International Journal for Quality Research 19(2) 451-470
ISSN 1800-6450

Ayça Maden [1]
Emre Alptekin

Article info:
Received 24.06.2024.
Accepted 20.01.2025.

DOI – 10.24874/IJQR19.02-06

SOFTWARE QUALITY EVALUATION USING SQUARE SERIES AND SCENARIO-BASED FUZZY COGNITIVE MAP

Abstract: *Software quality assessment requires a thorough examination of product quality, usability, and data integrity. While conventional quality attributes have their merits, international standards often offer broad quality frameworks that can pose challenges when applied to specific domains. This study adopts a case study approach to analyse the software needs of a company and its supply chain, utilizing ISO/IEC 25010 and ISO/IEC 25012 standards. Through scenario analysis using Fuzzy Cognitive Maps, the research delves into the relationships among ISO software characteristics and their impact on outcomes. The study's significance lies in evaluating trade-offs among software quality attributes, evaluating them within specific domains, and presenting a comprehensive assessment spanning product, user, and data dimensions. This research enhances comprehension of how software quality characteristics shape outcomes and provides valuable insights for tailored evaluations in diverse industries.*

Keywords: *software quality, fuzzy cognitive map, scenario analysis, SQuaRE, ISO standards*

1. Introduction

Software product quality reflects the degree to which essential requirements, such as the software's purpose, performance, usability, and maintainability, are fulfilled (Nakai et al. 2017). If the quality falls short, it may result in increased development or maintenance expenses and a disregard for user requirements. Additionally, comparing non-standard quality frameworks, metrics, and models specific to organizations is challenging, as they often focus solely on particular quality characteristics. However, while some software quality models offer standard characteristics, they are often too generalized and fail to address domain-specific requirements for a particular field or industry. Therefore, there is a need to shift focus towards domain-specific applications in this regard. Moreover, the characteristics used to evaluate software quality are often narrowly focused. For instance, when assessing the quality of software products, it is crucial to consider user perspectives. However, there remains a scarcity of research focusing on evaluating the quality of software products from the user's standpoint. Additionally, integrating characteristics related to data into the evaluation of software quality is of paramount importance. In this study, a case study application was conducted, considering the software needs of a specific company in conjunction with its supply chain. To accomplish this, ISO/IEC 25010 (ISO/IEC 2011) and ISO/IEC 25012 (ISO/IEC n.d.) standard quality

[1] Corresponding author: Ayça Maden
Email: aycamaden@beykent.edu.tr

characteristics, covering product quality, quality in use, and data quality characteristics, were employed. This approach facilitated the swift identification of problematic software quality characteristics within the case company from a broader perspective, utilizing a standardized language.

A trade-off refers to a characteristic that affects multiple attributes simultaneously. In the realm of software architecture, it is essential to acknowledge that every decision made within a system involves balancing various characteristics (Maden & Yücenur 2024; Paik et al. 2019). For instance, enhancing privacy in a particular application might necessitate the implementation of additional controls, potentially impacting its performance. Hence, recognizing both positive and negative relationships among the software characteristics used in assessing software quality is crucial. In this study, a scenario-based FCM enabled us to evaluate trade-offs among the software quality characteristics. The contribution of this study to the literature can be summarized in three main points. Firstly, it involves the assessment of trade-offs among software quality characteristics. Secondly, with the aid of scenario analysis, it evaluates standard software quality characteristics in a domain-specific manner, focusing on the supply chain. Thirdly, it offers a comprehensive assessment of software quality from the perspectives of software product, user, and data.

The structure of this paper is outlined as follows: Section 1 introduces the introduction, Section 2 discusses literature review, Section 3 details the methodology, Section 4 presents a case study, and Section 5 concludes the paper.

2. Literature Review

Various software quality models exist, including McCall's (McCall, 1977), Boehm's (Boehm, 1978), FURPS (Grady, 1992), Dromey's (Dromey, 1996), and ISO 25010 (ISO/IEC 2011). Among these models, ISO 25010 (ISO/IEC 2011) model stands out as the most comprehensive, encompassing a wide array of characteristics. Standard quality attributes offer several advantages in the field of software development and evaluation. These attributes, as defined by international standards such as ISO/IEC 25010, provide a well-established and widely accepted framework for assessing software quality. They offer a common language and set of criteria that facilitate communication among stakeholders, including developers, testers, and end-users. By adhering to standard quality attributes, software professionals can ensure a systematic and comprehensive approach to quality assurance. Furthermore, these standards aid in establishing benchmarks and best practices, promoting a more consistent and reliable evaluation process across diverse projects. While standard quality attributes offer certain advantages, existing international standards provide broad quality models and guidelines that can be challenging to implement effectively in specific domains (Bertoa, Troya, and Vallecillo, 2006; Yuen, & Lau 2011; C. Yue, 2020). Each software development domain may possess unique requirements, constraints, and user expectations that might not be adequately addressed by the standardized models. Tailoring these broad standards to suit specific contexts necessitates a deep understanding of the domain-specific nuances and a careful balancing act to prioritize the most relevant quality attributes. The assessment of software quality is a multifaceted undertaking that involves consideration of numerous dimensions. Despite the complexity inherent in evaluating software, the current body of literature has not extensively explored the application of Multi-Criteria Decision-Making (MCDM) methods in this context. MCDM methodologies provide a systematic approach to decision-making by considering

multiple criteria and their interdependencies. By employing MCDM methods, researchers and practitioners can simultaneously take into account various criteria, such as performance, reliability, usability, and security, offering a more holistic understanding of software quality. Exploring this approach may unveil new insights and contribute to a more sophisticated and adaptive framework for assessing the multifaceted nature of software quality. Software quality evaluation studies using MCDM methods are represented in Table 1.

