Automated Software Engineering (2023) 30:22
https://doi.org/10.1007/s10515-023-00389-7

# A systematic decision-making framework for tackling quantum software engineering challenges

**Muhammad Azeem Akbar**[1] · **Arif Ali Khan**[2] · **Saima Rafi**[3]

Received: 12 April 2023 / Accepted: 23 June 2023


## Abstract

Quantum computing systems harness the power of quantum mechanics to execute computationally demanding tasks more effectively than their classical counterparts. This has led to the emergence of Quantum Software Engineering (QSE), which focuses on unlocking the full potential of quantum computing systems. As QSE gains prominence, it seeks to address the evolving challenges of quantum software development by offering comprehensive concepts, principles, and guidelines. This paper aims to identify, prioritize, and develop a systematic decision-making framework of the challenging factors associated with QSE process execution. We conducted a literature survey to identify the challenging factors associated with QSE process and mapped them into 7 core categories. Additionally, we used a questionnaire survey to collect insights from practitioners regarding these challenges. To examine the relationships between core categories of challenging factors, we applied Interpretive Structure Modeling (ISM). Lastly, we applied fuzzy TOPSIS to rank the identified challenging factors concerning to their criticality for QSE process. We have identified 22 challenging factors of QSE process and mapped them to 7 core categories. The ISM results indicate that the 'resources' category has the most decisive influence on the other six core categories of the identified challenging factors. Moreover, the fuzzy TOPSIS indicates that 'complex programming', 'limited software libraries', 'maintenance complexity', 'lack of training and workshops', and 'data encoding issues' are the highest priority challenging factor for QSE process execution. Organizations using QSE could consider the identified challenging factors and their prioritization to improve their QSE process.

**Keywords** Quantum computing · Quantum software engineering (QSE) · Challenging factors · Prioritization

Extended author information available on the last page of the article

Published online: 26 July 2023

## 1 Introduction

Quantum computing (QC) has gained significant attention from industry and policymakers due to its potential to revolutionize various industrial areas. This is evidenced by heavy investments from technology giants such as IBM, Google, and Microsoft to implement QC as a service offering for complex computational problems (Moguel et al. 2022; Qiskit 2021). However, developing QC applications is a challenging endeavor that requires a systematic engineering lifecycle perspective to manage complexity effectively. This approach takes precedence over ad-hoc implementation techniques and relying on individual development skills, which can lead to errors and high maintenance costs (Azeem Akbar et al. 2022). To realize the full potential of QC, advanced software engineering methods are necessary (Piattini et al. 2021; Khan et al. 2022). This will ensure that QC can live-up to its promising potential and provide the desired impact.

Quantum software engineering (QSE) is becoming increasingly important as the field of quantum computing continues to grow and attract more interest from both academia and industry. Quantum computing has the potential to revolutionize many industries, from finance to healthcare, by solving complex problems that are beyond the capabilities of classical computers (Piattini et al. 2021; Agarwal and Alam 2021). However, the development of quantum software is a complex and challenging endeavor that requires specialized tools, techniques, and expertise (Khan et al. 2022; Akbar et al. 2022a; Ali et al. 2022). Without proper QSE practices, the development of quantum software can be error-prone, difficult to maintain, and ultimately fail to achieve its full potential (Ali et al. 2022; Failed 2022). Luis et al. (Hevia et al. 2022) indicated that QSE practices can help to address challenges such as scalability, debugging, testing, and integration with classical systems, as well as help ensure the security and privacy of sensitive information through quantum-safe encryption techniques.

Akbar et al. 2022; Failed 2022) and Khan et al. (Khan et al. 2023a) stated that QSE deal with additional challenges to classic software engineering as it has fundamental differences between classical and quantum computing. These challenges include limited scalability, lack of standardized tools and frameworks, lack of software optimization techniques, and a shortage of skilled quantum software engineers (Khan et al. 2023b, b ; Sarkar 2212). Debugging and testing quantum software is also challenging, as it requires specialized tools and techniques that are different from classical software engineering (Arias et al. 2023). Moreover, the development of quantum software requires specialized resources such as quantum simulators and quantum compilers, which are not widely available (Arias et al. 2023; Barrera et al. 2022a).

We found some research studies have been conducted to address the challenges in quantum software engineering (Cruz-Lemus and Serrano 2022). Cruz-Lemus et al. (Cruz-Lemus and Serrano 2022) proposes the use of a high-level quantum programming language called Q# to simplify the development and testing of quantum software. Another study conducted by Mintz et al. (Mintz et al. 2020) focuses on the use of software design patterns to develop scalable and maintainable quantum software. In addition, Mitarai et al. (Mitarai et al. 2018) proposes the use of machine learning techniques to optimize quantum circuits and improve their performance. Similarly, Enrique et al. (Moguel et al. 2020) discusses the need for new software development processes and methodologies to effectively manage the complexity of quantum software development. Benjamin et al. (Weder et al. 2022) also emphasizes the importance of developing a standardized quantum software development environment to enable interoperability and facilitate the exchange of knowledge and tools between researchers and practitioners.

However, to the best of knowledge, no empirical study has been conducted to explore the key challenging factors facing by QSE practitioners. Hence, we are motivated to conduct a study that can help the software organization to focus on challenging factors that could have significant negative influence on successful execution of QSE process. This study aim (1) to identify the QSE challenging factors reported in existing literature, (2) to conduct questionnaire survey study with industry practitioners to get their insight concerning to the criticality of the identified challenging for QSE process, and (3) to develop a systematic decision-making framework by applying fuzzy TOPSIS and ISM approaches. The results and analysis of this study give a deep understanding about QSE challenging factors and their significance on the success of QSE process. We believe that the findings of this study can help in developing effective strategies for tackling problems associated with QSE process which is significant to the success and progression of QSE process. In particular, this study focuses on the following research questions (RQ):

**RQ1** What are the most important challenging factors of QSE process reported in literature?

**RQ2** What would be the systematic decision-making framework of the QSE challenging factors?

The remainder of this paper is organized as follows: Sect. 2 discusses the background and motivation for the study. Section 3 describes the research used methodology and Sect. 4 presents the results and analysis. The decision-making framework and a summary of the results are provided in Sect. 5. Section 6 delves into the study's implications and potential threats to validity. Finally, Sect. 7 concludes the paper and outlines future work related to this study.

## 2 Background and motivation

Quantum technology has emerged as a new field of science and technology, encompassing the principles of quantum mechanics to develop new technologies and devices. The technology uses the peculiar properties of quantum mechanics, such as superposition and entanglement, to create new devices that are faster and more efficient than their classical counterparts (Moguel et al. 2022; Guzik et al. 2015). The potential applications of quantum technology are numerous, including quantum communication, quantum cryptography, quantum sensing, and quantum computing (Orús et al. 2019). Quantum computing, in particular, is seen as a game-changer in the field of computing, with the potential to solve complex problems that are beyond the capabilities of classical computers (Poonia and Kalra 2016; Zidan et al. 2021). As such, governments, private companies, and research institutions have invested heavily in the development of quantum technology to drive innovation and advance science (MacQuarrie et al. 2020).

The field of QSE has been gaining increasing attention in recent years, with several research initiatives and academic programs dedicated to this area. The aim of QSE is to provide a systematic approach to the development and maintenance of quantum software, taking into account the unique characteristics of quantum computing, such as superposition, entanglement, and interference (Zhao 2007; Serrano et al. 2022).

There have been several research papers published in recent years that have contributed to the development of quantum software engineering. For instance, (Hoo Teo et al. 2021) conducted a study for developing software engineering framework for quantum computing and to apply quantum in traffic simulation. Their proposed framework integrates classical and quantum software components. Another study conducted by Zhao 2007 develop a roadmap for quantum software engineering. The developed roadmap provides the guidelines for the development of quantum-based software by highlighting the major challenges and research directions in this field.

Other studies have focused on developing new programming languages and tools for quantum software engineering. For example, Fingerhuth et al. (Fingerhuth et al. 2018) describes Qiskit, an open-source software framework for quantum computing that provides a high-level interface for programming quantum circuits. Similarly, Mykhailova and Soeken (Mykhailova and Soeken 2021) describes a set of coding exercises that teach quantum programming using the Microsoft Quantum Development Kit.

In summary, QSE is a critical aspect of quantum computing, with significant potential to enable the development of efficient and scalable quantum software. The challenges faced by QSE are being addressed by ongoing research initiatives, and there are several promising approaches that have been proposed to overcome these challenges (Li et al. 2021). However, the development of quantum software poses several challenges due to its inherited differences with classical computing and immature QSE practices (Ali and Yue 2020). For instance, lack of standardized tools and techniques, the need for specialized programming languages, limited availability of skilled quantum software engineers, and the complexity of quantum algorithms. Research in this area has focused on developing new programming languages, tools, and methodologies to address these challenges.

In recent years, the multicriteria decision making (MCDM) techniques have emerged as a significant research area, with the aim of identifying critical features and developing models that enable software organizations to effectively manage complex development activities and mitigate risk factors on priority. Various existing studies using MCDM technique are using in software engineering research and fuzzy TOPSIS is one of them.

Fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is a multi-criteria decision-making technique that has gained significant attention in software engineering research. It can effectively handle imprecise and uncertain data and provide a ranking of alternatives based on their proximity to the ideal solution.

The technique has been applied to various problems in software engineering, including team evaluation, agile assessment, outsourcing destination selection, testing technique prioritization, technical debt evaluation, and open-source software project assessment.

For instance, Huang and Tsai (Huang and Tsai 2021) proposed a fuzzy TOPSIS-based framework for evaluating the quality of software development teams. Similarly, Aksu and Ulucay (Aksu and Uluçay 2021) used fuzzy TOPSIS to evaluate the agility of software development teams based on multiple criteria. In a study by Sun et al. (Sun et al. 2020), fuzzy TOPSIS was used to select the best software development outsourcing destination based on several criteria. Gao et al. (Gao et al. 2020) used fuzzy TOPSIS to prioritize software testing techniques based on various criteria. Pan et al. (Pan et al. 2020) applied fuzzy TOPSIS to evaluate the impact of technical debt on software quality. Mohsin et al. (Mohsin et al. 2019) used fuzzy TOPSIS to evaluate the quality of open-source software projects based on different criteria. Singh et al. (Singh and Benyoucef 2011) used fuzzy TOPSIS to prioritize software testing techniques based on multiple criteria.

Recent studies have also shown the effectiveness of fuzzy TOPSIS in addressing DevOps and DevSecOps challenges. Akbar et al. (Akbar et al. 2022b) used fuzzy TOPSIS and ISM approach to determine the priority order and significance level of the DevSecOps process improvement challenging factors. Rafi et al. (Rafi et al. 2020) applied fuzzy TOPSIS and ISM approach for leveling and prioritizing the DevOps data quality assessment challenges. To summarize, it is noted that the application of fuzzy TOPSIS and ISM approach in software engineering research has shown promising results and its potential in aiding decision-making processes in this field (Rafi et al. 2022; Failed 2023a).

Despite the importance of QSE in recent years, there is a lack of research on addressing the multicriteria decision making problem of QSE challenging factors. To fill this gap, our study has developed a novel framework based on fuzzy TOPSIS and ISM approaches to prioritize QSE challenges. The proposed framework will provide organizations with a tool to assess the impact of QSE related challenges on the successful implementation of QSE practices and identify areas that require more attention. By identifying the key challenging factors that significantly influence the execution of QSE processes, our study contributes to the field by providing a unique approach through a systematic decision-making framework. Hence, our proposed framework is a valuable tool for organizations to understand the challenges they face when implementing QSE practices and prioritize their efforts accordingly. This approach can ultimately lead to successful and efficient execution of QSE process.

## 3 Methodology

The research methodology followed to achieve the research goals and to give the answers to the research questions described in Sect. 1. The research has been divided into three stages, and they have been briefly discussed in the following sections. The Fig. 1 shows the roadmap of methodology.

**Step–1** Identification of QSE challenging factors from existing literature.

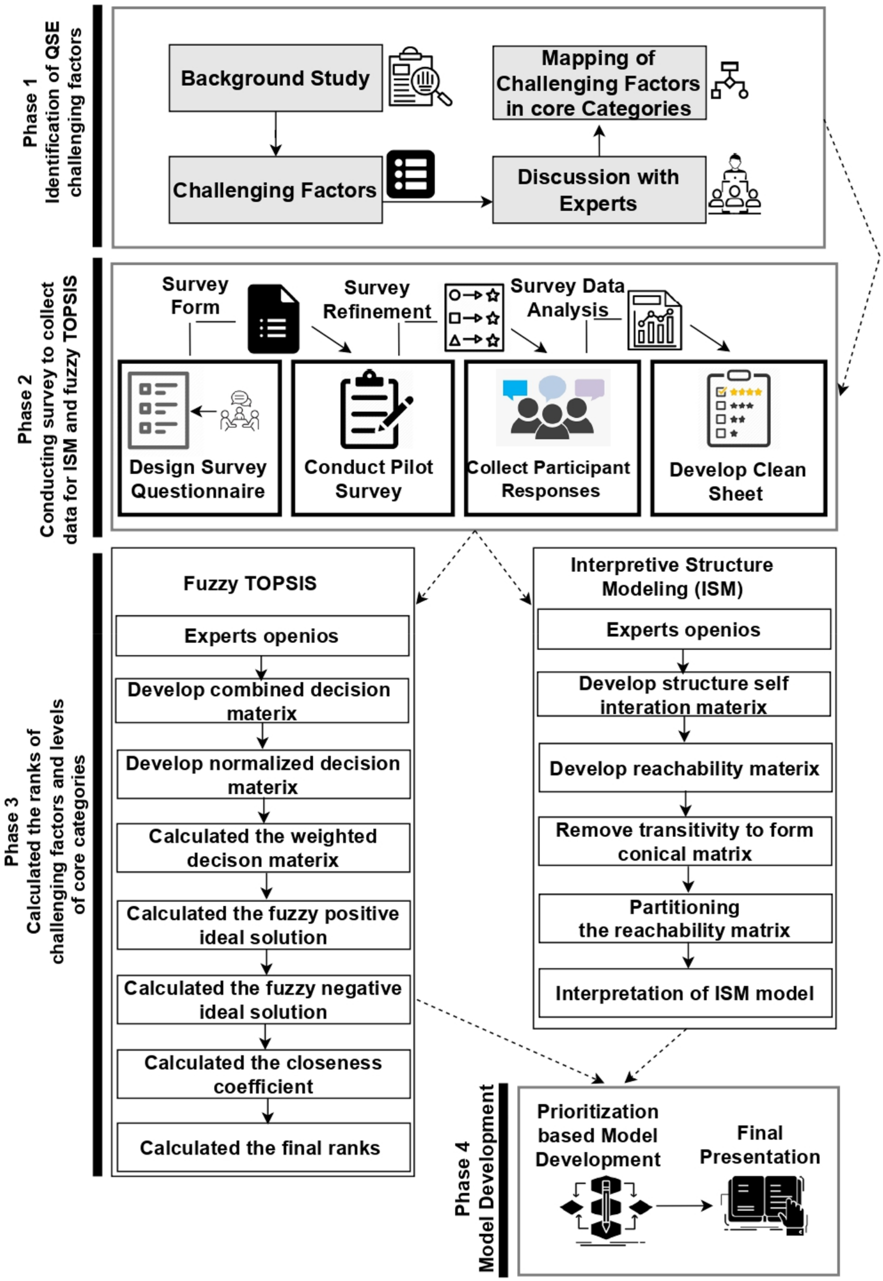


**Fig. 1** Proposed research setting

**Step–2** Conduct survey to collect data for ISM and fuzzy TOPSIS.

**Step–3** Calculating the ranks of challenging factors and levels of the core categories.

**Step–4** Develop the final prioritization-based model.

### 3.1 Literature survey

To identify the challenging factors of quantum software engineering (QSE), we conducted a literature survey by exploring journals articles, conference proceeding and other relevant literature (Akbar et al. 2020b, 2021a, b; Shameem et al. 2020). We used Google Scholar to search for literature studies, as it provides a user-friendly interface to access scholarly articles from various digital libraries (e.g., Springer Link, IEEE Xplore, ACM Digital Library). This comprehensive approach ensured that no relevant digital library was overlooked. Relevant keywords were used to identify appropriate published articles. Additionally, the snowballing data sampling approach was used to collect potential literature related to the study objective, involving examining reference sections of relevant publications (backward snowballing) and studies citing the selected literature (forward snowballing) (Wohlin 2014). This method led to an increasing sample size as more references and citations were explored (Wohlin 2014).

