Computer and Decision Making

An International Journal

www.comdem.org
eISSN: 3008-1416

A Decision-Support Framework Integrating ISM, Modified Fuzzy MICMAC, and Optimization for Software Requirement Dependency Analysis

Hamdi Bashir[1, 2, *], Messa Alhammadi[3], Hassan Ahmed Al Zarooni[4], Salah Haridy[1, 2, 5], Mohammad Shamsuzzaman[1, 2]

1 Department of Industrial Engineering and Engineering Management, College of Engineering, University of Sharjah, UAE
2 Sustainable Engineering Asset Management Research Group, University of Sharjah, Sharjah, UAE
3 Emirates Group, Dubai, UAE
4 Sharjah Electricity, Water and Gas Authority, UAE
5 Benha Faculty of Engineering, Benha University, Benha, Egypt

ARTICLE INFO

Article history:
Received 2 June 2026
Received in revised form 10 July 2026
Accepted 19 July 2026
Available online 21 July 2026

Keywords:
Software requirements; Requirement dependencies; Interpretive Structural Modeling (ISM); Modified Fuzzy MICMAC; Lexicographic optimization

ABSTRACT

Several studies have developed methods for modeling and analyzing dependencies among requirements in software development projects. However, these methods provide limited support for decision-making regarding software requirements (SR) changes. Extending this line of research, the present study proposes a three-stage decision-support framework to more systematically model, analyze, and manage dependencies among SRs. First, Interpretive Structural Modeling (ISM) is employed to organize SRs into a simple hierarchical form. Next, a modified version of the Fuzzy Cross-Impact Matrix Multiplication Applied to Classification (MICMAC) analysis, which considers both the likelihood of occurrence and the impact of change, is employed to categorize SRs based on their criticality. Building on the outputs of the ISM and fuzzy MICMAC analyses, a lexicographic multi-objective optimization model is then developed to support the selection of SR changes that satisfy customer requirements while minimizing the breadth and intensity of change propagation. The applicability of the proposed framework is demonstrated through a real-world software development project involving 28 SRs. This demonstration illustrates that the proposed framework provides a systematic and robust decision-support tool for software requirements change management, helping to reduce the risk of project cost and schedule overruns.

1. Introduction

Software requirements (SRs) changes, also known as “requirement volatility” [1], are an unavoidable phenomenon of software development projects [2]. According to Yin et al. [3], the development process itself is continuously evolving; thus, SR changes can occur at any stage of a project. Indeed, Bhatti et al. [4] reported that more than 50% of SRs change during the project

* *Corresponding author.*
E-mail address: hbashir@sharjah.ac.ae

lifecycle. Such changes have a substantial influence on project outcomes, often leading to schedule delays and budget overruns. These operational disruptions translate directly into substantial financial burdens. Ingram [5] reported that SR changes could make up 40% to 90% of the total cost of an industrial project, highlighting the profound economic implications of requirement volatility. Therefore, managing change effectively is important to a project's success. In a large-scale literature survey conducted by Iriarte and Bayona [6], a total of 263 different factors of project success were identified in the literature survey; in this context, the second most mentioned was change management.

Research on managing changes has investigated various practices. Studies frequently focus on project management practices to address SR volatility, via frameworks like change control boards [7], early choice of execution strategies and process models [8], methodologies like Joint Application Design, configuration management [7, 9], establishing baseline requirements [10], or comprehensive change management planning [11]. Nevertheless, as Thakurta [1] pointed out, "the success of these practices is known to vary significantly across regions, suggesting that best practices are not universal."

A key challenge limiting the effectiveness of these practices is the complex interdependencies among SRs. Knowledge boundaries and coordination challenges further exacerbate this complexity during requirements determination, thereby hindering effective decision-making [12]. Empirical evidence suggests that approximately 80% of SRs exhibit interdependencies [13], meaning that a change in one requirement can trigger cascading effects across others. Such propagation increases uncertainty and may lead to significant cost and schedule overruns. Consequently, it becomes essential to systematically analyze and manage these interdependencies, taking into account their mutual interactions and influence [14]. To address this challenge, several methods have been proposed to model and analyze SR dependencies; however, as discussed in Section 3, these methods suffer from notable limitations.

To overcome these limitations, this study proposes an integrated, three-stage decision-support framework. First, Interpretive Structural Modeling (ISM) is implemented to organize SRs into a hierarchical structure that reveals multilevel dependency relationships. Second, a modified version of the Fuzzy Cross-Impact Matrix Multiplication Applied to Classification (MF-MICMAC) analysis is used to classify SRs by criticality, considering both the likelihood and impact of change propagation. Third, building on the outputs of the ISM and MF-MICMAC analyses, an optimization-based decision-support model is developed to systematically select SR changes that satisfy customer requirements while minimizing the breadth and intensity of change propagation.

2. Related Research

Jayatilleke and Lai [14], in their systematic review, identified three main components of change management in SRs: determining changes, understanding their impact, and quantifying the effort or cost. This article is about the second component: change impact analysis.

Change impact analysis involves modeling and analyzing dependencies among requirements to help trace how changes could spread and what implications they might have early in the development lifecycle. Given the complexity of these interdependencies, decision-support techniques have been increasingly developed to assist in improving planning, monitoring, and control in software projects [15].

Over time, various methods have been developed to model these interdependencies. Liu and Smolka [16] proposed Directed Graphs (DGs) with hyperedges to capture cause-effect relationships,

providing the groundwork for visualizing software dependencies. Other methods have used models of links and traceability to connect criteria and show how they are related in meaning [17-19]. Sangal et al. [20] used the Design Structure Matrix (DSM), sometimes called the dependency structure matrix, to identify interdependencies in software design by analyzing source code. Later, Widiastuti and Siahaan [21] built on this idea to show the order of changes in requirements, making it easier for stakeholders to see how changes were made and their effects.

At the same time, goal-oriented requirements engineering facilitated the modeling of deliberate and goal-oriented linkages across requirements, hence enhancing impact reasoning at an elevated level [22–24]. Fu et al. [25] also used DSM to examine software architecture, showing how easily changes can spread through a system.

Later, more studies broadened the reach. Ali and Lai [26] examined changes in SR within global development contexts. Jayatilleke et al. [27] employed DSM to examine the intricacy of need modifications. Yin et al. [3] built a class-class matrix at the code level to show how modifications could spread and what concerns they might bring. Priyadi et al. [28] built on this work by employing text processing and algorithms to make DGs from SR specification documents. Researchers have also studied probabilistic and risk-based models to determine how likely change would spread based on dependence data [29, 30]. Addressing limitations identified in prior research [17, 18, 31–34], Arvanitou et al. [35] proposed the Requirements Ripple Effect Metric for predicting the probability of a requirement causing changes elsewhere in the system, offering a more precise way of estimating change impact

Most recently, Bashir et al. [36] introduced an approach grounded in Social Network Analysis (SNA) to model and analyze interdependencies among SRs. Using a real-world case comprising 28 SRs, the study applied structural network measures, including density, in-degree and out-degree centrality, and betweenness centrality, to determine requirement criticality and assess the potential propagation of changes across the requirements network.

3. Research Gaps and Study Justification

DMs have been widely used in various studies to represent dependencies among SRs for change impact analysis. However, aside from Fu et al. [25], most of these studies rely on binary representations of relationships, which may oversimplify the nature and intensity of SR dependencies, thereby limiting their usefulness for decision-making.

A major problem with both binary and non-binary DMs is that they do not capture accumulated or multilevel dependencies. One way to address this shortcoming is to use DGs, in which each SR is represented as a node and dependencies are shown as directed links between nodes. DGs allow for a more detailed view of how changes might propagate. However, as the number of requirements grows, DGs often become increasingly complex, making it hard to follow dependency paths or understand interactions across different levels. Also, both DMs and DGs lack robust analytical tools to systematically examine how SRs interact or classify them by the nature of their dependencies. This kind of classification is very important for managing software projects because it enables decision makers to identify the most critical requirements and prioritize them accordingly.