Table 1. Software quality evaluation studies using MCDM methods

Study	Method	Standard Quality Attributes
(Yuen and Lau, 2011)	Fuzzy group Analytical Hierarchy Process (AHP)	ISO/IEC9126-1:2001
(C. Yue, 2020)	Intuitionistic fuzzy vector, TOPSIS	N/A
(Khan et al., 2020)	ANP, TOPSIS	N/A
(Botchway et al., 2021)	AHP	Some existing software quality models
(Chang et al.,2008)	FAHP	ISO 9126-1
(Thapar and Sarangal, 2020)	FAHP	N/A
(Dayanandan and Kalimuthu, 2018)	AHP, Buckley method	ISO 25010
(Z. L. Yue et al., 2023)	VIKOR	N/A

*N/A: Not Available

After thoroughly reviewing the research presented in Table 1, it is evident that researchers have predominantly employed the AHP as the approach for evaluating software quality criteria. However, it is acknowledged that AHP becomes challenging when addressing issues that encompasses more than seven (or seven plus or minus two) criteria or alternatives (Kaviani et al. 2020; Saaty and Ozdemir 2003). This challenge arises from the potential disruption of consistency and redundancy in judgments within the matrices utilized for pairwise comparisons. When examining studies in the literature, it becomes apparent that the criteria used to evaluate software quality have generally assessed it through a narrow framework. However, software quality evaluation necessitates a comprehensive examination from various perspectives, encompassing product quality, quality in use, and data considerations. Product quality focuses on the inherent characteristics of the software, ensuring that it meets specified requirements and functions reliably. Quality in use assesses how well the software performs in real-world scenarios, considering aspects such as usability and user satisfaction. Additionally, the evaluation of data quality is vital, as software often operates on and generates data, necessitating accuracy, integrity, and security. By addressing these broad perspectives, the evaluation process ensures a holistic understanding of software quality, encompassing both its intrinsic attributes and its performance in practical applications. Furthermore, acknowledging the importance of data quality is essential in supporting the software's overall functionality and reliability. In literature, limited research exists on assessing the quality of software products from the standpoint of user perspectives. However, the quality of a product directly impacts its usability. Put simply, the effectiveness of a product in use is contingent upon its inherent quality. Therefore, a comprehensive evaluation of software quality should consider both the intrinsic characteristics of the software and its usability from the user's perspective. This study employed the ISO/IEC 25010 (ISO/IEC 2011) and ISO/IEC 25012 (ISO/IEC n.d.) standards for

evaluating software quality from broad perspectives, including product quality, quality in use, and data considerations.

This research employs a scenario-based Fuzzy Cognitive Map (FCM) methodology due to several factors: the presence of cause-and-effect relationships among software characteristics, the occurrence of conflicting characteristics, the existence of both positive and negative relations among these characteristics, and the lack of precise data. In contrast to conventional MCDM approaches like Analytic Network Process (ANP) and Decision-Making Trial and Evaluation Laboratory (DEMATEL), FCM offers the advantage of integrating negative relationships between criteria, enabling a more comprehensive examination. Furthermore, FCMs assist decision-makers in simulating various scenarios and assessing the consequences of different interventions. The creation of scenarios, which acknowledges the uncertainty of the future, highlights the potential pitfalls of the undue precision often found in software-related roadmaps. Additionally, the scenarios developed in FCM, specific to the supply chain case, provide the opportunity for domain-specific evaluation.

3. Methodology

This study assesses the software quality of a case company, aligning it with the requirements of its stakeholders, with a particular focus on its supply chain. A case study application was conducted, considering the problematic software quality metrics and characteristics of the case supply chain, utilizing a scenario-based FCM approach. In this study, a committee comprising three stakeholders from the case company, representing fields such as software development, software engineering, and cybersecurity, was formed. Through a combination of literature review and consultations with software experts, it was determined to focus on ISO software models and their associated characteristics. In this study, FCMapper (FCMapper n.d.), an Excel macro with an open license, is utilized for conducting FCM analysis. The aggregated fuzzy influence matrix is copied and pasted into the matrix sheet of the FCMapper tool to execute the model. The "check matrix" button is employed to verify the general characteristics of the normalized matrix, ensuring that values fall within the range [0,1]. The FCM Analysis section runs the simulation and applies Eq. (3) to the data using the sigmoidal function. Additionally, the FCM_Scenarios screen facilitates the simulation of diverse scenarios, enabling the exploration of various potential outcomes. The graphical user interface (GUI) of the FCMapper Excel tool and the methodology are illustrated in Figures 1 and 2, respectively.

3.1. SQuaRE Standards Family

SQuaRE, which stands for Software Quality Requirements and Evaluation, is a family of standards divided into five parts (ISO/IEC 2011). This study specifically focuses on two divisions within the SQuaRE framework: the Quality Model Division 2501n and the Quality Measurement Division 2502n. These standards provide a consistent terminology for specifying, measuring, and evaluating system and software product quality. Within the SQuaRE framework, there are three primary quality models: the quality in use model, the product quality model, and the data quality model. In this study, problematic metrics related to product quality, quality in use, and data quality characteristics were considered. These problematic characteristics were chosen based on the identified gaps between the metrics and their targets, aiming to address areas where the software quality fell short of desired standards or expectations. There are numerous studies in the literature that have utilized ISO software standards (Abran, Khelifi, and Suryn 2003; Baggen et al. 2012; Baldassarre et al. 2012; Côté et al. 2005; Chang et al., 2008; Paulk 1993; Sánchez-

Gordón and O'Connor 2016; Stefani and Xenos 2008; Van Zeist 1996); however, there is a need for domain-specific applications to foster wider adoption of these standards.

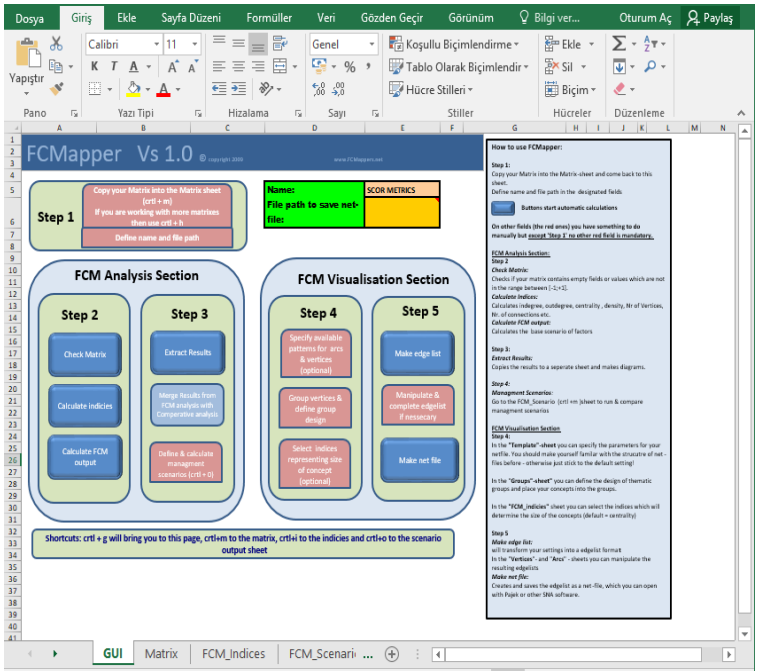


Figure 1. GUI interface of the FCMapper (FCMapper n.d.)