The survey included literature studies discussing QSE process implementation challenges. Additionally, studies that did not explicitly discuss challenging factors were considered (Niazi 2012, 2015), provided they presented relevant QSE lessons learned and experience reports. Extracting challenging factors from such reports proved difficult and required thorough, in-depth reviews (Niazi 2015; Khan et al. 2021). The study selection process was primarily carried out by the first and second authors, with disagreements resolved through discussion and input from all authors. In total, 65 studies were shortlisted using forward and back snowballing.

These studies were used to structure this article and addressed the research question (RQ1) discussed in Sect. 1. The first and second authors reviewed the selected studies to identify key themes, concepts, and demotivators or challenges related to QSE processes. The third and fourth authors subsequently conducted a second review to refine the initial findings and report any missing information. Following Kitchenham and Charters' guidelines (Kitchenham and Charters 2007), we employed a quantitative method to analyze the extracted data (themes and concepts). We used a coding scheme (Hsieh and Shannon 2005) for quantitative analysis of the extracted data. Applying the coding scheme steps (i.e., "code," "sub-categories," "categories and theory"), we mapped and rephrased the identified concepts to develop the final list of QSE challenging factors. After completing the data extraction process, we rephrased the identified themes and concepts, resulting in a list of 22 challenges critical for the adoption and improvement of QSE processes.

Moreover, we mapped the identified challenging factors to their related core categories. The mapping process was carried out by three authors (author numbers 1, 2, and 4), with the possibility of bias and uncertainty. To address this concern, author number 3 randomly participated in the mapping process to check for bias and validate the results. We also invited three external experts in empirical software engineering research to confirm the validity of the mapping process (Akbar et al. 2022). These experts randomly selected 15 concepts from the identified list and mapped them into core categories of challenging factors. To ensure the reliability of the mapping results, we performed an inter-rater reliability test to compare the mappings of the external experts and study authors. We calculated a non-parametric Kendall's coefficient of concordance (W) (Akbar et al. 2022b), where W = 1 indicates perfect agreement, and W = 0 indicates no agreement. The test outcomes for 15 randomly selected concepts showed a W value of 0.93 ($p$ = 0.004), indicating an acceptable level of agreement between the mappings of study authors and external experts. This confirms that the mapping process was unbiased and consistent. The detailed results of the test are presented in Table 1, and the used code is provided below.

```
"library(qseTools)
qse <- data.frame
(external_ex=c(3,3,3,4,3,4,2,3,3,2),
external_ex=(3,4,3,5,3,4,2,4,3,3),
external_ex3=(3,3,4,4,3,4,1,3,3,3)
authors_abc=(2,3,3,4,2,4,3,3,2,3)
)
KendallW(qse, TRUE)
KendallW(qse, TRUE, test=TRUE)
)
KendallW(t(d.att[, -1]), test = TRUE)
friedman.test(y=as.matrix(d.att[,-1]), groups = d.att$id)"
```

### 3.2 Industrial empirical study

The questionnaire survey approach effectively collects data from a large and targeted population (Lenarduzzi and Taibi 2016; Akbar et al. 2020a). The following steps were adopted to perform the empirical investigation.

#### 3.2.1 Designing the questionnaire

To collect the training data, initially, we employed unstructured interviews with 5 experienced practitioners of software engineering and quantum computing.

**Table 1** Kendall's coefficient of concordance test

| Data Set | Kendall Chi-Squared | df | Subjects | Raters | $p$-value | W |
|---|---|---|---|---|---|---|
| QSE | 33.316 | 13 | 15 | 3 | 0.004267 | 0.93212 |

These interviews were conducted face-to-face via Google Meet and Zoom and lasted approximately 30 min each (Akbar et al. 2022b, c). Based on the results of the interviews and the challenging factors of QSE process mentioned in the literature, we developed a closed-ended survey questionnaire to collect data for applying fuzzy TOPSIS and ISM approaches. We chose this data collection approach as it is a standard method for collecting information from a large and potential population. The questionnaire was divided into two categories: the first category focused on demographic questions, while the second category consisted of closed-ended questions related to project attributes. We utilized a 6-point linguistic scale, which included the following levels: just equal (JE), equally important (EI), weakly important (WI), strongly more important (SMI), very strongly more important (VSMI), and absolutely more important (AMI) (Bozbura et al. 2007). The questionnaire survey method has been widely recognized as an effective means for obtaining information that is difficult to gather using observational techniques, as demonstrated in numerous studies (Ikart 2019; Sanchez 1992).

#### 3.2.2 Pilot assessment of the questionnaire

To ensure the reliability and consistency of the survey instrument, a pilot evaluation was conducted after designing the questionnaire. Pre-testing of the survey was deemed significant in previous research, as it helps to enhance the quality of the questionnaire and collect appropriate responses from the population (Khan et al. 2021; Failed 2023b). Ten experts, including five from the previous unstructured interviews and four new ones, were asked to participate in the pilot testing. The experts belonged from the Nanjing University of Aeronautics and Astronautics, China and Griffith University, Australia, respectively, and three industrial experts from Virtual Force, Pakistan; Integrio Systems, Canada; and Startup Development House, Poland, respectively. After receiving experts' suggestions, the questionnaire was finalized, which finally includes three sections: demographic details, closed-ended queries on attributes, and fuzzy TOPSIS and ISM related information for implementing specific attributes. The questionnaire items were paraphrased to improve readability, and one expert suggested presenting survey questions in the form of a table.

## 4 Analysis of the empirical data

We used the frequency analysis approach to analyze the data, which is suitable for analyzing the descriptive types of data (Kitchenham and Pfleeger 2002). It comparatively analyzes the survey variables and computes the agreement level between the survey participants based on the selected Likert scale. In this study, we used the fuzzy TOPSIS and ISM approaches for ranking the challenges concerning to QSE process (Niazi et al. 2016; Ali and Khan 2016; Akbar et al. 2018; Keshta et al. 2017; Mahmood et al. 2017).

### 4.1 Phase 2: ISM approach

Sage (Sage 1977) defined Interpretive Structural Modeling (ISM) as a methodology that brings structure and direction to the intricate relationships among factors and systems, resulting in a comprehensive and systematic model. ISM is an interactive learning technique that aids in organizing related factors, either directly or indirectly, within a holistic model. This model provides a coherent and graphical representation of the concept (Sage 1977; Ravi and Shankar 2005). By addressing complexities surrounding the relationships among various factors, ISM enhances our understanding of these relationships. Several studies have employed this approach to develop conceptual models that illustrate the connections between factors (Kannan et al. 2009; Sharma and Gupta 1995; Agarwal and Vrat 2017).

However, ISM may be influenced by interpersonal bias in expert opinions, potentially affecting the results. Furthermore, ISM does not assign weights for analyzing the ranking of each factor at a level. To address these concerns, we employed the fuzzy TOPSIS approach in subsequent steps, prioritizing QSE challenging factors based on their relationships with the seven core categories of QSE challenging factors. The ISM approach was used to explore the interactions among core categories of QSE challenging factors (programming, standards, Technical, Resources, Standardization, expertise, Responsiveness, and management). Figure 1 displays the detailed steps followed to implement the ISM approach, drawing from the study by Raj and Attri (Raj and Attri 2011).

### 4.2 Phase 3: fuzzy TOPSIS

In 1981, Hwang and Yoon (Yoon and Hwang 1985) introduced the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), with the goal of identifying positive and negative ideal solutions. A positive ideal solution maximizes beneficial criteria while minimizing costly ones, whereas a negative ideal solution does the opposite, maximizing costs and minimizing benefits (Yoon and Hwang 1985). The most desirable solution is one that closely approximates the positive ideal solution and is distant from the negative ideal solution.

To address the challenges of multi-criteria decision-making, Chen and Tsao (Chen and Tsao 2008) developed Fuzzy TOPSIS, which incorporates expert opinions on a specific topic. Decision-makers assign weights to each criterion using linguistic variables, which are subsequently transformed into fuzzy triangular numbers (TNFs). TNFs effectively manage the inherent vagueness of linguistic terms expressed by decision-makers (Rafi et al. 2020; Kannan et al. 2014; Krohling and Campanharo 2011). The algorithms for Fuzzy TOPSIS, as applied to multi-criteria decision-making, are outlined below.

**Step 1**. Consider a decision-making problem that involves m alternatives and *n* evaluation criteria. This problem can be represented as a matrix, where A1, A2,…, Am denote the alternatives, and $E_1$, $E_2$,…, $E_n$ represent the evaluation criteria. The performance rating of alternative Fi with respect to criterion $E_j$, denoted by $F_{ij}$,

**Table 2** Fuzzy triangular scale

| Linguistic terms | Triangular fuzzy scale |
|---|---|
| Just Equal (JE) | (1,1,1) |
| Equally Important (EI) | (0.5,1,1.5) |
| Weakly Important (WI) | (1,1.5,2) |
| Strongly More Important (SMI) | (1.5,2,2.5) |
| Very Strongly More Important (VSMI) | (2,2.5,3) |
| Absolutely more important (AMI) | (2.5,3,3.5) |

is assessed by the decision-makers. Furthermore, each criterion $E_j$ is assigned a weight, represented by $W_j$.

$$E_1,\ E_2,\ \dots E_n$$

$$D = \left(F_{ij}\right)_{m*n} = \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{pmatrix} \tag{1}$$

**Step 2**. Revise the given alternatives and their associated weighted criteria based on Eq. (1). Allocate ratings to each specific criterion and corresponding alternatives utilizing Bozbura et al.'s (Bozbura et al. 2007) fuzzy triangular scale, which is depicted in Table 2.

**Step 3**. Determine the aggregate fuzzy rating of *K* decision-makers for each criterion by using Eqs. (2) and (3).

$$A_{ij} = K^{\min}\left\{x_{ijk}\right\},\ b = \frac{1}{K}\sum_{k=1}^{K} y_{ijK},\ c = K^{\max}\left\{z_{ijK}\right\} \tag{2}$$

$$W_{j1} = K^{\min}\left\{x_{jK1}\right\},\ b = \frac{1}{K}\sum_{k=1}^{K} y_{jK2},\ c = K^{\max}\left\{z_{jK3}\right\} \tag{3}$$

where $A_{ij} = (x_{ij}, y_{ij}, z_{ij})$ and i = 1,2,3,…m, and j = 1,2,3…,n, and weight of each criterion is calculated as $W_j = (W_{j1}, W_{j2,} W_{j3})$.

**Step 4**. Calculate the normalized decision matrix "R" using linear scale transformation. After normalization, the resulting matrix will be represented as:

$$R^{\sim} = \left[r_{ij}\right]_{m*n} \tag{4}$$

Equations (5) and (6), given below, are used to calculate each alternative's cost and benefit criteria.

$$r_{ij} = \left( \frac{x_{ij}}{z_j^+}, \frac{y_{ij}}{z_j^+}, \frac{z_{ij}}{z_j^+} \right) \textit{ and } z_j^+ = \max_i z_{ij} (\textit{benefit criteria}) \tag{5}$$

$$r_{ij} = \left( \frac{x_j^-}{z_{ij}}, \frac{x_j^-}{y_{ij}}, \frac{x_j^-}{x_{ij}} \right) \textit{ and } x_j^- = \max_i l_{ij} (\cos t \textit{ criteria}) \tag{6}$$

**Step 5**. Calculate the weighted normalized fuzzy decision matrix V ~ by multiplying the normalized fuzzy decision matrix values with the corresponding criteria weights $W_j$.

$$V \cong \left[ V_{ij} \right]_{m*n} \tag{7}$$

where $v_{ij}$ is calculated by using Eq. (8) and (7)

$$v_{ij} = A_{ij} * W_j \tag{8}$$

**Step 6**. In this stage, the fuzzy positive and negative ideal solutions are computed as demonstrated in Eqs. (9) and (10).

$$A^+ = \left[ v_{1,}^+ v_{j,}^+ \dots ..v_m^+ \right] \tag{9}$$

$$A^- = \left[ v_{1,}^- v_{j,}^- \dots ..v_m^- \right] \tag{10}$$

The values of positive and negative ideal solutions range between [0.1].

**Step 7**. Determine the distance of each alternative from a positive and negative ideal solution.

$$D_i^+ = \sum_{j=1}^{n} D_v \left( v_{ij}, \ v_j^+ \right) \tag{11}$$

$$D_i^- = \sum_{j=1}^{n} D_v \left( v_{ij}, \ v_j^- \right) \tag{12}$$

where D represents the distance between two fuzzy numbers.

**Step 8**. To compute the ranks of alternatives, the CCi value is calculated by considering a fuzzy positive and negative ideal solution.

$$CCi = \frac{D_i^-}{D_i^+ + D_i^-} \tag{13}$$

**Step 9**. Provide a ranking of all alternatives based on their CCi value in order of priority, where alternatives with higher CCi values receive a higher rank.

**Table 3** Identified list of QSE challenging factors

| Categories | Sr.# | Challenges |
|---|---|---|
| Programming (C1) | Ch1 | Complex programming |
| | Ch2 | Limited simulation resources |
| | Ch3 | High error rates |
| Technical (C2) | Ch4 | Lack of debugging tools |
| | Ch5 | Limited software libraries |
| | Ch6 | Maintenance complexity |
| Resources (C3) | Ch7 | Complexity of quantum algorithms |
| | Ch8 | Data encoding issues |
| | Ch9 | Difficult to optimize quantum software |
| Standardization (C4) | Ch10 | Lack of standardization |
| | Ch11 | Lack of interoperability |
| | Ch12 | Lack of requirements engineering strategies |
| Expertise (C5) | Ch13 | Lack of expertise |
| | Ch14 | Integration with classical computing |
| | Ch15 | Lack of training and workshops |
| Responsiveness (C6) | Ch16 | Security issue in QSE |
| | Ch17 | Ethical issue in QSE |
| Management (C7) | Ch18 | Verification and validation issues |
| | Ch19 | QSE scalability issues |
| | Ch20 | Budget constraints |
| | Ch21 | Lack of commercial applications |
| | Ch22 | Project management issues |

## 5 Results and discussions

In this section, we present the results and analysis of our study. The identified list of QSE challenging factors is explained in Sect. 4.1. The results of the ISM and fuzzy TOPSIS analyses are presented in Sect. 4.2. By examining these findings in-depth, we aim to shed light on the key challenging factors that impact QSE success and provide insights for practitioners and researchers alike.

### 5.1 Identified challenges of QSE

Quantum software engineering (QSE) is an emerging field that aims to develop quantum-intensive software by leveraging the engineering processes, reference architectures, patterns, tools, and frameworks of traditional software engineering. QSE focuses on activities such as quantum domain engineering, quantum system co-design, quantum algorithm design, and source coding and quantum information simulation, and these activities facing several critical challenges. A list of most significant QSE challenges is given in Table 3 and describe below.

**Ch1 (Complex programming):** Quantum software engineering involves programming for quantum computers that operate using quantum phenomena such as superposition, entanglement, and interference (Nielsen and Chuang 2010). These phenomena result in complex behavior that is different from classical computers, requiring new techniques and algorithms for developing quantum software. For example, quantum algorithms like Shor's algorithm for factoring large numbers, Grover's algorithm for searching, and quantum error correction require understanding and utilizing quantum complexity (Botsinis et al. 2016). Developing such algorithms and techniques remains an active area of research.

**Ch2 (Limited simulation resources):** Quantum computers are currently limited in terms of the number of qubits, gate fidelity, and coherence time, which poses significant challenges for developing scalable and efficient quantum software (Shi et al. 2020). As a result, simulations of quantum systems are used to test and optimize quantum software, but these simulations also have limitations in terms of the number of qubits they can simulate accurately (Shi et al. 2020). Therefore, it is difficult to predict how quantum algorithms and software will perform on larger systems without access to larger and more powerful quantum computers.

**Ch3 (High error rates)** Quantum computers are susceptible to noise and errors due to their inherent fragility, which makes it challenging to develop reliable quantum software (Shi et al. 2020). The errors can arise from various sources, including decoherence, environmental noise, and hardware imperfections, and can impact the accuracy and efficiency of quantum algorithms (Shi et al. 2020; Devitt et al. 2013). Error correction techniques are essential for reducing these errors, but they require additional resources and overhead (Devitt et al. 2013).

**Ch4 (Lack of debugging tools):** Debugging quantum software is challenging due to the lack of mature debugging tools and techniques (Nagori and Varadarajan 2023). Traditional debugging techniques for classical software are not always applicable to quantum software, as the behavior of quantum systems is fundamentally different from classical systems (Nagori and Varadarajan 2023). Moreover, the complexity of quantum algorithms and the presence of quantum effects such as entanglement and superposition make it difficult to trace and isolate errors (Moll et al. 2021). There is a need for developing specialized debugging tools and techniques for quantum software engineering (Moll et al. 2021).

