Received: 25 November 2023 | Revised: 2 April 2024 | Accepted: 22 April 2024

IET Quantum Communication

DOI: 10.1049/qtc2.12096

The Institution of Engineering and Technology WILEY

REVIEW

# Quantum computing challenges and solutions in software industry—A multivocal literature review

**Masaud Salam** | **Muhammad Ilyas**

Department of Computer Science and IT, Software Engineering Research Group (SERG‐UOM), University of Malakand, Chakdara, Khyber Pakhtunkhwa, Pakistan

**Correspondence**
Muhammad Ilyas, Department of Computer Science and IT, University of Malakand, Dir Lower, Chakdara, KP 18800, Pakistan.
Email: milyasmkd@uom.edu.pk

**Abstract**

Quantum computing (QC) hinged upon the bedrock principles of quantum theory and holds promise for reforming a large number of industries. The researcher in this area aims to deliver a comprehensive understanding of the current state of the art and future trajectories of QC. The authors have discovered that most academic studies have concentrated upon dissecting specific aspects of QC. This discernment underscores the exigency of identifying challenges that might impede the seamless integration of QC within the software industry. Moreover, it becomes crucial to ascertain the panoply of solutions/practices required to overcome these barriers. A comprehensive multi‐vocal literature review was performed and culled a total of 49 academic papers for data extraction. A total of 13 challenges encountered by organisations were identified during the adoption of QC. Subsequently, these challenges were examined deeply and determined that five of them are the most critical, these are 'Lack of quantum specific algorithms, dev and testing methodologies', 'Difficult compilation and debugging', 'Lack of development tools and technology', 'Lack of development guidelines & Quality Assurance Standards' and 'Lack of professional expert', together founding over 30% of occurrences. These challenges from various perspectives were evaluated, including time frame, methodology, geographical region and publication platform. To address these barriers and implement the QC in software industry effectively, a total of 53 practices/solutions. This research aims to share valuable knowledge to simplify and amplify quantum application development.

**KEYWORDS**

quantum computing, quantum information

## 1 | INTRODUCTION

The working of Quantum computing (QC) hinged upon the bedrock principles of quantum theory [1], which were laid out by eminent scholars Max Planck and Niels Bohr during the early 20th century, gravitates towards a distinctive [2, 3]. QC harnesses quantum phenomena such as entanglement and superposition to achieve enhanced computational speed. Entanglement pertains to the profound interconnection between objects without direct interaction [4], whereas superposition enables objects to exist in multiple states simultaneously [5]. Entangled qubits, which serve as fundamental constituents of quantum systems, demonstrate a peculiar quantum state in which alterations made to one qubit exert an influence on the others, irrespective of their spatial separation. Consequently, when measuring and collapsing the superposition of one qubit into a singular state, the remaining qubits within the system are subject to analogous modifications [6].

Compared to classical computing (CC), QC provides a substantial enhancement in information processing capabilities, delivering significantly faster computational performance. QC excels particularly in executing specific tasks with exponential speed [7]. The potency of QC resides within the foundational unit of computation, namely the qubit. It is crucial to acknowledge that QC should not be perceived merely as an incremental advancement of conventional computing systems.

On the contrary, QC heralds a revolutionary breakthrough poised to fundamentally reshape the very fabric of the computing paradigm [8]. Traditional computing relies on bits, which are binary digits capable of holding either a '0' or a '1' at any given time, representing the 'off' or 'on' states of transistors [9]. Classical computers are composed of interconnected wires and circuits that process, store, and transmit information using electrical power. In contrast, quantum computers utilise qubits as the fundamental unit to encode and manipulate information, departing from the conventional storage mechanism based on 0 and 1s. There exist diverse problem types that are computationally infeasible to solve efficiently using CC, requiring an exponential amount of time and resources [10]. However, employing quantum computations enables the achievement of efficient solutions through a polynomial approach [11].

QC exhibits superiority over CC in two primary application areas [12]. The first area pertains to tasks demanding significant parallel computation, such as big data analysis, encryption, machine learning, and optimisation [13–16]. The second application area, such as those found in chemistry, materials science, and physics [17–19].

Despite being in its nascent stage, QC is experiencing rapid expansion. The market size for QC is projected to grow from US $0.5 billion in 2021 to US $ 1.7 billion in 2026, showcasing a remarkable compound annual growth rate of 30.2%. To enable QC to effectively tackle real-world challenges, the development of high-quality software with diverse capabilities is paramount. Consequently, the exploration of integrating Software Engineering (SE) principles into the QC community becomes indispensable, as it aids QC developers in generating larger and more robust code [20].

In recent years, the advancements and influence of QC technology have undergone substantial growth, underscoring its potential to revolutionise the field of SE. The significance of QC is increasingly acknowledged within the domains of SE and information science [21]. Given its tremendous potential, both industry professionals and academic scholars are devoting more attention to improving its reliability [22]. While research on QC has expanded, the focus has predominantly been on advancing technical components like quantum hardware and software tools and technologies [23]. However, there exists a dearth of research endeavours specifically aimed at identifying the challenges and opportunities that lie ahead in expanding QC knowledge, with insufficient emphasis placed on non-technical aspects [23].

Academic researchers possess limited understanding regarding the potential applications of QC in diverse industrial domains, including but not limited to project management, quality enhancement, and service delivery [21]. The paucity of research dedicated to comprehending the potential applications of QC across a broad spectrum of sectors, including energy, finance, and healthcare [24] is a critical gap that needs to be addressed. This gap is of utmost importance as QC has the potential to significantly enhance the effectiveness and efficiency of various business processes [25]. For instance, quantum technologies have the capability to expedite the development of novel materials and pharmaceuticals, as well as drive advancements in manufacturing processes [26].

Therefore, it is paramount to scrutinise the key factors that QC applications encounter in real-world scenarios. The evaluation and resolution of challenges posed by QC to the software industry have not been extensively explored, receiving inadequate attention and lacking a comprehensive understanding. In order to devise future strategies in this field, it is imperative to comprehend the obstacles that impede the effective utilisation of QC applications [27].

Addressing these challenges entails the consolidation of fragmented knowledge on QC and the delineation of its limitations. Furthermore, given that the practical implementation of QC in everyday industrial operations is not yet fully comprehended, it is imperative to enhance our understanding of its practical applications [28].

QC is an emerging field of research [29] that garners attention from both academic and industrial researchers. A plethora of knowledge pertaining to QC is available in diverse forms of literature, encompassing published/formal literature as well as grey literature (GL). Grey literature encompasses a broad spectrum of sources, including blogs, videos, technical reports, and white papers, which are not formally published in journals or conferences [30]. The literature revealed that QC is a newly emerging field for research [31]. Researchers, therefore, necessitate insights into the present-day advancements and forthcoming guidelines in the domain of QC. The available literature indicates that the majority of studies concentrate on specific aspects or dimensions of QC, underscoring the importance of identifying barriers that could hinder the adoption of QC in the software industry.

In order to bridge these gaps, our initial approach involved conducting a comprehensive multi-vocal literature review (MLR) with the aim of providing a thorough overview of the challenges associated with the adoption of QC. Moreover, it is imperative to explore potential solutions or practices that can surmount these identified barriers. These solutions or practices may entail the development of new algorithms, software, or techniques that can be integrated into QC systems to achieve specific objectives or address particular challenges.

The primary objective of our research work is to integrate QC into the software industry, driven by the perceived benefits that can be attained through its adoption. To pave the way for QC, we have established the following two main objectives: (1) Identification of the barriers encountered by the software industry in adopting QC. (2) Identification of solutions or practices to overcome these barriers in the implementation of QC in the software industry. The following research questions have been formulated to accomplish the aforementioned objectives.

> ***RQ 1***: What are the barriers, as identified through MLR, in the adoption of QC technology in the context of software industry?:
>
> ***RQ 2***: What are the solutions/practices, as identified through MLR, for addressing the barriers, identified in RQ- 1?:

The paper is structured as follows: Section 2 provides a background overview of the research. Section 3 outlines the research methodology employed. Section 4 presents the outcomes of the MLR, while Section 5 offers a detailed description of analysis and discussion. Finally, Section 6 concludes the paper and provides insights into future directions.

## 2 | BACKGROUND

It is imperative to examine the primary hurdles that arise in practical situations when implementing QC applications. The comprehensive exploration and evaluation of these challenges posed by QC to the software industry have been lacking, receiving inadequate attention and remaining incompletely understood. To shape future strategies in this field, comprehending the barriers that impede the utilisation of QC applications becomes necessary [27]. Effectively addressing these challenges entails consolidating fragmented knowledge on QC and defining its boundaries. Moreover, considering the limited understanding of the practical application of QC in everyday industrial operations, it is essential to enhance our comprehension of its practical implementation [28].

Zanni Junior et al. [7] conducted a systematic mapping study that primarily focused on the impact of quantum technologies on the software development process. The study explored various aspects of QC related to programming languages, such as the programming platforms used for QC, potential advancements that quantum computers can bring to software development, and identified domains where quantum computers would be particularly suitable. However, it is important to acknowledge that the study had a limited scope and did not provide a comprehensive overview of the current state of QC research or propose new directions for future research.

Piattini et al. [21] propose a roadmap for software engineers to prepare for the emerging era of QC. They suggest that software engineers should start by acquiring a solid understanding of the fundamental concepts of QC and its potential applications. Additionally, they recommend collaborating with experts in quantum physics and hardware to develop new software development tools and techniques. The authors delve into the potential impact of QC on SE, asserting that it will revolutionise the field and open up new avenues for innovation. The paper provides an overview of the fundamental principles of QC and explores its potential applications in various domains, including cryptography, machine learning, and optimisation. The authors also address the challenges that software engineers may encounter when developing software for quantum computers, such as the lack of standardised development tools and the need for specialised skills. The paper concludes by emphasising the importance of investing in research and development in both QC and SE. The authors argue that this new era of QC presents a unique opportunity for software engineers to create novel applications and solve complex problems in areas such as finance, healthcare, and energy.

Awan et al. [27] conducted a literature review using a fuzzy analytical hierarchy process (AHP) methodology to identify the challenges faced by the software industry due to the emergence of QC. The paper presents an approach that prioritises the identified challenges through the application of fuzzy AHP. Through a systematic literature review (SLR), the authors identified six challenges in the software industry related to QC: quantum algorithm design, quantum software development, quantum hardware development, quantum simulation, quantum-safe cryptography, and quantum education and training. The authors utilised the fuzzy AHP approach to rank these challenges based on their significance.

The findings revealed that quantum algorithm design held the highest level of importance, followed by quantum software development, quantum hardware development, quantum simulation, quantum-safe cryptography, and quantum education and training. The authors concluded that the software industry should prioritise quantum algorithm design, quantum software development, and quantum hardware development in order to effectively address the challenges posed by QC. Additionally, they recommended that companies invest in training their employees in QC to maintain a competitive edge.

Jianjun Zhao [31] conducted a survey in the field of quantum SE, aiming to investigate the current state of the art and explore various stages of the quantum software life cycle. The survey focused on different phases, including requirements, design, coding, testing, maintenance, and software reuse, with specific attention given to those phases that are unique to QC.

Laszlo Gyongyosi and Sandor Imre [32] conducted a survey on QC technologies, wherein they identified various problem types relevant to the field. These problems included: the quantum machine learning, computational problems, implementation of quantum algorithms and optimisation problems. The researchers proposed several algorithms tailored for QC, leveraging the effects of quantum mechanics to achieve significant speed enhancements compared to classical algorithms, both polynomial and exponential. They also highlighted that certain problems that are currently unsolvable using classical algorithms can be addressed through algorithms designed for QC. The primary focus of their study revolved around the conversion of classical algorithms into algorithms suitable for quantum technology.

Weder et al. [33] conducted research in the field of quantum software development life cycle (SDLC). The study unveiled that the quantum SDLC consists of 10 distinct phases: quantum-classical splitting, hardware implementation independency, quantum circuit enhancement, hardware-independent optimisation, quantum hardware selection, readout error mitigation preparation, compilation and hardware dependency optimisation, execution, integration, and analysis of results. Each phase was precisely defined in terms of its objectives, methodologies, available tools, and potential inputs/outputs. The primary aim of the study was to establish a hybrid SDLC capable of operating independently of hardware, functioning on both classical and QC platforms.