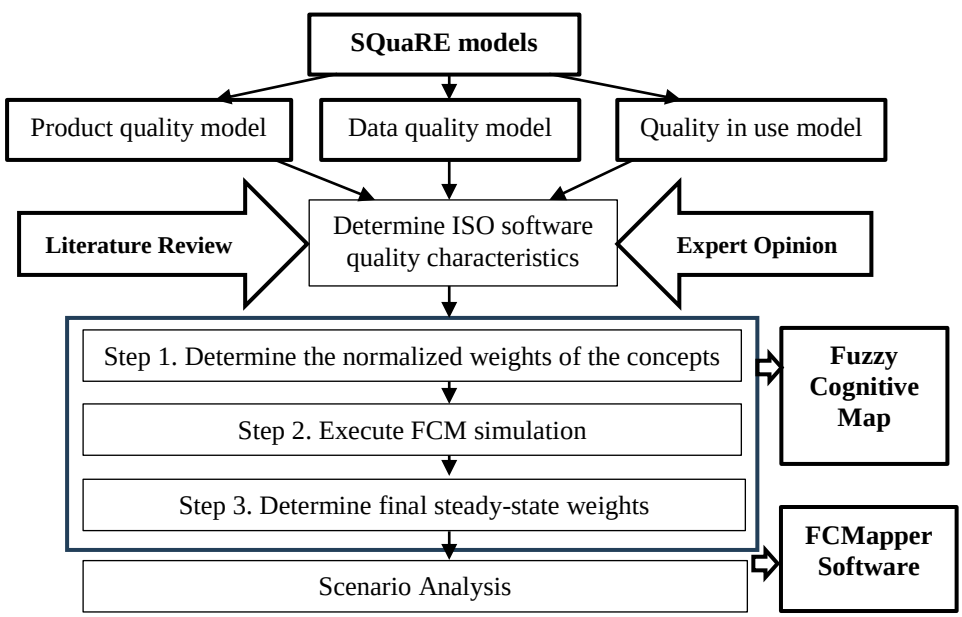


Figure 2. Methodology

3.2 Fuzzy Cognitive Map

Fuzzy Cognitive Mapping (FCM) involves constructing a network consisting of nodes and weighted arcs that symbolize relationships among various concepts. In the early versions of cognitive maps, concepts and connection matrices were assigned negative, positive, or neutral values. However, Kosko (Kosko 1986) expanded on this concept by integrating fuzzy causal relationships into both the directed edges and concepts. The procedural steps of applying FCM can be succinctly outlined as described in (Baykasoğlu & Gölcük 2015; Kosko 1986):

Step 1 Initial values for concepts are defined, and the normalized weight vector is established as the initial concept values, following the formulation in Eq. (1),

$$(C_{MA_1}^{t=0}..C_{MA_p}^{t=0}..C_{MA_n}^{t=0})=(nw_1..nw_p..nw_n) \quad (1)$$

where $C_{MA_i}^{t=0}$ is the concept value of the *i*th main attribute when t = 0. Next, fuzzy influence matrices are obtained, with each matrix represented as $\tilde{e}_{jiu}^{MA}$, signifying the influence matrix derived from the main attributes provided by the *u*th decision-maker. Subsequently, these fuzzy influence matrices are consolidated using Eq. (2).

$$\tilde{e}_{ji}^{MA} = \frac{\sum_{u=1}^{S} \tilde{e}_{jiu}^{MA}}{S} \quad (2)$$

where $\tilde{e}_{ji}^{MA}$ denoted the aggregated fuzzy influence matrix of the main attributes. The fuzzy weights resulting from the aggregation process are subsequently transformed into crisp values through defuzzification, resulting in the acquisition of the aggregated crisp influence matrix e_{ji}^{MA}.

Step 2 The parameters of the activation function are specified, and FCMs are simulated using an appropriate threshold function to reveal persistent influences among the main attributes. The simulation of the FCM is executed through the application of Eq. (3).

$$C_{MA_i}^{t+1}=\mathrm{f}(C_{MA_i}^{t}+\sum_{\substack{j=1 \\ j=1}}^{n} e_{ji}^{MA}\times C_{MA_j}^{t}) \quad (3)$$

where $\mathrm{C}_{\mathrm{MA_i}}^{\mathrm{t}}$ represents the concept value of the *i*th main attribute at time *t.*

Step 3 The concept weights in the steady-state undergo normalization, and the final steady-state weights for the primary attributes are determined using the Eq. (4).

$$w_i=\frac{c_{MA_i}}{\sum_{i=1}^{n} c_{MA_i}}, \quad (4)$$

For this research, sigmoidal FCMs were chosen as the preferred option among the various threshold functions available for use in FCMs. Sigmoidal functions offer a smooth transition between states and are well-suited for capturing gradual changes or nonlinear relationships between concepts in the cognitive map. This choice aligns with the study's objectives and the complexity of the relationships being modelled, allowing for a more accurate representation of the dynamics within the system under study.

