

# Strategic Risk Management in Software Projects: ISM-MICMAC Framework

Mangesh Joshi<sup>1\*</sup> & Shweta Dixit Joshi<sup>2\*</sup>

<sup>1</sup>School of Engineering Sciences, Ramdeobaba University, Nagpur 440 013, Maharashtra, India

<sup>2</sup>Dr. Babasaheb Ambedkar College of Business Management, Nagpur 440 010, Maharashtra, India

*Received 23 May 2024; revised 19 September 2024; accepted 13 November 2024*

In the dynamic field of software development, identifying and managing critical risk factors is essential for project success. This study aims to model the primary risk factors influencing software projects using a novel approach. The research employs Interpretive Structural Modelling (ISM) combined with the Matriced' Impacts Croisés Multiplication Appliquée à un Classement (MICMAC) method to analyze and categorize these factors. ISM provides a structured framework for understanding the interrelationships between risk factors, while MICMAC offers insights into their driving power and dependence. This integrated methodology reveals a hierarchical structure of risk factors, classifying them into driving, linkage, and dependence categories. Key findings show that driving factors, such as External Dependencies and Requirements Volatility, are crucial to project success, necessitating proactive risk management strategies. Linkage factors, like Technical Complexity and Project Management, also play significant roles, while dependence factors, such as Technology Risks and Scope Creep, underscore the interconnected nature of these risks. The novelty of this study lies in its integration of ISM and MICMAC, offering a comprehensive framework to model and understand risk factors in software development. This approach provides valuable insights for project managers and stakeholders to better mitigate risks and improve project outcomes, contributing meaningfully to existing literature on software project risk management.

**Keywords:** Hierarchical structure, ISM, MICMAC, Risk factors, Software development

## Introduction

### Background

In the realm of software development, projects are intricate and multifaceted endeavors that demand careful attention to various uncertainties and risks.<sup>1</sup> These risks, if left unmanaged, can significantly impede the success of a project, affecting its timely delivery, cost-effectiveness, and overall quality. Therefore, it is crucial for project managers and stakeholders to prioritize identifying and mitigating these risks effectively. When critical risk factors in software development projects are discussed, reference is made to those key variables that have the potential to disrupt project progress or compromise its objectives. These factors can manifest in various forms, such as inadequate requirements gathering, technological complexities, resource constraints, unrealistic timelines, and external dependencies, among others.<sup>2</sup> Consequences such as project delays, budget overruns, diminished quality, or even project failure can result from failure to address these critical risk factors. For instance, confusion, rework, and

project scope creep can be caused by poorly defined or constantly changing requirements. Similarly, unforeseen challenges requiring additional time and resources to resolve may be encountered in projects involving cutting-edge or unfamiliar technologies.

Hindrance to progress and impact on project outcomes can be caused by resource constraints, such as limited availability of skilled personnel, hardware, or software.<sup>3</sup> Rushed developments, sacrificing quality and increasing the likelihood of errors, may result from setting overly ambitious or unrealistic deadlines. Moreover, projects relying on third-party components, APIs, or services are susceptible to disruptions caused by changes or failures in those dependencies.

Effective risk management involves proactive identification, assessment, and mitigation of these critical risk factors.<sup>4</sup> This often entails thorough risk identification by analyzing project requirements, stakeholders' expectations, and external factors. Subsequently, evaluation of the likelihood and impact of identified risks to prioritize mitigation efforts and allocate resources effectively is carried out. Developing strategies and contingency plans to address identified risks, such as adopting agile

\*Authors for Correspondence

E-mail: shweta.9590@gmail.com, joshimp@rknc.edu

practices, conducting thorough testing, or diversifying resource allocation, is also crucial. Continuous monitoring of project progress and reassessment of risks throughout the development lifecycle are essential to adapt mitigation strategies as needed. By modeling critical risk factors and implementing robust risk management methodologies, software development projects can enhance their resilience to uncertainties, improve decision-making, and increase the likelihood of successful outcomes.

#### **Purpose**

The primary objective of this research is to model critical risk factors in software development projects. In today's fast-paced and dynamic software development landscape, understanding and managing risks effectively is essential for project success.<sup>5</sup> Therefore, the aim of this study is to identify, map, and prioritize critical risk factors that may impact the outcome of software development projects. By modeling these risk factors, project managers and stakeholders can gain valuable insights into potential threats and take proactive measures to mitigate them, ultimately improving project outcomes and ensuring successful delivery.

