysis of context histories. *Information and Software Technology*, 131(1), Article 106497. <https://doi.org/10.1016/j.infsof.2020.106497>
- Foldnes, N., Moss, J., & Grønneberg, S. (2025). Improved goodness of fit procedures for structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, 32(1), 1–13. <https://doi.org/10.1080/10705511.2024.2372028>
- Fortuny-Sicart, A., Pansera, M., & Lloveras, J. (2024). Directing innovation through confrontation and democratisation: The case of platform cooperativism. *Journal of Responsible Innovation*, 11(1), Article 2414512. <https://doi.org/10.1080/23299460.2024.2414512>
- Gouthaman, P., & Sankaranarayanan, S. (2021). Prediction of risk percentage in software projects by training machine learning classifiers. *Computers & Electrical Engineering*, 94(2), Article 107362. <https://doi.org/10.1016/j.compeleceng.2021.107362>
- Hsieh, M. Y., Hsu, Y. C., & Lin, C. T. (2018). Risk assessment in new software development projects at the front end: A fuzzy logic approach. *Journal of Ambient Intelligence and Humanized Computing*, 9(2), 295–305. <https://doi.org/10.1007/s12652-016-0372-5>
- Hussain, S., Anwaar, H., Sultan, K., Mahmud, U., Farooqui, S., Karamat, T., & Toure, I. K. (2024). Mitigating software vulnerabilities through secure software development with a policy-driven waterfall model. *Journal of Engineering*, 2024(1), Article 9962691. <https://doi.org/10.1155/2024/9962691>
- Islam, G., & Storer, T. (2020). A case study of agile software development for safety-critical systems projects. *Reliability Engineering & System Safety*, 200(10), Article 106954. <https://doi.org/10.1016/j.ress.2020.106954>
- Jingwen, Y., Rahman, A. A., Tong, T., Kamarulzaman, N. H., & Sidek, S. B. (2025). A study of Chinese enterprises' business models to determine the impact of dynamic capabilities on innovation and performance. *PLOS ONE*, 20(1), Article e0310854. <https://doi.org/10.1371/journal.pone.0310854>
- Johny, D., Padhi, S. S., & Cheng, T. C. E. (2026). Drone selection for disaster responses: An application of the stratified-best-worst method. *Management Decision*, 64(1), 333–374. <https://doi.org/10.1108/MD-07-2024-1658>
- Kern, C. (2024). Developer ecosystems for software safety. *Communications of the ACM*, 67(6), 52–60. <https://doi.org/10.1145/3651621>

- Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A. M., & Luo, B. (2023). SQL and NoSQL database software architecture performance analysis and assessments—A systematic literature review. *Big Data and Cognitive Computing*, 7(2), Article 97. <https://doi.org/10.3390/bdcc7020097>
- Kumar, D., Hamidi, P., Rana, S. K., Rana, A. K., & Ansarullah, S. I. (2026, April). A Unified Security Framework for SDN-Enabled 5G Core Networks with Secure Network Slicing Enforcement. In *2026 IEEE 15th International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 7-12). IEEE. <https://doi.org/10.1109/CSNT69054.2026.11502394>
- Laporte, C. Y., Verret, G., & Muñoz, M. (2023). A software project that partially failed: A small organization that ignored the management and technical practices of software standards. *Computer*, 56(5), 138–144. <https://doi.org/10.1109/MC.2023.3253979>
- Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L., & Ramasamy, R. K. (2023). Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. *Sustainability*, 15(2), Article 1121. <https://doi.org/10.3390/su15021121>
- Li, R., Liang, P., Soliman, M., & Avgeriou, P. (2022). Understanding software architecture erosion: A systematic mapping study. *Journal of Software: Evolution and Process*, 34(3), Article e2423. <https://doi.org/10.1002/smr.2423>
- Ma, D., & Wu, W. (2024). Does artificial intelligence drive technology convergence? Evidence from Chinese manufacturing companies. *Technology in Society*, 79(5), Article 102715. <https://doi.org/10.1016/j.techsoc.2024.102715>
- Mamman, H., Balogun, A. O., Basri, S., Capretz, L. F., Adeyemo, V. E., Imam, A. A., & Kumar, G. (2023). Software requirement risk prediction using enhanced fuzzy induction models. *Electronics*, 12(18), Article 3805. <https://doi.org/10.3390/electronics12183805>
- Masur, P. K., Epstein, D., Quinn, K., Wilhelm, C., Baruh, L., & Lutz, C. (2025). Comparative privacy research: Literature review, framework, and research agenda. *The Information Society*, 41(2), 69–90. <https://doi.org/10.1080/01972243.2025.2451863>
- Metin, B., Duran, S., Telli, E., Mutlutürk, M., & Wynn, M. (2024). IT risk management: Towards a system for enhancing objectivity in asset valuation that engenders a security culture. *Information*, 15(1), Article 55. <https://doi.org/10.3390/info15010055>
- Ming, C., Yu, X., & Zhang, B. (2024). Assessing the infringement risk of patent portfolios using network analysis and IF-TOPSIS: A case of standard-essential patent portfolios in the ICT industry. *Technology in Society*, 78, Article 102663. <https://doi.org/10.1016/j.techsoc.2024.102663>
- Opricovic, S., & Tzeng, G. H. (2007). Extended VIKOR method in comparison with outranking methods. *European Journal of Operational Research*, 178(2), 514–529. <https://doi.org/10.1016/j.ejor.2006.01.020>