Another gap in existing research is that only a few studies, including Fu et al. [25] and Jayatilleke and Lai [14], have addressed changes in SRs that originate at higher levels of abstraction. Most methods still look at changes at the source code level. This is especially a problem during the early stages of software development, when building new systems from scratch, because the source code may not yet be available or sufficiently reliable for decision support.

The work of Bashir et al. [36] is a step forward as it provides a network-based view of SR dependencies but has some drawbacks. First, it adopts a binary network model that does not account for the strength of connections between SRs. Second, the network becomes denser and harder to read as more SRs are added, especially when tracking cascading or multilevel dependencies. Third, the conventional centrality measures used in this approach only yield a flat ranking of importance; they don't group SRs into meaningful hierarchical layers that reveal deeper, cumulative dependencies, thereby limiting decision support for SR change management.

The limitations discussed above highlight the need for a decision-support framework that (i) structures SRs into a hierarchical model capable of representing multilevel and accumulated dependencies in a simplified and interpretable form, (ii) incorporates fuzziness into dependency analysis by accounting for both the likelihood of occurrence and the impact of change, and (iii) supports systematic decision-making by enabling the selection of requirement changes that balance functional objectives with change propagation effects.

Based on these gaps, the novelty of this study lies in proposing an integrated framework that combines hierarchical modeling, fuzzy dependency analysis, MICMAC classification, and optimization-based decision support. Unlike previous DM-, DG-, and SNA-based approaches, the proposed framework captures accumulated, multilevel SR dependencies while considering both the likelihood and the impact of change propagation. More importantly, it extends dependency analysis by incorporating an optimization model to select requirement changes that maximize functional benefits while minimizing change propagation.

4. Proposed Decision-Support Framework

Decision-making in software projects often involves multiple interrelated factors, necessitating structured techniques to support effective planning and control and to understand system relationships [15, 37]. Systems theory offers various techniques for identifying and analyzing relationships within complex systems, including Interpretive Structural Modeling (ISM) [38]. This technique has three main characteristics. First, it is interpretive: specialists and analysts make judgment decisions about the relationships between elements. Second, it is structural: a comprehensive structure is derived from a set of variables (elements) based upon their associations. Last, it is a modeling technique in which a graphical model with nodes and directed arcs represents a comprehensive structure. This graphic transformation is viewed as expedient for managing poorly defined or unstructured mental models [39].

Dependencies among variables identified through ISM are commonly further analyzed using MICMAC analysis [40] to determine the most critical system variables based on their dependence and driving powers and to categorize them accordingly. Scholars have typically used variants of MICMAC analysis: the classical version (e.g., [41–44]) and the fuzzy version (e.g., [45, 46]).

While ISM and MICMAC-based dependency analysis methods (classical or fuzzy) can provide valuable insights into dependency structures and SR criticality, they remain primarily analytical and therefore require extension to explicitly account for both the likelihood and the impact of change propagation and to support optimization-based decision-making for systematic selection of requirement changes. Accordingly, the proposed framework adopts a modified fuzzy MICMAC analysis that accounts for both the likelihood of occurrence and the impact of change. It integrates these results into an optimization-based decision-support model to guide the systematic selection of changes to requirements**.**

As illustrated in Figure 1, the framework is implemented in three sequential stages. In the first stage, ISM is applied to structure SRs into a hierarchical model that reveals multilevel and accumulated dependencies. In the second stage, the modified fuzzy MICMAC (MF-MICMAC) analysis is employed to assess dependency strength and classify requirements based on their driving and dependence characteristics. In the third stage, the outputs of the ISM and MF-MICMAC analyses are operationalized through an optimization-based decision-support model that evaluates alternative combinations of requirement changes and identifies solutions that satisfy functional objectives while minimizing the risk of change propagation.

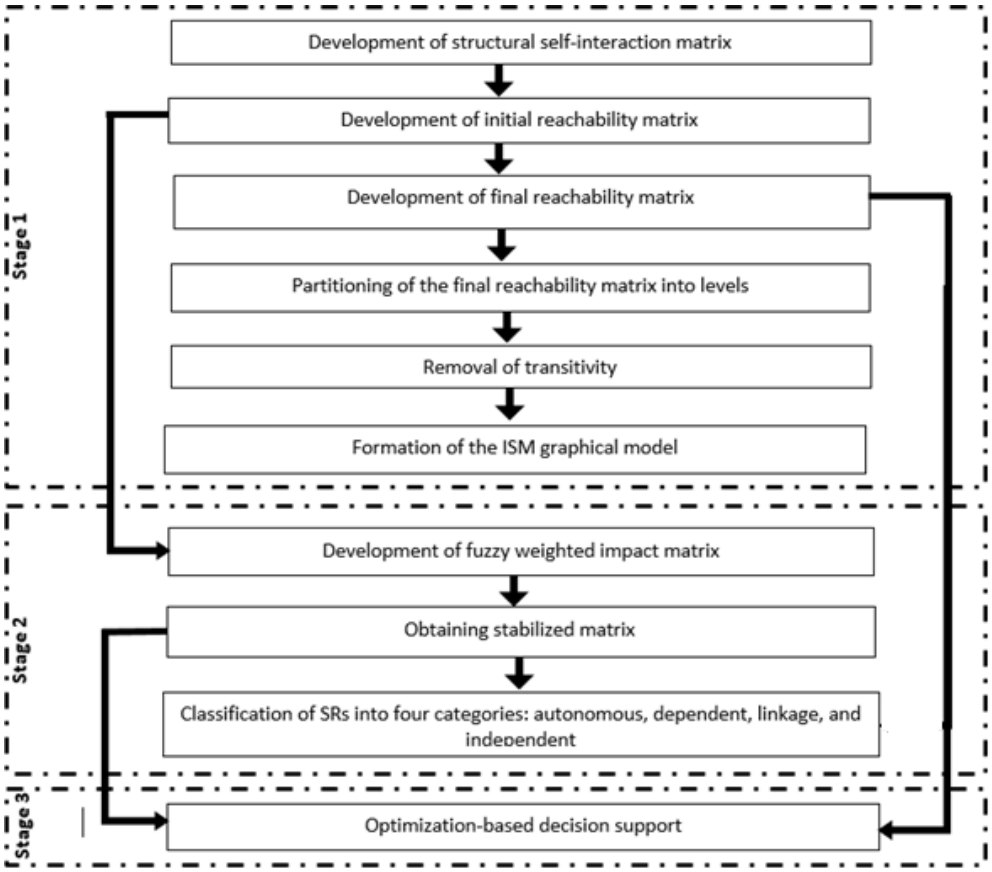

Fig. 1. Stages of the proposed ISM–MF-MICMAC–optimization framework

To further position the proposed framework relative to existing dependency-analysis methods, Table 1 compares representative DM-, DG-, ISM-, MICMAC-, and SNA-based approaches with the proposed framework in terms of modeling, classification, and decision-support capabilities for SR dependencies.

To demonstrate its practicality, the framework is applied to 28 SRs from a real-world human resource management system (HRMS) development project. The HRMS supports the main functions of the human resources department, such as benefits administration, payroll, performance evaluation, recruitment, and training. This same project was used in Bashir et al. [36], where an SNA-based approach was implemented to model and analyze SR interrelationships. Using the same HRMS project in the present study enables a direct comparison between the proposed framework and the SNA-based approach developed by Bashir et al. [36].