4. Case Study

In this study, the case company operates in the field of mechanical and plant engineering, providing cutting-edge technology and industry knowledge to support works worldwide. The company specializes in customer-specific integrated automation solutions and systems, offering a comprehensive range of products and services tailored to the needs of the industry. Additionally, company experts provide advisory services for various projects, utilizing unique resources and expertise throughout the project lifecycle, from design to project realization. However, the involvement of multiple stakeholders in their software systems can lead to challenges and potential problems. Coordination among various stakeholders, each with their own requirements and priorities, may introduce complexities that need to be carefully managed to ensure the successful development and deployment of software

solutions. Indeed, proper authorization poses a significant challenge, especially in light of faulty entries that may compromise system integrity and security. Furthermore, a high proportion of modifications can lead to degradation in the quality of software systems over time. Ensuring that data comply with regulations and standards is also a critical concern for the case company. Given the software problems encountered within the case supply chain, encompassing issues related to software product quality, data product quality, and quality in use, it becomes imperative to evaluate existing software systems from a broader perspective. This holistic evaluation will enable the identification of underlying issues and the implementation of effective solutions to enhance the overall quality and performance of the software systems within the supply chain.

Defining or assessing every sub-characteristic for each component of an extensive computer system or software product can indeed be challenging. Not all quality characteristics within quality models may be relevant to every stakeholder's needs or priorities. In this study, for product quality assessment, the focus was placed on specific sub-characteristics that are particularly pertinent to the case company's objectives and requirements. These sub-characteristics included performance efficiency, reliability, compatibility, usability, maintainability, functional suitability, and security characteristics. The selection of these sub-characteristics was informed by their related metrics and performance gaps from the targets, ensuring a targeted and meaningful evaluation of the software product. Similarly, for quality in-use assessment, the focus was narrowed down to satisfaction and freedom from risk characteristics, reflecting the end-users' perspectives and priorities in terms of usability and safety. Given the technological domain of the case company, only system-dependent data quality characteristics were considered, ensuring that the assessment aligns with the specific context and requirements of the organization. This focused approach allows for a more efficient and effective evaluation of software quality, addressing key areas of concern while avoiding unnecessary complexity and redundancy. By employing the three quality models, a total of 30 problematic software characteristics were identified based on their associated metrics and related performance gaps from their targets. Initially, the company identified software issues and the metrics associated with them. Subsequently, the software characteristics relevant to these metrics were determined. To accomplish this, ISO/IEC 25022 (ISO/IEC 2015a), ISO/IEC 25023 (ISO/IEC 2015b), and ISO/IEC 25024 (ISO/IEC n.d.) standards were utilized to define quality metrics for quantitatively evaluating the software quality characteristics. ISO/IEC 25022 (ISO/IEC 2015a) provided guidance for the measurement of quality in use, ISO/IEC 25023 (ISO/IEC 2015b) offered insights for measuring system and software product quality, and ISO/IEC 25024 (ISO/IEC n.d.) was instrumental in determining measures for data quality. The determined 30 problematic software characteristics (Maden, 2021) encompassing product quality, quality in use, and data product quality are detailed in Table 2, Table 3, and Table 4.

The selection of metrics and related software characteristics has been determined as follows: For the time behaviour sub-characteristic within the performance efficiency category, all metrics are considered suitable for use in their software systems.However, mean turnaround time presents a greater challenge, displaying a performance gap from the target. Mean response time is also deemed appropriate for calculations.

Table 2. Software product quality characteristics and sub-characteristics

Performance Efficiency	Time Behaviour
	Resource Utilization
	Capacity
Reliability	Maturity
	Availability
	Fault Tolerance
	Recoverability
Security	Confidentiality
	Integrity
	Non-repudiation
	Accountability
	Authenticity
Compatibility	Interoperability
	Co-existence
Usability	Operability
Maintainability	Modularity
	Modifiability
Functional Suitability	Functional Completeness

Table 3. Quality in use characteristics and sub-characteristics

Satisfaction	Trust
Freedom from risk	Economic Risk Mitigation

Table 4. Data product quality characteristics

Data Product Quality Characteristics	Accessibility
	Compliance
	Confidentiality
	Efficiency
	Precision
	Traceability
	Understandability
	Availability
	Portability
	Recoverability

It's important to note that mean response time is based on the first response time, whereas mean turnaround time focuses on the entire task completion. This distinction is crucial, as mean turnaround time provides a comprehensive view of the time taken for task completion, while Mean Response Time specifically addresses the time taken for the initial response. Both metrics offer valuable insights into the performance efficiency of the software system and should be considered in tandem to ensure a comprehensive evaluation.

In terms of the capacity sub-characteristic of performance efficiency, all measures including transaction processing capacity, user access capacity, and user access increase adequacy showed similarity in target achievement. However, user access increase adequacy holds particular significance for their system, especially considering potential issues that may arise when more stakeholders are involved. Regarding the confidentiality sub-characteristic, access controllability, which measures the percentage of data items requiring access control solely reached using

appropriate authorization, presented challenges. This is primarily due to significant problems with proper authorization within their software systems, particularly considering the presence of faulty entries. Addressing these challenges in access controllability is crucial for ensuring data confidentiality and maintaining system security, especially in environments where multiple stakeholders are involved and proper authorization is essential for controlling access to sensitive information.

In the context of traceability, user access traceability exhibited performance gaps from the target. This metric evaluates the degree to which users' access to data values is tracked using data from the data repository or other system capabilities. Software stakeholders recognized the importance of user access traceability, specifying which files require traceability regarding access to data values. The base measure for user access traceability was calculated for each data file in the repository by determining the ratio of records with traceable data regarding users who accessed its value (for reading or modification) over the total number of records for which user access traceability is expected. Similar decision-making processes were applied to other software characteristics. Positive and negative relations between ISO software characteristics can be seen in Table 5. To focus on the essential problematic sub-characteristics, a Fuzzy Cognitive Map was simulated, and final weights were obtained. Subsequently, steady-state values were calculated in 12 iterations using FCMapper software to determine the long-term influences among the ISO software characteristics, as presented in Table 6. This process allows for a deeper understanding of the interrelations among different software characteristics and their impacts on overall software quality. By leveraging FCM and iterative calculations, the study aims to identify key areas for improvement and optimize the performance of the software system. In Table 6, columns indicate software characteristics, while rows indicate iterations. After reaching the steady-state concept weights for ISO software characteristics, the values were normalized, as depicted in Table 7, to obtain the final normalized weights of the ISO software characteristics. The data presented in Tables 5, 6, and 7 of this study were taken from the corresponding author's doctoral dissertation (Maden, 2021).