The combination of Interpretive Structural Modeling (ISM) and the Matriced' Impacts Croisés Multiplication Appliquée à un Classement (MICMAC) approach presents a robust framework for comprehensively analyzing critical risk factors<sup>6</sup> within software development projects. ISM serves as a foundational tool in this process by allowing project managers to construct a hierarchical representation of risk factors, thereby visually mapping out their intricate interrelationships.<sup>7</sup> This hierarchical structure not only provides a holistic understanding of the complex network of risks but also highlights the key drivers that significantly influence project outcomes. With ISM, project managers can systematically categorize risk factors based on their causal dependencies, enabling them to prioritize and focus their attention on the most impactful risks. By positioning risk factors within the hierarchy according to their level of influence and dependence, ISM facilitates the identification of critical risks that pose the greatest threat to project success. This prioritization is invaluable in guiding resource allocation and decision-making processes, ensuring that efforts are directed towards mitigating the most consequential risks.

Furthermore, the integration of the MICMAC approach enhances the analytical capabilities of ISM

by providing a structured method for categorizing risk factors according to their driving power and dependence. The MICMAC approach distinguishes between different types of risk factors, such as autonomous, dependent, linkage, and autonomous-dependent factors, offering deeper insights into their relative importance and impact on project outcomes. Autonomous factors, for instance, possess high driving power and exert significant influence on project objectives independently of other factors. Identifying and addressing these autonomous risks are crucial for mitigating their potentially disruptive effects on the project. Dependent factors, on the other hand, have low driving power but high dependence on other factors, necessitating proactive management to minimize their impact. Linkage factors represent critical risks that both influence and are influenced by other factors within the risk landscape. Understanding the interconnected nature of these risks is essential for developing holistic risk mitigation strategies that address the underlying dynamics of the project environment. Finally, autonomous-dependent factors have low driving power and dependence, indicating minimal influence on project outcomes.

By systematically categorizing risk factors using the MICMAC approach, project managers can gain a nuanced understanding of the risk landscape and tailor their risk management strategies accordingly. This detailed analysis enables project managers to prioritize their efforts and resources effectively, focusing on mitigating the most critical risks while allocating adequate attention to other risk factors as needed.

#### **Related Works**

The systemic approach of Interactive Management (IM) has been utilized by researchers to identify and address key risk factors threatening Zimbabwean IT projects. Specifically, six main risks—limited computer literacy, poor communication, lack of executive support, design complexity, bureaucracy, and employee turnover—have been identified and analyzed through a causal model to demonstrate their interconnections and implications for risk mitigation efforts.<sup>8</sup> Interpretive Structural Modelling (ISM) was used to explore and understand the interrelationships among 23 critical risk factors in software engineering projects.<sup>9</sup> The application of risk analysis methodologies was carried out by using the DEMATEL-ISM model to establish hierarchical relationships among IT project risk factors, with

findings emphasizing the significant causal influence of factors such as inaccurate need analysis, lack of user support, and contract risks on other risks. Their work underscores the critical importance of accurate risk control mechanisms for IT project development.<sup>10,11</sup> Additionally, it was categorized into human, technology, and process dimensions, employing ISM to clarify their complex interrelations. By integrating 26 risk factors into seven broader elements, their model offers a structured framework that simplifies the analysis of risks, aiding personnel in comprehending intricate risk relationships and providing practical guidance for similar risk assessments in other domains.

#### Research Gap

The research gap in the domain of modeling critical risk factors in software development projects using ISM and MICMAC approach lies in the limited exploration and application of these methodologies in real-world software development contexts. While ISM and MICMAC have been extensively used in various fields for analyzing complex systems and understanding interrelationships among factors, their application specifically in the context of software development projects is relatively sparse.

One aspect of the research gap pertains to the need for empirical studies that validate the effectiveness and practical utility of ISM and MICMAC in modeling critical risk factors in software development projects. While theoretical discussions and conceptual frameworks exist, there is a lack of empirical evidence demonstrating the applicability and benefits of these methodologies in real-world project settings. Empirical studies could involve case studies or longitudinal analyses of software development projects where ISM and MICMAC are applied to model critical risk factors and evaluate their impact on project outcomes.

Furthermore, there is a gap in the literature regarding the integration of ISM and MICMAC with existing risk management frameworks and methodologies in software development. While ISM and MICMAC offer valuable insights into the hierarchical structure and categorization of risk factors, their integration with established risk management practices. Investigating how ISM and MICMAC can complement existing risk management processes and enhance their effectiveness could provide valuable insights for practitioners.