Table 1
Comparison of the proposed framework with relevant dependency-analysis approaches

Approach	Main focus	Treatment of dependency strength	Treatment of multilevel / accumulated dependencies	Classification capability	Decision-support capability
DM/DSM-based methods	Represent pairwise dependencies among SRs in matrix form	Usually binary; some studies use non-binary values	Limited, as dependencies are mainly represented at the pairwise level	Limited	Limited; mainly supports dependency visualization and tracing
DG-based methods	Represent SRs as nodes and dependencies as directed links	Usually binary, unless weighted links are added	Can show change propagation paths, but graphs become complex as the number of SRs increases	Limited	Limited; mainly supports visual analysis of dependency paths
ISM-based methods	Structure dependencies into hierarchical levels	Usually based on binary relationships	Strong ability to represent hierarchical and multilevel dependencies	Provides hierarchical structuring, but not dependency-strength classification	Limited if used alone
MICMAC-based methods	Classify elements based on driving and dependence power	Usually based on binary reachability relationships	Limited if not integrated with hierarchical modeling	Classifies elements into groups such as driving, dependent, linkage, and autonomous	Limited if used alone
SNA-based methods	Analyze SR interrelationships using network measures	Usually binary or weighted, depending on the network design	Can identify central SRs, but often provides flat rankings rather than hierarchical levels	Provides centrality-based ranking but not hierarchical classification	Limited; does not directly support selection of requirement changes
Proposed framework	Integrates hierarchical modeling, fuzzy dependency analysis, MICMAC classification, and optimization-based decision support	Considers dependency strength using fuzzy likelihood and impact values	Captures direct, accumulated, and multilevel dependencies in a simplified hierarchical structure	Classifies SRs based on driving and dependence characteristics	Supports selection of requirement changes by balancing functional benefits and change propagation effects

The following are the SRs:

1. Login feature based on users with username and password
2. Admin creates users
3. Users can log out of the HRMS system
4. A failure message is shown if a user does not exist in the database or if the user has not been authorized by the admin yet
5. Users can reset the password using the phone number saved in his/her record
6. Interface and data based on user role
7. Admin creates roles (HR's role/Manager's role/Employee's role)
8. Admin assigns permissions to each role
9. Admin assigns a role to a user
10. The user role can be checked from the database
15. Users with a manager's role can edit his/her employee's info
16. Users with a manager's role can search for the employees who are under his/her coverage
17. HR and admin roles can search all the employees' information
18. Search feature works on specific keywords, showing employees' characteristics, peculiarities, skills, features, etc.
19. The report button filters the contents of the search option
20. Admin can update the role type of a specific user
21. HR users can add a new employee to the database
22. Admin can edit user details
23. Admin can list all users

11. Users with defined roles can display the content of the database
12. Users with an employee role can view his/her personal info
13. Users with an employee role can edit his/her personal info
14. Users with a manager's role can view his/her employee info
24. Admin can search for users
25. Admin can edit the user's role
26. Admin can list all roles
27. Admin can display all users who have a specific role
28. Admin can manage permissions for each role

4.1 Stage I: ISM Model Development

Developing an ISM model comprises three main steps:

1. Determining binary relationships among SRs;
2. Identifying the hierarchical level of each SR; and
3. Building a hierarchical graphical model

4.1.1 Determining binary relationships among SRs

Binary relationships among SRs are determined using the Structural Self-Interaction Matrix (SSIM), following the procedure presented in Bashir et al. [36]. In this step, the development team examines the directional dependency between each pair of SRs using four standard symbols: *V*, *A*, *X*, and *O*. The symbols are defined as follows: *V* denotes that a change in SR_i would trigger a change in SR_j (a forward link), *A* denotes that a change in SR_j would trigger a change in SR_i (a reverse link), *X* denotes a mutual dependency between SR_i and SR_j, and *O* denotes that SR_i and SR_j are independent.

The SSIM is then converted into the initial reachability matrix using established substitution rules [38]. This matrix is equivalent to the adjacency matrix used in Bashir et al. [36], with the only difference being that the diagonal entries are set to zero in the adjacency matrix, whereas self-reachability is retained in the initial reachability matrix. Next, transitivity is applied to generate the final reachability matrix, which represents both direct and indirect associations among SRs. This is achieved by iteratively multiplying the initial reachability matrix by itself via Boolean matrix multiplication until stabilization is reached. Table 2 shows the resulting final reachability matrix for the demonstrative development project.

4.1.2 Identifying the hierarchical level of each SR

After the reachability matrix is created, each SR's level in the ISM hierarchical graphical model is determined by finding the intersection of the antecedent and reachability and sets first. For every SRi, the antecedent set $A(SR_i)$ is the set of SRs that reach SR_i. Similarly, for each SR_i, the reachability set $R(SR_i)$ is the set of SRs reachable from SR_i. Any SR_i where the antecedent set and the intersection set were identical was classified as a bottom-level SR_i in the ISM hierarchical model. All bottom-level SRs are then removed from $A(SR_i)$ and $R(SR_i)$ to identify the next level of SRs utilizing the same process. This process continues until all SRs are assigned to hierarchy levels. Table 3 shows all SRs for the demonstrative development project, assigned to seven levels, using this process.

Table 2
Final reachability matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
2	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
8	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
9	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
10	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
11	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

4.1.3 Constructing a hierarchical graphical model

The final step in this stage involves visualizing the associations among the SRs by first removing transitive links, thereby constructing a simplified hierarchical graphical model composed of nodes and directed arcs. In this model, the nodes representing the SRs are arranged according to the levels identified in the preceding step. If a direct association is present between SR_i and SR_j, an arc directing from SR_i to SR_j is plotted. Figure 2 shows the hierarchical graphical model constructed for the demonstrative development project.

4.2 Stage II: MF-MICMAC Analysis

This stage comprises two main steps:

1. Measuring the strength of relationships among SRs and
2. Classifying the SRs

4.2.1 Measuring the strength of relationships among SRs

In this step, the off-diagonal elements with a value of “1” in the initial reachability matrix are replaced by weights that represent the direct relationships among the SRs, based on the likelihood of occurrence and the impact of changes on each other, using fuzzy set theory [47]. Fuzzy set theory allows for the gradual assessment of an element's membership in a set. Different shapes are related to membership functions, but triangular membership functions are most frequently used [48]. A triangular fuzzy number is defined by three parameters, (*a*, *m*, *b*), where *a*, *m* and *b* represent the lower bound, the most probable (modal) value, and the upper bound, respectively. The corresponding membership function “$\mu_{\tilde{A}}(x)$” is given in Eq. (1).

Table 3
Hierarchical levels of SRs

SR	Reachability set	Antecedent set	Level
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28	1	Level 1
2	2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28	1, 2, 21	Level 2
3	3	1, 2, 3, 21	Level 7
4	4	1, 2, 4, 21	Level 7
5	5	1, 2, 5, 21	Level 7
6	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 6	Level 2
7	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 2, 6, 7, 8, 9, 10, 11, 21	Level 3
8	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 2, 6, 7, 8, 9, 10, 11, 21	Level 3
9	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 2, 6, 7, 8, 9, 10, 11, 21	Level 3
10	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 2, 6, 7, 8, 9, 10, 11, 21	Level 3
11	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28	1, 2, 6, 7, 8, 9, 10, 11, 21	Level 3
12	12, 13	1, 2, 6, 7, 8, 9, 10, 11, 12, 21	Level 6
13	13	1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 21	Level 7
14	14, 15, 16, 17, 18, 19	1, 2, 6, 7, 8, 9, 10, 11, 14, 21	Level 4
15	15	1, 2, 6, 7, 8, 9, 10, 11, 14, 15, 21	Level 7
16	16, 18, 19	1, 2, 6, 7, 8, 9, 10, 11, 14, 16, 21	Level 5
17	17, 18, 19	1, 2, 6, 7, 8, 9, 10, 11, 17, 21	Level 5
18	18, 19	1, 2, 6, 7, 8, 9, 10, 11, 14, 16, 17, 18, 21	Level 6
19	19	1, 2, 6, 7, 8, 9, 10, 11, 14, 16, 17, 18, 19, 21	Level 7
20	20	1, 2, 6, 7, 8, 9, 10, 11, 20, 21	Level 7
21	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28	21	Level 1
22	22	1, 2, 21, 22	Level 7
23	23	1, 2, 21, 23	Level 7
24	24	1, 2, 21, 24	Level 7
25	25	1, 2, 6, 7, 8, 9, 10, 11, 21, 25	Level 7
26	26	1, 2, 6, 7, 8, 9, 10, 11, 21, 26	Level 7
27	27	1, 2, 6, 7, 8, 9, 10, 11, 21, 27	Level 7
28	28	1, 2, 6, 7, 8, 9, 10, 11, 21, 28	Level 7

$$\mu_{\tilde{A}}(x) = \begin{bmatrix} 0 & x < a \\ \frac{x-a}{m-a} & a \leq x < m \\ \frac{b-x}{b-m} & m \leq x \leq b \\ 0 & x > b \end{bmatrix} \quad (1)$$