**Ch5 (Limited software libraries):** Developing complex applications for quantum computing is challenging due to the limited availability of software libraries. The current state of quantum software development is still in its early stages, and there is a need for more comprehensive software libraries that can support a wide range of applications (Veryazov et al. 2004). Although there are several open-source quantum software development platforms available, the libraries they provide are often limited in scope and functionality. The development of new software libraries is crucial for advancing the field of quantum software engineering (Devitt et al. 2013).

**Ch6 (Maintenance complexity):** Maintaining quantum software is a challenging task due to the fast-paced development of quantum hardware and software, as well as the complexity of quantum algorithms (Moll et al. 2021). The rapid evolution of quantum hardware can lead to changes in the optimal software design and the need for frequent updates (O'Riordan and Jerger 2019). Moreover, the complexity of quantum algorithms and the lack of standardization can make it challenging to maintain and update quantum software. Efforts are being made to develop best practices for quantum software maintenance, such as version control and testing frameworks (O'Riordan and Jerger 2019).

**Ch7 (Complexity of quantum algorithms):** Quantum algorithms are fundamentally different from classical algorithms, which makes it challenging for developers to understand and implement them. The development of quantum algorithms requires expertise in both quantum mechanics and computer science (Moll et al. 2021). Moreover, quantum algorithms are highly dependent on the specific characteristics of the quantum hardware, which adds an additional layer of complexity. To address these challenges, efforts are being made to develop more intuitive and user-friendly quantum programming languages and tools (Nielsen and Chuang 2010).

**Ch8 (Data encoding issues):** Quantum computers require data to be encoded in quantum states, which can be challenging to implement in practice (Nielsen and Chuang 2010). The choice of data encoding can significantly impact the performance of quantum algorithms, and there is no one-size-fits-all approach (Coles et al. 2018). Moreover, the encoding process can be sensitive to noise and errors, which can lead to a degradation in the performance of the quantum algorithm. Efforts are being made to develop new data encoding techniques that are robust to noise and errors (Coles et al. 2018).

**Ch9 (Difficult to optimize quantum software):** Optimizing the performance of quantum software is challenging due to the complexity of quantum algorithms and hardware. Quantum software needs to be optimized for the specific hardware architecture on which it will run, which can be difficult due to the rapid pace of hardware development (Gambetta and Cross 2018). Techniques such as circuit optimization, compiler optimization, and algorithmic improvements are being developed to address these performance optimization challenges (Pednault et al. 2019).

**Ch10 (Lack of standardization):** Standardization refers to the process of defining common interfaces, protocols, and data formats that allow different systems to work together seamlessly. In quantum software engineering, the lack of standardization creates challenges for developers in creating software that can work across different quantum computing platforms (Fingerhuth et al. 2018). This makes it difficult to create interoperable software that can be easily used by other developers and organizations (Gill et al. 2022). The absence of standardization also leads to higher costs and longer development times as developers must create custom solutions for each platform. Standardization efforts are currently underway in the quantum computing industry to address this issue (Gill et al. 2022).

**Ch11 (Lack of Interoperability):** Interoperability between different quantum hardware and software platforms is a major challenge in quantum software engineering. The lack of standardization in quantum computing makes it difficult to create interoperable software (Helsen and Raedt 2020). Developers need to consider hardware-specific details such as gate sets, noise models, and connectivity when designing quantum software, which can limit interoperability (Coveney and Highfield

2019). Efforts are being made to develop standardization efforts to address these interoperability issues (Coveney and Highfield 2019).

**Ch12 (lack of requirements engineering strategies):** Requirements engineering, and management are critical aspects of software development (Pandey et al. 2010). However, there is a lack of tools and techniques specific to quantum software engineering, making it challenging for developers to gather and manage requirements effectively (Sodhi and Kapur 2021). The unique features of quantum computing, such as entanglement and superposition, also require new approaches to requirements engineering (Sodhi and Kapur 2021). Addressing these challenges is crucial to ensure the development of reliable and effective quantum software.

**Ch13 (Lack of expertise):** Quantum computing is a relatively new and rapidly evolving field, and there are limited experts available to guide and mentor developers in quantum software engineering (Moll et al. 2021). The field of quantum software engineering requires expertise in both quantum mechanics and computer science, making it challenging for developers to acquire the necessary skills and knowledge (Shaydulin et al. 2020). Moreover, the current shortage of quantum computing experts limits the availability of experienced mentors and educators (Shaydulin et al. 2020). Efforts are being made to address this shortage, including training programs, online courses, and academic research.

**Ch14 (Integration with classical computing)** Quantum software engineering requires seamless integration with classical computing systems, which adds complexity to the software engineering process (Bravyi et al. 2022). The integration problem arises because quantum computers have different architectures and programming paradigms than classical computers (Bravyi et al. 2022). Moreover, hybrid quantum–classical algorithms, which are essential for solving practical problems, require efficient communication between the classical and quantum components (Moll et al. 2021). Efforts are being made to address the integration problem, including the development of quantum–classical interfaces and middleware (Moll et al. 2021).

**Ch15 (Lack of training and workshops)** Quantum software engineering is an emerging field that is rapidly developing with the advancement of quantum computing. However, there is a lack of training and workshops available for individuals interested in pursuing a career in this field (Lanzagorta et al. 2020). This is due to the novelty and complexity of the subject matter, as well as the limited number of experts in the field. The lack of resources for training and workshops in quantum software engineering is a significant challenge that needs to be addressed to facilitate the growth of the field and to prepare a workforce for quantum software engineering jobs (Lanzagorta et al. 2020; Cao and Romero 2018). To overcome this challenge, organizations and academic institutions are developing training programs and workshops to teach individuals how to develop quantum software.

**Ch16 (Security issue in QSE):** Quantum computers have the potential to break many of the currently used cryptographic protocols, which poses a significant challenge for the development of secure quantum software (Bravyi et al. 2022). The cryptographic protocols that are secure on classical computers may become vulnerable to attacks on quantum computers due to their ability to perform certain mathematical operations exponentially faster (Veryazov et al. 2004). To develop secure quantum software, new cryptographic protocols and algorithms that are resistant to quantum attacks need to be developed (Veryazov et al. 2004).

**Ch17 (Ethical issue in QSE)**: The development of quantum software raises ethical considerations related to data privacy, security, and access. As quantum computing has the potential to break many of the currently used cryptographic protocols, the development of quantum-resistant encryption algorithms is necessary to protect sensitive information (Poczatek et al. xxxx). Additionally, the development of quantum software for military applications raises concerns about the potential misuse of quantum technology (Singh and Walia 2020).

**Ch18 (Verification and validation issues):** Verification and validation of quantum software is challenging due to the complexity of quantum systems and the lack of reliable simulation tools (Altman et al. 2021). In addition, quantum hardware is susceptible to noise and errors, which further complicates the verification and validation process (Cross et al. 2018). Efforts are being made to develop new verification and validation techniques specifically tailored for quantum software, such as quantum circuit optimization and error correction (Altman et al. 2021).

**Ch19 (QSE Scalability issues):** Developing scalable quantum software is challenging due to the limitations of current quantum hardware. Quantum computers are currently limited in terms of the number of qubits, gate fidelity, and coherence time, which restricts the size and complexity of quantum algorithms that can be run on them (Gambetta et al. 2017). Moreover, the lack of standardization and limited expertise in quantum software engineering adds to the challenge of developing scalable quantum software. Efforts are being made to develop new hardware and software solutions to improve the scalability of quantum computing (Moll et al. 2021).

**Ch20 (Budget constraints):** Quantum hardware and software development is currently expensive, which makes it challenging for organizations to invest in quantum technology (Bravyi et al. 2022). The high cost of quantum hardware and the limited availability of quantum experts also contribute to the high cost of quantum software engineering (Cross et al. 2018). However, efforts are being made to reduce the cost of quantum technology, such as developing cloud-based quantum computing services and open-source quantum software libraries (Cross et al. 2018).

**Ch21 (Lack of commercial applications):** The lack of commercial applications of quantum computing can be attributed to the current limitations of quantum hardware and software, as well as the high costs involved in quantum research and development (Cross et al. 2018). However, there are efforts underway to address these challenges and bring quantum computing to industry. As more organizations invest in quantum technology and new quantum applications are developed, the commercial potential of quantum computing is expected to grow (Bova et al. 2021). Nevertheless, the lack of commercial applications is still a challenge for developers in terms of applying their skills and knowledge in real-world scenarios (Awschalom et al. 2021).

**Ch22 (Management issues)**: Quantum software engineering (QSE) is an emerging field that involves the development of software for quantum computers. However, managing QSE projects can be challenging due to the complexity and novelty of the technology, as well as the lack of established best practices (Yigitbasioglu and Kocaturk 2021). Some of the management problems in QSE include the difficulty of estimating project timelines, the need for specialized expertise, and the challenge of testing and debugging quantum software (Babbush et al. 2018). Effective management strategies and tools are needed to address these issues and ensure the successful development of quantum software (Chong et al. 2017).

## 5.2 Empirical analysis

In this section, we present the results and analysis of the questionnaire survey study. Section 4.2.1 introduces the demographic information of the survey participants. The step-by-step implementation of the Interpretive Structural Modeling (ISM)

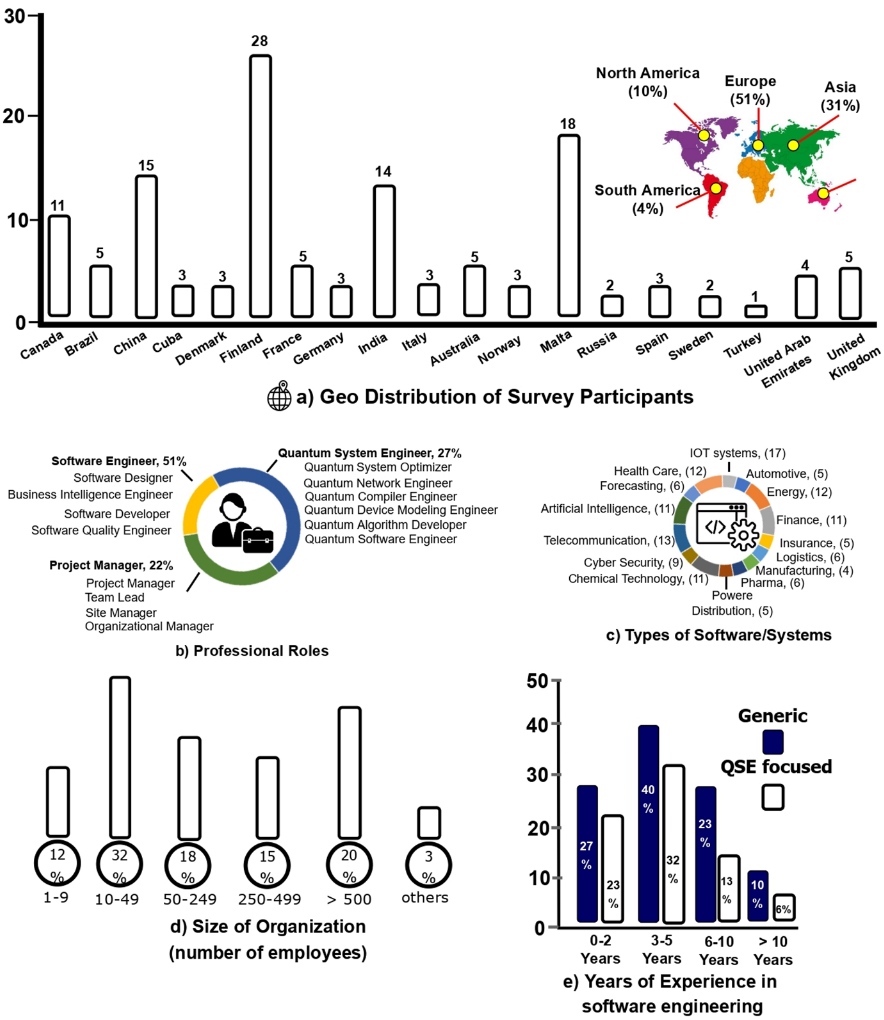


**Fig. 2** Survey Participants demographics

approach is explored in Sect. 4.2.2. Lastly, the application of the fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) approach is discussed in Sect. 4.2.3.

### 5.2.1 Demographics

Demographic information is crucial in questionnaire survey studies as it helps to identify patterns and trends among different population groups, enabling researchers to better understand and generalize their findings across diverse contexts (Saris and Gallhofer 2014; Alderman and Salem 2010). We performed a frequency analysis to assess the descriptive data, an effective technique for analyzing various types of variables, including both numeric and ordinal data. After conducting the survey, we received 133 complete responses. However, the computational complexity of the method increases with the size of the dataset. Therefore, to handle large data set we have used excel. As depicted in Fig. 2a, the survey participants represented 19 different countries across 5 continents. Notably, a majority of the participants (50%) were from Europe, suggesting that the region is at the forefront of quantum technology adoption and implementation. This also implies that Europe is an ideal location for sourcing quantum-related information, research, and industry positions.

We categorized the roles of the survey participants into three groups: software engineer, quantum system engineer, and project manager. Our analysis revealed that 51% of the participants were software engineers. The project types of the survey participants were also analyzed, with 15 distinct projects mentioned (see Fig. 2c). Additionally, we evaluated the organization sizes of the participants and found that 32% of their organizations had 10–49 employees (Fig. 2d).

Regarding the experience of the survey participants, we inquired about their general experience in the software engineering field and their specific experience in quantum applications. The results (Fig. 2e) showed that 40% of the participants had 3–5 years of general experience in software development organizations, while 32% had experience in the quantum domain (Fig. 2e).

## 5.3 Results of ISM approach

The ISM approach has proved useful in analyzing the complex interactions among the major categories of QSE challenging factors. Numerous studies have utilized this approach to investigate the contextual interactions of different elements (Kannan et al. 2009; Sharma and Gupta 1995; Agarwal and Vrat 2017). In order to establish a comprehensive understanding of how the criteria interact with each other, it is crucial to develop a structural-self-interaction matrix (SSIM). The SSIM is a powerful tool for visualizing and evaluating the relationships between the criteria, and its construction and application will be discussed in the following sections.

**Table 4** SSIM matrix

| | C7 | C6 | C5 | C4 | C3 | C2 | C1 |
|---|---|---|---|---|---|---|---|
| C1 | V | O | X | V | O | O | * |
| C2 | O | O | O | O | O | * | * |
| C3 | V | O | O | O | * | * | * |
| C4 | V | X | X | * | * | * | * |
| C5 | O | V | * | * | * | * | * |
| C6 | O | * | * | * | * | * | * |
| C7 | * | * | * | * | * | * | * |

#### 5.3.1 Structural-self-interaction matrix

We used the ISM approach to analyze the contextual relationships among key categories of QSE challenging factors, with input from industry and R&D experts. The survey participants demographics are presented in Fig. 2. Using their input, we constructed the SSIM matrix. While we acknowledge that our sample size may limit the generalizability of our findings, previous studies, such as Kannan et al. (Kannan et al. 2009), Soni et al. (Soni 2015), and Attri et al. (Attri et al. 2013a), successfully employed similar sample sizes of experts in their decision-making processes. Thus, we deemed 133 experts sufficient for our ISM model. The direction of the relationship between the QSE challenging factors (m and n) is indicated by the following symbols.

- 'V' indicates the relationship from m enabler to n enabler.
- 'O' indicates the relationship from n enabler to m enabler.
- 'X' indicates when both enablers m and n reach each other.
- 'O' indicates a situation in which there is no relationship between enabler m and enabler n.

Based on experts' opinions, we developed the SSIM presented in Table 4.

Table 4 presenting the identified several core categories of challenging factors that impact QSE process improvement. We analyzed the relationships between these categories and represented them using symbols such as 'V', 'X', 'O', and 'A'. For example, we found that there was no relationship between C1 (Programming) and C2 (Limited simulation resources), as their relationship is represented with an 'O'. Similarly, C2 (Limited simulation resources) has no relationship with C7 (Management). However, we noted that C1 helps to improve C7 as they are in a 'V' type relationship. Furthermore, according to our results, C1 and C5 have an 'X' relationship, which indicates that both categories can help to improve QSE process improvement. Interestingly, we found no 'A' type relationships between the core categories of QSE challenging factors, meaning that the relationships were clear and unambiguous. Our findings suggest that addressing these challenging factors in a coordinated manner can lead to significant improvements in QSE process improvement.