Another research gap relates to the scalability and adaptability of ISM and MICMAC to different types and sizes of software development projects. While these methodologies have demonstrated utility in various domains, their applicability to projects of varying complexities and scopes within the software development domain requires further exploration. Research could focus on identifying the factors that influence the effectiveness of ISM and MICMAC in different project contexts and developing guidelines or best practices for their application in diverse software development settings.

Overall, addressing these research gaps would contribute to advancing the state of knowledge in risk management in software development projects and provide valuable insights for practitioners seeking to enhance their risk management practices. By bridging the divide between theory and practice and providing empirical evidence of effectiveness, future research could help establish ISM and MICMAC as valuable tools for mitigating risks and improving project outcomes in the software development domain.

#### Research Objectives

The specific objectives of the research are outlined as follows:

1. **Identifying Critical Risk Factors:** The first objective of the research is to identify critical risk factors inherent in software development projects. This involves conducting a comprehensive analysis of potential risk factors that could impact project success, considering factors such as requirements volatility, technical complexity, resource constraints, and external dependencies.
2. **Analyzing Interrelationships:** The second objective is to analyze the interrelationships among identified risk factors. This involves examining how different risk factors influence each other and understanding the causal dependencies between them. By analyzing these interrelationships, the research aims to gain insights into the complex dynamics of risk in software development projects.
3. **Prioritizing Risks:** The third objective is to prioritize identified risks based on their impact and influence on project outcomes. This involves developing a systematic methodology for quantitatively assessing the importance of each risk factor and prioritizing them accordingly. By prioritizing risks, project managers can allocate resources more effectively and focus their efforts on mitigating the most critical threats.

**Methodology**

The ISM is a methodological approach used to analyze and understand complex systems by identifying the relationships among different elements or factors within the system. It was developed by John N. Warfield in the 1970s and has since been applied in various fields, including engineering, management, and systems analysis. The research flow diagram for the study is shown in Fig. 1.

The ISM methodology involves several key steps:

1. **Identifying factors:** The first step in ISM is to identify the elements or factors that constitute the system being analyzed. These elements could be components of a technology, aspects of an organization, variables in a process, or any other relevant entities within the system. The list of such factors is shown in Table 1.

2. **Develop Structural Self-Interaction Matrix (SSIM):** The Structural Self-Interaction Matrix

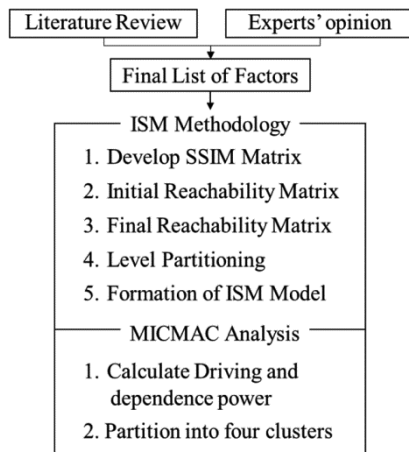


Fig. 1 — Research Flow diagram

(SSIM) is a key output of the Interpretive Structural Modeling (ISM) methodology. It is a square matrix that represents the contextual relationships between a set of variables or factors. Each row and column in the matrix correspond to a single factor, and the cells in the matrix indicate the degree of influence that each factor has on itself and on the other factors in the system. The SSIM is developed through a process of expert consultation, where practitioners are asked to identify the factors that are relevant to the system and to discuss the relationships between them. To establish a contextual relationship between variables in the ISM methodology, experts' opinions are solicited through management approaches such as brainstorming and nominal group technique. In this study, virtual meetings were conducted with practitioners to discuss the contextual relationship for each CSF and determine the existence and direction of influence between any two factors *i* and *j*. Their evaluation was used to construct a contextual relationship known as SSIM, as shown in Table 2. This relationship indicates that one variable contributes to the enhancement of another, creating a contextual relationship between the selected parameters. The analysis of factors in the development of SSIM utilized the following four variables to indicate the relationship between factors *i* and *j*:

- V- Factor *i* influences Factor *j*;
- A- Factor *i* is influenced by Factor *j*;
- X- Factor *i* and *j* influences each other; and
- O- No relation between the Factor *i* and *j*

3. **Constructing a Reachability Matrix:** After establishing the relationships between elements, a