The initial task in this step is to create linguistic likelihood and impact matrices. The linguistic likelihood matrix is constructed by replacing each element e_{ij} having value "1" in the initial reachability matrix with the likelihood that a change in SR_i would trigger a change in SR_j, employing the linguistic variables described in Table 4. Similarly, the linguistic impact matrix is constructed by replacing each element e_{ij} with value "1" in the initial reachability matrix with the average impact of a change in SR_i on SR_j (should a change in SR_i occur), using the linguistic variables described in Table 5. Each linguistic term in the likelihood and impact matrices is represented by a corresponding triangular fuzzy number (Tables 4 and 5). Multiplying the corresponding elements of the direct impact matrices and fuzzy direct likelihood creates a fuzzy weighted impact matrix. Afterward, this matrix is repeatedly multiplied by itself using fuzzy matrix multiplication principles until a stabilized matrix is obtained. This matrix is called the MF-MICMAC-stabilized matrix. For the demonstrative development project, the resulting fuzzy-weighted impact matrix and the corresponding MF-MICMAC-stabilized matrix are presented in Tables 6 and 7, respectively.

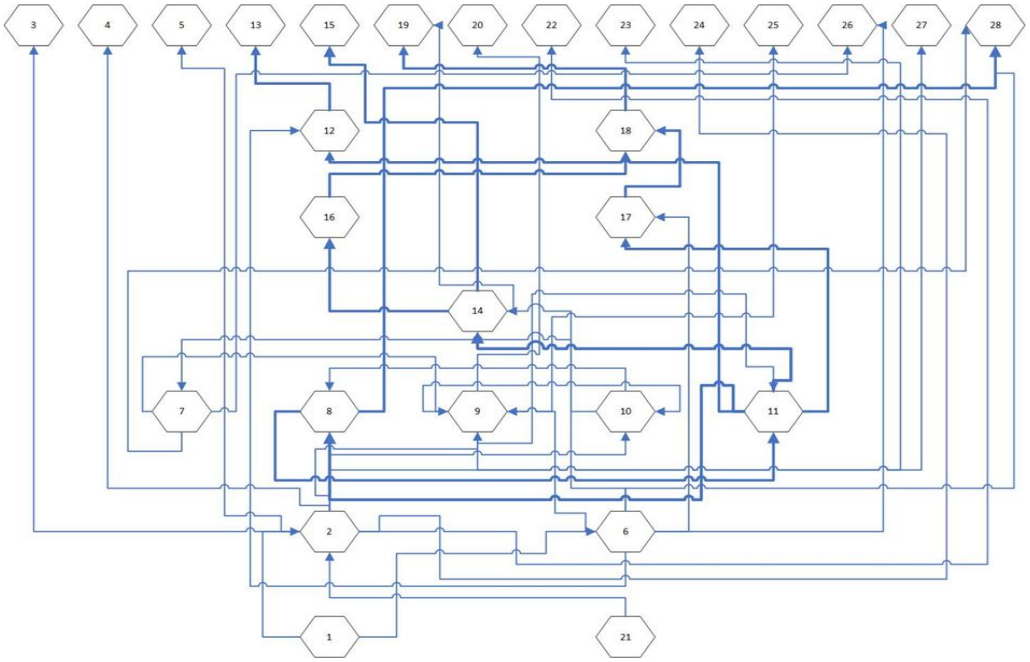

Fig.2. Hierarchical graphical model

Table 4
The fuzzy linguistic scale for the likelihood

Linguistic terms	Triangular fuzzy number
Very Unlikely (VU)	(0.0, 0.1, 0.3)
Unlikely (U)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
Likely (L)	(0.5, 0.7, 0.9)
Very Likely (VL)	(0.7, 0.9, 1.0)

Table 5
The fuzzy linguistic scale for the impact

Linguistic terms	Triangular fuzzy number
Very Low (VL)	(0.0, 0.1, 0.3)
Low (L)	(0.1, 0.3, 0.5)
Moderate (M)	(0.3, 0.5, 0.7)
High (H)	(0.5, 0.7, 0.9)
Very High (VH)	(0.7, 0.9, 1.0)

4.2.2 Classification of SRs

The second step in this stage is to categorize the SRs based on their dependence and driving powers. The sum of all values in column j of the MF-MICMAC-stabilized matrix determines the dependence power of SR_j, whereas the sum of all values in row *i* of the MF-MICMAC-stabilized matrix determines the driving power of SR_i. The computed values of dependence and driving powers are then plotted on a four-quadrant graph termed the "driving–dependence power diagram."

Table 6
Fuzzy weighted impact matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0.00	0.75	0.09	0.61	0.35	0.61	0.00	0.00	0.35	0.04	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.75	0.75	0.75	0.15	0.00	0.00	0.00

2	0.00	0.00	0.09	0.25	0.25	0.00	0.00	0.00	0.21	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.12	0.12	0.12	0.09	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.35	0.35	0.12	0.75	0.26	0.26	0.26	0.26	0.26	0.26	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.12	0.12	0.12	0.12
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.61	0.75	0.75	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.61	0.61	0.61	0.49
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.75	0.00	0.75	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.00	0.00	0.00	0.75	0.75	0.75	0.75	0.75	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.43	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 7

Stabilized matrix

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
1	0.00	0.75	0.09	0.61	0.35	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.49	0.49	0.61	0.00	0.75	0.75	0.75	0.61	0.61	0.61	0.61
2	0.00	0.00	0.00	0.25	0.25	0.00	0.09	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.25	0.00	0.12	0.12	0.12	0.21	0.09	0.21	0.21
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.61	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.49	0.49	0.75	0.00	0.00	0.00	0.00	0.75	0.61	0.75	0.61
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.61	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.49	0.49	0.75	0.00	0.00	0.00	0.00	0.75	0.61	0.75	0.61
8	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.09	0.09	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.09	0.00	0.00	0.00	0.00	0.09	0.09	0.09	0.75
9	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.49	0.00	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.49	0.49	0.75	0.00	0.00	0.00	0.00	0.75	0.09	0.75	0.49
10	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	0.09	0.00	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.00	0.00	0.00	0.09	0.09	0.09	0.09
11	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.49	0.09	0.09	0.00	0.75	0.75	0.75	0.75	0.75	0.75	0.49	0.49	0.09	0.00	0.00	0.00	0.00	0.09	0.09	0.09	0.49
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.43	0.00	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.04	0.00	0.04	0.04	0.00	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.00	0.04	0.04	0.04	0.04	0.04	0.04	0.04
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

- The first quadrant comprises autonomous SRs with weak driving and dependence powers (i.e., they have weak direct and indirect associations with other SRs).
- The second quadrant comprises dependent SRs, which have weak driving power and strong dependence power (i.e., they have a strong dependency on other SRs either directly or indirectly, whereas other SRs have weak dependencies on them).
- The third quadrant comprises linkage SRs, which have strong driving power and dependence power (i.e., they have a strong dependency on other SRs either directly or indirectly, while other SRs also have strong dependencies on them).