Garhwal et al. [34] conducted a literature survey with a specific focus on quantum programming languages. The survey analysed the high-level characteristics and compared various programming languages used in the context of quantum computers. The authors aimed to address inquiries regarding the current status of programming languages in the field of QC. However, it is important to acknowledge that this study had a restricted scope, focusing solely on programming languages specifically designed for QC.

Piattini et al. [35] propose a novel approach to SE that takes into account the unique characteristics of QC. They argue that conventional SE methods are not well-suited for the development of quantum software and advocate for new approaches to ensure the reliability, maintainability, and scalability of such software.

The paper begins by providing an overview of the current state of QC and the challenges that must be addressed to create practical quantum software. These challenges include the development of new algorithms and data structures that leverage the distinctive properties of QC, as well as resolving issues related to quantum error correction, noise, and decoherence. Subsequently, the authors present a new framework for quantum SE that integrates principles from both classical SE and quantum information science. This framework consists of guidelines for designing, implementing, and testing quantum software, along with a suite of tools and techniques to manage complexity and ensure the reliability of quantum software. Furthermore, the authors emphasise the significance of collaboration between quantum information scientists and software engineers in the development of quantum software. They argue that effective collaboration between these two communities is crucial for the success of quantum SE.

Akbar et al. [36] have proposed a new field of study called 'Quantum Software Engineering (Q-SE)' that focuses on the unique challenges and opportunities brought about by QC. The paper emphasises the need for innovative methodologies and tools to ensure the reliability and scalability of quantum software. It also highlights the importance of collaboration between quantum information scientists and software engineers in the development of quantum software. While the paper introduces a new perspective on SE for QC, it lacks comprehensive practical guidelines or concrete examples for implementing the proposed framework.

de la Barrera et al. [37] provide an overview of the current state of quantum software testing and highlight the unique challenges associated with testing quantum software due to the complexity and unpredictability of quantum systems.

The paper discusses the limitations of classical software testing methods for quantum software and proposes new techniques and tools, such as randomised benchmarking, gate set tomography, and quantum volume, for testing quantum software. It also emphasises the importance of collaboration between quantum information scientists and software engineers in developing effective testing methods for quantum software. However, the focus of the paper is primarily on summarising existing approaches and proposals for quantum software testing, and it does not provide detailed experimental results or practical guidelines for implementing the proposed techniques. Additionally, the paper does not address the issue of testing quantum software in the presence of noise and de-coherence, which is a critical challenge in the development of practical quantum software.

De Stefano et al. [38] conducted a survey on the current state of SE for quantum programming and discussed the progress and limitations in the field. The survey identified key challenges in developing reliable and scalable quantum software, including the need for new programming languages, compilers, and testing techniques. The paper emphasised the importance of collaboration between quantum information scientists and software engineers to address these challenges and accelerate progress in the field of quantum SE. However, one limitation of the study is that it provided a broad survey of the current state without delving into specific techniques or tools for addressing the discussed challenges. Additionally, the paper primarily focused on the technical challenges of quantum software development and did not address broader issues such as the societal and ethical implications of QC.

Moguel et al. [39] present an overview of the current state of quantum service-oriented computing, which is an approach to developing distributed computing systems using quantum resources. The paper discusses the challenges and opportunities involved in developing quantum services, including the need for new programming models and architectures that can leverage the unique properties of QC.

The authors emphasise the importance of collaboration between quantum information scientists and software engineers in the development of quantum services and explore potential applications of quantum service-oriented computing in areas, such as finance, cryptography, and optimisation. However, the paper does not delve into the technical implementation details of quantum service-oriented computing, such as specific algorithms, data structures, or programming languages.

The literature [23, 40, 41] on QC reveals dreams, such as improved cyber security, machine learning, optimised and stable artificial intelligence, accurate and timely weather forecasting, transportation route optimisation, environment friendly energy efficient systems, driverless automobiles, timely prediction of market instability, design automation, and fast estimation/prediction of financial risks, improvements in healthcare and medicine such as fast radiotherapies, virtual labs for discovery of new drugs, improved disease detection, non-invasive and high-precision surgeries, and creation of vaccines (like COVID-19). Similarly improved simulations of complex molecules, new materials discoveries and advanced molecular design in the field of Chemistry. These dreams, if come true, will change and improve the life style of human society. The available literature indicates that QC is an emerging field of research [31]. Consequently, researchers seek to gain insights into the current state of the art and future prospects within the realm of QC. It has been observed that the majority of studies concentrate on specific aspects or dimensions of QC, highlighting the necessity to identify barriers that could impede the adoption of QC within the software industry. Additionally, there is a need to identify corresponding solutions or practices that can effectively address the identified barriers.

## 3 | METHODOLOGY

We have conducted our research using the concept of MLR, which involved a comprehensive examination of relevant literature to identify the factors or challenges faced in the adoption of QC in the software industry. Moreover, it was also our objective to identify effective practices or solutions that can be employed to overcome these factors or challenges.

MLR is an emerging trend in SE and is a type of SLR. It encompasses not only published literature such as conference papers and journal articles but also include informal literature such as blog posts, videos, and white papers. MLR differs from both SLR and ordinary literature reviews because it aims to be unbiased.

MLR provides significant advantages in specific fields of SE where there are ongoing developments and a lack of institutional studies.

Garousi et al. [42] highlighted the importance of conducting MLR as primary studies, particularly in the field of SE. They emphasised that developers in SE heavily rely on GL as a common way to disseminate knowledge, guidance, and practices related to new methods, methodologies, and skill-driven advancements. Relying solely on traditional research literature may result in the exclusion of significant up-to-date data on rapidly evolving real-world occurrences of interest.

MLR has the potential to bridge the gap between industry and academic research by incorporating a wider range of sources and perspectives [42]. MLRs are valuable for researchers and practitioners alike as they present a summary of both the current state-of-the-art & state-of-practice in a particular field [43, 44]. The primary motivations for including GL in our research are lack of intuitional research in the area [44] of QC as it is an emerging research area & complementary suggestion in GL. White papers and articles available on the Internet are the main sources of GL data used. Table 1 presents different types of literature that is, Published, Grey and Black literature.

We present a standard MLR procedure in Figure 1, which is based on the SLR method described in the guidelines by Kitchenham and Charters [45]. We have adopted three phases for conducting MLRs from the SLR guidelines [45], as they have been proven to be well-organised and pertinent. The three phases include Review Planning, Carrying out the review, and reporting the results of the review.

**TABLE 1** Range of white, grey and black literature.

| Publish/white literature | Unpublished/ grey literature | Black literature |
|---|---|---|
| Formal literature | Audio-video (AV) media | Concepts |
| Published journal papers conference proceedings | Blogs data sets | Feelings |
| | e-Prints | Ideas |
| Books | Lectures | Opinions |
| | Preprints | Thoughts |
| | Reports | |
| | Technical | |

### 3.1 | Constructing search term

Kai Petersen and the research team [46] developed a methodology known as PICO, an acronym representing 'Population', 'Intervention', 'Comparison', and 'Outcomes', to facilitate the identification of keywords [47]. The framework, recommended by Kitchenham and Charters in 2007, was employed to devise search terms or strings aligned with the research questions under investigation. In the following sections, we outline a selection of terms that will aid us in constructing the search terms for our research questions.

**Population**: Software Engineering & QC.
**Interventions**: Challenges/risks/factors/problems.
**Outcome of relevance**: Smooth adoption of QC in software industry.
**Experimental design**: MLR, empirical studies, case studies, theoretical studies and expert opinions.

### 3.2 | Search strategy and searching

The steps involved in the MLR are as follows:

- Determine the population, intervention, and outcome to create search terms.
- Identify alternative spellings and synonyms.
- Verify the keywords in any relevant papers.

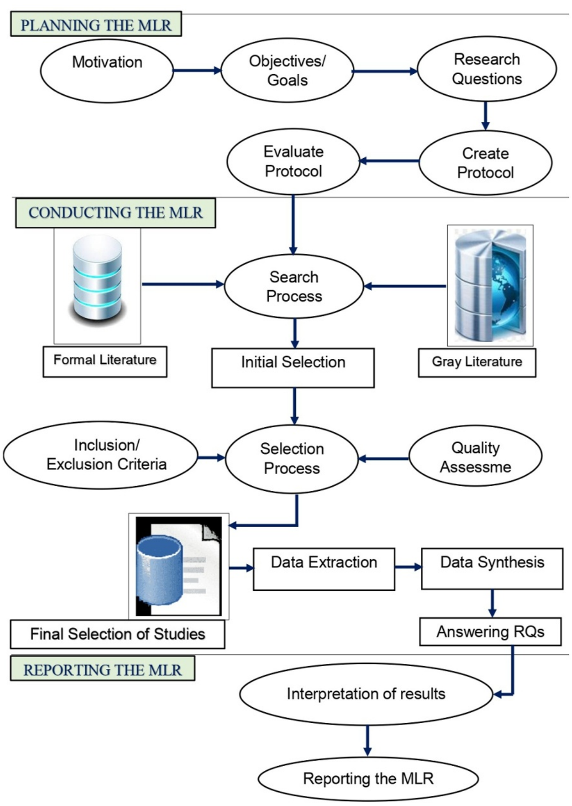


**FIGURE 1** Process steps for multi-vocal literature review (MLR).

- Employ Boolean Operators.
- Reduce the search string into a summarise form if necessary.

To create the search string for resource searching, we utilised Boolean operators (OR and AND) to combine the keywords. The 'OR' operator was used to link synonyms and alternative spellings, while the 'AND' operator was used to connect the main terms. Consequently, the following search string was formulated: (('Quantum Computing') AND (Software OR Programming OR 'Software Development' OR 'Application Development') AND (Challenge OR Barrier OR Risk OR Threat OR Solution OR Practice)).

## 3.3 | Selection of literature resources

In order to identify literature resources for this research, we divided this step into two sub-steps: searching for white literature (WL) resources and searching for GL resources.

### 3.3.1 | Resources to be searched for WL

Following the guidelines outlined by Kitchenham and Charters in 2007, two search approaches will be employed to find relevant GL. Firstly, an automated search will be conducted using digital databases. Additionally, the snowballing technique will be utilised as an additional method for gathering relevant sources.

### 3.3.2 | Automatic search (digital database)

We have selected 'IEEE Xplore', 'ACM Portal', 'ScienceDirect', 'SpringerLink', and 'Wiley Online Library' as the digital libraries, databases, and search engines for searching published resources. These databases were chosen because they provide access to content that aligns with the services offered by our institution. Additionally, these databases encompass a wide range of impactful full-text journals and conference proceedings, providing comprehensive coverage of the field of SE as a whole (Kitchenham & Charters, 2007).

*Snowballing*

In accordance with the guidelines established by Kitchenham and Charters (2007) for SLRs, both forward and backward snowballing techniques utilised and ensured the inclusion of all relevant sources to the maximum extent possible.

*Resources searched for grey literature*

To identify suitable literature resources, we adopted a specific search strategy for GL [48], following the guidelines proposed by Garousi et al. in their MLR guidelines [43]. The strategy involves an advanced search on both the regular Google search engine and digital database the 'ProQuest Dissertations and Thesis Global' database to identify relevant PhD and master's theses.

## 3.4 | Criteria for the selection of the literatures

Figure 2 presents the criteria for the selection of articles. The selection of publications is based on the objective of including only those that are directly relevant to the research questions. Our focus is specifically on papers related to QC software/applications, while excluding any papers that primarily deal with QC. The selection process involved assessing titles, abstracts, reading full-text articles, and evaluating their quality. The aim was to gather a comprehensive set of papers that meet specific criteria for inclusion and exclusion, ensuring their relevance to our MLR. We have defined a set of inclusion and exclusion criteria, which are illustrated in the accompanying Figure 2, to guide the publication selection process.

## 3.5 | Primary selection

The first step we have performed in selecting key sources for the study was assessing their titles, keywords, and abstracts. The objective was to eliminate any irrelevant results and concentrate solely on those that are directly related to the research questions and the area of interest. After this initial selection process, the chosen primary sources was then passed from thorough evaluation where a comprehensive review of their full text conducted, taking into account the inclusion and exclusion criteria specified below. In situations where there was uncertainty about whether a particular source should be included or excluded, we directly referred it to a secondary reviewer for further assessment.