**Table 5** Reachability matrix

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| C1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| C2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| C3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| C4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| C5 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| C6 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| C7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Table 6** Transitivity check

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | DIV | RANK |
|---|---|---|---|---|---|---|---|---|---|
| C1 | 1 | 0 | 0 | 1 | 1 | 1* | 1 | 5 | 4 |
| C2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| C3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 |
| C4 | 1* | 0 | 0 | 1 | 1 | 1 | 1 | 5 | 4 |
| C5 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 4 | 3 |
| C6 | 1* | 0 | 0 | 1 | 1* | 1 | 1* | 5 | 4 |
| C7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| DEP | 4 | 1 | 1 | 4 | 4 | 4 | 5 | 23 | |
| RANK | 2 | 1 | 1 | 2 | 2 | 2 | 3 | | |

*The identity of a specific challenging factor

### 5.3.2 Reachability matrix

To develop the reachability matrix, we converted the values of V, A, X, and O into binary digits (0, 1), using the following rules:

- When m and n in the SSIM have a value of V, replace it with 1; otherwise, assign a value of 0.
- When m and n in the SSIM have a value of A, replace it with 0; otherwise, assign a value of 1.
- When m and n in the SSIM have a value of X, replace it with 1 and assign 1 to both n and m entries.
- When m and n in the SSIM have a value of O, replace it with 0 and assign 0 to both n and m entries.

We develop the reachability matrix (Table 5) by applying these protocols. To obtain the final reachability matrix, we established transitivity as detailed in Sect. 3.3. We employed the 1* value to incorporate transitivity, which helps to fill in gaps in the data collected from experts during SSIM development. The integration of the transitivity check is presented in Table 6.

Table 6 outlines the criteria for driving power, dependence power, and their corresponding ranks. Driving power represents the set of essential criteria required to successfully address a specific category of QSE challenging factors. On the other hand, dependence power refers to the criteria that may contribute to achieving these objectives. Utilizing both driving and dependence power, we can conduct a cross-impact matrix multiplication applied to classification (MICMAC) analysis. This process allows us to categorize the criteria into four distinct clusters: autonomous, dependent, linkage, and independent clusters.

#### 5.3.3 Partitioning the reachability matrix

Warfield (Warfield 1974) stated that the reachability set comprises the element itself and any other elements it may contribute to achieving. In contrast, the antecedent set includes the element itself and any other elements that may facilitate its achievement. Subsequently, the intersection of these sets is calculated for all elements. Elements with identical reachability and intersection sets are positioned at the top level of the ISM hierarchy. These top-level elements do not assist in achieving any other elements above their level.

After identifying the top-level element, it is isolated from the remaining elements. The process is then repeated to ascertain the elements in the subsequent level. This procedure continues until the level of each element is determined. These levels aid in constructing the diagraph and the ISM model. In this study, we examine 7 criteria (categories of QSE challenging factors) and present their reachability set, antecedent set, intersection set, and levels in Table 7.

**Table 7** Leveling of final reachability matrix

| | Level Partitions | | | |
|---|---|---|---|---|
| | Lteration One | | | |
| Categories | Reachability Set | Antecedent Set | Intersection Set | Levels |
| C1 | 1,4,5,6,7 | 1,4,5,6 | 1,4,5,6 | |
| C2 | 2 | 2 | 2 | Level 1 |
| C3 | 3,7 | 3 | 3 | |
| C4 | 1,4,5,6,7 | 1,4,5,6 | 1,4,5,6 | |
| C5 | 1,4,5,6 | 1,4,5,6 | 1,4,5,6 | Level 2 |
| C6 | 1,4,5,6,7 | 1,4,5,6 | 1,4,5,6 | |
| C7 | 7 | 1,3,4,5,7 | 7 | Level 3 |
| | Lteration Two | | | |
| C3 | 3 | 3 | 3 | Level 4 |

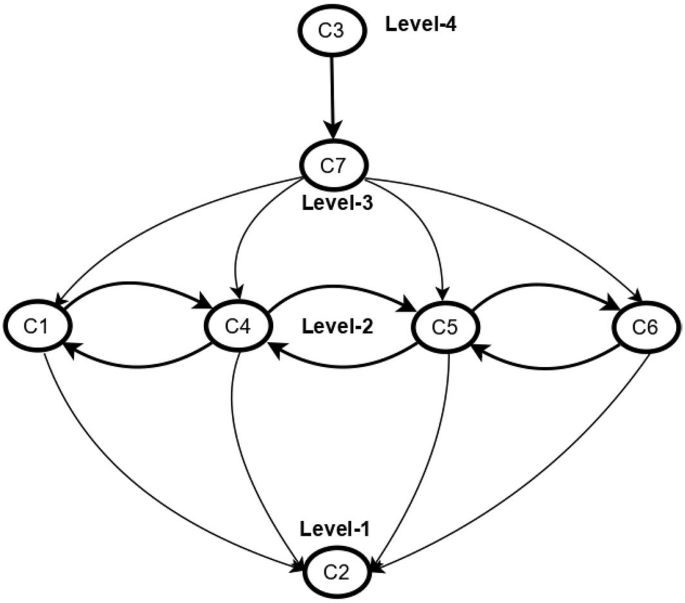


**Fig. 3** Leveling of core categories of QSE challenging factors

#### 5.3.4 Interpretation of the ISM model

The final ISM model was created based on the outcomes of the reachability matrix analysis (Table 7). The interconnections between categories of QSE challenging factors are depicted using arrows that indicate the direction of influence from one criterion to another. Transitivity analysis was conducted to eliminate ambiguity in the data, which led to the transformation of the diagraph into the ISM model (Fig. 3).

The C3 (Resources) category emerges as the most influential factor among the identified QSE challenging factors, signifying that it is an independent category. All other categories rely on C3. Notably, the C7 (Management) category depends solely on C3 (Resources), while the remaining categories are influenced by C7 (Management). As shown in Fig. 3, C1(Programming), C4 (Standardization), C5 (Expertise), and C6 (Responsiveness) belong to level-2 and are influenced by both C3 and C7, exhibiting strong interrelationships. These categories (C1, C4, C5, and C6) demonstrate a high degree of dependence and driving power for C2 (Technical), which is situated at level 1. C2 (Technical) has a strong dependence on all other categories of QSE challenging factors but lacks any driving power.

In summary, C3 (Resources) possesses a potent driving force without any dependence, while C2 (Technical) exhibits a strong dependency but lacks driving power. Practitioners should take these findings into account when developing QSE process implementation strategies, considering the relationships among the core categories of QSE challenging factors.

#### 5.3.5 MICMAC analysis

The MICMAC analysis is a useful tool for understanding the categories that drive a system. Attri et al. (Attri et al. 2013a) define the MICMAC as an analysis that examines the driving and dependence power of categories. By using this analysis, the influencing factors of QSE can be classified into four clusters based on their driving and dependence power. Overall, the MICMAC analysis can provide valuable insights into the categories that drive a system and their level of impact on the overall system.

- **Autonomous cluster:** The first cluster is the autonomous cluster, which consists of categories with weak driving and dependence power. These categories are largely disconnected from the system due to weak links, and therefore have a minor impact on the overall system.
- **Linkage cluster:** The second cluster is the linkage cluster, which consists of categories with strong driving and dependence power. These categories have a significant impact on the system and affect other enablers due to their strong linkage.
- **Dependent cluster:** The third cluster is the dependent cluster, which consists of enablers with strong dependence power but weak driving power. These categories rely heavily on other enablers to function effectively.
- **Independent cluster:** The independent cluster consists of enablers with weak dependence power but strong driving power, also known as "key enablers." These categories have a significant impact on the system and are crucial for its success.

#### 5.3.6 Development of conical matrix

The primary objective of developing a conical matrix is to conduct a MICMAC analysis. The conical matrix, presented in Table 6, is created by analyzing the data provided in Tables 7 and 8. To create the conical matrix, the categories are first ordered based on their level number, as shown in Table 8. Then, the values of each category are taken from Table 7. For example, the value of C2 (Technical) across

**Table 8** Conical matrix after clustering categories of QSE challenging factors

| | C3 | C7 | C1 | C4 | C5 | C6 | C2 | DIV |
|---|---|---|---|---|---|---|---|---|
| C3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| C7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| C1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 5 |
| C4 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 4 |
| C5 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 4 |
| C6 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 5 |
| C2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| DEP | 1 | 5 | 4 | 4 | 4 | 4 | 1 | |

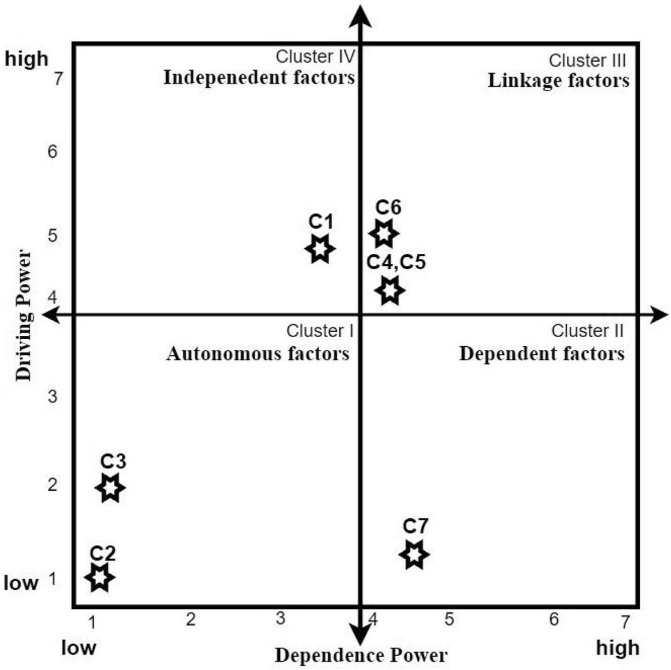


**Fig. 4** Graphical view of MICMAC analysis

rows and columns of the transitivity matrix in Table 7 indicates that it has no relationship with any other criterion, except itself (C2), which has a value of "0." Similarly, the value of C7 indicates a relationship of "1" with C1, C3, C4, C6, and C7, and a relationship of "0" with the rest of the criteria. This procedure is repeated for all categories, and the resulting matrix is presented in Table 8. Therefore, the conical matrix allows for a more comprehensive understanding of the relationships between categories and their driving and dependence power in the system, which is crucial for conducting a thorough MICMAC analysis.

To classify the categories of QSE challenging factors, we used the approach proposed by (Kannan et al. 2009) and presented the MICMAC analysis results in Fig. 4. The categories were classified into four clusters based on their driving and dependence power: autonomous, dependent, and independent clusters, and key categories. The first cluster includes autonomous categories, and C3 (Resources) was found to be part of this cluster. This suggests that C3 is largely disconnected from the system due to weak links with other categories of QSE challenging factors. The second cluster includes dependent categories, and C7 (Management) belongs to this cluster. This indicates that C7 has strong dependence power but weak driving power. C4 (Standardization), C5 (Expertise), and C6 (Responsiveness) were found to have strong driving and dependence power and were classified into the third and fourth clusters. These categories have a significant impact on the system and affect other categories of QSE challenging factors due to their strong linkage. C1 (Programming) was classified as an independent cluster, indicating that it has weak dependence power but strong driving power. This category is also considered a key category of QSE challenging factors. Thus, this MICMAC analysis provides valuable insights into the driving and dependence power of the categories of QSE challenging factors, which can be used to develop effective strategies to address these challenges.

### 5.4 Application of fuzzy TOPSIS

While ISM is helpful in identifying key categories of QSE challenging factors, it may not fully account for uncertainties and vagueness in decision-making. To address this, we used the fuzzy TOPSIS approach to rank the challenging factors by their priority in contributing to the success and progression of QSE processes. The fuzzy TOPSIS approach is a well-established method used in various engineering domains for tackling multicriteria decision-making problems (Rafi et al. 2020; Junior et al. 2014; Liao and Kao 2011; Zouggari and Benyoucef 2012).

To apply the fuzzy TOPSIS approach, we requested feedback from experts who participated in the ISM analysis (see Fig. 2 for details). We designed a questionnaire (Appendix 1) that allowed each expert to rank the challenging factors based on their own experience and understanding. To obtain more representative responses, each participant was allowed to consult with colleagues while ranking the complex factors. The fuzzy TOPSIS approach was then applied using the ranked challenging factors, and the resulting fuzzy TOPSIS results were computed. These results give practitioners a ranking of the highest priority challenging factors for the success and progression of QSE processes. Therefore, the fuzzy TOPSIS approach provides a useful tool for addressing uncertainties and vagueness in decision-making when considering QSE challenging factors. The opinions of experts are valuable in obtaining a comprehensive understanding of the effectiveness of each challenging factor and determining the highest priority factors for QSE success.

**Steps 1 & 2:** To get the insights of experts regarding the effectiveness of QSE challenging factors, we used the fuzzy triangular scale, which yields linguistic values (Table 2) (Kannan et al. 2009; Sharma and Gupta 1995; Agarwal and Vrat 2017).

**Step 3:** We calculated the combined decision matrix using Eq. (1) (Sect. 3.4). The study involved a total of 22 challenging factors that are related to the seven core categories of QSE challenging factors. The resulting combined decision matrix, which includes the collective opinions of all the experts involved in decision-making, is presented in Table 9.

**Step 4:** In this step, we calculated the normalized decision matrix by using Eqs. (2) and (3) (Sect. 3.4). Normalizing the decision matrix is a crucial step in the fuzzy TOPSIS approach as it enables us to compare the challenging factors with different units and scales. To normalize the decision matrix, we need to consider cost and benefit criteria (Attri et al. 2013b). This is a systematic process that allows us to measure the strengths and weaknesses of alternatives and identify the best options for achieving benefits while performing a specific task. In our case, we used "resources" as the cost criterion, as it includes the challenging factors related to "complexity of quantum algorithms, data encoding issues and difficult to