- The fourth quadrant comprises independent SRs, which have weak dependence power and strong driving power (i.e., they have weak or no dependencies on other SRs, whereas other SRs have high dependencies on them either directly or indirectly).

The dependence and driving power values determined for the demonstrative development project were used to construct the driving–dependence power diagram shown in Figure 3.

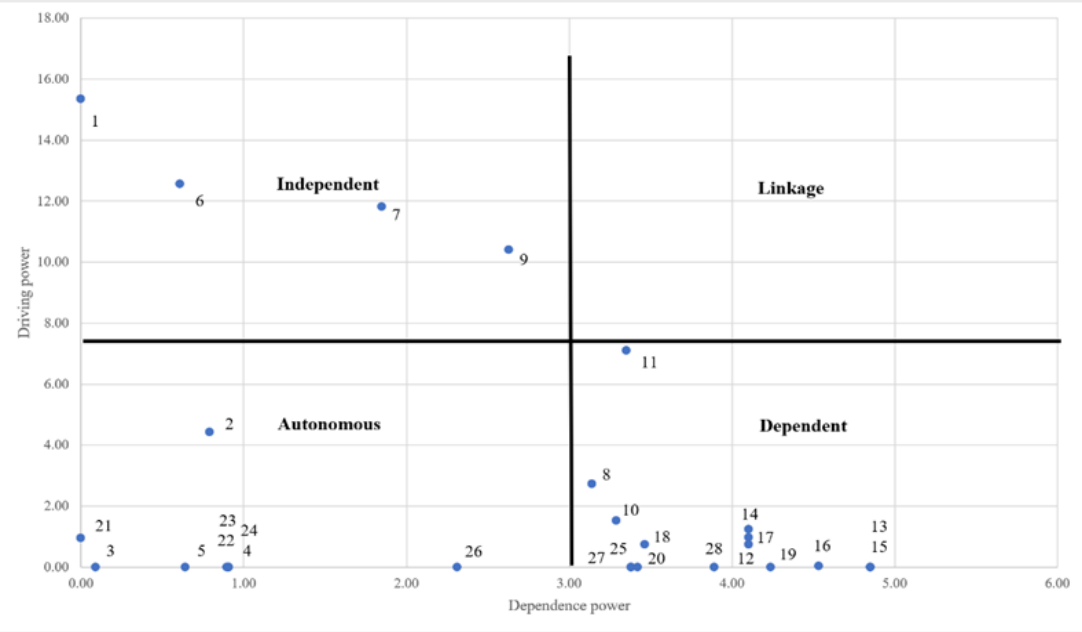

Fig.3. 'Driving-dependence power' diagram

4.3 Stage III: Optimization Model Development

Selecting SRs for change involves a fundamental trade-off between delivering stakeholder value and preserving architectural stability. Modifying an SR may trigger cascading impacts on other SRs, thereby increasing system volatility. Similar trade-offs have been addressed in dependency-aware SRs selection models that integrate optimization techniques to account for interdependencies [49]. Building on this concept, this study formulates the problem as a multi-objective combinatorial optimization problem that integrates the outputs of the ISM and MF-MICMAC analyses. Given the hierarchical nature of the objectives, where limiting change propagation is prioritized over secondary implementation considerations, a lexicographic optimization approach is adopted.

Model components

Let *I* denote the set of SRs.

The model comprises one decision variable, x_i, defined as follows:

$$x_i \in \{0,1\}, \forall i \in I$$

where x_i =1 if SR_i is selected to be changed, and x_i = 0 otherwise.

Let v_i denote the stakeholder value, or satisfaction contribution, obtained from changing SR_i.

Let r_{ij} denote the element of the final reachability matrix derived from ISM, where:

$$r_{ij} = \begin{cases} 1, & \text{if a change in } \mathrm{SR}_i \text{ can propagate to } \mathrm{SR}_j \\ 0, & \text{otherwise} \end{cases}$$

The binary impact variable Y_j is defined as follows:

$$Y_j \in \{0,1\}, \forall_j \in I$$

where Y_j =1 if SR_j is impacted by at least one selected change, as determined from the final reachability matrix, and Y_j = 0 otherwise.

Let FI_i denote the fuzzy driving power of SR, calculated as the row sum of the stabilized fuzzy reachability matrix obtained from the MF-MICMAC analysis.

Constraints

The minimum satisfaction constraint is formulated as:

$$\sum_{i \in I} v_i x_i \geq S_{min} \quad (2)$$

$$\text{where} \quad S_{min} \in [S_{lower}, S_{upper}]$$

This constraint ensures that the selected set of SRs achieves a minimum acceptable requirement satisfaction level ($S_{\underline{min}}$), while allowing flexibility to explore alternative satisfaction thresholds.

The change propagation constraints are formulated as:

$$Y_j \geq x_i r_{ij} \quad \forall i \in I, \ j \in J \quad (3)$$
$$Y_j \leq \sum_{i \in I} x_i r_{ij} \quad \forall j \in J \quad (4)$$

These constraints jointly define Y_j as a union-based impact indicator, ensuring that an SR is classified as impacted if and only if at least one selected SR propagates a change to it.

Optimization structure

Stage 1: Minimization of propagation breadth

$$\min \mathrm{Z}_1 = \sum_{j \in \mathrm{I}} Y_j \quad (5)$$

This objective minimizes the number of impacted SRs, thereby limiting the breadth of change propagation and preserving architectural stability.

Stage 2: Minimization of propagation intensity

$$\min \mathrm{Z}_2 = \sum_{i \in I} FI_i x_i \quad (6)$$

This objective minimizes the total intensity of change propagation by selecting SRs with less fuzzy driving power while maintaining the minimum propagation breadth achieved in Stage 1.

5.0 Discussion

This study proposes a three-stage decision-support framework for modeling, analyzing, and managing dependencies among SRs by integrating ISM, modified fuzzy MICMAC (MF-MICMAC) analysis, and a lexicographic optimization model. ISM visualizes SR interrelationships through a hierarchical graphical structure that explicitly reveals both direct and multilevel dependency paths. Building on this structure, the MF-MICMAC analysis evaluates the criticality of these dependencies by simultaneously considering the likelihood of change propagation and its potential impact. Finally, the optimization model operationalizes the outputs of the first two stages by identifying SR change combinations that satisfy stakeholder needs while minimizing both the breadth and intensity of change propagation.

5.1 Managing Change Propagation through ISM

The ISM hierarchical model provides an intuitive visualization of SR dependencies, allowing development teams to understand how SR changes propagate in the system and to identify both direct and indirect propagation paths. Supporting this, studies show that graphical feedback leads to more complete and faster learning in multidimensional information-processing tasks than numerical feedback [50]. In addition, graphical representations enhance the ability of decision-makers to identify dependency patterns and relationships between variables [51, 52].

The developed ISM model for the demonstrative development project comprises seven levels of hierarchy. SR1 and SR21 are at the lowest level of the hierarchy, whereas SR3–SR5, SR13, SR15, SR19, SR20 and SR22–SR28 are at the highest level. As a consequence, changes made at lower hierarchical levels may propagate through several intermediate SRs before they reach higher levels. This hierarchical model allows project teams to identify likely propagation paths for proposed SR changes. For example, Figure 2 shows an initial change to SR11 that can cause direct and indirect changes to multiple SRs, with the respective change propagation paths indicated by the bold arrows. Such information helps project managers to understand the consequences of proposed changes and evaluate the potential impact before implementation.