## 3.6 | Inclusion criteria

We have established specific criteria to determine the literature to be used for data extraction based on our search terms. Only

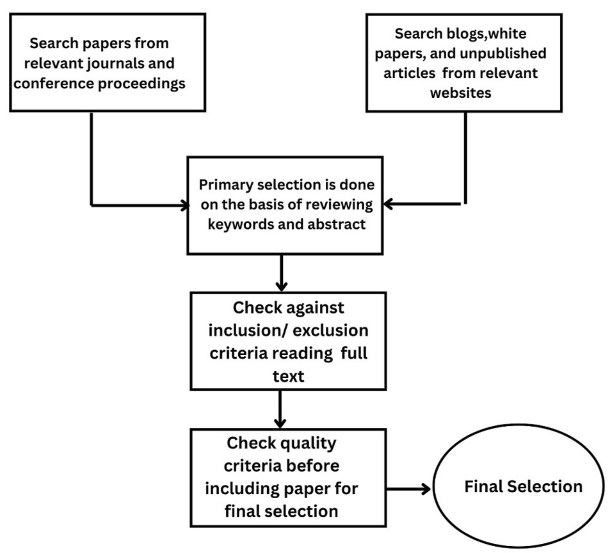


**FIGURE 2** Publication selection criteria.

formal and informal papers related to QC and software/applications, written in English, were considered. The focus was on studies that:

- Discuss the general development of quantum applications.
- Describe the challenges faced by developers in adopting QC.
- Describe the solutions and challenges related to the adoption of QC.
- The search included both published and unpublished GL available online.
- Research papers, articles, books, white papers, & review papers written in English were included in the search.

## 3.7 | Exclusion criteria

To identify which literature will be excluded or disregarded from the search results, the following exclusion criteria were established:

- Research that does not address the research questions.
- Research that does not outline the challenges related to QC from a software perspective.
- Studies that do not meet the criteria specified in the inclusion criteria mentioned earlier.
- Duplicate articles found through multiple search engines and resources were identified and removed from the final set of papers selected for analysis.
- Publications that are not written in English, but in another language, were excluded.

## 3.8 | Quality assessment

All the papers included in the final selection underwent a thorough quality evaluation. We established specific criteria for assessing the quality and used a three-point Likert scale ('yes', 'partially', 'no') to rate each component of the reviewed articles. To ensure meaningful and consistent results, we assigned a value of 2 to 'yes', 1 to 'partially', and 0 to 'no' for each component. Any paper that achieved an average score of $\geq 0.5$ was considered to have acceptable quality and was included in the MLR. The quality list containing the relevant questions is provided in Table 2.

## 3.9 | Procedure for study selection

To refine the research articles and papers identified through search strings in digital libraries and relevant websites for GL, we employed the tollgate approach as recommended by Afzal et al. [49] in our MLR study. This approach consists of five phases:

- We conducted searches in digital libraries and databases to locate relevant articles.
- We assessed the titles and abstracts of the papers to determine their inclusion or exclusion.
- We reviewed the introduction and conclusions of the papers to make further decisions regarding inclusion or exclusion.
- We thoroughly read the full text of the papers to determine their suitability for our study.
- We selected the primary studies and papers that would be included in the MLR study.

After executing the search strings and applying the inclusion and exclusion criteria, a total of 10,366 research publications were extracted from online digital libraries, databases, and websites. Figure 3 presents the detailed process of the finally selected articles.

The selection process consisted of two stages. In the first stage, the papers were initially screened based on their titles and abstracts, resulting in the identification of 400 articles/papers that had the potential to meet the selection criteria. The second stage involved a comprehensive reading of the full text

**TABLE 2** Criteria for quality assessment.

| S. No. | Criteria for quality assessment | Likert scale |
|---|---|---|
| Q.1 | Is the sample size of the study being conducted is appropriate? | Yes = 2<br>Partially = 1<br>No = 0 |
| Q.2 | Does the paper point out any challenge/factor regarding QC? | Yes = 2<br>Partially = 1<br>No = 0 |
| Q.3 | Solutions to overcome the challenges faced by the QC during adoption? | Yes = 2<br>Partially = 1<br>No = 0 |

Abbreviation: QC, quantum computing.

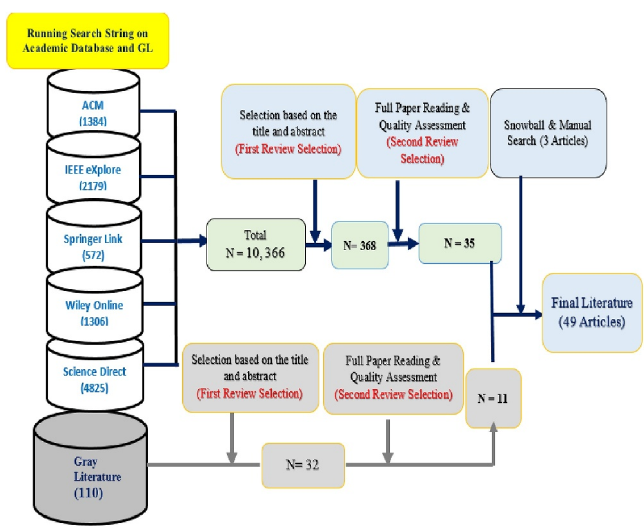


**FIGURE 3** Final literature selection phases.

of the papers that passed the initial screening, and the final selection was made from this list.

The final selection from the MLR consisted of 49 articles that were chosen to address the research questions mentioned above, as depicted in Table 3 below, detailed list of these articles is affixed in Appendix A for ready reference.

## 3.10 | Data extraction

The purpose of this phase is to extract the necessary data from the selected papers to address the research questions. The papers that successfully met the inclusion, exclusion, and quality assessment criteria discussed above resulted in a final set of $N = 49$ publications. The data extraction process followed a predefined data extraction form, which include fields such as Serial Number, Paper-ID, Review-Date, Title of the paper, Author's name, Reference, Database, Methodology (survey, report, interview, case study etc.), Population, Organisation Type, Company Size (small/medium/large), Analysis Location/Country, Publication Year, and Challenges faced by organisations in adopting QC in the software industry.

## 3.11 | Data storage

We have stored the challenges/factors and solutions recognised from every paper in MS Word electronically.

## 3.12 | Data synthesis

The extracted data was synthesised, leading to the identification of 37 challenges. Following this, some of specific challenges were grouped, ultimately 13 distinct challenges were then finalised. The detail overview is shown in Table 4.

# 4 | RESULTS

In this section, we present the outcomes of the MLR and analyse the identified challenges. The results are then examined to address Research Question-1 (RQ1) and Research Question-2 (RQ2) as discussed in Section 1. Further subsections provide more detailed information about these findings.

**TABLE 3** Total number of finally selected papers.

| S. No. | Resource | Final selection |
|---|---|---|
| 1. | ACM portal | 14 |
| 2. | IEEE eXplore | 12 |
| 3. | Springer-link | 06 |
| 4. | Science-direct | 02 |
| 5. | Wiley online library | 01 |
| 6. | Google search engine | 11 |
| 7. | Snowballing | 03 |
| **Total papers finally selected** | | **49** |

## 4.1 | Challenges identified in the adoption of QC in software industry through MLR

Upon the culmination of the MLR process, our discerning analysis brought to the fore 13 formidable challenges that organisations may confront during their quest for QC adoption. These challenges, standing as daunting impediments, necessitate immediate and sagacious intervention to foster the broad-based embrace of QC.

These challenges were prioritised based on their frequency within the selected publications. The chart in Figure 4 represents the visible 13 threatening challenges along with highest frequency. The subsequent section delves into a comprehensive exploration of these challenges encountered by organisations throughout the development process of quantum application.

### 4.1.1 | Challenge # 1: Lack of quantum specific algorithms, dev and testing methodologies

According to our study, the most significant challenge identified among the 13 factors is the ‘Lack of quantum-specific algorithms, development, and testing methodologies’, with

**TABLE 4** List of challenges.

| S. No. | Name of challenge | Freq. | % Age |
|---|---|---|---|
| 1 | Lack of quantum specific algorithms, dev and testing methodologies | 32 | 65 |
| 2 | Difficult compilation and debugging | 20 | 41 |
| 3 | Lack of development tools and technology | 19 | 39 |
| 4 | Lack of development guidelines and Quality Assurance Standards | 17 | 35 |
| 5 | Lack of professional expert | 15 | 31 |
| 6 | Lack of management support and incorporation of QC in the curriculum | 14 | 29 |
| 7 | Challenging design and validation process | 14 | 29 |
| 8 | Limited development environment and resources availability | 11 | 22 |
| 9 | Huge dev, deployment, maintenance, operation and sustainability cost | 9 | 18 |
| 10 | Lack of QC API & UI | 8 | 16 |
| 11 | Quantum domain complexity | 8 | 16 |
| 12 | Scalability, portability, integration and interoperability issues | 8 | 16 |
| 13 | Lack of roadmap for migration from classical to QC | 8 | 16 |

Abbreviations: API, application programming interface; QC, quantum computing; UI, user interface.

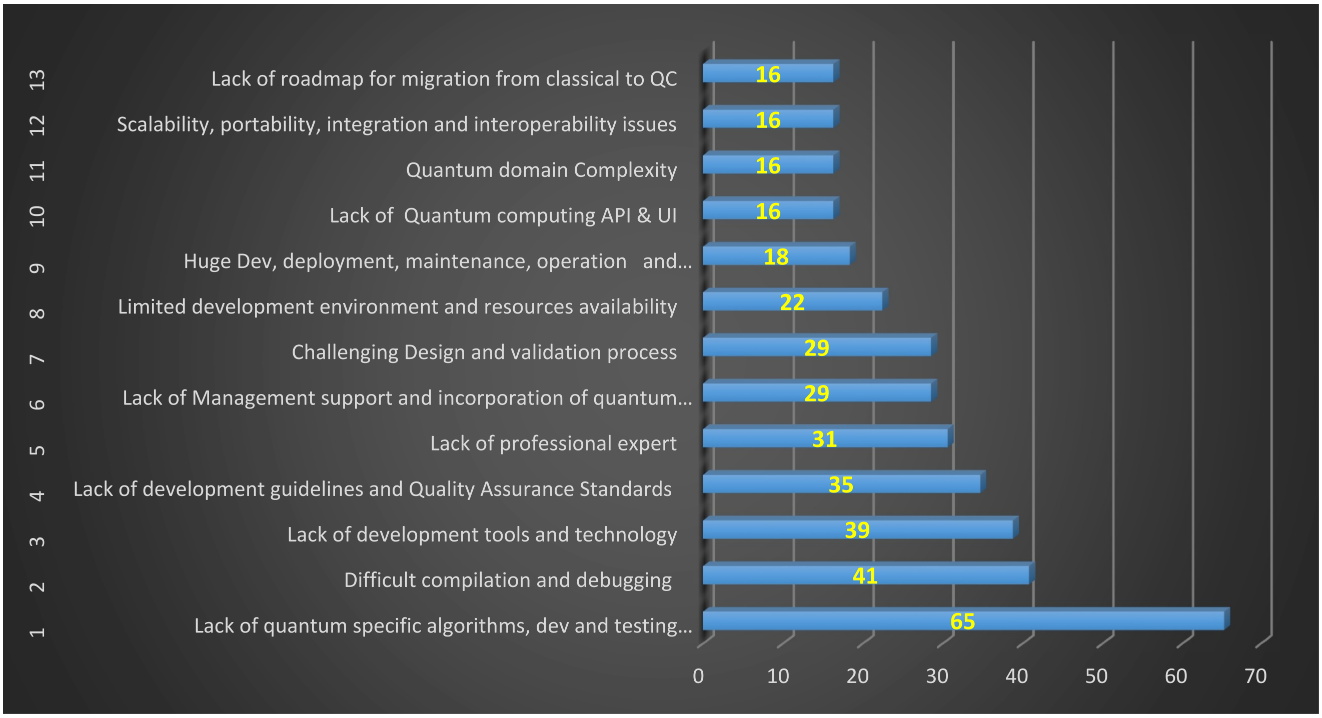


**FIGURE 4** Challenges with higher frequency from bottom to top.

65% frequency. This indicates that the progress on QC is impeded not only by hardware limitations but also by software, particularly the algorithms that facilitate the effective operation of the software. It is essential to develop methodologies, techniques, and tools for reusing quantum software systems in order to advance in the field [31]. Furthermore, the specification, verification, debugging, and testing of quantum programs are extremely challenging tasks [50].