Scenario Analysis

Utilizing scenario analysis through Fuzzy Cognitive Maps enables the generation of multiple hypothetical scenarios, each representing different states of concepts within the software system. By comparing the values of new and old concepts, one can ascertain whether the system is progressing or diminishing in significance through its relationships. To observe the evolution of scenarios, we utilized FCMapper to modify concept values, focusing solely on the values for concepts under management. These concept values are constrained within the interval [0; 1], with each concept having a value of 1 or 0 based on whether it is enhanced or not. This approach allows for the exploration of various potential states of the software system and the evaluation of how changes in specific concepts may impact overall system performance and quality. It provides valuable insights into the dynamics of the system and helps inform decision-making processes for improving software quality. The scenarios were specifically focused on a particular aspect: the impact of ISO software characteristics' relationships on the outcome. To understand how variations in one concept could influence other variables and observe changes in the case study's results, five scenarios were developed based on the unique circumstances of the case company and its supply chain, as represented in Table 8.

By examining the results of these scenarios, the study aims to gain insights into the dynamics of the software system and

identify strategies for improving software quality and performance within the context of the case company and its supply chain. The baseline scenario (Scenario 1), signifying no change, was chosen as the initial vector for comparison. This scenario provides a reference point against which the effects of changes in other scenarios can be measured.

Scenario 2.

As the backbone of modern supply chain operations, the case company's software infrastructure plays a crucial role in safeguarding critical information, ensuring transaction integrity, and maintaining the confidentiality of sensitive data. In the dynamic and interconnected landscape of the supply chain, security is not just a requirement but a fundamental pillar. Within this context, the sub-characteristic of confidentiality holds special significance, with access controllability emerging as a key aspect in securing their software system. Recognizing the paramount importance of maintaining secure access control mechanisms, the company has initiated a focused examination of the impact of decreasing confidentiality within their supply chain software.

Table 5. Positive and negative relations between ISO software characteristics

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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	-	-	-	+			+	-							
29															
30															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
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										+	+	+			
29															
30															

Table 6. Steady-state concept weights of the ISO software characteristics

	1	2	3	4	5	6	7	8	9	10
1	0,917	0,622	0,891	0,769	0,786	0,731	0,948	0,881	0,750	0,731
2	0,872	0,543	0,838	0,711	0,741	0,675	0,923	0,835	0,696	0,676
3	0,854	0,527	0,818	0,697	0,731	0,663	0,914	0,822	0,683	0,664
4	0,848	0,524	0,811	0,693	0,728	0,660	0,911	0,818	0,680	0,661
5	0,846	0,523	0,809	0,692	0,727	0,659	0,911	0,817	0,679	0,660
6	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
7	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
8	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
9	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
10	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
11	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660
12	0,846	0,523	0,808	0,692	0,727	0,659	0,910	0,817	0,679	0,660

	11	12	13	14	15	16	17	18	19	20
1	0,731	0,786	0,858	0,769	0,769	0,769	0,786	0,731	0,786	0,731
2	0,677	0,734	0,806	0,715	0,714	0,714	0,735	0,675	0,733	0,675
3	0,665	0,721	0,791	0,701	0,700	0,700	0,722	0,663	0,719	0,663
4	0,663	0,718	0,787	0,698	0,697	0,697	0,719	0,660	0,716	0,660
5	0,662	0,717	0,786	0,697	0,696	0,696	0,718	0,659	0,715	0,659
6	0,662	0,717	0,786	0,697	0,696	0,696	0,718	0,659	0,715	0,659
7	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
8	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
9	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
10	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
11	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
12	0,662	0,717	0,786	0,697	0,696	0,696	0,717	0,659	0,714	0,659
	21	22	23	24	25	26	27	28	29	30
1	0,769	0,750	0,769	0,786	0,769	0,931	0,931	0,846	0,786	0,786
2	0,724	0,696	0,722	0,741	0,719	0,914	0,914	0,808	0,738	0,740
3	0,713	0,684	0,712	0,730	0,707	0,909	0,909	0,797	0,726	0,729
4	0,711	0,681	0,710	0,728	0,704	0,907	0,907	0,795	0,722	0,726
5	0,710	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,722	0,725
6	0,710	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
7	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
8	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
9	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
10	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
11	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725
12	0,709	0,680	0,709	0,727	0,703	0,907	0,907	0,794	0,721	0,725

Table 7. Final normalized weights of the ISO software characteristics

1	2	3	4	5	6	7	8	9	10
0,0386	0,0239	0,0369	0,0316	0,0332	0,0301	0,0416	0,0373	0,0310	0,0301
11	12	13	14	15	16	17	18	19	20
0,0302	0,0327	0,0359	0,0318	0,0318	0,0318	0,0328	0,0301	0,0326	0,0301
21	22	23	24	25	26	27	28	29	30
0,0324	0,0310	0,032	0,0324	0,0321	0,0414	0,0414	0,0362	0,0329	0,0331

Table 8. Scenario Analysis

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Resource Utilization	0,846	1,000	1,000	1,000	1,000
Capacity	0,523	0,867	0,853	0,870	0,878
Time Behavior	0,808	0,530	0,530	0,576	0,596
Maturity	0,692	0,888	0,876	0,894	0,902
Availability	0,727	0,712	0,712	0,711	0,662
Accountability	0,659	0,715	0,715	0,714	0,715
Fault Tolerance	0,910	0,659	0,659	0,659	0,659
Confidentiality	0,817	0	0,913	0,912	0,885
Recoverability	0,679	0,798	0,787	0,801	0,832
Integrity	0,660	0,680	0,680	0,679	0,679
Non-repudiation	0,662	0,637	0,665	0,665	0,664
Authenticity	0,717	0,638	0,665	0,665	0,664
Compliance	0,786	0,712	0	0,712	0,712
Accessability	0,697	0,787	0,792	0,791	0,791
Understandability	0,696	0,703	0,703	0,703	0,705