5.2 Prioritizing SR Changes Using MF-MICMAC

Although the ISM model reveals dependency structures, it does not indicate which SRs are most critical from a change management perspective. The proposed MF-MICMAC analysis addresses this limitation by classifying SRs according to their driving and dependence characteristics while evaluating dependency strength based on both the likelihood and impact of change propagation. This dual-dimensional assessment provides a more comprehensive evaluation of SR criticality than the traditional fuzzy MICMAC analysis, which considers only the impact dimension.

The classification of SRs into autonomous, dependent, linkage, and independent groups provides practical guidance for SR change management. Independent SRs represent the most influential group because changes to these SRs are likely to trigger extensive change propagation throughout the

system. Consequently, changes to these SRs should be carefully evaluated and implemented only when necessary to satisfy stakeholder needs. For the demonstrative project, four SRs (SR1, SR6, SR7, and SR9) were classified as independent, whereas no linkage re SRs were identified.

In contrast, autonomous and dependent SRs generally have relatively limited propagation effects and therefore present lower implementation risks. In the demonstrative project, nine SRs (SR2–SR5, SR21–SR24, and SR26) were classified as autonomous, whereas fifteen SRs (SR10–SR20, SR25, SR27, and SR28) were classified as dependent. These classifications provide useful guidance when evaluating alternative changes. When two alternative SRs belong to the same category, selection can primarily be based on their contribution to SRs that satisfy stakeholder requirements, since their propagation characteristics are expected to be similar. However, when competing alternatives belong to different categories but provide comparable stakeholder value, preference should generally be given to the SRs in the less critical category to reduce the risk of widespread propagation of change.

5.3 Supporting SR Change Decisions through Optimization

While the ISM and MF-MICMAC analyses provide valuable structural and criticality information, they do not directly determine the optimal combination of SRs to change when multiple feasible alternatives exist. The third stage of the proposed framework addresses this limitation by integrating the outputs of the ISM and MF-MICMAC analyses within a lexicographic optimization model. Specifically, the Final Reachability Matrix generated during the ISM analysis determines the propagation breadth for each candidate SR, whereas the fuzzy propagation intensity values from the MF-MICMAC analysis quantify the severity of change propagation.

To demonstrate the practical application of this third stage of the proposed framework, consider a scenario in which the objective is to satisfy the high-level customer need to strengthen access control and administrative functionality in HRM. This objective can be achieved through changes to one or more of five candidate SRs (SR1, SR6, SR10, SR11, and SR25). The corresponding stakeholder satisfaction values (v_i) and fuzzy propagation intensity values (FI_i), obtained from the MF-MICMAC analysis, are summarized in Table 8.

5.3.1 Construction of optimization inputs

The optimization model is based on three types of information generated in the preceding stages of the proposed framework. First, the stakeholder satisfaction values (v_i) show how much each candidate RS contributes to the defined customer need. Secondly, the Final Reachability Matrix obtained from the ISM analysis determines the scope of propagation for each candidate SR by identifying all SRs that would be directly or indirectly affected by implementing changes in the candidate SR. Third, the MF-MICMAC analysis provides the fuzzy intensity of propagation values (FI_i) that represent the potential severity of propagated changes associated with each candidate SR. The three sources of information serve as inputs to the proposed lexicographic optimization model.

Table 8
Candidate SRs data

SR$_i$	**Description**	**Satisfaction (v_i)**	**Fuzzy intensity (FI_i)**
SR1	Login feature based on username and	0.4	15.34

SR_i	Description	Satisfaction (v_i)	Fuzzy intensity (FI_i)
SR6	Interface and data based on user role	0.7	12.56
SR10	The user role can be checked from the	0.3	1.53
SR11	Users with defined roles can display database	0.5	7.10
SR25	Admin can edit the user's role	0.2	0.00

Analysis of the Final Reachability Matrix reveals three important propagation characteristics. First, selecting SR1 results in the broadest propagation effect, impacting 26 SRs, corresponding to a propagation breadth of $Z_1 = 26$. Second, selecting SR6 produces a considerably smaller propagation footprint, affecting only 18 SRs ($Z_1 = 18$). Finally, the reachability sets of SR10, SR11, and SR25 are completely contained within the reachability set of SR6. Consequently, selecting any of these SRs together with SR6 does not increase the overall propagation breadth beyond 18 impacted SRs.

These observations suggest that different candidate SRs may contribute similarly to meeting stakeholder needs but have very different propagation characteristics. Therefore, the choice of the most appropriate SRs cannot rely only on the analysis of stakeholder satisfaction or dependency. Instead, it needs an optimization procedure to simultaneously balance stakeholder satisfaction, propagation breadth, and propagation intensity.

5.3.2. Decision Analysis

The proposed lexicographic optimization model was solved for three minimum thresholds of stakeholder satisfaction. During the first optimization stage, the model minimizes the propagation breadth (Z_1). Subsequently, among all solutions having the same minimum propagation breadth, the second stage minimizes the total propagation intensity (Z_2). The resulting optimal SR combinations are presented in Table 9.

For Case A, several combinations meet the stakeholders' minimum satisfaction level of 0.9. However, the optimization model finds that {SR6, SR25} is the optimal solution, as it simultaneously achieves the smallest propagation breadth (18 impacted SRs) and the smallest propagation intensity (12.56). While SR1 could also be used to satisfy the required stakeholder value, it is a less desirable alternative due to its significantly larger propagation footprint (26 impacted SRs).

Table 9
Optimal SR combinations for different stakeholder satisfaction thresholds

Case	Minimum stakeholder satisfaction (S_{min})	Optimal SR set	Breadth (Z_1)	Intensity (Z_2)
A	0.9	{SR6, SR25}	18	12.56
B	1.0	{SR6, SR10}	18	14.09
C	1.2	{SR6, SR11}	18	19.66

When the minimum stakeholder satisfaction threshold is increased to 1.0 (Case B), the combination {SR6, SR25} is no longer feasible. Among other feasible alternatives, the optimization model chooses {SR6, SR10}, as it has the minimum propagation breadth, and produces the smallest propagation intensity (14.09). This solution is therefore the best trade-off between higher stakeholder satisfaction and risk of change propagation.

For Case C, the required stakeholder satisfaction threshold is further increased to 1.2. Under this constraint, the optimal solution becomes {SR6, SR11}. Although this solution maintains the same minimum propagation breadth, the overall propagation intensity increases to 19.66, reflecting the

greater influence of SR11 on propagation compared with SR10 and SR25. These results illustrate the expected trade-off between achieving higher stakeholder satisfaction and limiting the severity of propagated changes.

5.4 Comparative Discussion with Related Study

Bashir et al. [36] conducted a related study using a binary SNA-based approach to model and analyze the same set of 28 SRs from the HRMS project. Based on in-degree, out-degree, and betweenness centrality measures, the study identified SRs 1, 2, 6, 7, and 11 as critical. One of the limitations of the approach by Bashir et al. [36], is that as the number of SRs increases, the resulting network graphs become complex. In contrast, the present study adopts an integrated ISM and MF-MICMAC approach to model dependencies using a simple hierarchical graph. Moreover, based on both the likelihood of occurrence and the impact of change, the MICMAC analysis identified SRs 1, 6, 7, and 9 as critical.

The consistent identification of SRs 1, 6, and 7 across both approaches confirms their role as stable drivers within the system rather than artifacts of a particular analytical approach, indicating robustness in the underlying dependency structure. The noted difference, specifically the prominence of SRs 2 and 11 in the SNA results and SR 9 in the MICMAC results, arises from the unique analytical viewpoints of the two approaches. While SNA highlights structural significance and mediation in the dependency network, ISM and MICMAC concentrate on hierarchy and cumulative influence. Consequently, the ISM–MF-MICMAC approach provides a more effective basis for prioritizing SR changes and supporting change management decisions. Importantly, the inclusion of the optimization model in the present study translates these analytical insights into actionable decision support by identifying SR changes that satisfy customer needs while minimizing change propagation. As a result, the proposed ISM–MF-MICMAC–optimization framework is better suited to managing SR changes in practice.