#### 4.1.2 | Challenge # 2: Difficult compilation and debugging

The second most significant barrier in the field of QC, with a frequency of 41%, is the challenging process of compilation and debugging. Debugging and verification of quantum programs are identified as the primary barriers in the adoption of QC, as highlighted in previous studies [38, 51]. Additionally, issues related to cost optimisation, compilation time, memory usage, and compile time checking are prevalent concerns in this field [52].

#### 4.1.3 | Challenge # 3: Lack of development tools and technology

Another barrier that severely affects the software industry in embracing QC is the availability of suitable programming languages, tools, and techniques. The development of quantum software is significantly hindered by these obstacles. The lack of development tools and technology ranks as the third highest critical challenge, with 35% frequency. The need for new tools arises from the requirement to write quantum algorithms in the form of program code. Currently, there is no comprehensive methodology or lifecycle that encompasses all the relevant phases occurring during the development and execution process [33, 53, 54].

#### 4.1.4 | Challenge # 4: Lack of development guidelines and quality assurance standards

Developing new methodologies and performance standards that focus on modularity, maintainability, and portability is essential for the successful implementation of quantum programs. Furthermore, it is necessary to establish recommendations and exemplary methods for programming quantum applications.

#### 4.1.5 | Challenge # 5: Lack of professional expert

Since QC is a relatively new field of study, there is a scarcity of qualified professionals in the domain, which hampers and slows down the progress of QC. Due to the limited understanding of SE methodologies among quantum computer scientists, there is a higher likelihood of repeating mistakes and experiencing costly 'rediscoveries' [55, 56]. A programmer will not have the fundamental understanding of how to write code without bugs. The ability of testers to create adequate test cases is lacking [57].

#### 4.1.6 | Challenge # 6: Lack of management support and incorporation of QC in the curriculum

Organisational barriers to the adoption of QC include the need for stable management that can effectively drive the organisation's long-term research and development vision [8]. Unfortunately, management often prioritises short-term profits over making long-term commitments, leading to a short-sighted approach. Additionally, in the field of QC, a significant issue is the lack of collaboration and communication between academia and industry [27, 36, 58]. Experts predict that the next generation of software development may face a shortage of resources, as many universities and institutions still lack a framework to integrate QC into their curriculum. This limited integration results in a constrained pool of resources for the industry to compete [12].

#### 4.1.7 | Challenge # 7: Challenging design and validation process

One of the key challenge preventing the software industry from embracing QC is the absence of engineering designs for modernising QC software and inadequate resources for software development that is, insufficient resources are available for software design [21]. Only a limited number of research studies have addressed the design and validation challenges associated with QC technologies [59]. Many design issues, including simulation, compilation, verification and validation, exist in the field of QC [58].

#### 4.1.8 | Challenge # 8: Limited development environment and resources availability

As QC is a recent and rapidly evolving field, it faces numerous challenges, among one of them is the limited accessibility of quantum development environments. While there may be a pool of resources available within certain organisations, they are still scarce, which raises concerns regarding the early adoption of QC [8]. A notable resource disparity persists between practical quantum algorithms and actual machines also affecting the adoption of QC.

#### 4.1.9 | Challenge # 9: Huge development, deployment, maintenance, operation and sustainability cost

Since the QCs are expensive [60], they suffer due to the lack of allocation of stable budget and long-term planning. The development, deployment, maintenance, operation, and sustainability of QC are the key challenges that hinder its implementation in the software industry.

#### 4.1.10 | Challenge # 10: Lack of QC API & UI

Creating user interfaces and application programming interfaces (APIs) for quantum programming languages would require a significant amount of time and money [60]. QC faces similar SE challenges as other domains, such as issues related to API usage and changes. Additionally, the lack of established tools to aid in quantum program development is a significant concern [61].

#### 4.1.11 | Challenge # 11: Quantum domain complexity

The interdisciplinary nature of QC adds to the domain complexity challenge [31, 55]. Furthermore, the new complexity introduced in quantum programming makes it difficult to find bugs in the quantum source code [62]. Testing quantum programs is particularly difficult because of the intricate nature of quantum features such as superposition and entanglement. Moreover, writing quantum software is a complex task.

#### 4.1.12 | Challenge # 12: Scalability, portability, integration and interoperability issues

QC faces significant challenges related to scalability, technology for reliable and realistic implementation [63], the integration of QC systems with classical computer systems [38], the creation and integration of libraries, and the lack of code portability [52].

#### 4.1.13 | Challenge # 13: Lack of roadmap for migration from classical to QC

The lack of a clear plan for transitioning from classical to quantum SE can lead to changes in project management [64]. This is because there are currently no tools or frameworks available to assess an organisation's readiness to transition from traditional to quantum software development methods. In the maintenance phase, quantum software may require adjustments, modifications, or updates to meet customer requirements. A new research area in quantum SE is re-engineering, which seeks to integrate classical information systems with quantum algorithms [65].

### 4.2 | Solutions and practices to overcome the identified challenges/barriers in implementation of QC within the software industry

Our assiduous inquiry has engendered the recognition of 53 salient practices, gleaned from an expansive and all-encompassing MLR. These practices, undergirded by sagacity and innovation, span a diverse gamut of pragmatic approaches and strategies, effectively furnishing organisations with the wherewithal to circumnavigate the obstacles encountered during the adoption of QC. By adeptly implementing these astute solutions and practices, organisations actively involved in quantum application development shall be endowed with the fortitude and acumen to forge ahead, ultimately engendering a pantheon of auspicious quantum applications that augur a propitious future. The list of practices for each of the 13 challenges is presented in Tables 5–17.

**TABLE 5** Practices for challenge # 1: Lack of quantum specific algorithms, dev and testing methodologies.

| Practice no. | Practice (total = 14) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 1.1 | QC systems require appropriate structures and frameworks to facilitate the formulation, verification, and coordination of quantum algorithms | 10, 14, 13, 61 | (4) |
| P# 1.2 | Eigenvalues and eigenvectors can be used to forecast code quality and identify high risk applications | 3 | (1) |
| P# 1.3 | The static analysis, statistical assertion, and runtime assertion scheme methods are suggested to detect errors in quantum programs. These approaches can aid in guaranteeing platform accuracy and reducing the likelihood of bugs in QC | 5, 8 | (2) |
| P# 1.4 | It has been suggested that the LoC metric is useful for determining the size of quantum software as well as the development effort. This is similar to how LoC is used to evaluate classical software | 26 | (1) |
| P# 1.5 | At each stage of the creation of the application, the software engineer must apply the most suitable quantum software technique | 61 | (1) |
| P# 1.6 | The influence of faults in the quantum programme is reduced using a readout errors mitigation method | 8 | (1) |
| P# 1.7 | When fitting data in SE applications like cost estimate, software reliability predictions, and defect prediction, the least squares method is effective | 3 | (1) |
| P# 1.8 | Testing quantum programs by utilising metamorphic relations to solve the quantum measurement problem is proposed | 19 | (1) |
| P# 1.9 | QuCAT (QUantum CombineAtorial Testing) [21] and (QUantum InpuTOutput coverage) [26] and 'QuSBT [49]', Quito [26] and quantum software mutation-based testing method [62], fault model [61] are techniques proposed for verifying the correctness of quantum programs through systematic and automated testing. Its aim is to ensure the accuracy of quantum programs | 21, 31, 49, 61 | (4) |
| P# 1.10 | Researchers should advise the QC sector to give open source software top priority in order to promote QC innovation, dissemination, and adoption | 38 | (1) |
| P# 1.11 | In order to extend the advantages of cloud and service computing to quantum software and services, it is necessary to create new methodologies, tools, and techniques | 42 | (1) |
| P# 1.12 | We propose that CBQSE might be a promising strategy for the design and creation of systems based on quantum computers | 61 | (1) |
| P# 1.13 | To include the effects of QC in software architectures, it may be necessary to devise novel techniques for defining the characteristics, functionalities, constraints, and attributes attributed to architectural components | 61 | (1) |
| P# 1.14 | It is essential to create a community for quantum SE that is focused on coming up with effective strategies, instruments, and procedures for creating quantum software systems | 61 | (1) |

Abbreviations: CBQSE, Component-Based Quantum Software Engineering; LoC, lines of code; MLR, multi-vocal literature review; QC, quantum computing; SE, software engineering.

**TABLE 6** Practices for challenge # 2: Difficult compilation and debugging.

| S. No. | Practice (total practices = 1) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 2.1 | It is essential to create a quantum ecosystem that includes a consistent quantum programming language, compilers, and debuggers, as well as a quantum hardware abstraction layer. This will enable the compilation of a single programme for numerous quantum hardware platforms | 41 | (1) |

Abbreviation: MLR, multi-vocal literature review.

**TABLE 7** Practices for challenge # 3: Lack of development tools and technology.

| S. No. | Practice (total practices = 7) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 3.1 | The vendors should coordinate their efforts to solve the industry's lack of skilled workers problem by offering suitable tools and trained staff | 1 | (1) |
| P# 3.2 | Platform independence and the ability to perform numerous functions, such as the production, validation, and verification of quantum codes, are requirements for quantum software models | 14 | (1) |
| P# 3.3 | Quantum modelling notations like QML or architectural languages [9] can be used to describe the model in a semi-formal manner | 30 | (1) |
| P# 3.4 | To support quantum software, one needs a quantum software stack that consists of operating systems, compilers, and programming languages | 63 | (1) |
| P# 3.5 | Google's Cirq, Xanadu's Pennylane, and IBM's Qiskit are examples of software development kits for QC. These kits enable developers and researchers to use a common language that can be implemented on different quantum platforms. Essentially, they provide a standardised way for people to work with quantum computers, regardless of the specific hardware they are using | 73, 11 | (2) |
| P# 3.6 | To carry out computational experiments and validate language and algorithm designs, it is crucial to have reliable quantum computer simulators in place | 77, 19, 71 | (3) |
| P# 3.7 | Presently, there is a requirement for a fresh programming approach and software stack to enable the implementation of QC | 78 | (1) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing.

**TABLE 8** Practices for challenge # 4: Lack of development guidelines and quality assurance standards.

| S. No. | Practice (total = 4) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 4.1 | Professional associations should assess their methods, techniques, and practices and incorporate the principles of quantum SE to an appropriate extent. Furthermore, they can contribute to promoting the field by disseminating information about its advantages and challenges | 1 | (1) |
| P# 4.2 | Guidelines are needed to enable the development of well-engineered quantum software systems given the maturity of quantum software development approaches | 61 | (1) |
| P# 4.3 | Practitioners and the research community must focus on standardising representations for quantum software components | 63 | (1) |
| P# 4.4 | To avoid spending time on unnecessary effort, quantum computer scientists need to comprehend modern SE ideas and practices | 82 | (1) |

Abbreviations: MLR, multi-vocal literature review; SE, software engineering.

**TABLE 9** Practices for challenge # 5: Lack of professional experts.

| S. No. | Practice (total = 5) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 5.1 | The lack of skilled workers in the disciplines of QC and quantum SE should be addressed by educational institutions | 1, 2, 38, 73, 80, 81 | (6) |
| P# 5.2 | Governments should encourage educational institutions to offer training in quantum SE and computing, physics, linear algebra, and mathematics | 1, 2, 3, 8, 65 | (5) |
| P# 5.3 | Technical specialists are required to create a QC-friendly atmosphere and a robust ecosystem that can transcend organisational boundaries | 38 | (1) |
| P# 5.4 | It is suggested that practitioners begin understanding the fundamentals of QC | 65 | (1) |
| P# 5.5 | It is advised that cooperation between businesses and academia must be promoted along with an integrated strategy for the study and creation of quantum hardware and software. A good example of such a collaboration is the project 'Quantum' | 38, 79, 83 | (3) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing; SE, software engineering.