**Table 9** Combined decision matrix

| Sr | Programming | | | Technical | | | Resourses | | | Standardization | | | Experties | | | Responsivness | | | Management | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weights | 0,5 | 1 | 2 | 1,2 | 2,5 | 3,5 | 0,5 | 1,4 | 2,5 | 1,2 | 2,5 | 3,5 | 0,5 | 1 | 1,5 | 0,5 | 1,4 | 2,5 | 0,5 | 1,2 | 2,5 |
| Ch1 | 0,5 | 2 | 3 | 1,5 | 2,4 | 3,5 | 1,5 | 2,4 | 3,5 | 1 | 1,2 | 2,5 | 0,5 | 1 | 1,5 | 1,5 | 2,3 | 3 | 0,5 | 1,3 | 2,5 |
| Ch2 | 0,5 | 1,6 | 3 | 0,5 | 2,1 | 3 | 1,5 | 2 | 3 | 0,5 | 1,3 | 3 | 0,5 | 1,4 | 3 | 1,5 | 2,3 | 3 | 1 | 1,3 | 2 |
| Ch3 | 0,5 | 2 | 3 | 1 | 1,2 | 2,5 | 0,5 | 1,7 | 3 | 1,5 | 1,2 | 3 | 0,5 | 1,2 | 2,5 | 1,5 | 2,4 | 3,5 | 1 | 1,3 | 2 |
| Ch4 | 1,5 | 2,5 | 3 | 0,5 | 2,1 | 3 | 1,5 | 2,3 | 3,5 | 0,5 | 1,2 | 2,5 | 0,5 | 1,2 | 2,5 | 0,5 | 2,1 | 3 | 1 | 1,3 | 2 |
| Ch5 | 0,5 | 2 | 3 | 0,5 | 2,1 | 3 | 0,5 | 1,7 | 3 | 0,5 | 1,2 | 2,5 | 0,5 | 1 | 1,5 | 0,5 | 2,1 | 3 | 0,5 | 1,3 | 2,5 |
| Ch6 | 0,5 | 1,4 | 3 | 0,5 | 2,1 | 3 | 1,5 | 2,3 | 3,5 | 1,5 | 2,3 | 3 | 0,5 | 1 | 1,5 | 0,5 | 2,1 | 3 | 1,5 | 2,4 | 3,5 |
| Ch7 | 0,5 | 1,4 | 3 | 1,5 | 2,3 | 3 | 0,5 | 2,1 | 3,5 | 0,5 | 1,3 | 3 | 0,5 | 1 | 1,5 | 1,5 | 2,4 | 3,5 | 0,5 | 1 | 1,5 |
| Ch8 | 0,5 | 1,4 | 3 | 1 | 2 | 3 | 0,5 | 1,7 | 3 | 1,5 | 1,2 | 3 | 1,3 | 3 | 3,5 | 1,5 | 2,5 | 3,5 | 1 | 1,3 | 2 |
| Ch9 | 1,5 | 2,1 | 3 | 1 | 1,4 | 2 | 0,5 | 2,1 | 3,5 | 0,5 | 1,2 | 2,5 | 0,5 | 1,2 | 2,5 | 1,5 | 2,2 | 3,5 | 0,5 | 1 | 1,5 |
| Ch10 | 0,5 | 2 | 3 | 0,5 | 1,4 | 2,5 | 1,5 | 2,4 | 3,5 | 1,5 | 2,3 | 3 | 0,5 | 1 | 1,5 | 1,5 | 2,4 | 3,5 | 0,5 | 1 | 2 |
| Ch11 | 1,5 | 2,1 | 3 | 1 | 1,4 | 2 | 0,5 | 1,7 | 3 | 0,1 | 1,8 | 3 | 0,5 | 1,2 | 2,5 | 0,5 | 2,1 | 3 | 1 | 1,3 | 2 |
| Ch12 | 0,5 | 1 | 1,5 | 1 | 1,4 | 2 | 1,5 | 2,3 | 3,5 | 1,5 | 2,3 | 3 | 0,5 | 1,3 | 2 | 1,5 | 2,3 | 3 | 0,5 | 1 | 1,5 |
| Ch13 | 0,5 | 1 | 1,5 | 0,5 | 1,4 | 2,5 | 0,5 | 2,1 | 3,5 | 0,1 | 1,8 | 3 | 0,5 | 1 | 1,5 | 0,5 | 1 | 1,5 | 0,5 | 1,7 | 3 |
| Ch14 | 0,5 | 2 | 3 | 1 | 1,4 | 2 | 0,5 | 2,1 | 3,5 | 0,1 | 1,8 | 2 | 0,5 | 1 | 1,5 | 0,5 | 1,7 | 3 | 0,5 | 1,7 | 3 |
| Ch15 | 0,5 | 2 | 3 | 0,5 | 2,1 | 3 | 1,5 | 2,4 | 3,5 | 0,1 | 1,8 | 3 | 1,3 | 3 | 3,5 | 1,5 | 2,4 | 3,5 | 1 | 1,3 | 2 |
| Ch16 | 0,5 | 1,4 | 3 | 1,5 | 2,3 | 3 | 1,5 | 2,4 | 3,5 | 1 | 1,4 | 2 | 0,5 | 2 | 3 | 0,5 | 2,1 | 3 | 0,5 | 1,7 | 3 |
| Ch17 | 0,5 | 1,4 | 3 | 0,5 | 2,1 | 3 | 0,5 | 1,7 | 3 | 0,1 | 1,8 | 3 | 0,5 | 1 | 1,5 | 0,5 | 1,7 | 3 | 0,5 | 1 | 1,5 |
| Ch18 | 1,5 | 2,1 | 3 | 0,5 | 2,1 | 3 | 1,5 | 2,3 | 3,5 | 1 | 1,4 | 2 | 1,3 | 3 | 3,5 | 1,5 | 2,4 | 3,5 | 1 | 1,3 | 2 |
| Ch19 | 0,5 | 1 | 1,5 | 0,5 | 2,1 | 3 | 0,5 | 2,1 | 3,5 | 0,1 | 1,8 | 2 | 0,5 | 1,3 | 2 | 0,5 | 1,7 | 3 | 0,5 | 1 | 1,5 |
| Ch20 | 1,5 | 2,5 | 3 | 1,5 | 2,3 | 3 | 0,5 | 2,1 | 3,5 | 0,1 | 1,8 | 3 | 0,5 | 1 | 1,5 | 0,5 | 2,1 | 3 | 1 | 1,3 | 2 |
| Ch21 | 0,5 | 2 | 3 | 1 | 1,4 | 2 | 0,5 | 2,1 | 3,5 | 1 | 1,4 | 2 | 1,3 | 3 | 3,5 | 0,5 | 1,7 | 3 | 0,5 | 1 | 1,5 |
| Ch22 | 0,5 | 1,4 | 3 | 0,5 | 2,1 | 3 | 0,5 | 1,7 | 3 | 0,1 | 1,8 | 3 | 0,5 | 1 | 1,5 | 0,5 | 1,7 | 3 | 0,5 | 1 | 1,5 |

**Table 10** Normalized decision matrix

| Sr | Programming | | | Technical | | | Resourses | | | Standardization | | | Experties | | | Responsivness | | | Management | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weights | 0,5 | 1 | 2 | 1,2 | 2,5 | 3,5 | 0,5 | 1,4 | 2,5 | 1,2 | 2,5 | 3,5 | 0,5 | 1 | 1,5 | 0,5 | 1,4 | 2,5 | 0,5 | 1,2 | 2,5 |
| Ch1 | 0,17 | 0,25 | 1,00 | 0,43 | 0,69 | 1,00 | 0,43 | 0,69 | 1,00 | 0,33 | 0,40 | 0,83 | 0,14 | 0,29 | 0,43 | 0,43 | 0,66 | 0,86 | 0,14 | 0,37 | 0,71 |
| Ch2 | 0,17 | 0,31 | 1,00 | 0,14 | 0,60 | 0,86 | 0,43 | 0,57 | 0,86 | 0,17 | 0,43 | 1,00 | 0,14 | 0,40 | 0,86 | 0,43 | 0,66 | 0,86 | 0,29 | 0,37 | 0,57 |
| Ch3 | 0,17 | 0,25 | 1,00 | 0,29 | 0,34 | 0,71 | 0,14 | 0,49 | 0,86 | 0,50 | 0,40 | 1,00 | 0,14 | 0,34 | 0,71 | 0,43 | 0,69 | 1,00 | 0,29 | 0,37 | 0,57 |
| Ch4 | 0,17 | 0,20 | 0,33 | 0,14 | 0,60 | 0,86 | 0,43 | 0,66 | 1,00 | 0,17 | 0,40 | 0,83 | 0,14 | 0,34 | 0,71 | 0,14 | 0,60 | 0,86 | 0,29 | 0,37 | 0,57 |
| Ch5 | 0,17 | 0,25 | 1,00 | 0,14 | 0,60 | 0,86 | 0,14 | 0,49 | 0,86 | 0,17 | 0,40 | 0,83 | 0,14 | 0,29 | 0,43 | 0,14 | 0,60 | 0,86 | 0,14 | 0,37 | 0,71 |
| Ch6 | 0,17 | 0,36 | 1,00 | 0,14 | 0,60 | 0,86 | 0,43 | 0,66 | 1,00 | 0,50 | 0,77 | 1,00 | 0,14 | 0,29 | 0,43 | 0,14 | 0,60 | 0,86 | 0,43 | 0,69 | 1,00 |
| Ch7 | 0,17 | 0,36 | 1,00 | 0,43 | 0,66 | 0,86 | 0,14 | 0,60 | 1,00 | 0,17 | 0,43 | 1,00 | 0,14 | 0,29 | 0,43 | 0,43 | 0,69 | 1,00 | 0,14 | 0,29 | 0,43 |
| Ch8 | 0,17 | 0,36 | 1,00 | 0,29 | 0,57 | 0,86 | 0,14 | 0,49 | 0,86 | 0,50 | 0,40 | 1,00 | 0,37 | 0,86 | 1,00 | 0,43 | 0,71 | 1,00 | 0,29 | 0,37 | 0,57 |
| Ch9 | 0,17 | 0,24 | 0,33 | 0,29 | 0,40 | 0,57 | 0,14 | 0,60 | 1,00 | 0,17 | 0,40 | 0,83 | 0,14 | 0,34 | 0,71 | 0,43 | 0,63 | 1,00 | 0,14 | 0,29 | 0,43 |
| Ch10 | 0,17 | 0,25 | 1,00 | 0,14 | 0,40 | 0,71 | 0,43 | 0,69 | 1,00 | 0,50 | 0,77 | 1,00 | 0,14 | 0,29 | 0,43 | 0,43 | 0,69 | 1,00 | 0,14 | 0,29 | 0,57 |
| Ch11 | 0,17 | 0,24 | 0,33 | 0,29 | 0,40 | 0,57 | 0,14 | 0,49 | 0,86 | 0,03 | 0,60 | 1,00 | 0,14 | 0,34 | 0,71 | 0,14 | 0,60 | 0,86 | 0,29 | 0,37 | 0,57 |
| Ch12 | 0,33 | 0,50 | 1,00 | 0,29 | 0,40 | 0,57 | 0,43 | 0,66 | 1,00 | 0,50 | 0,77 | 1,00 | 0,14 | 0,37 | 0,57 | 0,43 | 0,66 | 0,86 | 0,14 | 0,29 | 0,43 |
| Ch13 | 0,33 | 0,50 | 1,00 | 0,14 | 0,40 | 0,71 | 0,14 | 0,60 | 1,00 | 0,03 | 0,60 | 1,00 | 0,14 | 0,29 | 0,43 | 0,14 | 0,29 | 0,43 | 0,14 | 0,49 | 0,86 |
| Ch14 | 0,17 | 0,25 | 1,00 | 0,29 | 0,40 | 0,57 | 0,14 | 0,60 | 1,00 | 0,03 | 0,60 | 0,67 | 0,14 | 0,29 | 0,43 | 0,14 | 0,49 | 0,86 | 0,14 | 0,49 | 0,86 |
| Ch15 | 0,17 | 0,25 | 1,00 | 0,14 | 0,60 | 0,86 | 0,43 | 0,69 | 1,00 | 0,03 | 0,60 | 1,00 | 0,37 | 0,86 | 1,00 | 0,43 | 0,69 | 1,00 | 0,29 | 0,37 | 0,57 |
| Ch16 | 0,17 | 0,36 | 1,00 | 0,43 | 0,66 | 0,86 | 0,43 | 0,69 | 1,00 | 0,33 | 0,47 | 0,67 | 0,14 | 0,57 | 0,86 | 0,14 | 0,60 | 0,86 | 0,14 | 0,49 | 0,86 |
| Ch17 | 0,17 | 0,36 | 1,00 | 0,14 | 0,60 | 0,86 | 0,14 | 0,49 | 0,86 | 0,03 | 0,60 | 1,00 | 0,14 | 0,29 | 0,43 | 0,14 | 0,49 | 0,86 | 0,14 | 0,29 | 0,43 |
| Ch18 | 0,17 | 0,24 | 0,33 | 0,14 | 0,60 | 0,86 | 0,43 | 0,66 | 1,00 | 0,33 | 0,47 | 0,67 | 0,37 | 0,86 | 1,00 | 0,43 | 0,69 | 1,00 | 0,29 | 0,37 | 0,57 |
| Ch19 | 0,33 | 0,50 | 1,00 | 0,14 | 0,60 | 0,86 | 0,14 | 0,60 | 1,00 | 0,03 | 0,60 | 0,67 | 0,14 | 0,37 | 0,57 | 0,14 | 0,49 | 0,86 | 0,14 | 0,29 | 0,43 |
| Ch20 | 0,17 | 0,20 | 0,33 | 0,43 | 0,66 | 0,86 | 0,14 | 0,60 | 1,00 | 0,03 | 0,60 | 1,00 | 0,14 | 0,29 | 0,43 | 0,14 | 0,60 | 0,86 | 0,29 | 0,37 | 0,57 |
| Ch21 | 0,17 | 0,25 | 1,00 | 0,29 | 0,40 | 0,57 | 0,14 | 0,60 | 1,00 | 0,33 | 0,47 | 0,67 | 0,37 | 0,86 | 1,00 | 0,14 | 0,49 | 0,86 | 0,14 | 0,29 | 0,43 |
| Ch22 | 0,17 | 0,36 | 1,00 | 0,14 | 0,60 | 0,86 | 0,14 | 0,49 | 0,86 | 0,03 | 0,60 | 1,00 | 0,14 | 0,29 | 0,43 | 0,14 | 0,49 | 0,86 | 0,14 | 0,29 | 0,43 |

**Table 11** Weighted normalized decision matrix

| Sr | Programming | | | Technical | | | Resourses | | | Standardization | | | Experties | | | Responsivness | | | Management | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weights | 0,5 | 1 | 2 | 1,2 | 2,5 | 3,5 | 0,5 | 1,4 | 2,5 | 1,2 | 2,5 | 3,5 | 0,5 | 1 | 1,5 | 0,5 | 1,4 | 2,5 | 0,5 | 1,2 | 2,5 |
| Ch1 | 0,08 | 0,25 | 2,00 | 0,51 | 1,71 | 3,5 | 0,21 | 0,96 | 2,50 | 0,4 | 1,00 | 2,92 | 0,07 | 0,29 | 0,64 | 0,21 | 0,92 | 7,50 | 0,07 | 0,45 | 1,79 |
| Ch2 | 0,08 | 0,31 | 2,00 | 0,17 | 1,50 | 3 | 0,21 | 0,8 | 2,14 | 0,2 | 1,08 | 3,50 | 0,07 | 0,40 | 1,29 | 0,21 | 0,92 | 0,00 | 0,14 | 0,45 | 1,43 |
| Ch3 | 0,08 | 0,25 | 2,00 | 0,34 | 0,86 | 2,5 | 0,07 | 0,68 | 2,14 | 0,6 | 1,00 | 3,50 | 0,07 | 0,34 | 1,07 | 0,21 | 0,96 | 0,00 | 0,14 | 0,45 | 1,43 |
| Ch4 | 0,08 | 0,20 | 0,67 | 0,17 | 1,50 | 3 | 0,21 | 0,92 | 2,50 | 0,2 | 1,00 | 2,92 | 0,07 | 0,34 | 1,07 | 0,07 | 0,84 | 0,00 | 0,14 | 0,45 | 1,43 |
| Ch5 | 0,08 | 0,25 | 2,00 | 0,17 | 1,50 | 3 | 0,07 | 0,68 | 2,14 | 0,2 | 1,00 | 2,92 | 0,07 | 0,29 | 0,64 | 0,07 | 0,84 | 6,25 | 0,07 | 0,45 | 1,79 |
| Ch6 | 0,08 | 0,36 | 2,00 | 0,17 | 1,50 | 3 | 0,21 | 0,92 | 2,50 | 0,6 | 1,92 | 3,50 | 0,07 | 0,29 | 0,64 | 0,07 | 0,84 | 2,14 | 0,21 | 0,82 | 2,50 |
| Ch7 | 0,08 | 0,36 | 2,00 | 0,51 | 1,64 | 3 | 0,07 | 0,84 | 2,50 | 0,2 | 1,08 | 3,50 | 0,07 | 0,29 | 0,64 | 0,21 | 0,96 | 2,14 | 0,07 | 0,34 | 1,07 |
| Ch8 | 0,08 | 0,36 | 2,00 | 0,34 | 1,43 | 3 | 0,07 | 0,68 | 2,14 | 0,6 | 1,00 | 3,50 | 0,19 | 0,86 | 1,50 | 0,21 | 1 | 2,50 | 0,14 | 0,45 | 1,43 |
| Ch9 | 0,08 | 0,24 | 0,67 | 0,34 | 1,00 | 2 | 0,07 | 0,84 | 2,50 | 0,2 | 1,00 | 2,92 | 0,07 | 0,34 | 1,07 | 0,21 | 0,88 | 2,14 | 0,07 | 0,34 | 1,07 |
| Ch10 | 0,08 | 0,25 | 2,00 | 0,17 | 1,00 | 2,5 | 0,21 | 0,96 | 2,50 | 0,6 | 1,92 | 3,50 | 0,07 | 0,29 | 0,64 | 0,21 | 0,96 | 2,14 | 0,07 | 0,34 | 1,43 |
| Ch11 | 0,08 | 0,24 | 0,67 | 0,34 | 1,00 | 2 | 0,07 | 0,68 | 2,14 | 0,04 | 1,50 | 3,50 | 0,07 | 0,34 | 1,07 | 0,07 | 0,84 | 2,14 | 0,14 | 0,45 | 1,43 |
| Ch12 | 0,17 | 0,50 | 2,00 | 0,34 | 1,00 | 2 | 0,21 | 0,92 | 2,50 | 0,6 | 1,92 | 3,50 | 0,07 | 0,37 | 0,86 | 0,21 | 0,92 | 2,50 | 0,07 | 0,34 | 1,07 |
| Ch13 | 0,17 | 0,50 | 2,00 | 0,17 | 1,00 | 2,5 | 0,07 | 0,84 | 2,50 | 0,04 | 1,50 | 3,50 | 0,07 | 0,29 | 0,64 | 0,07 | 0,4 | 2,50 | 0,07 | 0,58 | 2,14 |
| Ch14 | 0,08 | 0,25 | 2,00 | 0,34 | 1,00 | 2 | 0,07 | 0,84 | 2,50 | 0,04 | 1,50 | 2,33 | 0,07 | 0,29 | 0,64 | 0,07 | 0,68 | 2,50 | 0,07 | 0,58 | 2,14 |
| Ch15 | 0,08 | 0,25 | 2,00 | 0,17 | 1,50 | 3 | 0,21 | 0,96 | 2,50 | 0,04 | 1,50 | 3,50 | 0,19 | 0,86 | 1,50 | 0,21 | 0,96 | 2,50 | 0,14 | 0,45 | 1,43 |
| Ch16 | 0,08 | 0,36 | 2,00 | 0,51 | 1,64 | 3 | 0,21 | 0,96 | 2,50 | 0,4 | 1,17 | 2,33 | 0,07 | 0,57 | 1,29 | 0,07 | 0,84 | 2,14 | 0,07 | 0,58 | 2,14 |
| Ch17 | 0,08 | 0,36 | 2,00 | 0,17 | 1,50 | 3 | 0,07 | 0,68 | 2,14 | 0,04 | 1,50 | 3,50 | 0,07 | 0,29 | 0,64 | 0,07 | 0,68 | 2,14 | 0,07 | 0,34 | 1,07 |
| Ch18 | 0,08 | 0,24 | 0,67 | 0,17 | 1,50 | 3 | 0,21 | 0,92 | 2,50 | 0,4 | 1,17 | 2,33 | 0,19 | 0,86 | 1,50 | 0,21 | 0,96 | 1,07 | 0,14 | 0,45 | 1,43 |
| Ch19 | 0,17 | 0,50 | 2,00 | 0,17 | 1,50 | 3 | 0,07 | 0,84 | 2,50 | 0,04 | 1,50 | 2,33 | 0,07 | 0,37 | 0,86 | 0,07 | 0,68 | 2,14 | 0,07 | 0,34 | 1,07 |
| Ch20 | 0,08 | 0,20 | 0,67 | 0,51 | 1,64 | 3 | 0,07 | 0,84 | 2,50 | 0,04 | 1,50 | 3,50 | 0,07 | 0,29 | 0,64 | 0,07 | 0,84 | 2,50 | 0,14 | 0,45 | 1,43 |
| Ch21 | 0,08 | 0,25 | 2,00 | 0,34 | 1,00 | 2 | 0,07 | 0,84 | 2,50 | 0,4 | 1,17 | 2,33 | 0,19 | 0,86 | 1,50 | 0,07 | 0,68 | 2,14 | 0,07 | 0,34 | 1,07 |
| Ch22 | 0,08 | 0,36 | 2,00 | 0,17 | 1,50 | 3 | 0,07 | 0,68 | 2,14 | 0,04 | 1,50 | 3,50 | 0,07 | 0,29 | 0,64 | 0,07 | 0,68 | 2,14 | 0,07 | 0,34 | 1,07 |
| A + | 0,17 | 0,50 | 2,00 | 0,51 | 1,71 | 3,50 | 0,21 | 0,96 | 2,50 | 0,60 | 1,92 | 3,50 | 0,19 | 0,86 | 1,50 | 0,21 | 1,00 | 7,50 | 0,21 | 0,82 | 2,50 |
| A- | 0,08 | 0,20 | 0,67 | 0,17 | 0,86 | 2,00 | 0,07 | 0,68 | 2,14 | 0,04 | 1,00 | 2,33 | 0,07 | 0,29 | 0,64 | 0,07 | 0,40 | 0,00 | 0,07 | 0,34 | 1,07 |