6.0 Conclusions

Encountering changes in SRs is typical in any software development process. A critical challenge in dealing with these changes is evaluating their consequences during requirements analysis and at the start of the stage, especially in projects that involve developing software from scratch. Addressing this challenge requires tools that enable development teams to model, analyze, and manage dependencies among SRs at early stages of the development process.

To this end, this study proposes a three-stage decision-support framework for systematically modeling, analyzing, and managing dependencies among SRs. In the first stage, ISM is employed to develop a hierarchical graphical model that provides clear insight into multilevel interactions among SRs. In the second stage, integrating ISM with MF-MICMAC analysis enables the classification of SRs into autonomous, dependent, independent, and linkage groups by jointly considering the likelihood of occurrence and the impact of change propagation.

Autonomous SRs are identified as the least critical, whereas independent SRs are the most critical in terms of their potential cascading effects. Thus, changes to independent SRs should not be performed unless they are necessary to meet customer requirements. When such changes are unavoidable, these SRs require close attention, as changes may lead to substantial rework and increased risks of cost overruns and schedule delays. Although this classification offers useful guidance for understanding SR criticality, the framework goes beyond analytical classification by

incorporating an optimization-based decision support model, which provides a systematic evaluation of alternative combinations of SR changes and supports the selection of changes that satisfy functional requirements while minimizing the breadth and intensity of change propagation, thus enabling more informed and robust change management decisions.

The applicability of the proposed framework is demonstrated by its use in a real-world software development project involving 28 SRs. This scale is widely used in the literature for demonstration and validation because it allows for a clear and systematic presentation of methodological procedures and results. However, the computational complexity of the framework increases as the number of SRs increases. Since the dependency matrix is based on pairwise assessment, its construction grows quadratically with the number of SRs. In addition, the ISM and MF-MICMAC stages involve matrix-based operations, while the optimization stage becomes more complex as the number of decision variables and dependency constraints increases. Therefore, although the framework is practical for small- and medium-sized projects, applying it to projects with hundreds or thousands of SRs may require decomposing the requirements into modules, subsystems, or functional groups. Accordingly, empirical studies are required to validate the robustness and generalizability of the framework across projects of different sizes, complexity levels, and application domains. Although the framework is demonstrated using an HRMS development project, its logic is not limited to this domain. The framework can be generalized to other software domains, such as healthcare, finance, and embedded systems, provided that the relevant SRs and their dependency relationships can be identified by domain experts or extracted from requirement documentation. Moreover, when applying the framework in industry, users can further assess the stability of the optimization results by changing the assigned likelihood and impact values and comparing the resulting requirement-change selections.

Further validation is also required, and the framework needs to be integrated with automated tools or existing software engineering workflows to improve scalability and support practical implementation. Future research may also explore the use of machine learning and large language models to support the automated extraction of dependencies among SRs, thereby reducing reliance on manual expert assessment and improving the scalability of the proposed framework. Such integration would allow more SRs to be managed more efficiently, provide real-time decision support for SR change management, and enable more effective management of dynamic software development environments.

Author Contributions

Conceptualization, H.B.; methodology, H.B., M.A., H.A.A, S.H., and M.S.; software, H.B. and M.A.; validation, H.B.; formal analysis, H.B. and M.A.; investigation, M.A.; data curation, M.A.; writing—original draft preparation, H.B.; writing—review and editing, H.B., H.A.A, S.H., and M.S.; visualization, M.A.; supervision, H.B.; project administration, H.B. All authors have read and agreed to the published version of the manuscript.” Authorship must be limited to those who have contributed substantially to the work reported.

Funding

This research received no external funding.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] Thakurta, R. (2015). Projects as temporary organizations: insights from requirement volatility', *International Journal of Project Organization and Management*, 7(2), 184–202, https://doi.org/10.1504/IJPOM.2015.069617

[2] da Cruz Mello, O., & Manzoni Fontoura, L. (2023). Improving the evaluation of change requests using past cases. *International Journal of Information Systems and Project Management*, 11(1), Article 5. https://doi.org/10.12821/ijispm110104

[3] Yin, L., Liu, Y., Shen, Z., & Li, Y. (2018, August). A technique to predict software change propagation. In *International Conference on Computer Engineering and Networks* (pp. 538–543). Cham: Springer International Publishing.

[4] Bhatti, M. W., Hayat, F., Ehsan, N., Ishaque, A., Ahmed, S., & Sarwar, S. Z. (2010). An investigation of changing requirements with respect to development phases of a software project. *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 323–327. IEEE. https://doi.org/10.1109/CISIM.2010.5643639

[5] Ingram, C. (2011). *Using requirements and design information to predict volatility in software development* (Doctoral dissertation, Newcastle University).

[6] Iriarte, C., & Bayona, S. (2020). IT projects success factors: A literature review. *International Journal of Information Systems and Project Management*, 8(2), 49–78.

[7] Jones, C.T. (2007). Estimating software costs (2nd ed.). New York.

[8] Thakurta, R., & Ahlemann, F. (2011). Understanding requirement volatility in software projects. *Engineering Management Journal*, 23(3), 3–7. https://doi.org/10.1080/10429247.2011.11431900

[9] Hashmi, S. I., Lane, S., Karastoyanova, D., & Richardson, I. (2010). A CMMI-based configuration management framework to manage the quality of service-based applications. In *EuroSPI: Industrial Proceedings of the 17th European Conference on Software Process Improvement*, Grenoble, France, September 1–3.

[10] Wiegers, K.E. (2003). *Software requirements* (2nd ed.). Microsoft Press.

[11] Young, R.R. (2001). *Effective requirements practices*, Addison-Wesley, Boston.

[12] O'Raghallaigh, P., Lane, S., Adam, F., & Sammon, D. (2020). Difficult knowledge boundaries during requirements determination. *Journal of Decision Systems*, 29(sup1), 270–284. https://doi.org/10.1080/12460125.2020.1848342

[13] Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., & Natt och Dag, J. (2001). An industrial survey of requirements interdependencies in software product release planning. *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, 84–91. https://doi.org/10.1109/ISRE.2001.948547

[14] Jayatilleke, S., & Lai, R. (2018). A systematic review of requirements change management. *Information and Software Technology*, 93, 163–185. https://doi.org/10.1016/j.infsof.2017.09.004

[15] Marchwicka, E. (2020). A technique for supporting decision process of global software project monitoring and rescheduling based on risk analysis. *Journal of Decision Systems*, 29(sup1), 398–412. https://doi.org/10.1080/12460125.2020.1790825

[16] Liu, X., & Smolka, S. A. (1998, July). Simple linear-time algorithms for minimal fixed points. In *International Colloquium on Automata, Languages, and Programming* (pp. 53-66). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0055040

[17] Goknil, A., Kurtev, I., & van den Berg, K. (2008, June). Change impact analysis based on formalization of trace relations for requirements. In *ECMDA Traceability Workshop*, ECMDA-TW 2008 (pp. 59-75). SINTEF Report.

[18] Goknil, A., Kurtev, I., van den Berg, K., & Spijkerman, W. (2014). Change impact analysis for requirements: A metamodeling approach. *Information and Software Technology*, 56(8), 950–972. https://doi.org/10.1016/j.infsof.2014.03.002

[19] Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.) (2012). *Software and systems traceability*. Springer. https://doi.org/10.1007/978-1-4471-2239-5

[20] Sangal, N., Jordan, E., Sinha, V., & Jackson, D. (2005, October). Using dependency models to manage complex software architecture. In *Proceedings of the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 167-176). https://doi.org/10.1145/1094811.1094824

[21] Widiastuti, M., & Siahaan, D. (2008). Mapping the impact of requirement changes using LT-RC. In *Proceedings of the 4th International Conference on Information & Communication Technology and Systems (ICTS 2008)* (pp. 315–319).