**TABLE 10** Practices for challenge # 6: Lack of management support and incorporation of QC in the curriculum.

| S. No. | Practice (total = 4) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 6.1 | The degree programs for SE and other related fields of study should include courses and curricula related to quantum SE | 1, 2 | (2) |
| P# 6.2 | Quantum SE should be included in the strategic plans of the government and funding agencies, with adequate funding and dissemination programs provided to support its development | 1 | (1) |
| P# 6.3 | A quantum programming language must provide high-level abstractions to shorten the development time and enable portability across diverse quantum hardware backbends | 37 | (1) |
| P# 6.4 | It is essential for countries to collaborate with the private sector and other friendly nations to tackle the key challenges facing QC. Additionally, there should be an increase in financial support for the advancement of this technology | 38, 69 | (2) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing; SE, software engineering.

**TABLE 11** Practices for challenge # 7: Challenging design and validation process.

| S. No. | Practice (total = 4) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 7.1 | The cross-talk-aware algorithm is a method for lowering the quantum programmers' mistake rate | 19 | (1) |
| P# 7.2 | By analysing programs, the identification of bug patterns can aid in the creation of effective methods for locating bugs in the source code of quantum programs | 23 | (1) |
| P# 7.3 | In order to make quantum software and services more accessible through cloud and service computing, it is necessary to create new approaches, tools, and methods | 42 | (1) |
| P# 7.4 | Since it is currently unclear what debugging methods are best for QC, new strategies must be developed | 61 | (2) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing.

**TABLE 12** Practices for challenge # 8: Limited development environment and resources availability.

| S. No. | Practice (total = 4) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 8.1 | The utilisation of QSA allows creators and developers of quantum software to establish connections between the actions of qubits and the various architectural components and connectors necessary for implementing quantum software | 30 | (1) |
| P# 8.2 | Tools for evaluating architecture (as exemplified by the Software Architecture Analysis Method or SAAM) are employed to perform an objective assessment of the architecture and identify any defects or inconsistencies present in its source code | 30 | (1) |
| P# 8.3 | We propose that CBQSE may be an effective strategy for the design and creation of systems based on quantum computers | 61 | (1) |
| P# 8.4 | To enhance understanding, design, and implementation, design techniques for visualising Q-algorithm execution must be used | 77 | (2) |

Abbreviations: CBQSE, Component-Based Quantum Software Engineering; MLR, multi-vocal literature review; QSA, Quantum Software Architecture.

The underlying impetus driving this research endeavour resides in the dissemination of invaluable knowledge, poised to simplify and amplify the process of quantum application development. By doing so, it is envisaged that this body of knowledge will afford practitioners enhanced convenience and efficacy in their pursuit of realising the potential of QC.

## 5 | ANALYSIS AND DISCUSSION

This section presents diverse analyses of the challenges that hinder in the adoption of QC in the software industry, gathered through a MLR. These analyses consider various characteristics such as continent/country, methodologies mentioned in the

**TABLE 13** Practices for challenge # 9: Huge dev, deployment, maintenance, operation and sustainability cost.

| S. No. | Practice (total = 3) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 9.1 | It is important for individuals involved in the QSE lifecycle, such as algorithm designers and code developers, to have a good understanding of the software tool chains used in the process | 30 | (1) |
| P# 9.2 | Technical specialists are required to create a QC-friendly atmosphere and a robust ecosystem that can transcend organisational boundaries | 38 | (1) |
| P# 9.3 | As QC computing is a rapidly evolving field, it requires new engineering methodologies to address issues such as unforeseen changes in requirements and restricted budgets, which are often encountered in scientific software projects | 61 | (1) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing; QSE, Quantum Software Engineering.

**TABLE 14** Practices for challenge # 10: Lack of quantum computing API & UI.

| S. No. | Practice (total = 1) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 10.1 | The focus of the research community should be on developing integrated development environment which is commercially viable and can aid in the implementation of quantum software | 63 | (1) |

Abbreviations: API, application programming interface; MLR, multi-vocal literature review; UI, user interface.

**TABLE 15** Practices for challenge # 11: Quantum domain complexity.

| S. No. | Practice (total = 1) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 11.1 | It is recommended to develop a readiness model to support software development organisations in evaluating, modifying, and enhancing their workflow as they transition from the conventional to the quantum software development paradigm. The preparedness model will consist of the assessment portion, which includes factors such as process areas, challenges, and enablers as well as the rules portion | 47 | (1) |

Abbreviation: MLR, multi-vocal literature review.

**TABLE 16** Practices for challenge # 12: Scalability, portability, integration and interoperability issues.

| S. No. | Practice (total = 1) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 12.1 | Quipper is a declarative, higher-order quantum programming language designed for usability, scalability, and correctness, and addressing the shortage of practical quantum programming languages. It has a monadic operational semantics, and offers scalability, expressiveness, and a sound theoretical foundation | 13 | (1) |

Abbreviation: MLR, multi-vocal literature review.

**TABLE 17** Practices for challenge # 13: Lack of roadmap for migration from classical to QC.

| S. No. | Practice (total = 2) | MLR (%) $N = 49$ Paper ID | Frequency |
|---|---|---|---|
| P# 13.1 | To ensure a successful transition from classic to quantum software development, practitioners need a dependable roadmap and guidelines. The readiness model serves as a crucial tool for software development organisations to aid in this process | 47 | (1) |
| P# 13.2 | In order to facilitate requirement engineering, particularly in integrating quantum mechanics into requirement analysis, the development of use cases, and the corresponding UML models, there is a need for novel and effective hybrid guidelines | 63 | (1) |

Abbreviations: MLR, multi-vocal literature review; QC, quantum computing; UML, unified modeling language.

research papers, publication's venue and time/period of publication. The objective of these studies is to investigate the consistency of the identified success variables across different continents, research methods, venue and period wherein the papers published techniques, in order to explore any variations.

We conducted four distinct analyses on the factors, each aiming to examine them from a unique perspective. To analyse the data, we utilised a linear-by-linear chi-square test (LBL-CST) in SPSS, considering the ordinal nature of the collected data. Research literature suggests that this test is preferred and more effective than the Pearson chi-square test when assessing differences between ordinal variables. The following theory was tested:

- Null hypothesis (H0): There is no statistically significant difference in the occurrence of communication and coordination issues across different study approaches employed for a specific task.
- Alternative hypothesis (H1): There is a statistically significant difference in the occurrence of communication and coordination issues among the different study approaches employed for a specific subject. If the *p*-value exceeds 0.05, we will accept the null hypothesis (H0) and focus on studying the problems clients encounter in quantum software development. Conversely, if the *p*-value is equal to or <0.05, we reject the null hypothesis (H0) and consider the alternative hypothesis (H1) as supported.

## 5.1 | Continent-based analysis of the identified challenges

In our study, the analysis of our research indicates that the selected articles were sourced from diverse continents. The study involves a comparison of challenges identified in Asia, Europe, and North America by examining 49 research papers. Table 18 and the visual representation mentioned in Figure 5 displays the distribution of these papers, among them 5 selected from Asia, 25 from Europe, 18 from North America and only one from other continents. The primary objective of this study is to explore potential variations in these challenges among different continents. By understanding similarities and differences, developers and organisations can effectively address these challenges in specific regions. To identify significant differences across the three continents, the study employed the linear-by-linear association chi-square test. The analysis reveals that the challenges exhibit more similarities than differences among these continents. The only notable difference found is the 'Lack of a roadmap for migration from classical to QC'.

The most common challenges identified across all three continents are the 'Lack of quantum-specific algorithms, development, and testing methodologies', 'Lack of professional experts', and 'Challenging design and validation process'.

In Europe and North America, additional common challenges includes 'Lack of development tools and technology', 'Limited development environment & resource availability', and 'Huge development, deployment, maintenance, operation, and sustainability costs'. Table 18 highlights that organisations in North America and Europe face challenges such as 'Scalability, portability, integration, and interoperability issues', 'Huge development, deployment, maintenance, operation, and sustainability costs', and 'Lack of management support and incorporation of QC in the curriculum'. In the software industry of North America, challenges also include 'Lack of development guidelines and Quality Assurance Standards', 'Lack of professional experts', and 'Limited development environment and resource availability'. On the other hand, in Europe, the most common challenges for software development organisations are 'Huge development, deployment, maintenance, operation, and sustainability costs', 'Challenging design and validation process', and 'Scalability, portability, integration, and interoperability issues'. Meanwhile, the adoption of QC in Asia presents the challenge of 'Lack of quantum-specific algorithms, development, and testing methodologies'.

## 5.2 | Methodology-based analysis of the identified factors

The challenges have been categorised into four commonly employed research methodologies as outlined in Table 19. Methodologies consist upon experiments, ordinary literature review (OLR); survey/case study/empirical study, and others, which encompass websites, reports, and exploratory studies mentioned graphically in Figure 6. Based on our examination of 49 research papers, we observed that OLR was the most frequently employed methodology, occurring 16 times. The second most utilised methodology was survey, case study, and empirical study, with a frequency of 13. Experiments were conducted in 9 papers, while other methodologies were employed in 11 papers.

To address Research Question-1 (RQ-1), we have examined the identified challenges using a range of research methodologies, including experiments, case studies, empirical research, surveys, interviews and questionnaires. A Linear-by-Linear Chi-Square test was conducted to determine if there were any significant differences among the challenges identified through these various research strategies/methodologies.

Our analysis revealed a single significant difference, which pertained to the challenge of 'Challenging Design and validation process'. The first three challenges, namely 'Lack of quantum-specific algorithms, development, and testing methodologies', 'Difficult compilation and debugging', and 'Lack of development tools and technology', were identified as critical across all utilised methodologies.

Our findings indicate that a total of 13 critical challenges were identified through the literature review. However, certain methodologies did not encompass all these critical challenges, as their frequencies in specific challenges were 0. Challenges with a frequency of 40% or higher were considered common challenges, including 'Lack of quantum-specific algorithms, development, and testing methodologies', reported by all four groups of methodologies in Table 19.

**TABLE 18** Continent based analysis of challenges.

| Continent base analysis of challenges | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Asia ($N = 5$) | | Europe ($N = 25$) | | North America ($N = 18$) | | Chi-square test (linear-by-linear association) $\alpha = 0.05$, df = 1 | |
| **S. No.** | **Challenges** | $F$ | % | $F$ | % | $F$ | % | $\chi^2$ | $P$ |
| 1 | Lack of quantum specific algorithms, dev and testing methodology | 4 | 80 | 15 | 60 | 12 | 67 | 0.417 | 0.518 |
| 2 | Difficult compilation and debugging | 0 | 0 | 6 | 24 | 3 | 17 | 0.017 | 0.895 |
| 3 | Lack of development tools and technology | 0 | 0 | 6 | 24 | 5 | 28 | 0.669 | 0.413 |
| 4 | Lack of development guidelines and Quality Assurance Standards | 0 | 0 | 2 | 8 | 6 | 33 | 0.065 | 0.800 |
| 5 | Lack of professional expert | 1 | 20 | 7 | 28 | 6 | 33 | 0.109 | 0.741 |
| 6 | Lack of management support and incorporation of quantum computing in the curriculum | 1 | 20 | 5 | 20 | 2 | 11 | 0.672 | 0.412 |
| 7 | Challenging design and validation process | 1 | 20 | 10 | 40 | 5 | 28 | 0.122 | 0.727 |
| 8 | Limited development environment and resources availability | 0 | 0 | 7 | 28 | 6 | 33 | 0.2955 | 0.086 |
| 9 | Huge dev, deployment, maintenance, operation and sustainability cost | 1 | 20 | 11 | 44 | 4 | 22 | 0.716 | 0.397 |
| 10 | Lack of quantum computing API & UI | 1 | 20 | 3 | 12 | 4 | 12 | 0.097 | 0.755 |
| 11 | Quantum domain complexity | 0 | 0 | 5 | 20 | 3 | 17 | 0.097 | 0.755 |
| 12 | Scalability, portability, integration and interoperability issues | 1 | 20 | 7 | 28 | 2 | 11 | 0.812 | 0.451 |
| 13 | **Lack of roadmap for migration from classical to QC** | **1** | **20** | **2** | **8** | **1** | **6** | **3.934** | **0.047** |

Abbreviations: API, application programming interface; QC, quantum computing; UI, user interface.