Table 1 — List of identified factors x

Factor	Description
Technical complexity <sup>12</sup>	The level of intricacy and difficulty associated with the technical aspects of the software development process
Scope creep <sup>13</sup>	The tendency for project scope to expand beyond initial expectations, leading to increased project complexity
Resource constraints <sup>14</sup>	Limitations in the availability or allocation of resources, such as budget, time, or manpower
Team competency <sup>15</sup>	The collective skills, knowledge, and expertise of the project team in executing tasks and achieving goals
Communication Challenges <sup>16</sup>	Obstacles or barriers hindering effective communication among project stakeholders
Technology risks <sup>17</sup>	Potential threats or uncertainties associated with the adoption and use of technology in the project
Requirements volatility <sup>18</sup>	The frequency and extent of changes to project requirements over the course of the project
Quality assurance <sup>19</sup>	Practices and processes aimed at ensuring the quality and reliability of the software product
Project management <sup>20</sup>	The planning, coordination, and control of project activities to achieve project objectives
Stakeholder engagement <sup>21</sup>	The involvement and collaboration of stakeholders throughout the project lifecycle
External dependencies <sup>22</sup>	External factors, entities, or dependencies that impact the project but are beyond the project's control

Table 2 — Structural self interaction matrix (SSIM)

S N	Factors/problems	1	2	3	4	5	6	7	8	9	10	11
1	Technical complexity	–	V	O	A	V	O	A	V	V	O	O
2	Scope creep		–	A	A	A	O	A	A	A	A	A
3	Resource constraints			–	V	V	O	A	V	A	O	O
4	Team competency				–	A	V	V	V	A	V	O
5	Communication challenges					–	O	A	O	A	V	O
6	Technology risks						–	O	O	A	O	O
7	Requirements volatility							–	V	V	V	V
8	Quality assurance								–	A	A	A
9	Project management									–	A	A
10	Stakeholder engagement										–	A
11	External dependencies											–

reachability matrix is constructed to represent the connections between each pair of elements. The reachability matrix is a binary matrix that indicates whether there is a direct or transitive relationship between each pair of elements.

4. **Building a Digraph:** Based on the reachability matrix, a directed graph (digraph) is constructed to visually represent the relationships among the elements. In the digraph, each element is represented as a node, and the relationships between elements are represented as directed edges.

5. **Level Partitioning:** The stage of level partitioning involves organizing the elements or factors into distinct levels based on their hierarchical relationships within the system. This process begins by identifying the most influential or fundamental elements, often those with no incoming edges, and assigning them to the highest level. Subsequent levels are then determined based on the direct influence of higher-level elements, with lower-level elements influenced by those at higher levels. Through iterative refinement and analysis of connectivity patterns, level partitioning helps establish a clear hierarchical structure that accurately reflects the dependencies and interactions within the system. This structured organization facilitates the interpretation and analysis of the ISM model, enabling stakeholders to gain insights into the system's architecture and dynamics for informed decision-making.

6. **Interpretation and Analysis:** Once the digraph is constructed and organized into levels, it is interpreted to gain insights into the structure and dynamics of the system. This involves analyzing the relationships between elements, identifying key drivers or influencers within the system, and understanding how changes to one element may impact others.

**MICMAC Analysis**

The Matriced' Impacts Croisés Multiplication Appliquée à un Classement (MICMAC) analysis is a method used to categorize factors based on their driving power and dependence within a system. Here are the steps involved in conducting MICMAC analysis:

1. **Identifying Factors:** The first step is to identify and list all relevant factors or elements that contribute to the system under study. These factors can include variables, components, or entities that have an impact on the system's functioning or outcomes.

2. **Constructing a Structural Matrix:** Once the factors are identified, a structural matrix is constructed to represent the relationships between each pair of factors. The structural matrix typically consists of cells indicating the influence of one factor on another, with values assigned based on the strength and direction of the relationship.

3. **Calculating Driving Power and Dependence:** Using the structural matrix, the driving power and dependence of each factor are calculated. The driving power of a factor refers to its ability to influence other factors within the system, while dependence indicates the extent to which a factor is influenced by others. These calculations are often based on algorithms or mathematical formulas applied to the structural matrix.

4. **Categorizing Factors:** Based on their driving power and dependence values, factors are categorized into four groups: autonomous, dependent, linkage, and drivers. Driving factors have high driving power and low dependence, meaning they exert significant influence on other factors but are not strongly influenced by them. Dependent factors have low driving power and high dependence, indicating they are influenced by other factors but do not exert

significant influence themselves. Linkage factors have both high driving power and dependence, representing critical factors that both influence and are influenced by other factors. Autonomous factors have low driving power and dependence, suggesting they have minimal influence on other factors and are not significantly influenced by them.