optimize quantum software." By normalizing the decision matrix, we were able to assign equal weight to each criterion and rank the challenging factors based on their overall performance. The results of the normalized decision matrix are presented in Table 10.

**Step 5:** In this step, we computed the weighted decision matrix by multiplying the weights assigned by the group of experts for each criterion with their respective alternative (i.e., challenging factors), using Eq. (4). The resulting weighted decision matrix is presented in Table 11.

**Step 6:** In this step, we executed Eqs. (5) and (6) and calculated the fuzzy ideal solutions (positive and negative). The fuzzy positive and negative solutions are given in Tables 12 and 13.

**Step 7:** In this step, we determined the distance Di* and Di- for each QSE challenging factor for the criteria (Eq. 7, and 8), and the results are given in Tables 13 and 14.

**Step 8:** Using Eq. (9), the closeness coefficient was calculated; the results are given in Table 14.

**Step 9:** The ranks for each challenging factor were determined based on their closeness coefficient (CCi) values, with higher CCi values indicating a higher ranking (Table 14). The priority ranks were determined to assess the significance of the challenging factors within their specified category (local ranks) and in comparison, to all 22 identified challenging factors (global ranks), as presented in Table 14.

For example, Ch1 (Complex programming, with a CCi value of 0.95) was identified as the most significant challenging factor within its category (C1, programming) and ranked highest in the global ranks. Additionally, Ch5 (Limited software libraries, with a CCi value of 0.89) was ranked as the 1st priority challenging factor within its category (C2, Technical) and 2nd in the global ranks. The local and global ranks for all factors were determined using the same process, and the results are presented in Table 14.

## 6 Systematic decision-making framework of QSE challenging factors

Finally, we developed a systematic decision-making framework of the identified QSE challenging factors using the ISM and fuzzy TOPSIS approaches. The developed framework (Fig. 5) indicates the global ranking (GR) and local ranking (LR) of each of the identified challenging factors. The local ranks were calculated to check the priority of a challenging factor within its specific core category and the global ranking (GR) presents the priority order of challenging factors for overall QSE process.

According to the developed framework, the C3 (Resources) category is entirely independent, as it stands out at level four in the ISM analysis. The MICMAC analysis showed that C3 belongs to an autonomous cluster, implying that resources have a weak dependency on other core QSE categories. This indicates that resources may have a more self-sufficient role in addressing QSE challenges. According to Heng et al. (Li et al. 2021), prioritizing the challenges of the resources category could

**Table 12** Fuzzy positive ideal solution

| Sr | Program-ming | Technical | Resources | Standardiza-tion | Experties | Responsive-ness | Manage-ment | DI* |
|---|---|---|---|---|---|---|---|---|
| Ch1 | 0.15 | 0.00 | 0.00 | 0.64 | 0.60 | 0.05 | 0.47 | 1.91 |
| Ch2 | 0.12 | 0.37 | 0,23 | 0.53 | 0.30 | 4.33 | 0.66 | 6.54 |
| Ch3 | 0.15 | 0.77 | 0.27 | 0.53 | 0.39 | 4.33 | 0.66 | 7.10 |
| Ch4 | 0.79 | 0.37 | 0.02 | 0.67 | 0.39 | 4.33 | 0.66 | 7.23 |
| Ch5 | 0.15 | 0.37 | 0.27 | 0.67 | 0.60 | 0.73 | 0.47 | 3.27 |
| Ch6 | 0.10 | 0.37 | 0.02 | 0.00 | 0.60 | 3.10 | 0.00 | 4.18 |
| Ch7 | 0.10 | 0.29 | 0.11 | 0.53 | 0.60 | 3.09 | 0.87 | 5.59 |
| Ch8 | 0.10 | 0.35 | 0.27 | 0.53 | 0.00 | 2.89 | 0.66 | 4.79 |
| Ch9 | 0.79 | 0.96 | 0.11 | 0.67 | 0.39 | 3.09 | 0.87 | 6.89 |
| Ch10 | 0.15 | 0.74 | 0.00 | 0.00 | 0.60 | 3.09 | 0.68 | 5.26 |
| Ch11 | 0.79 | 0.96 | 0.27 | 0.40 | 0.39 | 3.10 | 0.66 | 6.57 |
| Ch12 | 0.00 | 0.96 | 0.02 | 0.00 | 0.47 | 2.89 | 0.87 | 5.22 |
| Ch13 | 0.00 | 0.74 | 0.11 | 0.40 | 0.60 | 2.91 | 0.26 | 5.02 |
| Ch14 | 0.15 | 0.96 | 0.11 | 0.78 | 0.60 | 2.89 | 0.26 | 5.76 |
| Ch15 | 0.15 | 0.37 | 0.00 | 0.40 | 0.00 | 2.89 | 0.66 | 4.47 |
| Ch16 | 0.10 | 0.29 | 0.00 | 0.81 | 0.22 | 3.10 | 0.26 | 4.77 |
| Ch17 | 0.10 | 0.37 | 0.27 | 0.40 | 0.60 | 3.10 | 0.87 | 5.72 |
| Ch18 | 0.79 | 0.37 | 0.02 | 0.81 | 0.00 | 3.71 | 0.66 | 6.36 |
| Ch19 | 0.00 | 0.37 | 0.11 | 0.78 | 0.47 | 3.10 | 0.87 | 5.71 |
| Ch20 | 0.79 | 0.29 | 0.11 | 0.40 | 0.60 | 2.89 | 0.66 | 5.74 |
| Ch21 | 0.15 | 0.96 | 0.11 | 0.81 | 0.00 | 3.10 | 0.87 | 6.01 |
| Ch22 | 0.10 | 0.37 | 0.27 | 0.40 | 0.60 | 3.10 | 0.87 | 5.72 |

be a more straightforward approach to QSE management. The prioritization-based ranking of C3 challenges shows that Ch8 (Data encoding issues), Ch7 (Complexity of quantum algorithms), and Ch9 (Difficulty optimizing quantum software) rank 5th, 8th, and 17th globally, suggesting that C3 (Resources) is independent but its related challenging factors are not ranked as the top significant challenging factors for QSE process. The results reveal that the C1 (Programming) category depends on C3 and C7, yet its challenge Ch1 (Complex programming) ranks as the most significant challenging factor locally and globally. In summary, the ISM-based leveling of core categories does not directly influence the priority of related challenging factors; instead, it indicates the dependency, interdependency, driving force, and dependence power of each category on other categories.

As depicted in Fig. 5, C7 (Management) is only dependent on C3 (Resources), while all other QSE challenging factors categories rely on it. Furthermore, C2 (Technical) is a fully dependent category, situated at level 1, indicating that all other QSE challenging factors categories can significantly impact the implementation of C2 challenging factors. Despite its complete dependency, Ch5 (Limited software

**Table 13** Fuzzy negative ideal solution

| Sr | Programming | Technical | Resources | Standardization | Experties | Responsiveness | Management | DI- |
|---|---|---|---|---|---|---|---|---|
| Ch1 | 1.03 | 1.52 | 1.36 | 0.15 | 0.20 | 3.64 | 0.30 | 3.19 |
| Ch2 | 1.03 | 1.26 | 0.60 | 0.02 | 0.79 | 0.17 | 0.08 | 3.96 |
| Ch3 | 1.03 | 1.04 | 0.14 | 0.18 | 0.79 | 0.19 | 0.08 | 3.46 |
| Ch4 | 0.00 | 0.24 | 0.62 | 0.09 | 0.20 | 0.11 | 0.08 | 1.34 |
| Ch5 | 1.03 | 1.26 | 0.58 | 0.01 | 0.20 | 2.66 | 0.30 | 2.05 |
| Ch6 | 1.04 | 1.26 | 0.62 | 0.74 | 0.79 | 2.76 | 1.32 | 8.54 |
| Ch7 | 1.04 | 1.45 | 0.59 | 0.09 | 0.79 | 2.84 | 0.00 | 6.81 |
| Ch8 | 1.04 | 1.23 | 0.58 | 0.18 | 0.98 | 3.83 | 0.08 | 7.92 |
| Ch9 | 0.00 | 0.03 | 0.01 | 0.09 | 0.20 | 2.80 | 0.00 | 3.13 |
| Ch10 | 1.03 | 1.04 | 0.20 | 0.74 | 0.79 | 2.84 | 0.07 | 6.71 |
| Ch11 | 0.00 | 0.03 | 0.00 | 0.14 | 0.79 | 2.76 | 0.08 | 3.81 |
| Ch12 | 1.08 | 1.06 | 0.05 | 0.74 | 0.79 | 3.78 | 0.00 | 7.49 |
| Ch13 | 1.08 | 1.04 | 0.16 | 0.22 | 0.79 | 3.61 | 0.70 | 7.59 |
| Ch14 | 1.03 | 1.06 | 0.01 | 0.22 | 0.00 | 3.65 | 0.70 | 6.67 |
| Ch15 | 1,.03 | 1.26 | 0.63 | 0.22 | 0.98 | 3.80 | 0.08 | 8.01 |
| Ch16 | 1.04 | 1.45 | 0.63 | 0.16 | 0.05 | 2.76 | 0.70 | 6.80 |
| Ch17 | 1.04 | 1.26 | 0.58 | 0.14 | 0.79 | 2.70 | 0.00 | 6.51 |
| Ch18 | 0.00 | 0.24 | 0.62 | 0.16 | 0.20 | 0.86 | 0.08 | 2.16 |
| Ch19 | 1.08 | 1.26 | 0.59 | 0.22 | 0.00 | 2.70 | 0.00 | 5.86 |
| Ch20 | 0.00 | 0.42 | 0.59 | 0.22 | 0.79 | 3.72 | 0.08 | 5.82 |
| Ch21 | 1.03 | 1.06 | 0.01 | 0.16 | 0.20 | 2.70 | 0.00 | 5.15 |
| Ch22 | 1.04 | 1.26 | 0.58 | 0.14 | 0.79 | 2.70 | 0.00 | 6.51 |

libraries) is identified as the second most important challenging factor for implementing the QSE process.

The development framework (Fig. 5) highlights that Ch1 (Complex programming) is the top-ranked challenging factor within its category and for the overall QSE process. Similarly,—Bettina et al. (Heim et al. 2020) state that quantum software programming complexity arises from the need to understand quantum mechanics principles like superposition and entanglement, which differ significantly from classical computing. Higher error rates in quantum computing necessitate robust error correction methods. Developing efficient quantum algorithms is challenging, as they often require inventive approaches to leverage quantum properties. The lack of mature quantum programming languages and libraries further complicates the process. Alexander et al. (McCaskey et al. 2018) also emphasize that the rapid evolution of quantum hardware demands continuous adaptation and optimization to fully harness its potential.

Ch5 (Limited software libraries) is ranked as 1st within their category and 2nd most important challenging for overall QSE process. This is also evident from the literature as indicted by Maria et al. (Belkhir et al. 2022) as the limited

**Table 14** Closeness coefficient values and ranks

| Categories | Sr | Challenging factors | CCi | Local ranks | Global ranks |
|---|---|---|---|---|---|
| Programming (C1) | Ch1 | Complex programming | 0.95 | 1 | 1 |
| | Ch2 | Limited simulation resources | 0.38 | 2 | 14 |
| | Ch3 | High error rates | 0.33 | 3 | 16 |
| Technical (C2) | Ch4 | Lack of debugging tools | 0.16 | 3 | 19 |
| | Ch5 | Limited software libraries | 0.89 | 1 | 2 |
| | Ch6 | Maintenance complexity | 0.67 | 2 | 3 |
| Resources (C3) | Ch7 | Complexity of quantum algorithms | 0.55 | 2 | 8 |
| | Ch8 | Data encoding issues | 0.62 | 1 | 5 |
| | Ch9 | Difficult to optimize quantum software | 0.31 | 3 | 17 |
| Standardization (C4) | Ch10 | Lack of standardization | 0.56 | 2 | 7 |
| | Ch11 | Lack of interoperability | 0.37 | 3 | 15 |
| | Ch12 | Lack of requirements engineering strategies | 0.59 | 1 | 7 |
| Experties (C5) | Ch13 | Lack of expertise | 0.60 | 2 | 6 |
| | Ch14 | Integration with classical computing | 0.54 | 3 | 9 |
| | Ch15 | Lack of training and workshops | 0.64 | 1 | 4 |
| Responsiveness (C6) | Ch16 | Security issue in QSE | 0.59 | 1 | 7 |
| | Ch17 | Ethical issue in QSE | 0.53 | 2 | 10 |
| Management (C7) | Ch18 | Verification and validation issues | 0.25 | 4 | 18 |
| | Ch19 | QSE scalability issues | 0.51 | 1 | 11 |
| | Ch20 | Budget constraints | 0.50 | 2 | 12 |
| | Ch21 | Lack of commercial applications | 0.46 | 3 | 13 |
| | Ch22 | Project management issues | 0.16 | 5 | 19 |

software libraries in quantum computing hinder the availability of pre-built tools and resources for developers. This scarcity forces quantum software engineers to create custom solutions, which can be both time-consuming and error prone. Stefano et al. (Stefano et al. 2022) also underlined the lack of standardized libraries hampers collaboration and interoperability between different quantum platforms. As a result, the overall progress and adoption of quantum software engineering face significant obstacles.