[22] van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. *Proceedings of the 5th IEEE International Symposium on Requirements Engineering*, 249–262. https://doi.org/10.1109/ISRE.2001.948567

[23] Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of the Third IEEE International Symposium on Requirements Engineering* (RE'97), 226–235. https://doi.org/10.1109/ISRE.1997.566873

[24] Horkoff, J., Aydemir, F. B., Cardoso, E., Li, T., Maté, A., Paja, E., Salnitri, M., & Giorgini, P. (2015). Goal-oriented requirements engineering: An extended systematic mapping study. *Requirements Engineering*, 20(2), 133–160. https://doi.org/10.1007/s00766-014-0180-z

[25] Fu, Y., Li, M., & Chen, F. (2012). Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management*, 30(3), 363–373. https://doi.org/10.1016/j.ijproman.2011.08.004

[26] Ali, N., & Lai, R. (2016). A method of requirements change management for global software development. *Information and Software Technology*, 70, 49–67. https://doi.org/10.1016/j.infsof.2015.09.005

[27] Jayatilleke, S., Lai, R., & Reed, K. (2018). A method of requirements change analysis. *Requirements Engineering*, 23, 493–508. https://doi.org/10.1007/s00766-017-0277-7

[28] Priyadi, Y., Djunaidy, A., & Siahaan, D. (2019). Requirements dependency graph modeling on software requirements specification using text analysis. In *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)* (Vol. 1, pp. 221–226). IEEE. https://doi.org/10.1109/ICORIS.2019.8874920

[29] Bohner, S. A. (1995). *A graph traceability approach for software change impact analysis* [Doctoral dissertation, George Mason University].

[30] Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., & Carley, K. M. (2006*).* Identification of coordination requirements: Implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work (pp. 353–362). ACM. https://doi.org/10.1145/1180875.1180929*

[31] Arora, C., Sabetzadeh, M., Goknil, A., Briand, L. C., & Zimmer, F. *(*2015, August). Change impact analysis for natural language requirements: An NLP approach. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)* (pp. 6–15). *IEEE. https://doi.org/10.1109/RE.2015.732040*

[32] Conejero, J. M., Figueiredo, E., Garcia, A., Hernández, J., & Jurado, E. (2012). On the relationship of concern metrics and requirements maintainability. *Information and Software Technology*, 54(2), 212–238. https://doi.org/10.1016/j.infsof.2011.09.003

[33] Hassine, J., Rilling, J., Hewitt, J., & Dssouli, R. (2005, September). Change impact analysis for requirement evolution using use case maps. In *Eighth International Workshop on Principles of Software Evolution (IWPSE'05)* (pp. 81–90). IEEE. https://doi.org/10.1109/IWPSE.2005.8

[34] Nejati, S., Sabetzadeh, M., Arora, C., Briand, L. C., & Mandoux, F. (2016, November). Automated change impact analysis between SysML models of requirements and design. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 242-253). https://doi.org/10.1145/2950290.2950293

[35] Arvanitou, E.-M., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P., & Tsiridis, N. (2022). A metric for quantifying the ripple effects among requirements. *Software Quality Journal*, 30. https://doi.org/10.1007/s11219-021-09581-y

[36] Bashir, H., Alhammadi, M., Ojiako, U., Shamsuzzaman, M., & Haridy, S. (2025). A social network analysis-based approach for modelling and analyzing dependencies among requirements in new software development projects. *International Journal of Project Organization and Management*, 17(4), 400–420. https://doi.org/10.1504/IJPOM.2025.150116

[37] Bendoly, E. (2014). System dynamics understanding in projects: Information sharing, psychological safety, and performance effects. *Production and Operations Management*, 23(8), 1352–1369. https://doi.org/10.1111/poms.12024

[38] Warfield, J. N. (1974). Developing interconnection matrices in structural modeling. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-4(1), 81–87. https://doi.org/10.1109/tsmc.1974.5408524

[39] Hughes, D. L., Dwivedi, Y. K., Rana, N. P., & Simintiras, A. C. (2016). Information systems project failure – analysis of causal links using interpretive structural modelling. *Production Planning & Control*, 27(16), 1313–1333. https://doi.org/10.1080/09537287.2016.1217571

[40] Duperrin, J. C., & Godet, M. (1973). *Méthode de hiérarchisation des éléments d'un système: essai de prospective du système de l'énergie nucléaire dans son contexte sociétal* [*Method for prioritizing elements of a system: A prospective analysis of the nuclear energy system in its societal context*.]. (Doctoral dissertation, Centre national de l'entrepreneuriat (CNE); CEA).

[41] Bagherian, A., Gershon, M., Kumar, S., & Mishra, M. K. (2024). Analyzing the relationship between digitalization and energy sustainability: A comprehensive ISM–MICMAC and DEMATEL approach. *Expert Systems with Applications*, 236, 121193. https://doi.org/10.1016/j.eswa.2023.121193

[42] Bashir, H., & Ojiako, U. (2020). An integrated ISM-MICMAC approach for modelling and analysing dependencies among engineering parameters in the early design phase. *Journal of Engineering Design*, 31(8-9), 461–483. https://doi.org/10.1080/09544828.2020.1817347

[43] Karamat, J., Shurong, T., Ahmad, N., Waheed, A., & Khan, S. (2018). Barriers to knowledge management in the health sector of Pakistan. *Sustainability*, 10(11), 4155. https://doi.org/10.3390/su10114155

[44] Sindhwani, R., & Malhotra, V. (2017). Modelling and analysis of agile manufacturing system by ISM and MICMAC analysis. *International Journal of System Assurance Engineering and Management*, 8(2), 253–263. https://doi.org/10.1007/s13198-016-0426-2

[45] Albastaki, F. M., Bashir, H., Ojiako, U., Shamsuzzaman, M., & Haridy, S. (2021). Modeling and analyzing critical success factors for implementing environmentally sustainable practices in a public utilities organization: A case study. *Management of Environmental Quality: An International Journal*, 32(4). https://doi.org/10.1108/meq-01-2021-0002

[46] Bashir, H., Ojiako, U., & Mota, C. (2020). Modeling and analyzing factors affecting project delays using an integrated social network-fuzzy MICMAC approach. *Engineering Management Journal*, 32(1), 26-36. https://doi.org/10.1080/10429247.2019.1656595

[47] Zadeh, L. A. (1999). Some reflections on the anniversary of Fuzzy Sets and Systems. *Fuzzy Sets and Systems*, 100, 1–3.

[48] Pedrycz, W. (1994). Why triangular membership functions? *Fuzzy Sets and Systems*, 64(1), 21–30. https://doi.org/10.1016/0165-0114(94)90003-5

[49] Mougouei, D., & Powers, D. M. W. (2021). Dependency-aware requirements selection using value dependency graphs and optimization. *Expert Systems with Applications*, 167, 113748.

[50] Hoffman, P. J., Earle, T. C., & Slovic, P. (1981). Multidimensional functional learning (MFL) and some new conceptions of feedback. *Organizational Behavior and Human Performance*, 27(1), 75–102. https://doi.org/10.1016/0030-5073(81)90040-4

[51] DeSanctis, G. (1984). Computer graphics as decision aids: Directions for research. *Decision Sciences*, 15(4), 463–487. https://doi.org/10.1111/j.1540-5915.1984.tb01236.x

[52] Dickson, G. W., DeSanctis, G., & McBride, D. J. (1986). Understanding the effectiveness of computer graphics for decision support: A cumulative experimental approach. *Communications of the ACM*, 29(1), 40–47. https://doi.org/10.1145/5465.5469