5. **Interpretation and Analysis:** Once factors are categorized, the results of the MICMAC analysis are interpreted and analyzed to gain insights into the dynamics of the system. This involves understanding the roles and relationships of different factors within the system, identifying critical factors that drive system behaviour, and recognizing dependencies that may impact system performance.

**Results & Discussion**

The analysis of critical risk factors in software development projects using ISM coupled with the MICMAC approach yielded valuable insights into the hierarchical structure and interrelationships among these factors.

In ISM, influence is measured in binary form i.e. in 0 and 1 only. There is no value in between indicating medium or weak influence. The experts were asked to map the relationship of influence between all the factors considering two factors at one time. The relationship was mapped using V, A, X and O letters indicating ith factor influencing jth factor, jth factor influencing ith factor, both factors trying to influence each other, and no relation between the factors. The data in the Table 3 represents Initial Reachability Matrix (IRM) which was formed by utilizing the rules mentioned Step 2 and Step 3 of ISM methodology section. The row-wise summation of values indicates driving power of that factor. The column-wise summation of values indicates the dependence of that factor on the other factors. Considering the rules for removing transitivity, the final reachability matrix was formed as shown in Table 4.

The level partitioning consisting of reachability set and antecedent and associated levels is given in the Table 5. To obtain the reachability set for factor 1, the

Table 3 — Initial reachability matrix (IRM)

S N	Factors/problems	1	2	3	4	5	6	7	8	9	10	11
1	Technical complexity	1	1	0	0	1	0	0	1	1	0	0
2	Scope creep	0	1	0	0	0	0	0	0	0	0	0
3	Resource constraints	0	1	1	1	1	0	0	1	0	0	0
4	Team competency	1	1	0	1	0	1	1	1	0	1	0
5	Communication challenges	0	1	0	1	1	0	0	0	0	1	0
6	Technology risks	0	0	0	0	0	1	0	0	0	0	0
7	Requirements volatility	1	1	1	0	1	0	1	1	1	1	1
8	Quality assurance	0	1	0	0	0	0	0	1	0	0	0
9	Project management	0	1	1	1	1	1	0	1	1	0	0
10	Stakeholder engagement	0	1	0	0	0	0	0	1	1	1	0
11	External dependencies	0	1	0	0	0	0	0	1	1	1	1

Table 4 — Final reachability matrix (FRM)

S N	Factors/problems	1	2	3	4	5	6	7	8	9	10	11	Driving power
1	Technical complexity	1	1	1	1	1	1	0	1	1	1	0	9
2	Scope creep	0	1	0	0	0	0	0	0	0	0	0	1
3	Resource constraints	1	1	1	1	1	1	1	1	0	1	0	9
4	Team competency	1	1	1	1	1	1	1	1	1	1	1	11
5	Communication challenges	1	1	0	1	1	1	1	1	1	1	0	9
6	Technology risks	0	0	0	0	0	1	0	0	0	0	0	1
7	Requirements volatility	1	1	1	1	1	1	1	1	1	1	1	11
8	Quality assurance	0	1	0	0	0	0	0	1	0	0	0	2
9	Project management	1	1	1	1	1	1	1	1	1	1	0	10
10	Stakeholder engagement	0	1	1	1	1	1	0	1	1	1	0	8
11	External dependencies	0	1	1	1	1	1	0	1	1	1	1	9
	Dependence power	6	10	7	8	8	9	5	9	7	8	3	80

Table 5 — Level partitioning

S N	Factors/problems	Reachability set	Antecedent set	Level
1	Technical complexity	1,2,3,4,5,6,8,9,10,	1,3,4,5,7,9,	IV
2	Scope creep	2,	1,2,3,4,5,7,8,9,10,11,	I
3	Resource constraints	1,2,3,4,5,6,7,8,10,	1,3,4,7,9,10,11,	IV
4	Team competency	1,2,3,4,5,6,7,8,9,10,11,	1,3,4,5,7,9,10,11,	III
5	Communication challenges	1,2,4,5,6,7,8,9,10,	1,3,4,5,7,9,10,11,	III
6	Technology risks	6,	1,3,4,5,6,7,9,10,11,	II
7	Requirements volatility	1,2,3,4,5,6,7,8,9,10,11,	3,4,5,7,9,	VII
8	Quality assurance	2,8,	1,3,4,5,7,8,9,10,11,	II
9	Project management	1,2,3,4,5,6,7,8,9,10,	1,4,5,7,9,10,11,	V
10	Stakeholder engagement	2,3,4,5,6,8,9,10,	1,3,4,5,7,9,10,11,	III
11	External dependencies	2,3,4,5,6,8,9,10,11,	4,7,11,	VI