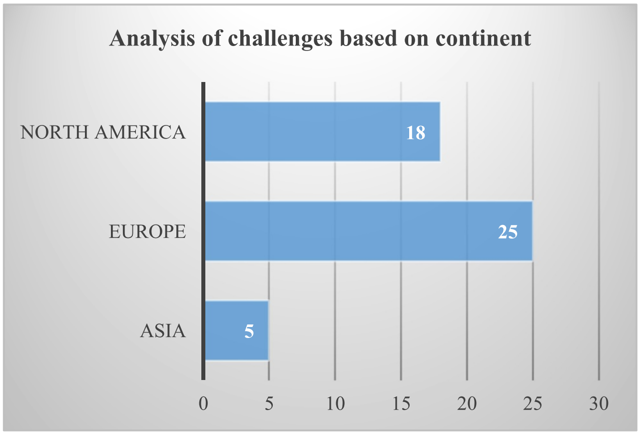


**FIGURE 5** Continent based analysis of challenges.

The challenges among 13 with frequencies exceeding 40% are considered common challenges. The context of the experiment methodology were 'Lack of quantum-specific algorithms, development, and testing methodologies' (78%), 'Challenging Design and validation process' (56%), 'Lack of development tools and technology' (44%), and 'Scalability, portability, integration, and interoperability issues' (44%).

## 5.3 | Time-based analysis

Table 20 presents a categorised compilation of identified challenges based on specific time periods. In order to gain insights into the range of challenges during distinct intervals, we examined two time periods: 2010–2015 and 2016–2023, and analysed the challenges identified within each period as shown in the chart mentioned in Figure 7. The linear-by-linear association chi-square test was utilised to assess if any significant differences existed among the challenges identified through the various research strategies and methodologies mentioned earlier.

Our analysis unveiled two notable differences related to the challenges of 'Challenging Design and Validation Process' and 'Limited Development Environment and Resource Availability'. Out of the 49 selected research papers, only 4 fell within the 2010–2015 timeframe, while 45 were published during the 2016–2023 period mentioned in chart in Figure 7. This indicates that QC remains a relatively new and emerging field of research, warranting further investigation and exploration.

Based on the criteria used to identify critical challenges, we have identified a total of 13 challenges. The challenge of 'Lack of quantum-specific algorithms, development, and testing methodologies' emerged as the most critical challenge in both periods, with frequencies of 75 and 62, respectively. However, it is noteworthy that the frequency of this challenge decreased in the second period, suggesting that it has garnered more attention from researchers and practitioners. Similarly, the challenge of 'Lack of development tools and technology', with frequencies of 50 and 20 in the two periods, indicates that this area has also received attention from researchers and developers. However, its higher frequency implies that it remains a critical challenge that necessitates solutions for wider adoption of QC.

In Table 20, Challenge number 5, which is the 'Lack of development guidelines and Quality Assurance Standard exhibits a frequency of 0 and 18 in the respective time periods'.

The challenge 'Lack of professional experts' has a frequency 25 in 2010–2015 period and 29 in 2016–2023 period. This analysis indicates that as research and practical work expanded during the second period, the demand for professional experts and the need for guidelines and quality assurance standards also increased over time. If these challenges are not effectively addressed on time then they could become more critical in the adoption of QC in the software industry.

The challenges 'Challenging Design and validation process' (100 and 29), 'Lack of development tools and technology' (50 and 20), and 'Scalability, portability, integration, and interoperability issues' (75 and 18) in both time periods indicate that they were significant challenges in the initial period but were significantly and actively addressed in the later period. This suggests that organisations recognised the importance of these challenges and made efforts to tackle them. However, as time progresses, these challenges will require continued attention and focus.

Certain challenges, such as 'Difficult compilation and debugging' (0 and 20) and 'Huge Dev, deployment, maintenance, operation, and sustainability cost' (25 and 33), are considered critical issues in both time periods and continue to present obstacles in the implementation of QC.

## 5.4 | Analysis of the identified challenges based on the venue of the publications

Table 21 presents an analysis of the publication outlets examined in our study. Out of the 49 selected papers used for data extraction through MLR, we found that these papers were published in various channels, including journals, workshops, conferences, and symposiums. To facilitate analysis, we categorised these channels into three groups: conferences, journals/workshops/symposiums, and 'other' (referring to GL, reports, website/webpages, and white papers). Table 21 indicates that the majority of our sample papers were published in conferences, followed by journals, workshops, and symposiums.

Based on the distribution of papers across these three channel groups, we identified several challenges. Our findings indicate that 22 papers in our study were related to conferences, 13 were related to journals/workshops/symposiums, and 14 fell into the 'other' category. We conducted a Linear-by-Linear Chi-Square test to identify significant differences among these challenges and found only one significant difference related to the 'lack of development guidelines and quality assurance standards'. According to our analysis shown in Table 21, the challenges of 'Lack of quantum specific algorithms, dev and testing methodologies' (73, 54 and 57), 'Difficult compilation and debugging' (23, 15 and 14), and 'Lack of development tools and technology' were found in research papers published in all

**TABLE 19** Methodology based analysis of challenges.

| Methodology based analysis of challenges | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Experiment (*N* = 9)** | | **OLR (*N* = 16)** | | **Survey + case study + empirical (*N* = 13)** | | **Other (*N* = 11)** | | **Chi-square test (linear-by-linear association) $\alpha = 0.05$, df = 1** | |
| **S. No.** | **Challenges** | ***F*** | **%** | ***F*** | **%** | ***F*** | **%** | ***F*** | **%** | $\chi^2$ | ***P*** |
| 1. | Lack of quantum specific algorithms, dev and testing methodologies | 7 | 78 | 10 | 63 | 8 | 62 | 6 | 55 | 0.960 | 0.327 |
| 2. | Difficult compilation and debugging | 3 | 33 | 2 | 13 | 2 | 15 | 2 | 18 | 0.395 | 0.530 |
| 3. | Lack of development tools and technology | 4 | 44 | 3 | 19 | 2 | 15 | 2 | 18 | 1.587 | 0.208 |
| 4. | Lack of development guidelines and Quality Assurance Standards | 3 | 33 | 3 | 19 | 2 | 15 | 0 | 0 | 3.779 | 0.052 |
| 5. | Lack of professional expert | 3 | 33 | 4 | 25 | 5 | 38 | 2 | 18 | 0.188 | 0.656 |
| 6. | Lack of management support and incorporation of quantum computing in the curriculum | 0 | 0 | 3 | 19 | 4 | 31 | 1 | 9 | 0.423 | 0.515 |
| 7. | **Challenging design and validation process** | **5** | **56** | **6** | **38** | **5** | **38** | **1** | **9** | **4.082** | **0.043** |
| 8. | Limited development environment and resources availability | 3 | 33 | 6 | 38 | 2 | 15 | 3 | 27 | 0.542 | 0.461 |
| 9. | Huge dev, deployment, maintenance, operation and sustainability cost | 3 | 33 | 6 | 38 | 5 | 38 | 2 | 18 | 0.529 | 0.467 |
| 10. | Lack of quantum computing API & UI | 2 | 22 | 2 | 13 | 3 | 23 | 1 | 9 | 0.213 | 0.645 |
| 11. | Quantum domain complexity | 3 | 33 | 3 | 19 | 2 | 15 | 0 | 0 | 3.779 | 0.052 |
| 12. | Scalability, portability, integration and interoperability issues | 4 | 44 | 5 | 31 | 4 | 31 | 3 | 27 | 1.212 | 0.201 |
| 13. | Lack of roadmap for migration from classical to QC | 3 | 33 | 5 | 31 | 5 | 38 | 1 | 9 | 1.081 | 0.299 |

Abbreviations: API, application programming interface; OLR, ordinary literature review; QC, quantum computing; UI, user interface.

three categories of publications, making them the most common challenges.

To compare these challenges between the categories of conferences and journals/workshops/symposiums, the most critical challenges reflected in most of the research papers are 'Lack of development tools and technology' (27, 23), 'Lack of development guidelines and Quality Assurance Standards' (27, 15), 'Lack of professional experts' (32, 31), 'Challenging Design and validation process' (32, 62), 'Huge Dev, deployment, maintenance, operation, and sustainability cost' (41, 38), and 'Lack of roadmap for migration from classical to QC' (27, 38).

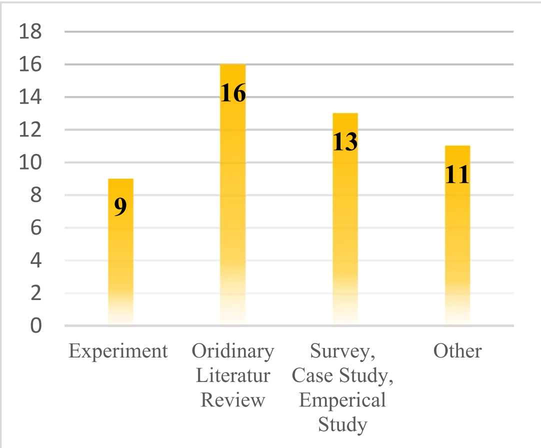


**FIGURE 6** Methodology based analysis of challenges.

## 6 | CONCLUSION AND FUTURE WORK

In this research, we performed a comprehensive review of existing white and grey literatures, known as a MLR, with the aim to identify the range of factors and challenges that hinder in the adoption of QC in software industry. Furthermore, we explored different practice and strategies to tackle these challenges. To address the research question RQ-1, we outlined a total of 13 challenges encountered by organisations during the adoption of QC. Subsequently, we examined these challenges deeply and determined that 5 of them were most critical. The critical challenges are the 'Lack of quantum specific algorithms, dev and testing methodologies', 'Difficult compilation and debugging', 'Lack of development tools and technology', 'Lack of development guidelines & Quality Assurance Standards' and 'Lack of professional expert', together founding over 30% of occurrences. Moreover, we evaluated these challenges from

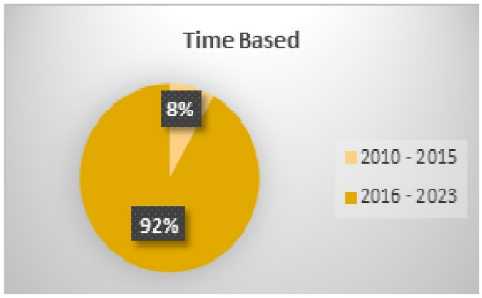


**FIGURE 7** Time base analysis.

**TABLE 20** Time based analysis of challenges.

| Time based analysis of challenges | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 2010–2015 ($N = 4$) | | 2016–2023 ($N = 45$) | | Chi-square test (linear-by-linear association) $\alpha = 0.05$, df = 1 | |
| **S. No.** | **Challenges** | ***F*** | **%** | ***F*** | **%** | $\chi^2$ | ***P*** |
| 1. | Lack of quantum specific algorithms, dev and testing methodologies | 3 | 75 | 28 | 62 | 0.253 | 0.615 |
| 2. | Difficult compilation and debugging | 0 | 0 | 9 | 20 | 0.960 | 0.327 |
| 3. | Lack of development tools and technology | 2 | 50 | 9 | 20 | 1.860 | 0.173 |
| 4. | Lack of development guidelines and Quality Assurance Standards | 0 | 0 | 8 | 18 | 0.833 | 0.362 |
| 5. | Lack of professional expert | 1 | 25 | 13 | 29 | 0.027 | 0.870 |
| 6. | Lack of management support and incorporation of QC in the curriculum | 0 | 0 | 8 | 18 | 0.833 | 0.362 |
| 7. | **Challenging design and validation process** | **4** | **100** | **13** | **29** | **8.031** | **0.005** |
| 8. | **Limited development environment and resources availability** | **3** | **75** | **11** | **24** | **4.507** | **0.034** |
| 9. | Huge dev, deployment, maintenance, operation and sustainability cost | 1 | 25 | 15 | 33 | 0.114 | 0.736 |
| 10. | Lack of quantum computing API & UI | 2 | 50 | 6 | 13 | 3.541 | 0.060 |
| 11. | Quantum domain complexity | 1 | 25 | 7 | 16 | 0.235 | 0.628 |
| 12. | Scalability, portability, integration and interoperability issues | 3 | 75 | 8 | 18 | 1.561 | 0.152 |
| 13. | Lack of roadmap for migration from classical to QC | 0 | 0 | 14 | 31 | 1.707 | 0.191 |

Abbreviations: API, application programming interface; QC, quantum computing; UI, user interface.