Availability (Data)	0,696	0,699	0,699	0,699	0,699
Efficiency	0,717	0,699	0,699	0,699	0,699
Precision	0,659	0,713	0,713	0,713	0,713
Traceability	0,714	0,659	0,659	0,659	0,659
Portability	0,659	0,716	0,716	0,716	0,716
Confidentiality (Data)	0,709	0,659	0,659	0,659	0,659
Recoverability (Data)	0,680	0,618	0,721	0,721	0,718
Interoperability	0,709	0,679	0,679	0,679	0,679
Co-existence	0,727	0,703	0,703	0,684	0,699
Operability	0,703	0,721	0,721	0,703	0,717
Modularity	0,907	0,708	0,708	0	0,659
Modifiability	0,907	0,895	0,895	0,894	0,769
Functional Completeness	0,794	0,895	0,895	0,894	0
Trust	0,721	0,786	0,786	0,770	0,782
Economic Risk Mitigation	0,725	0,677	0,726	0,726	0,725

The intricacies of access controllability, which are crucial for protecting sensitive supply chain data, are pivotal considerations in this analysis. To gain targeted insights into the consequences of altering confidentiality parameters, a domain-specific scenario has been meticulously crafted using the FCMapper scenario screen. This simulation delves into the nuanced effects of setting confidentiality to zero, offering a granular perspective on how this critical security characteristic influences the broader spectrum of ISO software qualities within the unique context of a supply chain. The ensuing analysis reveals domain-specific shifts in the weights of criteria, illuminating the intricate relationship between confidentiality and other software characteristics within the company's supply chain. The observed decrease in confidentiality not only impacts fault tolerance, compliance, and functional completeness but also introduces alterations in resource utilization, time behavior, modularity, and modifiability. This domain-specific scenario underscores the necessity of balancing security considerations with the dynamic requirements of the supply chain. It highlights the importance of maintaining a robust and adaptable software infrastructure that can effectively navigate the complexities of modern supply chain operations while ensuring the confidentiality and security of sensitive information.

Scenario 3.

As the pulse of modern supply chain operations, the case company's software ecosystem operates within a complex interplay of regulatory standards and technological advancements. In the ever-evolving landscape of the supply chain, ensuring compliance is not merely a necessity but a strategic imperative. Regulatory compliance, particularly in the realm of technology, serves as a cornerstone in safeguarding the integrity of the case company's software system Given the unique challenges posed by the supply chain domain, a comprehensive examination of compliance becomes imperative. This scenario meticulously explores the consequences of decreasing compliance to zero, offering a targeted perspective on how this critical characteristic influences the entire spectrum of ISO software qualities within the specific context of a supply chain. The ensuing analysis brings to the forefront the nuances of regulatory compliance within the supply chain operations of the company. The observed decrease in compliance, particularly concerning technology-driven regulatory mandates, has resulted in notable alterations in the weights of specific criteria. These changes have cascading effects, impacting not only the compliance aspect itself but also confidentiality, fault tolerance,

and performance efficiency. This domain-specific scenario serves as a strategic lens, offering the company valuable insights into the intricacies of compliance within their supply chain. It provides targeted considerations for refining their software systems, ensuring adherence to regulatory standards while optimizing performance and resilience within the dynamic landscape of supply chain operations. By understanding the nuanced impacts of compliance on various aspects of software quality, the company can make informed decisions to address compliance challenges effectively while maintaining the integrity and efficiency of their software ecosystem. This strategic approach ensures that the company remains compliant with regulatory requirements while staying agile and adaptable to changes in the regulatory landscape and technological advancements.

Scenario 4.

As the heartbeat of modern supply chain operations, the case company's software system plays a crucial role in orchestrating the seamless coordination of diverse components such as inventory management, order processing, and logistics. Recognizing the pivotal role of modularity in enhancing operational efficiency, the company has undertaken a detailed examination of their software's current state. Specifically, they have identified challenges stemming from the ratio of completely independent components, a critical aspect in the realm of supply chain management. This scenario methodically explores the consequences of decreasing modularity to zero, shedding light on the intricate interplay between modularity and other ISO software characteristics within the unique context of the supply chain. The ensuing analysis delves into the specific challenges faced by the company within the supply chain domain. The observed decrease in modularity, fueled by issues in the ratio of completely independent components, has far-reaching implications. Notably, the decrease in modularity has been found to adversely impact confidentiality, compliance, and functional completeness. Simultaneously, the reduction in modularity introduces an increase in resource utilization, a factor that is critical to address in the fast-paced and dynamic landscape of supply chain operations.

Scenario 5.

As the intricate operations of supply chain management continue to be shaped by technological advancements, the efficacy of software systems becomes paramount in ensuring seamless coordination and optimal functionality. The case company, a key player in the supply chain domain, has identified a pressing concern within their current software system – a notable shortfall in functional completeness, characterized by a substantial number of missing or incorrect functions. Recognizing the critical role of software in orchestrating the intricate dance of logistics, inventory management, and order processing, the company has initiated a focused examination of the consequences of this deficiency on various ISO software characteristics. This scenario rigorously explores the ramifications of decreasing functional completeness to zero, offering a granular perspective on how this deficiency resonates across the spectrum of ISO software characteristics within the unique context of a supply chain. The ensuing analysis illuminates the specific challenges faced by the company in the supply chain domain. The observed decrease in functional completeness, arising from the prevalence of missing or incorrect functions, has reverberations throughout the system. Notably, the decrease in functional completeness is found to increase resource utilization, and introduce potential vulnerabilities in the confidentiality of sensitive supply chain data. This domain-specific scenario serves as a tailored exploration, providing the company with valuable insights and strategic considerations

to address and rectify the deficiencies in functional completeness, ensuring their software systems are finely tuned to meet the unique demands and challenges of the supply chain industry.