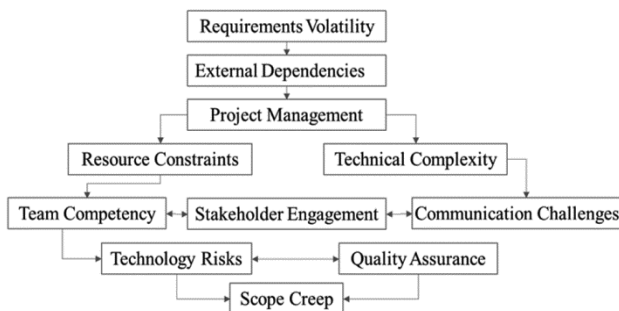


Fig. 2 — ISM model

corresponding row is searched for the value of ‘1’ in Table 4. To obtain the antecedent set for factor 1, the corresponding column is searched for the value of ‘1’ in Table 4. Reachability set /antecedent set consists of all such factors where it appears ‘1’. The intersection set is then evaluated by considering common elements of reachability set and antecedent set. The ISM model is shown in Fig. 2. Based on the values of driving and dependence power, Table 6 is formed representing ranking and grouping of the factors into driving, dependence, linkages and autonomous factors. The grouping based on driving and dependences powers of the factors is shown in Fig. 3.

In MICMAC analysis, External Dependencies are identified as a top driving factor with substantial influence and low dependence on internal factors. This suggests that external elements such as regulatory constraints, vendor relationships, and market shifts play a pivotal role in shaping project outcomes, while not being significantly affected by internal dynamics. Effectively managing these dependencies is crucial for minimizing risks and ensuring project success, given their capacity to impact other elements within the system. Requirements Volatility ranks second in the Driving Group, similarly demonstrating significant influence

while being minimally dependent on other factors. It impacts critical aspects such as project scope, resource allocation, and team dynamics. This underscores the need for agile project management strategies capable of adapting to evolving requirements, ensuring alignment with changing stakeholder expectations and maintaining project progress.

Ranked third in the Linkage Group, Technical Complexity balances significant driving power with moderate dependence. It influences key project areas such as timelines, resource management, and overall performance while being shaped by other system factors. Addressing technical complexity is vital for smooth project execution and minimizing disruptions. Project Management, ranked fourth, has a strong influence on various project elements, including resource optimization and coordination. However, its effectiveness is shaped by internal and external factors, making adaptive strategies essential for project success. Team Competency, in fifth place, emphasizes the importance of skilled teams in achieving project goals, though it is influenced by other elements, necessitating continuous development and empowerment efforts.

Resource Constraints (ranked sixth) highlight the need for strategic resource allocation, as balancing availability with project demands directly impacts delivery timelines and budgets. Communication Challenges, positioned seventh, play a vital role in fostering collaboration and resolving conflicts but are influenced by various internal and external dynamics, requiring robust communication management strategies. In the Dependence Group, Quality Assurance (ranked ninth) relies heavily on factors like resources and scope for maintaining standards, while Technology Risks (ranked tenth) necessitate proactive

Table 6 — MICMAC ranking and grouping

Factors	Dependence power	Driving Power	Driving/Dependence power	MICMAC rank	MICMAC group
Technical complexity	6	9	1.50	3	Linkage
Scope creep	10	1	0.10	11	Dependence
Resource constraints	7	9	1.29	6	Linkage
Team competency	8	11	1.38	5	Linkage
Communication challenges	8	9	1.13	7	Linkage
Technology risks	9	1	0.11	10	Dependence
Requirements volatility	5	11	2.20	2	Driving
Quality assurance	9	2	0.22	9	Dependence
Project management	7	10	1.43	4	Linkage
Stakeholder engagement	8	8	1.00	8	Linkage
External dependencies	3	9	3.00	1	Driving

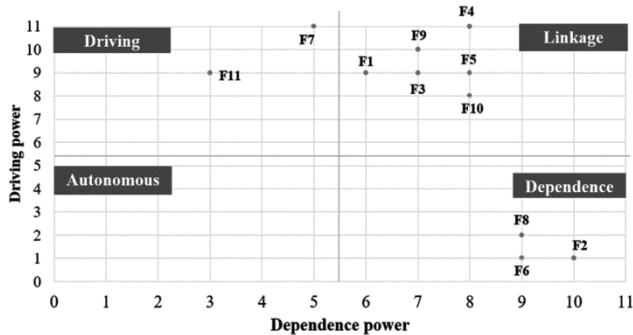


Fig. 3 — MICMAC analysis

risk management to prevent disruptions. Lastly, Scope Creep, ranked eleventh, underscores challenges in managing project scope, with high dependence on other factors that, if unmanaged, can lead to delays and cost overruns. Vigilant change control and scope management are essential for mitigating its impact.