**TABLE 21** Publication venue based analysis of challenges.

| Publication venue based analysis | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Conference ($N = 22$) | | Journal workshop/ symposium ($N = 13$) | | Others ($N = 14$) | | Chi-square test (linear-by-linear association) $\alpha = 0.05$, df = 1 | |
| S. No. | Challenges | *F* | % | *F* | % | *F* | % | $\chi^2$ | *P* |
| 1 | Lack of quantum specific algorithms, dev and testing methodologies | 16 | 73 | 7 | 54 | 8 | 57 | 1.049 | 0.306 |
| 2 | Difficult compilation and debugging | 5 | 23 | 2 | 15 | 2 | 14 | 0.441 | 0.507 |
| 3 | Lack of development tools and technology | 6 | 27 | 3 | 23 | 2 | 14 | 0.788 | 0.375 |
| 4 | Lack of development guidelines and Quality Assurance Standards | **6** | **27** | **2** | **15** | **0** | **0** | **4.554** | **0.033** |
| 5 | Lack of professional expert | 7 | 32 | 4 | 31 | 3 | 21 | 0.407 | 0.524 |
| 6 | Lack of management support and incorporation of quantum computing in the curriculum | 6 | 27 | 1 | 8 | 1 | 7 | 2.820 | 0.093 |
| 7 | Challenging design and validation process | 7 | 32 | 8 | 62 | 2 | 14 | 0.617 | 0.432 |
| 8 | Limited development environment and resources availability | 7 | 32 | 4 | 31 | 3 | 21 | 0.407 | 0.524 |
| 9 | Huge dev, deployment, maintenance, operation and sustainability cost | 9 | 41 | 5 | 38 | 2 | 14 | 2.472 | 0.116 |
| 10 | Lack of quantum computing API & UI | 3 | 14 | 4 | 31 | 1 | 7 | 0.100 | 0.752 |
| 11 | Quantum domain complexity | 5 | 23 | 1 | 8 | 2 | 14 | 0.593 | 0.441 |
| 12 | Scalability, portability, integration and interoperability issues | 3 | 14 | 4 | 31 | 1 | 7 | 0.100 | 0.752 |
| 13 | Lack of roadmap for migration from classical to QC | 6 | 27 | 5 | 38 | 3 | 21 | 0.071 | 0.790 |

Abbreviations: API, application programming interface; QC, quantum computing; UI, user interface.

various perspectives, including time frame, methodology, geographical region, and publication platform. To answer the research question RQ-2, we discovered a total of 53 practices from published as well as from GL. The identification of these challenges will help the software industry to tackle them seriously. Moreover, the industry may adapt the practices that we identified for addressing these challenges.

Our research was limited only to literatures review and this enabled us to light on various challenges and practices/solutions. However, our knowledge about the challenges faced by organisation/industries in practice while adopting QC is deficient. It is our future plan to carry out an empirical study and overcome this deficiency. Existing literature suggests a shortage of development tools and technologies for the practical implementation of QC, along with lack of suitable quantum SDLC and models, which are largely non-existent. Consequently, our future plans also involve the development of a model for quantum software development life cycles to cover these gaps.

## AUTHOR CONTRIBUTIONS

**Masaud Salam**: Conceptualization; formal analysis; investigation; visualization; writing – original draft; writing – review & editing. **Muhammad Ilyas**: Conceptualization; methodology; project administration; supervision; validation.

## CONFLICT OF INTEREST STATEMENT

We have no conflict of interest.

## DATA AVAILABILITY STATEMENT

Data will be available on request.

## ORCID

*Muhammad Ilyas* https://orcid.org/0000-0003-2531-6485

## REFERENCES


1. Weigold, M., et al.: Encoding patterns for quantum algorithms. IET Quant. Commun. 2(4), 141–152 (2021). https://doi.org/10.1049/qtc2.12032
2. Klein, M.J.: Max Planck and the beginnings of the quantum theory. Arch. Hist. Exact Sci. 1(5), 459–479 (1961). https://doi.org/10.1007/bf00327765
3. Heisenberg, W.: The development of the interpretation of the quantum theory. In: Niels Bohr and the Development of Physics, vol. 12 (1955)
4. Hassija, V., et al.: Forthcoming applications of quantum computing: peeking into the future. IET Quant. Commun. 1(2), 35–41 (2020). https://doi.org/10.1049/iet-qtc.2020.0026
5. Moguel, E., et al.: A roadmap for quantum software engineering: applying the lessons learned from the classics. In: Q-SET@ QCE, pp. 5–13 (2020)
6. Whurley: COMMUNITY LIBRARY, explore quantum computing insights, experiments, projects, and code. https://quantumcomputing.com/whurley/7-things-you-need-to-know-about-qubits (2017). Accessed 01 Oct 2021
7. Zanni Junior, P.E., Vieira de Camargo, V.: A systematic mapping on quantum software development in the context of software engineering. arXiv e-prints, arXiv: 2106.00926 (2021)
8. Piattini, M., Peterssen, G., Pérez-Castillo, R.: Quantum computing: a new software engineering golden age. Software Eng. Notes 45(3), 12–14 (2020). https://doi.org/10.1145/3402127.3402131
9. Hassija, V., et al.: Present landscape of quantum computing. IET Quant. Commun. 1(2), 42–48 (2020). https://doi.org/10.1049/iet-qtc.2020.0027
10. Chagneau, A., et al.: Quantum finite difference solvers for physics simulation. IET Quant. Commun. 4(2), 70–83 (2023). https://doi.org/10.1049/qtc2.12054
11. Hewitt-Horsman, C.: An introduction to many worlds in quantum computation. Found. Phys. 39(8), 869–902 (2009). https://doi.org/10.1007/s10701-009-9300-2
12. Martonosi, M., Roetteler, M.: Next steps in quantum computing: computer science's role. *arXiv preprint arXiv:1903.10541* (2019)
13. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Phys. Rev. Lett. 113(13), 130503 (2014). https://doi.org/10.1103/physrevlett.113.130503
14. Mosca, M.: Cybersecurity in an era with quantum computers: will we be ready? IEEE Secur. Priv. 16(5), 38–41 (2018). https://doi.org/10.1109/msp.2018.3761723
15. Biamonte, J., et al.: Quantum machine learning. Nature 549(7671), 195–202 (2017). https://doi.org/10.1038/nature23474
16. Guerreschi, G.G., Smelyanskiy, M.: Practical optimization for hybrid quantum-classical algorithms. *arXiv preprint arXiv:1701.01450* (2017)
17. McArdle, S., et al.: Quantum computational chemistry. Rev. Mod. Phys. 92(1), 015003 (2020). https://doi.org/10.1103/revmodphys.92.015003
18. Grimsley, H.R., et al.: An adaptive variational algorithm for exact molecular simulations on a quantum computer. Nat. Commun. 10, 1–9 (2019). https://doi.org/10.1038/s41467-019-10988-2
19. Childs, A.M., et al.: Toward the first quantum simulation with quantum speedup. Proc. Natl. Acad. Sci. USA 115(38), 9456–9461 (2018). https://doi.org/10.1073/pnas.1801723115
20. Miranskyy, A., et al.: Quantum computing for software engineering: prospects. In: Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering, pp. 22–25 (2022)
21. Piattini, M., Peterssen, G., Pérez-Castillo, R.: Quantum computing: a new software engineering golden age. Software Eng. Notes 45(3), 12–14 (2021). https://doi.org/10.1145/3402127.3402131
22. Schradle, N.: In algorithms we trust: magical thinking, super intelligent Ai and quantum computing: with Emanuelle Burton, "The nuts and bolts of transformation: science fiction's imagined technologies and the civic imagination"; Michelle A. Marvin, "Memory altering technologies and the capacity to forgive: Westworld and Volf in Dialogue"; Nathan Schradle,"In algorithms we trust: magical thinking, superintelligent AI, and quantum computing"; and Zhange Ni,"Reimagining Daoist alchemy, decolonizing transhumanism: the fantasy of immortality cultivation in twenty-first century China". Zygon® 55(3), 733–747 (2020). https://doi.org/10.1111/zygo.12637
23. Gill, S.S., et al.: Quantum computing: a taxonomy, systematic review and future directions. Software Pract. Ex. 52(1), 66–114 (2022). https://doi.org/10.1002/spe.3039
24. Bhavin, M., et al.: Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications. J. Inf. Secur. Appl. 56, 102673 (2021). https://doi.org/10.1016/j.jisa.2020.102673
25. Ilyas, M., Khan, S.U.: Software integration in global software development: challenges for GSD vendors. J. Software Evol. Process 29(8), e1875 (2017). https://doi.org/10.1002/smr.1875
26. Egger, D.J., et al.: Quantum computing for finance: state-of-the-art and future prospects. IEEE Trans. Quant. Eng. 1, 1–24 (2020). https://doi.org/10.1109/tqe.2020.3030314
27. Awan, U., et al.: Quantum computing challenges in the software industry. A fuzzy AHP-based approach. Inf. Software Technol. 147, 106896 (2022). https://doi.org/10.1016/j.infsof.2022.106896
28. Pérez-Castillo, R., Serrano, M.A., Piattini, M.: Software modernization to embrace quantum technology. Adv. Eng. Software 151, 102933 (2021). https://doi.org/10.1016/j.advengsoft.2020.102933
29. Purohit, A., et al.: Building a quantum-ready ecosystem. IET Quant. Commun. 5, 1–18 (2024). https://doi.org/10.1049/qtc2.12072
30. Al-Matouq, H., et al.: A maturity model for secure software design: a multivocal study. IEEE Access 8, 215758–215776 (2020). https://doi.org/10.1109/access.2020.3040220
31. Zhao, J.: Quantum software engineering: Landscapes and horizons. *arXiv preprint arXiv:2007.07047* (2020)
32. Gyongyosi, L., Imre, S.: A survey on quantum computing technology. Computer Sci. Rev. 31, 51–71 (2019). https://doi.org/10.1016/j.cosrev.2018.11.002
33. Weder, B., et al.: The quantum software lifecycle. In: Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software, pp. 2–9 (2020)
34. Garhwal, S., Ghorani, M., Ahmad, A.: Quantum programming language: a systematic review of research topic and top cited languages. Arch. Comput. Methods Eng. 28(2), 289–310 (2021). https://doi.org/10.1007/s11831-019-09372-6
35. Piattini, M., et al.: Toward a quantum software engineering. IT Prof. 23(1), 62–66 (2021). https://doi.org/10.1109/mitp.2020.3019522
36. Akbar, M.A., et al.: Quantum software engineering: a new Genre of computing. *arXiv preprint arXiv:2211.13990* (2022)
37. de la Barrera, A.G., et al.: Quantum software testing: current trends and emerging proposals. In: Quantum Software Engineering, pp. 167–191. Springer (2022)
38. De Stefano, M., et al.: Software engineering for quantum programming: how far are we? J. Syst. Software 190, 111326 (2022). https://doi.org/10.1016/j.jss.2022.111326
39. Moguel, E., et al.: Quantum service-oriented computing: current landscape and challenges. Software Qual. J. 30(4), 983–1002 (2022). https://doi.org/10.1007/s11219-022-09589-y
40. Joan Camps, S.Q.S.: How Will Quantum Computing Change the World? https://www.riverlane.com/news/2021/03/how-will-quantum-computing-change-the-world/ (2021, 23/02/2022)
41. Hollebeek, T.: The Impact of Quantum Computing on Society. https://www.digicert.com/blog/the-impact-of-quantum-computing-on-society (2021, 23/02/2022)
42. Garousi, V., Felderer, M., Mäntylä, M.V.: The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–6 (2016)
43. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf. Software Technol. 106, 101–121 (2019). https://doi.org/10.1016/j.infsof.2018.09.006
44. Neto, G.T.G., et al.: Multivocal literature reviews in software engineering: preliminary findings from a tertiary study. In: 2019 ACM/IEEE