5. Conclusion and Discussion

Software quality evaluation is crucial as it directly influences the overall performance and reliability of a software system. Assessing the quality of software helps identify and address potential issues, ensuring that the product meets the required standards and user expectations. By conducting thorough evaluations, developers can enhance the functionality, usability, and efficiency of the software, leading to a more robust and satisfactory user experience. The ISO software characteristics offer a standardized language and a diverse range of criteria for assessing any software architecture, facilitating the easy identification of software problems. In this research, a case study was carried out to address the software requirements of a case company along with its supply chain. To determine the most significant software characteristics, ISO/IEC 25010 (ISO/IEC 2011) and ISO/IEC 25012 (ISO/IEC n.d.) models were utilized, covering various aspects including product quality, quality in use, and data quality. This approach facilitated the swift identification of problematic software quality attributes within the case company from a broader standpoint, employing a standardized language. By leveraging ISO models for determining and evaluating software quality metrics, coupled with scenario-based Fuzzy Cognitive Maps, company stakeholders were provided with a shared set of performance metrics and a clear direction for change. To comprehend how changes in one software characteristic can impact others and observe alterations in the outcomes of the case study, five scenarios were formulated, taking into account the specific conditions of the case company and its supply chain. For instance, in scenario 2, compromised confidentiality not only affects fault tolerance, compliance, and functional completeness but also leads to modifications in resource utilization, time behavior, modularity, and modifiability. Similarly, in scenario 3, alterations not only influence the compliance aspect but also have a ripple effect on confidentiality, fault tolerance, and performance efficiency. In scenario 4, the decline in modularity has a negative impact on confidentiality, compliance, and functional completeness. Conversely, in scenario 5, the reduction in functional completeness is observed to elevate resource utilization and introduce potential vulnerabilities in the confidentiality of sensitive supply chain data. These scenarios present a range of potential futures, allowing various software stakeholders to understand their roles and take appropriate actions. Our approach proves cost-effective, given the scenario analysis and related calculations, eliminating the need to start from scratch. FCM allowed stakeholders to grasp system behavior through simple matrix operations, with the FCMapper software solution facilitating the necessary calculations. This integrated approach empowers stakeholders to make informed decisions and strategize effectively in response to potential changes and challenges in the software environment.

The findings of this study offer valuable insights for software development and management, offering significant implications for both practitioners and decision-makers. Managers in software development organizations can utilize the identified trade-offs among software quality characteristics to make informed decisions throughout the development and maintenance phases. Recognizing that enhancements in one aspect may entail

compromises in another enables strategic decision-making aligned with organizational priorities and user expectations. Practically, the study underscores the importance of adopting domain-specific perspectives in software quality evaluation. Organizations, particularly those with unique requirements like the supply chain industry in the presented case study, can derive benefit from customizing standardized quality frameworks, such as ISO/IEC 25010 and ISO/IEC 25012, to their specific domain needs. This customization ensures a more precise and relevant assessment of software quality, addressing industry-specific challenges, and optimizing software performance in real-world contexts. Furthermore, the study emphasizes the importance of integrating user perspectives and data-related characteristics into software quality evaluations. This insight directs practitioners to consider not only the technical aspects of software but also the end-users' experiences and the integrity of the underlying data. By incorporating user-centric evaluations and data quality assessments, software quality can be enhanced, leading to increased user satisfaction and ensuring the reliability of the processed information. In summary, the managerial and practical implications of the study highlight the significance of informed decision-making based on trade-offs, the customization of quality frameworks for specific domains, and the inclusion of user and data perspectives in software quality assessments. These insights offer guidance for software development practices and contribute to the delivery of high-quality software products that meet organizational objectives and end-user needs effectively.

This study is partially derived from the corresponding author's doctoral dissertation (Maden, 2021). While the initial data, calculations, and some scenarios are similar to those in the dissertation, the subsequent numerical results and analyses differ. In the dissertation, two particular aspects for scenarios were focused on: the impact of SCOR performance metrics' relations and ISO software characteristics' relations to the result (Maden, 2021). In contrast, in this article, the scenarios were specifically focused on a particular aspect: the impact of ISO software characteristics' relationships on the outcome. Some scenarios share similarities with those in the dissertation (Maden, 2021); however, the numerical findings from scenarios are distinct. Moreover, one of the previous studies derived from the dissertation (Maden & Alptekin, 2024) identified and discussed only the eight most critical ISO software characteristics without providing a detailed analysis of all characteristics. This study, however, comprehensively examines all 30 software characteristics through a numerical analysis, in alignment with the broader framework presented in the author's dissertation. Additionally, another previous publication derived from the dissertation (Maden & Alptekin, 2023) focused on procurement performance metric selection using SCOR modelling and scenario-based fuzzy cognitive mapping. Although related to performance metric analysis, it addresses a different context (procurement) and does not cover the comprehensive exploration of ISO software characteristics found in this study.

This study primarily assesses ISO metrics based on their understandability, measurability, and unambiguity. However, a more comprehensive evaluation could incorporate additional criteria such as specificity, relevancy, verifiability, and comparability. While specific company and supply chain scenarios were considered for scenario creation in this study, future research could explore all potential combinations by manipulating metric values to uncover a broader range of scenarios. Moreover, specific evaluation criteria like occurrence probability, usability, and influence could be taken into account when evaluating and selecting scenarios in FCMs. Additionally, future investigations can explore alternative methodologies or

combinations of methodologies, such as Fuzzy ANP and DEMATEL, to further enhance the evaluation process and capture different aspects of software quality characteristics and their interrelationships. The inherent uncertainty in this study arises from the subjectivity inherent in human judgments, especially when determining the increase or decrease in calculated ISO software metrics. Future research endeavours can address this uncertainty by examining the degree of deviation of metrics from their target values.

Declarations

No funding was obtained for this study.

Conflict of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Some parts of this chapter are generated based on Ayça Maden's PhD thesis supervised by Sadettin Emre Alptekin.

References:

Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*, *11*(4), 325-338.

Baggen, R., Correia, J. P., Schill, K., & Visser, J. (2012). Standardized code quality benchmarking for improving software maintainability. *Software Quality Journal*, *20*(2), 287-307..

Baldassarre, M. T., Caivano, D., Pino, F. J., Piattini, M., & Visaggio, G. (2012). Harmonization of ISO/IEC 9001: 2000 and CMMI-DEV: from a theoretical comparison to a real case application. *Software Quality Journal*, *20*, 309-335..

Baykasoğlu, A., & Gölcük, İ. (2015). Development of a novel multiple-attribute decision making model via fuzzy cognitive maps and hierarchical fuzzy TOPSIS. *Information Sciences*, *301*, 75-98.

Botchway, I. B., Emmanuel, A. A., Solomon, N., & Kayode, A. B. (2021). Evaluating software quality attributes using analytic hierarchy process (AHP). *International Journal of Advanced Computer Science and Applications*, *12*(3), 165–173.