Based on the results, several implications can be drawn for managing software development projects effectively:

**1. Strategic Planning and Risk Management:**

Recognizing driving factors such as External dependencies and Requirements volatility underscores the importance of robust strategic planning and risk management practices. Project managers should proactively identify and assess external dependencies and potential changes in project requirements, incorporating contingency plans to mitigate associated risks and ensure project resilience.

**2. Resource Allocation and Capacity Building:**

Linkage factors such as Technical complexity, Project management, and Team competency highlight

the critical role of resource allocation and capacity building initiatives. Project managers should prioritize investments in technical expertise, project management capabilities, and team development to effectively navigate project complexities and optimize project performance.

**3. Communication and Stakeholder Engagement:**

Addressing factors like Communication challenges and Stakeholder engagement is essential for fostering collaboration and alignment among project stakeholders. Project managers should implement effective communication strategies and stakeholder engagement plans to facilitate transparency, resolve conflicts, and maintain stakeholder support throughout the project lifecycle.

**4. Risk Mitigation and Contingency Planning:**

Dependence factors such as Technology risks, Quality assurance, and Scope creep highlight areas where proactive risk mitigation and contingency planning are essential. Project managers should conduct comprehensive risk assessments, implement quality assurance measures, and establish robust change control mechanisms to address potential disruptions and deviations from project scope.

**5. Agile and Adaptive Project Management:**

Given the dynamic nature of software development projects, adopting agile and adaptive project management methodologies is crucial. Agile approaches enable iterative development, continuous stakeholder feedback, and adaptive responses to changing project requirements, thereby enhancing project flexibility and responsiveness to evolving conditions.

## 6. Continuous Improvement and Learning Culture:

Emphasizing a culture of continuous improvement and learning is paramount for addressing the evolving challenges of software development projects. Project teams should encourage knowledge sharing, foster innovation, and embrace lessons learned from project experiences to drive ongoing performance enhancements and organizational growth.

By incorporating these implications into project management practices, organizations can effectively navigate the complexities of software development projects, mitigate risks, and maximize project success.

## Conclusions

This study provides valuable insights into the hierarchical structure and interrelationships among critical risk factors in software development projects. Using ISM to model complex risk interactions and MICMAC to categorize them based on driving power and dependence, it highlights key factors influencing project success. Driving factors, such as External dependencies and Requirements volatility, require strategic risk management, while linkage factors like Technical complexity emphasize resource allocation and collaboration. Dependence factors, including Technology risks and Scope creep, point to the importance of proactive mitigation. However, this approach has limitations, including reliance on data quality and the potential for subjective bias in risk factor analysis. Additionally, the static nature of ISM-MICMAC models may not fully capture the evolving dynamics of software projects. Future research could integrate quantitative methods for better accuracy and dynamic modelling to address the changing nature of projects. Exploring cross-industry applications and developing decision support systems may also expand its practical utility in managing software development risks.