Ch6 (Maintenance complexity) within C2 is ranked as the 2nd most significant challenge locally, and it holds the position of the 3rd most significant challenging factor in the global ranking. According to Zhao et al. (Zhao 2007) maintenance is complex in quantum software engineering due to the rapidly evolving landscape of quantum hardware, necessitating frequent updates and optimizations. Quantum error rates and susceptibility to environmental noise require ongoing improvements in error correction techniques (Azeem Akbar et al. 2022). The scarcity of experienced quantum software engineers contributes to the challenge of maintaining and enhancing quantum applications. Additionally, as new quantum algorithms emerge,

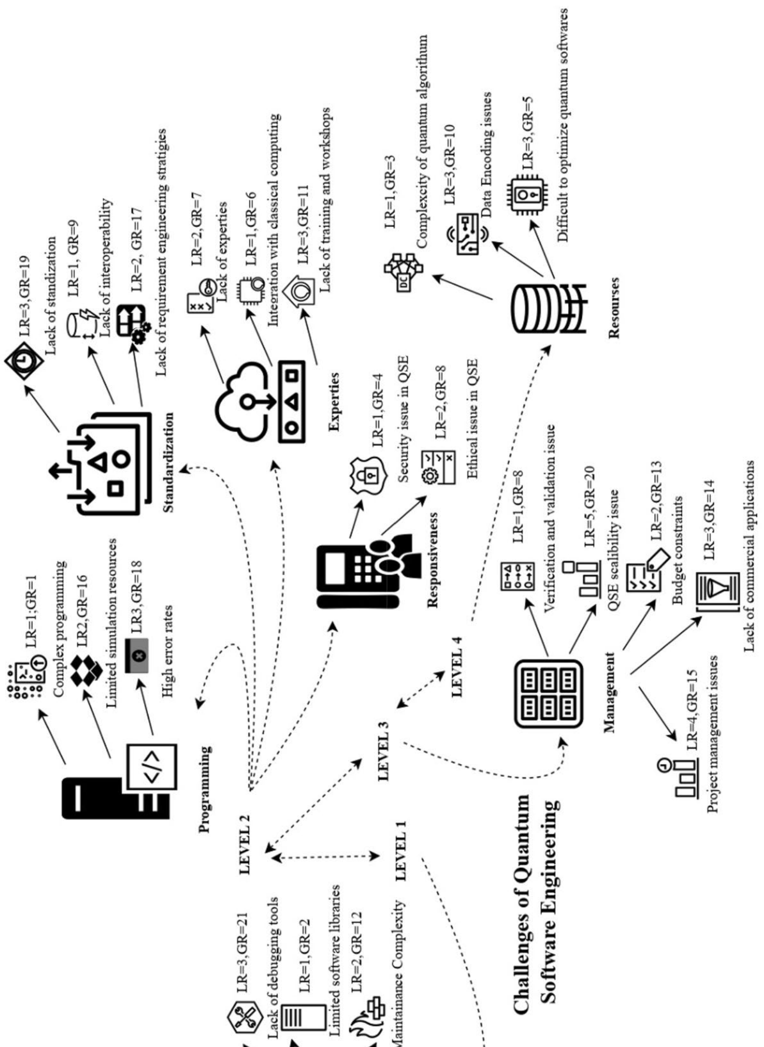


**Fig. 5** Systematic decision-making framework of QSE challenging factors

integrating them into existing systems can be intricate. Lastly, the lack of standardized practices and protocols in this nascent field complicates the maintenance process and hinders long-term stability (Pérez-Castillo et al. 2021).

Moreover, Ch15 (Lack of training and workshops), and Ch 8 (Data encoding issues) are ranked as 4th and 5th most important challenging factors for QSE process execution. Lastly, Ch4 (Lack of debugging tools) within C2 (Technical) and Ch22 (Project management issues) in C7 (Management) both rank at 19th and declared as the least significant challenging factors for successful QSE process implementation.

The proposed systematic decision-making framework assists QSE practitioners in revising or developing new, effective strategies for successful QSE process execution by taking into account the dependencies and driving forces of the core categories of QSE challenges. Additionally, practitioners should consider the critical levels of QSE challenges, as presented in Fig. 5. Researchers should also focus on the most important challenging factors of the QSE process in their future studies.

## 7 Study implications and threats to validity

Study implications and threats to validity are two essential aspects of this research study. The implications of a study are its potential real-world applications and consequences. Threats to validity, on the other hand, are expected limitation or bias that can undermine the accuracy and reliability of research findings. The potential implication of our research work and threats to validity are given in Sect. 6.1 and 6.2 respectively.

### 7.1 Implications

***For researchers***: This study offers a comprehensive overview of the challenging factors that are critical to the successful adoption and execution of QSE process. By conducting a thorough review of existing literature, this study identifies the 22 challenging factors that can negatively impact the implementation, execution, and improvement of QSE processes. This list of identified challenging factors serve as a knowledge base to provide valuable insights to researchers to inform their future strategies. Moreover, this study develops a systematic decision-making framework to assess the criticality of the identified challenging factors to the adoption of QSE processes. This framework can serve as a valuable guide to researchers in selecting the most critical challenges that need to be addressed first.

Additionally, this study examines the interrelationships and dependencies among the core categories of the identified challenging factors. By doing so, it provides a more nuanced understanding of how these factors can impact the QSE process.

In summary, the systematic decision-making framework developed in this study can provide researchers with useful guidelines to improve the adoption and implementation of QSE processes in their future work. This study's findings can also enhance researcher's understanding of the factors that influence QSE process adoption, and this will facilitate the researcher to the development of more effective strategies and roadmap for the success and progression of QSE process.

***For practitioners:*** The results of our study can be valuable for practitioners in various ways. Firstly, practitioners can use the list of identified challenging factors as a knowledge base to guide their QSE process. This can help to keep focusing on the most critical areas to make their QSE process effective.

Secondly, organizations can use the identified list of challenging factors to improve their project management capabilities by developing training opportunities targeted at areas where further skill development is needed. This can enhance the overall expertise of their team, leading to more successful QSE process.

Thirdly, practitioners can benefit from focusing on the highest priority challenging factors of each category, as this can help them better plan for the improvements of QSE process. This approach can lead to better outcomes and more efficient project management.

Organizations can also use the list of challenging factors as a basis for hiring software engineers with specific skills, as a risk mitigation strategy for QSE projects. This can help ensure that the right people with the right skills are involved in the project, addressing the critical challenging areas, and improving the chances of success.

The study findings can also be used to measure an organization's weak point to the adoption of QSE process. The developed systematic decision-making framework provides software practitioners with the ability to understand their current QSE related weaknesses within their organization.

Ultimately, the developed systematic decision-making framework software development organizations in a better position to adopt QSE process by preliminary addressing the important areas of QSE and this will lead to better outcomes and more efficient project management.

### 7.2 Threats to validity

There are several potential threats to the validity of this study. One of these threats is the study is the data extraction process from the existing literature and may have resulted in some relevant process areas being missed using informal literature review approach. However, it should be noted that the same literature review approach has been used in other studies for factor identification and classification and is not a systematic omission (Khan et al. 2019; Shameem et al. 2018). In future, we will consider a more systematic approach to ensure all relevant process areas are captured.

One possible threat is the limited time and resources available to conduct the empirical study with blockchain based software development experts, which may affect the validity of the reported process areas and their mapping. Additionally, the sample size of the survey questionnaire ($n = 133$) may not be sufficient to fully support the validity of the reported process areas, though it is representative data sample based on existing studies (Khan et al. 2019; Shameem et al. 2018; Akbar et al. 2019).

Another possible threat to this study findings is the potential differences in participants' understanding of the survey instruments. To address this threat, we followed Kitchenham and Pfleeger's guidelines for conducting surveys (Kitchenham et al. 2002) and piloted instrument to ensure their understandability. While our survey questionnaire was in English, we encountered some participants during the data collection process who found it challenging to convey their answers in English.

Another potential limitation of this study is the predominantly of Asian respondent pool, but data from other continents was also collected to increase representativeness. Similarly, a potential threat to the validity of this study is related to the survey conducted with QSE practitioners, which could be subject to response bias and may not accurately represent the actual population distribution. However, the study used a snowball sampling approach to locate experienced practitioners using web-based survey instruments. While it was not feasible to directly verify participants' experience, the study attempted to mitigate this limitation by targeting practitioners with at least two years of experience in QSE practices. Nonetheless, the participants' opinions may still be subject to inaccuracies regarding project outcomes. Future studies could consider alternative methods for verifying participants' experience and mitigating response bias.

Lastly, one of the potential threats of conclusion validity in this study is that the conclusions may be based solely on the understanding and experience of a single author, and conflicts between authors regarding the conclusions may not have been fully discussed or resolved. To address this threat, we took a collaborative approach. The first author extracted and analyzed the study data from the survey, and all other authors reviewed the data thoroughly through multiple meetings. Any conflicts on data analysis results were resolved through mutual discussions and brainstorming among all authors.

## 8 Conclusion and future work

Quantum software engineering (QSE) is a rapidly evolving field that involves the development, design, and implementation of software tailored for quantum computing systems. QSE is more challenging than classical software engineering due to the unique features of quantum computing and the lack of mature quantum development tools. In this study, we identify the 22 critical challenging factors of QSE by conducting a literature survey and mapped them into 7 core categories. Using the ISM approach, we obtained practitioners' views on the interrelationships between these core categories and found that the 'Resources' category has the highest driving power, while the 'Technical' category is fully dependent on the other six categories and has no drawing power. Finally, using the fuzzy-TOPSIS approach, we determined that 'complex programming,' 'limited software libraries,' 'maintenance complexity,' 'lack of training and workshops,' and 'data encoding issues' are the most critical challenging factors for the QSE process execution.

In the future, we plan to extend this study by empirically exploring additional challenging factors with QSE practitioners. Additionally, we aim to conduct a comprehensive study to identify success factors, barriers, and best practices for each phase of the software development process from requirement engineering to implementation. Ultimately, we aim to develop a readiness model to help software development organizations assess their readiness for adopting and improving the QSE process.

## Appendix 1

Sample of Used Questionnaire for Fuzzy-TOPSIS (https://tinyurl.com/3xxy9w2v).

**Author's contribution** Methodology, formal analysis and project administration done by MAA; data curation done by SR, and final revision and English polishing done by AAK. All authors have read and agreed to the published version of the manuscript.

**Funding** Open Access funding provided by University of Oulu including Oulu University Hospital.

**Data availability** The codes and data are available under request from the authors.

### Declarations

**Conflict of interest** The authors declare no competing interests.



## References

Agarwal P. and Alam M., "Exploring quantum computing to revolutionize big data analytics for various industrial sectors," in Big Data Analytics, ed: Auerbach Publications, 2021, pp. 113–130.

Agarwal, A., Vrat, P.: Modeling attributes of human body organization using ISM and AHP. Jindal J. Bus. Res. **6**, 44–62 (2017)

Akbar M. A., Rafi S., and Khan A. A. J. A. P. A., "Classical to quantum software migration journey begins: a conceptual readiness model," 2022.

Akbar, M.A., Khan, A.A., Khan, A.W., Mahmood, S.: Process Requirement change management challenges in GSD: An analytical hierarchy process approach. J. Softw. Evol. Proc. **32**, e2246 (2020a)

Akbar, M.A., Khan, A.A., Mahmood, S., Alsanad, A., Gumaei, A.: A robust framework for cloud-based software development outsourcing factors using analytical hierarchy process. J. Softw. Evolut. Process **33**, e2275 (2021b)

Akbar, M.A., Khan, A.A., Mahmood, S., Mishra, A.: SRCMIMM: the software requirements change management and implementation maturity model in the domain of global software development industry. Inform. Technol. Manag. (2022c). https://doi.org/10.1007/s10799-022-00364-ws

Akbar, M.A., Naveed, W., Alsanad, A.A., Alsuwaidan, L., Alsanad, A., Gumaei, A., et al.: Requirements change management challenges of global software development: an empirical investigation. IEEE Access. **8**, 203070–203085 (2020b)

Akbar, M.A., Sang, J., Khan, A.A., Amin, F.-E., Hussain, S., Sohail, M.K., et al.: Statistical analysis of the effects of heavyweight and lightweight methodologies on the six-pointed star model. IEEE Access **6**, 8066–8079 (2018)

Akbar, M.A., Sang, J., Khan, A.A., Mahmood, S., Qadri, S.F., Hu, H., et al.: Success factors influencing requirements change management process in global software development. J. Comput. Lang. **51**, 112–130 (2019)

Akbar, M.A., Shameem, M., Khan, A.A., Nadeem, M., Alsanad, A., Gumaei, A.: A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development. J. Softw. Evolut. Process **33**, e2292 (2021a)

Akbar, M.A., Smolander, K., Mahmood, S., Alsanad, A.: Toward successful DevSecOps in software development organizations: a decision-making framework. Inf. Softw. Technol. **147**, 106894 (2022b)

Aksu, M., Uluçay, O.: An integrated fuzzy AHP-TOPSIS approach for agile assessment in software development teams. Appl. Soft Comput. **201**, 103944 (2021)

Alderman, A.K., Salem, B.: Survey research. Plast. Reconstr. Surg. **126**, 1381–1389 (2010)

Ali S. and Yue T. "Modeling quantum programs: Challenges, initial results, and research directions," in Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software, 2020, pp. 14–21.

Ali, S., Khan, S.U.: Software outsourcing partnership model: an evaluation framework for vendor organizations. J. Syst. Softw. **117**, 402–425 (2016)

Ali, S., Yue, T., Abreu, R.: When software engineering meets quantum computing. Commun. ACM **65**, 84–88 (2022)

Akbar M. A., Rafi S., and Khan A. A., "Classical to quantum software migration journey begins: a conceptual readiness model," in product-focused software process improvement: 23rd international conference, PROFES 2022, Jyväskylä, Finland, November 21–23, 2022, Proceedings, 2022a, pp. 563–573.

Altman, E., Brown, K.R., Carleo, G., Carr, L.D., Demler, E., Chin, C., et al.: Quantum simulators: architectures and opportunities. PRX Quant. **2**, 017003 (2021)

Arias, D., Rodríguez, I.G., de Guzmán, M., Rodríguez, E.B., Terres, B.S., Gaviria, J., de la Puerta, I., Pastor, A.Z., Bringas, P.G.: Let's do it right the first time: survey on security concerns in the way to quantum software engineering. Neurocomputing **538**, 126199 (2023). https://doi.org/10.1016/j.neucom.2023.03.060

Attri, R., Dev, N., Sharma, V.: Interpretive structural modelling (ISM) approach: an overview. Res. J. Manag. Sci. **2319**, 1171 (2013b)

Attri, R., Grover, S., Dev, N., Kumar, D.: Analysis of barriers of total productive maintenance (TPM). Int. J. Syst. Assur. Eng. Manag. **4**, 365–377 (2013a)

Awschalom, D., Berggren, K.K., Bernien, H., Bhave, S., Carr, L.D., Davids, P., et al.: Development of quantum interconnects (quics) for next-generation information technologies. PRX Quantum **2**, 017002 (2021)

Azeem Akbar M., Khan A. A., Mahmood S., and Rafi S., "Quantum software engineering: a new genre of computing," arXiv e-prints, p. arXiv: 2211.13990, 2022.

Babbush R., Wiebe, N., & McClean, J. (2018). Resource estimation for quantum circuit compilation. arXiv preprint arXiv:1804.11085.

Belkhir, M., Benkaouha, H., Benkhelifa, E.: Quantum programming taxonomy. Ninth Int. Conf. Softw. Defin. Syst. (SDS) **2022**, 1–7 (2022)

Botsinis, P., Babar, Z., Alanis, D., Chandra, D., Nguyen, H., Ng, S.X., et al.: Quantum error correction protects quantum search algorithms against decoherence. Sci. Rep. **6**, 38095 (2016)

Bova, F., Goldfarb, A., Melko, R.G.: Commercial applications of quantum computing. EPJ Quant. Technol. **8**, 2 (2021)

Bozbura, F.T., Beskese, A., Kahraman, C.: Prioritization of human capital measurement indicators using fuzzy AHP. Expert Syst. Appl. **32**, 1100–1112 (2007)

Bravyi, S., Dial, O., Gambetta, J.M., Gil, D., Nazario, Z.: The future of quantum computing with superconducting qubits. J. Appl. Phys. **132**, 160902 (2022)

Cao Y., & Romero, J. (2018). Quantum software development: An overview. arXiv preprint arXiv:1806.06871. [3] Quantum Computing Education (QCE). (2021). Quantum Computing Education Resource Guide.

Chen, T.-Y., Tsao, C.-Y.: The interval-valued fuzzy TOPSIS method and experimental analysis. Fuzzy Sets Syst. **159**, 1410–1428 (2008)

Chong F. T., Frank, M. D., & Ross, N. J. (2017). A quantum software engineering manifesto. arXiv preprint arXiv:1709.03489.

Coles, P.J., Berta, M., Tomamichel, M., Wehner, S.: Entropic uncertainty relations and their applications. Rev. Mod. Phys. **90**(1), 015002 (2018)

Coveney, P.V., Highfield, R.R.: Quantum computing: a primer. CRC Press, Boca Raton (2019)

Cross A. W., Bishop, L. S., Smolin, J. A., & Gambetta, J. M. (2018). Open Quantum Assembly Language. arXiv preprint arXiv:1707.03429.