- Razzaq, A., & Ghayyur, S. A. (2023). A systematic mapping study: The new age of software architecture from monolithic to microservice architecture—Awareness and challenges. *Computer Applications in Engineering Education*, 31(2), 421–451. <https://doi.org/10.1002/cae.22586>
- Rezaei, J. (2015). Best-worst multi-criteria decision-making method. *Omega*, 53(2), 49–57. <https://doi.org/10.1016/j.omega.2014.11.009>
- Rezaei, J., Nispeling, T., Sarkis, J., & Tavasszy, L. (2016). A supplier selection life cycle approach integrating traditional and environmental criteria using the best worst method. *Journal of Cleaner Production*, 135(8), 577–588. <https://doi.org/10.1016/j.jclepro.2016.06.125>
- Roshdieh, N. (2024). The Effect of Monetary Policy Uncertainty on Stock Market Uncertainty with NARDL Approach. *Research Journal of Finance and Accounting*, 15(10), 1-9. <https://dx.doi.org/10.2139/ssrn.5110149>
- Sangaiah, A. K., Samuel, O. W., Li, X., Abdel-Basset, M., & Wang, H. (2018). Towards an efficient risk assessment in software projects—Fuzzy reinforcement paradigm. *Computers & Electrical Engineering*, 71, 833–846. <https://doi.org/10.1016/j.compeleceng.2017.07.022>
- Singal, P., Sharma, P., & Kumari, A. C. (2022). Integrating software effort estimation with risk management. *International Journal of System Assurance Engineering and Management*, 13(5), 2413–2428. <https://doi.org/10.1007/s13198-022-01652-y>

- Song, H., Chang, R., Cheng, H., Liu, P., & Yan, D. (2024). The impact of manufacturing digital supply chain on supply chain disruption risks under uncertain environment—Based on dynamic capability perspective. *Advanced Engineering Informatics*, 60(12), Article 102385. <https://doi.org/10.1016/j.aei.2024.102385>
- Suárez, J., & Vizcaíno, A. (2024). Stress, motivation, and performance in global software engineering. *Journal of Software: Evolution and Process*, 36(5), Article e2600. <https://doi.org/10.1002/smr.2600>
- Tam, C., da Costa Moura, E. J., Oliveira, T., & Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, 38(3), 165–176. <https://doi.org/10.1016/j.ijproman.2020.02.001>
- Tavakoli, N., Rafieian, M., Amirjani, R., & Shahinrad, M. (2026). Women's recreational walking preferences in urban streets: A structural equation modeling approach. *Cities*, 171, Article 106721. <https://doi.org/10.1016/j.cities.2025.106721>
- Tonellato, M., Tasselli, S., Conaldi, G., Lerner, J., & Lomi, A. (2024). A microstructural approach to self-organizing: The emergence of attention networks. *Organization Science*, 35(2), 496–524.
- Tuunanen, T., Vartiainen, T., Kainulainen, S., & Ebrahim, M. (2023). Development of an agile requirements risk prioritization method: A design science research study. *Communications of the Association for Information Systems*, 52(1), 609–637. <https://doi.org/10.17705/1CAIS.05226>
- Ullah, A., & Xinlei, L. (2024). Navigating complex interdependence: An in-depth analysis of Iran and Saudi Arabia's strategic engagement with the BRI in the Middle East. *Chinese Political Science Review*, 29(2), 257–281. <https://doi.org/10.1007/s11366-023-09860-4>
- Vujović, V., Denić, N., Stevanović, V., Stevanović, M., Stojanović, J., Cao, Y., & Radojkovic, I. (2020). Project planning and risk management as a success factor for IT projects in agricultural schools in Serbia. *Technology in Society*, 63(2), Article 101371. <https://doi.org/10.1016/j.techsoc.2020.101371>
- Wan, M., Yu, M., Lei, J., Yang, S., & Wang, Z. (2026). Optimization of diesel engine comprehensive economy and emissions based on the RSM-NSGA III-VIKOR method. *Energy Conversion and Management: X*, Article 101811. <https://doi.org/10.1016/j.ecmx.2026.101811>
- Wang, Y., Arora, C., Liu, X., Hoang, T., Malhotra, V., Cheng, B., & Grundy, J. (2025). Who uses personas in requirements engineering: The practitioners' perspective. *Information and Software Technology*, 178, Article 107609. <https://doi.org/10.1016/j.infsof.2024.107609>
- Waqar, A., Othman, I., Shafiq, N., & Mansoor, M. S. (2023). Applications of AI in oil and gas projects towards sustainable development: A systematic literature review. *Artificial Intelligence Review*, 56(11), 12771–12798. <https://doi.org/10.1007/s10462-023-10467-7>
- Wiesche, M. (2021). Interruptions in agile software development teams. *Project Management Journal*, 52(2), 210–222. <https://doi.org/10.1177/8756972821991365>
- Wu, D., Zhu, X., Wan, J., Bao, C., & Li, J. (2019). A multiobjective optimization approach for selecting risk response strategies of software project: From the perspective of risk correlations. *International Journal of Information Technology & Decision Making*, 18(1), 339–364. <https://doi.org/10.1142/S0219622019410013>
- Yalcintas, A., & Alizadeh, N. (2020). Digital protectionism and national planning in the age of the internet: The case of Iran. *Journal of Institutional Economics*, 16(4), 519–536. <https://doi.org/10.1017/S1744137420000077>
- Yasin, A., Fatima, R., JiangBin, Z., Ali Khan, J., & Ali Khan, A. (2024). Gamifying requirements: An empirical analysis of game-based technique for novices. *Journal of Software: Evolution and Process*, 36(6), Article e2617. <https://doi.org/10.1002/smr.2617>
- Zhang, P., Ding, S., & Zhao, Q. (2024). Exploiting blockchain to make AI trustworthy: A software development lifecycle view. *ACM Computing Surveys*, 56(7), 1–31. <https://doi.org/10.1145/3614424>