International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 1–6 (2019)
45. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Keele University (2007)
46. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. Inf. Software Technol. 64, 1–18 (2015). https://doi.org/10.1016/j.infsof.2015.03.007
47. Ilyas, M., et al.: Software integration model: an assessment tool for global software development vendors. J. Software Evol. Process 36(4), e2540 (2023). https://doi.org/10.1002/smr.2540
48. Antil, P.: Requirements engineering in global scaled agile software development environment: a multi-vocal literature review protocol. *arXiv preprint arXiv:2004.12647* (2020)
49. Afzal, W., Torkar, R., Feldt, R.: A systematic review of search-based testing for non-functional system properties. Inf. Software Technol. 51(6), 957–976 (2009). https://doi.org/10.1016/j.infsof.2008.12.005
50. Barbosa, L.S.: Software engineering for 'quantum advantage'. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, pp. 427–429 (2020)
51. N. A. P. W. DC: Quantum computing: progress and prospects (2019). https://doi.org/10.17226/25196
52. McCaskey, A., et al.: Hybrid programming for near-term quantum computing systems. In: 2018 IEEE International Conference on Rebooting Computing (ICRC), pp. 1–12 (2018)
53. E. NEWS: Quantum software testing: challenges, early achievements, and opportunities (2022)
54. Paltenghi, M., Pradel, M.: Bugs in quantum computing platforms: an empirical study. Proc. ACM Program. Lang. 6(OOPSLA1), 1–27 (2022). https://doi.org/10.1145/3527330
55. Shaydulin, R., Thomas, C., Rodeghero, P.: Making quantum computing open: lessons from open source projects. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, pp. 451–455 (2020)
56. Nodarse, G.P., Piattini Velthuis, M., Hevia, J.L.: The QPath blog: Practical quantum computing: challenges of quantum software development [English]. https://www.quantumpath.es/2022/01/31/practical-quantum-computing-challenges-of-quantum-software-development/ (2022). Accessed 31 Jan 2023
57. Möller, M., Vuik, C.: On the impact of quantum computing technology on future developments in high-performance scientific computing. Ethics Inf. Technol. 19(4), 253–269 (2017). https://doi.org/10.1007/s10676-017-9438-0
58. Wille, R., Hillmich, S., Burgholzer, L.: JKQ: JKU tools for quantum computing. In: Proceedings of the 39th International Conference on Computer-Aided Design, pp. 1–5 (2020)
59. Shukla, S.K., et al.: Nano, quantum, and molecular computing: are we ready for the validation and test challenges? In: Eighth IEEE International High-Level Design Validation and Test Workshop, pp. 3–7 (2003)
60. Miranskyy, A., Zhang, L.: On testing quantum programs. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), pp. 57–60 (2019)
61. Li, H., Khomh, F., Openja, M.: Understanding quantum software engineering challenges an empirical study on stack exchange forums and GitHub issues. In: 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 343–354 (2021)
62. Zhao, P., Zhao, J., Ma, L.: Identifying bug patterns in quantum programs. In: 2021 IEEE/ACM 2nd International Workshop on Quantum Software Engineering (Q-SE), pp. 16–21 (2021)
63. Metodi, T.S., et al.: A quantum logic array microarchitecture: scalable quantum data movement and computation. In: 38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05), pp. 12–318 (2005)
64. Akbar, M.A., Rafi, S., 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, pp. 563–573 (2022)
65. Ilyas, M., Khan, S.U.: Practices for software integration success factors in GSD environment. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6 (2016)

**How to cite this article:** Salam, M., Ilyas, M.: Quantum computing challenges and solutions in software industry—a multivocal literature review. IET Quant. Comm. 5(4), 462–485 (2024). https://doi.org/10.1049/qtc2.12096

## APPENDIX A

List of finally selected articles for MLR.

[1] M. Piattini, G. Peterssen, and R. Pérez-Castillo, "Quantum Computing: A New Software Engineering Golden Age," ACM SIGSOFT Software Engineering Notes, vol. 45, pp. 12–14, 2020.
[2] R. Shaydulin, C. Thomas, and P. Rodeghero, "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.
[3] A. Miranskyy, M. Khan, J. P. L. Faye, and U. C. Mendes, "Quantum computing for software engineering: prospects," in Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering, 2022, pp. 22–25.
[4] F. T. Chong, "Quantum computing is getting real: Architecture, PL, and OS roles in closing the gap between quantum algorithms and machines," in Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, 2018, pp. 285–285.
[5] M. Paltenghi and M. Pradel, "Bugs in quantum computing platforms: an empirical study," Proceedings of the ACM on Programming Languages, vol. 6, pp. 1–27, 2022.
[6] R. Wille, S. Hillmich, and L. Burgholzer, "JKQ: JKU tools for quantum computing," in Proceedings of the 39th International Conference on Computer-Aided Design, 2020, pp. 1–5.
[7] B. Weder, J. Barzen, F. Leymann, M. Salm, and D. Vietz, "The quantum software lifecycle," in Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software, 2020, pp. 2–9.
[8] M. A. Serrano, J. A. Cruz-Lemus, R. Perez-Castillo, and M. Piattini, "Quantum software components and platforms: Overview and quality assessment," ACM Computing Surveys, vol. 55, pp. 1–31, 2022.
[9] L. S. Barbosa, "Software engineering for quantum advantage," in Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, 2020, pp. 427–429.
[10] P. Zhao, J. Zhao, and L. Ma, "Identifying bug patterns in quantum programs," in 2021 IEEE/ACM 2nd

International Workshop on Quantum Software Engineering (Q-SE), 2021, pp. 16–21.
[11] S. Ali and T. Yue, "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.
[12] P. Das, S. S. Tannu, P. J. Nair, and M. Qureshi, "A case for multi-programming quantum computers," in Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019, pp. 291–303.
[13] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, "Quipper: a scalable quantum programming language," in Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation, 2013, pp. 333–342.
[14] S.-H. Hung, K. Hietala, S. Zhu, M. Ying, M. Hicks, and X. Wu, "Quantitative robustness analysis of quantum programs," Proceedings of the ACM on Programming Languages, vol. 3, pp. 1–29, 2019.
[15] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "A quantum logic array microarchitecture: Scalable quantum data movement and computation," in 38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05), 2005, pp. 12 pp.-318.
[16] H. Li, F. Khomh, and M. Openja, "Understanding Quantum Software Engineering Challenges An Empirical Study on Stack Exchange Forums and GitHub Issues," in 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2021, pp. 343–354.
[17] A. McCaskey, E. Dumitrescu, D. Liakh, and T. Humble, "Hybrid programming for near-term quantum computing systems," in 2018 IEEE international conference on rebooting computing (ICRC), 2018, pp. 1–12.
[18] P. B. Upama, M. J. H. Faruk, M. Nazim, M. Masum, H. Shahriar, G. Uddin, et al., "Evolution of quantum computing: A systematic survey on the use of quantum computing tools," in 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), 2022, pp. 520–529.
[19] R. Abreu, J. P. Fernandes, L. Llana, and G. Tavares, "Metamorphic testing of oracle quantum programs," in Proceedings of the 3rd International Workshop on Quantum Software Engineering, 2022, pp. 16–23.
[20] M. Schönberger, M. Franz, S. Scherzinger, and W. Mauerer, "Peel| pile? Cross-framework portability of quantum software," in 2022 IEEE 19th international conference on software architecture companion (ICSA-C), 2022, pp. 164–169.
[21] X. Wang, P. Arcaini, T. Yue, and S. Ali, "Application of combinatorial testing to quantum programs," in 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), 2021, pp. 179–188.
[22] A. Miranskyy and L. Zhang, "On testing quantum programs," in 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), 2019, pp. 57–60.
[23] P. Zhao, J. Zhao, and L. Ma, "Identifying bug patterns in quantum programs," in 2021 IEEE/ACM 2nd International Workshop on Quantum Software Engineering (Q-SE), 2021, pp. 16–21.
[24] A. Ahmad, A. A. Khan, M. Waseem, M. Fahmideh, and T. Mikkonen, "Towards process centered architecting for quantum software systems," in 2022 IEEE International Conference on Quantum Software (QSW), 2022, pp. 26–31.
[25] J. K. Vizzotto, "Quantum computing: State-of-art and challenges," in 2013 2nd Workshop-School on Theoretical Computer Science, 2013, pp. 9–13.
[26] S. Ali, P. Arcaini, X. Wang, and T. Yue, "Assessing the effectiveness of input and output coverage criteria for testing quantum programs," in 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST), 2021, pp. 13–23.
[27] M. Soeken, T. Haener, and M. Roetteler, "Programming quantum computers using design automation," in 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018, pp. 137–146.
[28] S. K. Shukla, R. Karri, S. C. Goldstein, F. Brewer, K. Banerjee, and S. Basu, "Nano, quantum, and molecular computing: Are we ready for the validation and test challenges?," in Eighth IEEE International High-Level Design Validation and Test Workshop, 2003, pp. 3–7.
[29] M. Möller and C. Vuik, "On the impact of quantum computing technology on future developments in high-performance scientific computing," Ethics and information technology, vol. 19, pp. 253–269, 2017.
[30] E. Moguel, J. Rojo, D. Valencia, J. Berrocal, J. Garcia-Alonso, and J. M. Murillo, "Quantum service-oriented computing: current landscape and challenges," Software Quality Journal, vol. 30, pp. 983–1002, 2022.
[31] M. A. Akbar, S. Rafi, and A. A. Khan, "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, 2022, pp. 563–573.
[32] X. Wang, P. Arcaini, T. Yue, and S. Ali, "Generating failing test suites for quantum programs with search," in International Symposium on Search Based Software Engineering, 2021, pp. 9–25.
[33] U. Awan, L. Hannola, A. Tandon, R. K. Goyal, and A. Dhir, "Quantum computing challenges in the software industry. A fuzzy AHP-based approach," Information and Software Technology, vol. 147, p. 106896, 2022.
[34] M. De Stefano, F. Pecorelli, D. Di Nucci, F. Palomba, and A. De Lucia, "Software engineering for quantum programming: How far are we?," Journal of Systems and Software, vol. 190, p. 111326, 2022.
[35] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, et al., "Quantum computing: A taxonomy,

systematic review and future directions," Software: Practice and Experience, vol. 52, pp. 66–114, 2022.
[36] J. Zhao, "Quantum software engineering: Landscapes and horizons," arXiv preprint arXiv: 2007.07047, 2020.
[37] A. G. de la Barrera, I. G.-R. de Guzmán, M. Polo, and J. A. Cruz-Lemus, "Quantum Software Testing: Current Trends and Emerging Proposals," in Quantum Software Engineering, ed: Springer, 2022, pp. 167–191.
[38] M. A. Akbar, A. A. Khan, S. Mahmood, and S. Rafi, "Quantum Software Engineering: A New Genre of Computing," arXiv preprint arXiv: 2211.13990, 2022.
[39] R. U. Shetty. (2000). Universal Group of Institutions Gateway to public service. Available: https://universal-institutions.com/the-challenges-of-quantum/
[40] J. Krupansky. (2018, 26.01.2024). The Greatest Challenges for Quantum Computing Are Hardware and Algorithms. Available: https://jackkrupansky.medium.com/the-greatest-challenges-for-quantum-computing-are-hardware-and-algorithms-c61061fa1210
[41] T. Yue. (2021, 21.01.2023). Quantum Software Testing: Challenges, Early Achievements, and Opportunities. Available: https://ercim-news.ercim.eu/en128/special/quantum-software-testing-challenges-early-achievements-and-opportunities
[42] J. Chow. (2022, 21.01.2023). State of Quantum Computing Building a Quantum Economy. Available: https://www3.weforum.org/docs/WEF_State_of_Quantum_Computing_2022.pdf
[43] Emily Grumbling and Mark Horowitz. (2019, 22.01.2023). Quantum Computing Progress and Prospects. Available: https://nap.nationalacademies.org/read/25196/chapter/8
[44] J. C. a. S. Stepney, "Proposed "Grand Challenge for Computing Research" Quantum Software Engineering," United Kingdom 30.09.2003 2003.
[45] D. V. Z. Chan, "Spotlight: Quantum Information Science and Technology," 26.08.2024 2020.
[46] C. d. A. Salil Gunashekar, et al, "Using Quantum Computers and Simulators in the Life Sciences Current Trends and Future Prospects," TAIWAN 05.10.2022 2022.
[47] M. Piattini. (2021, 23.01.2023). Quantum Software Engineering Challenges. Available: https://www.cutter.com/article/quantum-software-engineering-challenges
[48] M. Piattini. (2021, 25.01.2023). Requirements for a Robust Quantum Software Development Environment. Available: https://www.cutter.com/article/requirements-robust-quantum-software-development-environment
[49] M. P. V. a. J. L. H. Guido Peterssen Nodarse. (26.01.2023). Practical Quantum Computing: Challenges of Quantum Software Development. Available: https://www.quantumpath.es/2022/01/31/practical-quantum-computing-challenges-of-quantum-software-development/