Cruz-Lemus J. A. and Serrano M. A., Quantum software tools overview," in quantum software engineering, ed: Springer, 2022, pp. 229–250.

De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., De Lucia, A.: Software engineering for quantum programming: how far are we? J. Syst. Softw. **190**, 111326 (2022)

Devitt, S.J., Munro, W.J., Nemoto, K.: Quantum error correction for beginners. Rep. Prog. Phys. **76**(7), 076001 (2013)

de la Barrera A. G., de Guzmán I. G.-R.,Polo M., and Cruz-Lemus J. A., "Quantum software testing: current trends and emerging proposals," in quantum software engineering, ed: Springer, 2022, pp. 167–191.

Fingerhuth, M., Babej, T., Wittek, P.: Open source software in quantum computing. PLoS ONE **13**, e0208561 (2018)

Gambetta J. M., Chow J. M., and Steffen M., Building logical qubits in a superconducting quantum computing system. npj Quantum Information. 3: 2, (2017)

Gambetta, J.M., Cross, A.W.: Building logical qubits in a superconducting quantum computing system. npj Quant. Inform. **4**, 1–8 (2018)

Gao, Q., Song, Q., Jiang, W.: Prioritizing software testing techniques based on fuzzy TOPSIS method. J. Ambient. Intell. Humaniz. Comput. **11**(8), 3391–3401 (2020)

Gill, S.S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M., et al.: Quantum computing: a taxonomy, systematic review and future directions. Softw. Pract. Exp. **52**, 66–114 (2022)

Guzik V., Gushanskiy S., Polenov M., and Potapov V., Models of a quantum computer, their characteristics and analysis," in 2015 9th International Conference on Application of Information and Communication Technologies (AICT), 2015, pp. 583–587.

Heim, B., Soeken, M., Marshall, S., Granade, C., Roetteler, M., Geller, A., et al.: Quantum programming languages. Nature Rev. Phys. **2**, 709–722 (2020)

Helsen, J., De Raedt, H.: Interoperability of quantum software: state of the art and outlook. Quantum **4**, 306 (2020)

Hevia, J.L., Peterssen, G., Piattini, M.: Quantum path: a quantum software development platform. Softw. Pract. Exp. **52**, 1517–1530 (2022)

Hoo Teo, K., Zhang, Y., Chowdhury, N., Rakheja, S., Ma, R., Xie, Q., et al.: Emerging GaN technologies for power, RF, digital, and quantum computing applications: recent advances and prospects. J. Appl. Phys. **130**, 160902 (2021)

Hsieh, H.-F., Shannon, S.E.: Three approaches to qualitative content analysis. Qual. Health Res. **15**, 1277–1288 (2005)

Huang, C.W., Tsai, M.F.: Developing a fuzzy TOPSIS-based framework for evaluating the quality of software development teams. J. Syst. Softw. **173**, 110956 (2021)

Ikart, E.M.: Survey questionnaire survey pretesting method: an evaluation of survey questionnaire via expert reviews technique. Asian J. Soc. Sci. Stud. **4**, 1 (2019)

Junior, F.R.L., Osiro, L., Carpinetti, L.C.R.: A comparison between Fuzzy AHP and Fuzzy TOPSIS methods to supplier selection. Appl. Soft Comput. **21**, 194–209 (2014)

Kannan, D., de Sousa Jabbour, A.B.L., Jabbour, C.J.C.: Selecting green suppliers based on GSCM practices: Using fuzzy TOPSIS applied to a Brazilian electronics company. Europ. J. Operat. Res. **233**, 432–447 (2014)

Kannan, G., Pokharel, S., Kumar, P.S.: A hybrid approach using ISM and fuzzy TOPSIS for the selection of reverse logistics provider. Resour. Conserv. Recycl. **54**, 28–36 (2009)

Keshta, I., Niazi, M., Alshayeb, M.: Towards implementation of requirements management specific practices (SP1. 3 and SP1. 4) for Saudi Arabian small and medium sized software development organizations. IEEE Access **5**, 24162–24183 (2017)

Khan A. A., Ahmad A., Waseem M., Liang P., Fahmideh M., Mikkonen T., et al., "Software architecture for quantum computing systems-a systematic review," (2022)

Khan R. A., Akbar M. A., Rafi S., Almagrabi A. O., and Alzahrani M., "Evaluation of requirement engineering best practices for secure software development in GSD: an ISM analysis," Authorea Preprints, 2023a.

Khan, A.A., Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Mikkonen, T., et al.: Software architecture for quantum computing systems-a systematic review. J. Syst. Softw. **201**, 111682 (2023b)

Khan, A.A., Akbar, M.A., Fahmideh, M., Liang, P., Waseem, M., Ahmad, A., Niazi, M., Abrahamsson, P.: AI ethics: an empirical study on the views of practitioners and lawmakers. IEEE Trans. Computat. Soc. Syst. (2023c). https://doi.org/10.1109/TCSS.2023.3251729

Khan, A.A., Shameem, M., Kumar, R.R., Hussain, S., Yan, X.: Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development. Appl. Soft Comput. **83**, 105648 (2019)

Khan, A.A., Shameem, M., Nadeem, M., Akbar, M.A.: Agile trends in Chinese global software development industry: fuzzy AHP based conceptual mapping. Appl. Soft Comput. **102**, 107090 (2021)

Kitchenham B. and Charters S., Guidelines for performing systematic literature reviews in software engineering," ed: UK, 2007.

Kitchenham, B., Pfleeger, S.L.: Principles of survey research: part 5: populations and samples. ACM SIGSOFT Softw. Eng. Notes **27**, 17–20 (2002)

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., et al.: Preliminary guidelines for empirical research in software engineering. IEEE Trans. Software Eng. **28**, 721–734 (2002)

Krohling, R.A., Campanharo, V.C.: Fuzzy TOPSIS for group decision making: a case study for accidents with oil spill in the sea. Expert Syst. Appl. **38**, 4190–4197 (2011)

Lanzagorta M., Uhlmann, J., & Aspuru-Guzik, A. (2020). Quantum computing for the determined. MIT Press. .

Lenarduzzi V. and Taibi D., "MVP explained: A systematic mapping study on the definitions of minimal viable product," in 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2016, pp. 112–119.

Li, H., Khomh, F., Openja, M.: "Understanding quantum software engineering challenges an empirical study on stack exchange forums and Github issues. IEEE Int. Conf. Softw. Maint. Evolut. (ICSME) **2021**, 343–354 (2021)

Liao, C.-N., Kao, H.-P.: An integrated fuzzy TOPSIS and MCGP approach to supplier selection in supply chain management. Expert Syst. Appl. **38**, 10803–10811 (2011)

MacQuarrie, E.R., Simon, C., Simmons, S., Maine, E.: The emerging commercial landscape of quantum computing. Nature Rev. Phys. **2**, 596–598 (2020)

Mahmood, S., Anwer, S., Niazi, M., Alshayeb, M., Richardson, I.: Key factors that influence task allocation in global software development. Inf. Softw. Technol. **91**, 102–122 (2017)

McCaskey, A., Dumitrescu, E., Liakh, D., Humble, T.: Hybrid programming for near-term quantum computing systems. IEEE Int. Conf. Reboot. Comput. (ICRC) **2018**, 1–12 (2018)

Mintz, T.M., Mccaskey, A.J., Dumitrescu, E.F., Moore, S.V., Powers, S., Lougovski, P.: Qcor: a language extension specification for the heterogeneous quantum-classical model of computation. ACM J. Emerg. Technol. Comput. Syst. (JETC) **16**, 1–17 (2020)

Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**, 032309 (2018)

Moguel E., Berrocal J., García-Alonso J., and Murillo J. M., A Roadmap for Quantum Software Engineering: Applying the Lessons Learned from the Classics," in Q-SET@ QCE, 5–13 (2020)

Moguel, E., Rojo, J., Valencia, D., Berrocal, J., Garcia-Alonso, J., Murillo, J.M.: Quantum service-oriented computing: current landscape and challenges. Software Qual. J. **30**, 983–1002 (2022)

Mohsin, M., Zhang, J., Saidur, R., Sun, H., Sait, S.M.: Economic assessment and ranking of wind power potential using fuzzy-TOPSIS approach. Environ. Sci. Pollut. Res. **26**, 22494–22511 (2019)

Moll N., Barkoutsos P.K., Benjamin S.C. (2021) Quantum development beyond Qiskit. npj Quant. Inform. 7(1): 1–7

Mykhailova M. and Soeken M., Testing Quantum Programs using Q# and Microsoft Quantum Development Kit, in Q-SET@ QCE, 2021, pp. 81–88.

Nagori, V., Varadarajan, V.: Quantum computing posing a challenge to the businesses. Int. J. Res. Eng. Sci. Manag. **6**, 52–55 (2023)
Niazi, M.: An exploratory study of software process improvement implementation risks. J. Softw. Evolut. Process **24**, 877–894 (2012)
Niazi, M.: Do systematic literature reviews outperform informal literature reviews in the software engineering domain? An initial case study. Arab. J. Sci. Eng. **40**, 845–855 (2015)
Niazi, M., Mahmood, S., Alshayeb, M., Qureshi, A.M., Faisal, K., Cerpa, N.: Toward successful project management in global software development. Int. J. Project Manage. **34**, 1553–1567 (2016)
Nielsen M. A., & Chuang, I. L. (2010). Quantum computation and quantum information. Cambridge University Press. .
O'Riordan, D., Jerger, N.: Quantum software development: An emerging discipline. IEEE Softw. **36**(3), 15–20 (2019)
Orús, R., Mugel, S., Lizaso, E.: Quantum computing for finance: overview and prospects. Rev. Phys. **4**, 100028 (2019)
Pan, J., Shang, Y., Xu, J.: Evaluating the impact of technical debt on software quality: a fuzzy TOPSIS approach. Inf. Softw. Technol. **126**, 106349 (2020)
Pandey, D., Suman, U., Ramani, A.K.: An effective requirement engineering process model for software development and requirements management. Int. Conf. Adv. Rec. Technol. Commun. Comput. **2010**, 287–291 (2010)
Pednault E., Gunnels, J., Nannicini, G., Horesh, L., Johnson, M. W., Romero, J., ... & Chuang, I. (2019). Leveraging secondary storage to simulate deep 54-qubit Sycamore circuits. arXiv preprint arXiv:1910.06156.
Pérez-Castillo, R., Serrano, M.A., Piattini, M.: Software modernization to embrace quantum technology. Adv. Eng. Softw. **151**, 102933 (2021)
Piattini, M., Peterssen, G., Pérez-Castillo, R.: Quantum computing: a new software engineering golden age. ACM SIGSOFT Softw. Eng. Notes **45**, 12–14 (2021)
Poczatek A., Gariazzo C., Martinson S., Broomfield C., and Farley D., Quantum information science and its implications for international safeguards (xxxx)
Poonia, R.C., Kalra, M.: Bridging approaches to reduce the gap between classical and quantum computing. J. Inf. Optim. Sci. **37**, 279–283 (2016)
Qiskit, Qiskit global summer school on quantum machine learning, (2021)
Rafi, S., Akbar, M.A., Mahmood, S., Alsanad, A., Alothaim, A.: Selection of DevOps best test practices: a hybrid approach using ISM and fuzzy TOPSIS analysis. J. Softw. Evolut. Process **34**, e2448 (2022)
Rafi, S., Yu, W., Akbar, M.A., Alsanad, A., Gumaei, A.: Multicriteria based decision making of DevOps data quality assessment challenges using fuzzy TOPSIS. IEEE Access **8**, 46958–46980 (2020)
Raj, T., Attri, R.: Identification and modelling of barriers in the implementation of TQM. Int. J. Product. Qual. Manag. **8**, 153–179 (2011)
Ravi, V., Shankar, R.: Analysis of interactions among the barriers of reverse logistics. Technol. Forecast. Soc. Chang. **72**, 1011–1029 (2005)
Sage A. P., Methodology for large-scale systems, (1977)
Sanchez, M.E.: Effects of questionnaire design on the quality of survey data. Public Opin. q. **56**, 206–217 (1992)
Saris, W.E., Gallhofer, I.N.: Design, evaluation, and analysis of questionnaires for survey research. Wiley, New Jersey (2014)
Sarkar A., "Automated Quantum Software Engineering: why? what? how?," arXiv preprint arXiv:2212.00619, 2022.
Serrano, M.A., Cruz-Lemus, J.A., Perez-Castillo, R., Piattini, M.: Quantum software components and platforms: overview and quality assessment. ACM Comput. Surv. **55**, 1–31 (2022)
Shameem, M., Khan, A.A., Hasan, M.G., Akbar, M.A.: Analytic hierarchy process based prioritisation and taxonomy of success factors for scaling agile methods in global software development. IET Software **14**, 389–401 (2020)
Shameem, M., Kumar, R.R., Kumar, C., Chandra, B., Khan, A.A.: Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. J. Softw. Evolut. Process **30**, e1979 (2018)
Sharma, H., Gupta, A.: The objectives of waste management in India: a futures inquiry. Technol. Forecast. Soc. Chang. **48**, 285–309 (1995)

Shaydulin R., Thomas C., and Rodeghero P., Making quantum computing open: Lessons from open source projects, in Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, 2020, pp. 451–455.
Shi, Y., Gokhale, P., Murali, P., Baker, J.M., Duckering, C., Ding, Y., et al.: Resource-efficient quantum computing by breaking abstractions. Proc. IEEE **108**, 1353–1370 (2020)
Singh, R.K., Benyoucef, L.: A fuzzy TOPSIS based approach for e-sourcing. Eng. Appl. Artif. Intell. **24**, 437–448 (2011)
Singh, R., Walia, R.S.: Security challenges and ethical considerations in quantum computing. Int. J. Netw. Sec. **22**(6), 1046–1052 (2020)
Sodhi B., Kapur R., "Quantum computing platforms: assessing the impact on quality attributes and sdlc activities," in 2021 IEEE 18th International Conference on Software Architecture (ICSA), 2021, pp. 80–91.
Soni, M.: End to end automation on cloud with build pipeline: the case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery. IEEE Int. Conf. Cloud Comput. Emerg. Markets (CCEM) **2015**, 85–89 (2015)
Sun, J., Zhao, S., Zhao, Y., Xie, X.: A fuzzy TOPSIS-based method for selecting software development outsourcing destinations. J. Ambient. Intell. Humaniz. Comput. **11**(12), 5279–5289 (2020)
Veryazov, V., Widmark, P.O., Serrano-Andrés, L., Lindh, R., Roos, B.O.: 2MOLCAS as a development platform for quantum chemistry software. Int. J. Quantum Chem. **100**, 626–635 (2004)
Warfield J. N., "Developing interconnection matrices in structural modeling," IEEE Transactions on Systems, Man, and Cybernetics, pp. 81–87, 1974.
Weder B., Barzen J., Leymann F., and Vietz D., Quantum software development lifecycle," in Quantum Software Engineering, ed: Springer, 2022, pp. 61–83.
Wohlin C., "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in Proceedings of the 18th international conference on evaluation and assessment in software engineering, 2014, pp. 1–10.
Yigitbasioglu, O.M., Kocaturk, T.: Management problems in quantum software engineering. IEEE Softw. **38**(1), 100–105 (2021)
Yoon, K., Hwang, C.-L.: Manufacturing plant location analysis by multiple attribute decision making: Part I—single-plant strategy. Int. J. Prod. Res. **23**, 345–359 (1985)
Zhao J., "Quantum software engineering: Landscapes and horizons," arXiv preprint arXiv:2007.07047, 2020.
Zidan, M., Eleuch, H., Abdel-Aty, M.: Non-classical computing problems: toward novel type of quantum computing problems. Results Phys. **21**, 103536 (2021)
Zouggari, A., Benyoucef, L.: Simulation based fuzzy TOPSIS approach for group multi-criteria supplier selection problem. Eng. Appl. Artif. Intell. **25**, 507–519 (2012)



## Authors and Affiliations

**Muhammad Azeem Akbar[1] · Arif Ali Khan[2] · Saima Rafi[3]**

✉ Arif Ali Khan
arif.khan@oulu.fi

Muhammad Azeem Akbar
azeem.akbar@lut.fi

[1] Software Engineering Department, Lappeenranta-Lahti University of Technology, 53851 Lappeenranta, Finland

[2] M3S Empirical Software Engineering Research Unit, University of Oulu, 90014 Oulu, Finland

[3] School of Computing and Engineering and The Built Environment, Edinburgh Napier University, Edinburgh, UK