## References

- Joshi M, Mankar R & Shukla H, Identification and evaluation of factors influencing software quality using pythagorean fuzzy DEMATEL approach, *Int J Next-Gener Comput*, **13(5)** (2022), <https://doi.org/10.47164/ijngc.v13i5.938>.
- Verner J M & Abdullah L M, Exploratory case study research: Outsourced project failure. *Inf Softw Technol*, **54(8)** (2012) 866–886, <https://doi.org/10.1016/j.infsof.2011.11.001>.
- Trendowicz A & Münch J, Factors influencing software development productivity—state of the art and industrial experiences, *Adv Comput*, **77** (2009) 185–241, [https://doi.org/10.1016/S0065-2458\(09\)01206-6](https://doi.org/10.1016/S0065-2458(09)01206-6).
- Can Saglam Y, Yildiz Çankaya S & Sezen B, Proactive risk mitigation strategies and supply chain risk management performance: An empirical analysis for manufacturing firms in Turkey, *J Manuf Technol Manage*, **32(6)** (2021) 1224–1244, <https://doi.org/10.1108/JMTM-08-2019-0299>.
- Joshi M, Prioritizing performance evaluation factors of event-based information systems using interpretive structural modeling, *Suranaree J Sci Technol*, **30(3)** (2023) 1–10, <http://dx.doi.org/10.55766/sujst-2023-03-e01014>.
- Joshi M & Deshpande V, Application of interpretive structural modelling (ISM) for developing ergonomic workstation improvement framework, *Theor Issues Ergon Sci*, **24(1)** (2023) 88–110, <https://doi.org/10.1080/1463922X.2022.2044932>.
- Joshi M, Umredkar S & Das S, Application of interpretive structural modeling in user interface design, *Mater Today: Proc*, **72** (2023) 698–705, <https://doi.org/10.1016/j.matpr.2022.08.427>.
- Kanjanda T & Tuan N T, A systemic exploration of the risk factors in Zimbabwean information technology projects, *Systemic Pract Action Res*, **33** (2020) 77–93.
- Samantra C, Datta S, Mahapatra S S & Debata B R, Interpretive structural modelling of critical risk factors in software engineering project, *Benchmarking: Int J*, **23(1)** (2016) 2–24.
- Jun Y R, Shang L & Zhang H, Risk factor identification of the information technology project based on the DEMATEL-ISM Model, *J Eng Sci Technol Rev*, **15(4)** (2022) 53–59.
- Qu Y, Yuan M J & Liu F, The risk factor analysis for software project based on the interpretive structural modelling method, in *Int Conf Machine Learn Cybernet* (IEEE) 2012, 1019–1024.
- Zaman U, Jabbar Z, Nawaz S & Abbas M, Understanding the soft side of software projects: An empirical study on the interactive effects of social skills and political skills on complexity–performance relationship, *Int J Proj Manage*, **37(3)** (2019) 444–460, <https://doi.org/10.1016/j.ijproman.2019.01.015>.
- Komal B, Janjua U I, Anwar F, Madni T M, Cheema M F, Malik M N & Shahid A R, The impact of scope creep on project success: An empirical investigation, *IEEE Access*, **8** (2020) 125755–125775, <https://doi.org/10.1109/ACCESS.2020.3007098>.
- Akbar S, Ahmad I, Khan R, Lopes I O & Ullah R, Multi-skills resource constrained and personality traits based project scheduling, *IEEE Access*, **10** (2022) 131419–131429, <https://doi.org/10.1109/ACCESS.2022.3229867>.
- Assyne N, Ghanbari H & Pulkkinen M, The essential competencies of software professionals: A unified competence framework, *Inf Softw Technol*, **151** (2022) 107020, <https://doi.org/10.1016/j.infsof.2022.107020>.
- Alzoubi Y & Gill A, The critical communication challenges between geographically distributed agile development teams: empirical findings, *IEEE Trans Prof Commun*, **64(4)** (2021) 322–337, <https://doi.org/10.1109/TPC.2021.3110396>.

- 17 Ukalli K & Antovski L, Risks influencing software projects and managing them during the requirement engineering process, *Knowledge-Int J*, **23(1)** (2018) 321–328.
- 18 Kulk G P & Verhoef C, Quantifying requirements volatility effects, *Sci Comput Program*, **72(3)** (2008) 136–175, <https://doi.org/10.1016/j.scico.2008.04.003>.
- 19 Koru G, El Emam K, Neisa A & Umarji M, A survey of quality assurance practices in biomedical open source software projects, *J Med Internet Res*, **9(2)** (2007) e588, <https://doi.org/10.2196/jmir.9.2.e8>.
- 20 Keil M, Rai A, Mann J C & Zhang G P, Why software projects escalate: The importance of project management constructs, *IEEE Trans Eng Manage*, **50(3)** (2003) 251–261, doi: 10.1109/TEM.2003.817312.
- 21 Mysore K, Kirytopoulos K, Ahn S & Ma T, Adversarial factors in multi-stakeholders' engagement of global-IT projects, *Int J Manag Proj Bus*, **14(2)** (2021) 445–471, <https://doi.org/10.1108/IJMPB-01-2019-0014>.
- 22 Blincoe K, Harrison F, Kaur N & Damian D, Reference coupling: An exploration of inter-project technical dependencies and their characteristics within large software ecosystems, *Inf Softw Technol*, **110** (2019) 174–189, <https://doi.org/10.1016/j.infsof.2019.03.005>.