Bertoa, M. F., Troya, J. M., & Vallecillo, A. (2006). Measuring the Usability of Software Components. *Journal of Systems and Software*, *79*(3), 427–439.

Boehm, B. (1978). Characteristics of Software Quality. *Elsevier, North Holland*.

Chang, C. W., Wu, C. R., & Lin, H. L. (2008). Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality. *Software Quality Journal*, *16*, 263–276.

Côté, M. A., Suryn, W., Laporte, C. Y., & Martin, R.A. (2005). The Evolution Path for Industrial Software Quality Evaluation Methods Applying ISO/IEC 9126:2001 Quality Model: Example of MITRE's SQAE Method. *Software Quality Journal, 13, 17-30.*

Dayanandan, U., & Kalimuthu, V. (2018). Software Architectural Quality Assessment Model for Security Analysis Using Fuzzy Analytical Hierarchy Process (FAHP) Method. *3D Research, 9*(3), 1–14. https://doi.org/10.1007/s13319-018-0183-x.

Dromey, R. G. (1996). Cornering the chimera [software quality]. *IEEE software*, *13*(1), 33-43.

FCMapper. *Fuzzy Cognitive Mapping Software Solution*. Retrieved from http://www.fcmappers.net/joomla/.

Grady, R. B. (1992). *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall.

ISO/IEC. (2011). *ISO/IEC 25010:2011 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*.

ISO/IEC. (2015a). *ISO/IEC 25022:2016 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Measurement of Quality in Use*.

ISO/IEC. (2015b). *ISO/IEC 25023:2016 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Measurement of System and Software Product Quality*.

ISO/IEC. (n.d.) *ISO/IEC 25012-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Data Quality Model*.

ISO/IEC. (n.d.) *ISO/IEC 25024:2015 Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Measurement of Data Quality*.

Kaviani, M. A., Yazdi, A. K., Ocampo, L., & Kusi-Sarpong, S. (2020). An Integrated Grey-Based Multi-Criteria Decision-Making Approach for Supplier Evaluation and Selection in the Oil and Gas Industry. *Kybernetes, 49*(2), 406–441.

Khan, S. A., Alenezi, M., Agrawal, A., Kumar, R. & Khan, R. A. (2020). Evaluating Performance of Software Durability through an Integrated Fuzzy-Based Symmetrical Method of ANP and TOPSIS. *Symmetry*, *12*(4), 493.

Kosko, B. (1986). Fuzzy cognitive maps. *International journal of man-machine studies*, *24*(1), 65-75.

Maden, A. (2021). *Evaluation of blockchain technology for process improvements in supply chains using FUZZY QFD*. Galatasaray University.

Maden, A., & Alptekin, S. E. (2023, August). Procurement Performance Metric Selection Using SCOR Modelling and Scenario-Based Fuzzy Cognitive Map. In *International Conference on Intelligent and Fuzzy Systems* (pp. 465-471). Cham: Springer Nature Switzerland.

Maden, A., & Alptekin, E. (2024). Evaluation of Blockchain Technology for Supply Chains using an Integrated Fuzzy Cognitive Map-QFD Methodology. *Journal of Advanced Research in Natural and Applied Sciences*, *10*(2), 252-271.

Maden, A., & Yücenur, G. N. (2024). Evaluation of Sustainable Metaverse Characteristics Using Scenario-Based Fuzzy Cognitive Map. *Computers in Human Behavior*, *152*, 108090.

McCall, J. A. (1977). Factors in Software Quality. *US Rome Air development center reports*.

Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2017). A SQuaRE-Based Software Quality Evaluation Framework and Its Case Study. *IEEE Region 10 Annual International Conference, Proceedings/TENCON* (Qrs): 3704–7.

Paik, H. Y., Xu, X., Bandara, H. M. N, Lee, S. U., & Lo, S. K. (2019). Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance. *IEEE Access*, *7*, 186091-186107.

Paulk, M. C. (1993). Comparing ISO 9001 and the capability maturity model for software. *Software Quality Journal*, *2*, 245-256.

Saaty, T. L., & Ozdemir, M. S. (2003). Why the magic number seven plus or minus two. *Mathematical and computer modelling*, *38*(3-4), 233-244.

Sánchez-Gordón, M. L., & O'Connor, R. V. (2016). Understanding the gap between software process practices and actual practice in very small companies. *Software Quality Journal*, *24*, 549-570.

Stefani, A., & Xenos, M. (2008). E-Commerce System Quality Assessment Using a Model Based on ISO 9126 and Belief Networks. *Software Quality Journal*, *16*(1), 107–129.

Thapar, S. S., & Sarangal, H. (2020). Quantifying reusability of software components using hybrid fuzzy analytical hierarchy process (FAHP)-Metrics approach. *Applied Soft Computing*, *88*, 105997. https://doi.org/10.1016/j.asoc.2019.105997.

Van Zeist, R. H. J., & Hendriks, P. R. H. (1996). Specifying software quality with the extended ISO model. *Software Quality Journal*, *5*, 273-284.

Yue, C. (2020). An Intuitionistic Fuzzy Projection-Based Approach and Application to Software Quality Evaluation." *Soft Computing*, *24*(1) 429–43. https://doi.org/10.1007/s00500-019-03923-6.

Yue, Z. L., Yue, C., Peng, X. H., & Wu, W. (2023). VIKOR-based group decision-making method for software quality assessment. *Iranian Journal of Fuzzy Systems*, *20*(1), 53-70.

Yuen, K. K. F., & Lau, H. C. W. (2011). A Fuzzy Group Analytical Hierarchy Process Approach for Software Quality Assurance Management : Fuzzy Logarithmic Least Squares Method. *Expert Systems With Applications*, *38*(8), 10292–302. http://dx.doi.org/10.1016/j.eswa.2011.02.057.

Ayça Maden
Istanbul Beykent University
Istanbul,
Turkey
aycamaden@beykent.edu.tr
ORCID 0000-0002-8239-3084

Sadettin Emre Alptekin
Galatasaray University
Istanbul,
Turkey
alptekin@gsu.edu.tr
ORCID 0000-0003-3555-2684