

Systematic Review on Requirements Engineering in Quantum Computing: Insights and Future Directions

Samuel Sepúlveda ¹, Ania Cravero ^{1,*}, Guillermo Fonseca ² and Leandro Antonelli ^{3,4}

¹ Departamento de Ciencias de la Computación e Informática (DCI), Universidad de La Frontera, Temuco 4811230, Chile; samuel.sepulveda@ufrontera.cl

² Vicerrectoría de Investigación y Postgrado, Universidad de La Frontera, Temuco 4811230, Chile; g.fonseca02@ufromail.cl

³ Lifia, Facultad de Informática, Universidad Nacional de La Plata, Buenos Aires 1900, Argentina; lanto@lifia.info.unlp.edu.ar

⁴ CAETI, Facultad de Tecnología Informática, Universidad Abierta Interamericana, Buenos Aires 1900, Argentina

* Correspondence: ania.cravero@ufrontera.cl

Abstract: *Context:* Quantum software development is a complex and intricate process that diverges significantly from traditional software development. Quantum computing and quantum software are deeply entangled with quantum mechanics, which introduces a different level of abstraction and a deep dependence on quantum physical properties. The classical requirements engineering methods must be adapted to encompass the essential quantum features in this new paradigm. *Aim:* This study aims to systematically identify and analyze challenges, opportunities, developments, and new lines of research in requirements engineering for quantum computing. *Method:* We conducted a systematic literature review, including three research questions. This study included 105 papers published from 2017 to 2024. *Results:* The main results include the identification of problems associated with defining specific requirements for quantum software and hybrid system requirements. In addition, we identified challenges related to the absence of standards for quantum requirements engineering. Finally, we can see the advances in developing programming languages and simulation tools for developing software in hybrid systems. *Conclusions:* This study presents the challenges and opportunities in quantum computing requirements engineering, emphasizing the need for new methodologies and tools. It proposes a roadmap for future research to develop a standardized framework, contributing to theoretical foundations and practical applications.

Keywords: quantum computing; requirements engineering; software engineering; systematic review



Citation: Sepúlveda, S.; Cravero, A.; Fonseca, G.; Antonelli, L. Systematic Review on Requirements Engineering in Quantum Computing: Insights and Future Directions. *Electronics* **2024**, *13*, 2989. <https://doi.org/10.3390/electronics13152989>

Academic Editors: Ireneusz Miciuła, Łukasz Radliński, Aleksander Jarzębowski and Włodzimierz Wysocki

Received: 30 June 2024

Revised: 20 July 2024

Accepted: 26 July 2024

Published: 29 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Quantum computing (QC) is an emerging branch of computer technology that uses principles of quantum mechanics to perform computations. With the development of qubits and quantum gates, QC can solve complex problems much faster than traditional methods [1]. In this context, quantum software refers to applications and tools designed to interact with quantum infrastructure, allowing developers to take advantage of superposition, entanglement, and other features of quantum physics [2]. These unique features pose new challenges and opportunities for software engineers seeking to explore and exploit the capabilities of QC [3]. For instance, quantum software can be used in cryptography, optimization problems, and simulation of quantum systems. Thus, the authors of [4] explain that hybrid applications consisting of quantum and classical components require the development of appropriate quantum software architectures, which implies facing significant challenges throughout the software development lifecycle [5].

Quantum software development, in its essence, is a complex and intricate process that diverges significantly from traditional software development. Unlike conventional

software, quantum software is deeply intertwined with quantum mechanics, introducing a different level of abstraction and a profound reliance on quantum physical properties [2]. This unique characteristic of quantum software, where qubits can simultaneously represent multiple states and quantum operations are inherently probabilistic, poses a significant challenge for designers and implementers [6]. The challenge of quantum software engineering (QSE) lies in reworking and extending all classical software engineering (SE) to the quantum domain, enabling programmers to manipulate quantum programs as swiftly and confidently as they do with classical programs [7].

Challenges in QSE include creating specialized tools, designing software architectures adapted to quantum environments, and interpreting stochastic results [8]. One of the most complex challenges is designing and implementing software that can operate in a quantum environment while maintaining reliability and efficiency [1]. Error correction (due to the fragile nature of quantum states) and decoherence problems in quantum devices require innovative approaches and creative solutions. Furthermore, hardware limitations and the scarcity of quantum resources also complicate the quantum software development process. In this regard, Cartiere [9] explains that the characteristic difficulty of creating pure quantum software is mainly due to intermediate states' inaccessibility, making debugging virtually impossible. However, he argues that formal methods, which apply rigorous mathematical models to guarantee error-free software, can overcome this barrier. On the other hand, Barbosa [10] explains that it is difficult to design new quantum algorithms as quantum programming is complex. The same author mentions a greater need for adequate formal techniques in quantum software development.

Requirements engineering (RE) in the context of quantum software development poses unique challenges, as requirements must consider the inherent characteristics of QC [8,11]. The requirements analysis and specification process must address functional and non-functional aspects, considering overlap and entanglement [12]. However, it is crucial to thoroughly understand the application domain and how quantum algorithms can be applied to gain significant advantages over conventional software [5]. According to [11], the main challenges include defining requirements that encompass quantum and classical operations in hybrid applications, adapting to the stochastic nature of quantum results, and managing the integration between quantum and classical components.

Akbar et al. [13] underscore the pressing need for new modeling and specification techniques in RE for QC. These techniques should be tailored to the unique aspects of QC, such as logging functions and states. The authors stress that the RE research community must extend classical use cases, user stories, and goal modeling techniques to support the quantum RE process. This need arises from the unique characteristic of QC to operate on states that can represent multiple values simultaneously, which aligns poorly with traditional SE techniques and necessitates a renewed and specific approach to RE in this new field. This emphasis on the need for innovative approaches should inspire software engineers and developers, making them feel the importance of their work and the potential impact of their research.

No specific methods or models have been developed to help carry out the RE stage in QC. However, efforts have been made to advance definitions of functional and non-functional requirements [14]. The need for established approaches in this area indicates the need for research and development of tools and techniques to help software engineers define, document, and manage requirements in hybrid systems involving classical and quantum systems. Adapting and evolving RE methods to encompass quantum features are essential to developing software that can operate effectively in the quantum paradigm [13].

This study aims to identify and analyze the unique challenges systematically, advances, and further research lines in RE for QC software. The main research question is: *What are the challenges, opportunities, and future directions regarding requirements engineering for QC software?* Then, we review the current methodologies, projecting their evolution to meet the future demands of this rapidly developing field. It provides an insightful overview of cutting-edge RE practices tailored to QC, emphasizing the differences between traditional

and quantum software development. Additionally, it outlines strategic advances and suggests directions for further research, offering a comprehensive roadmap for navigating the complexities of this emerging paradigm. This study seeks to significantly enrich both the theoretical framework and practical applications in the field, serving as a crucial resource for researchers and practitioners.

The remainder of this paper is structured as follows. We present the QC and QSE background in Section 2. Section 3 presents some related work relative to quantum software development and quantum RE. Section 4 presents the methodology for the Systematic Literature Review (SLR). Sections 5 and 6 present the results and discussion, respectively. Finally, Section 7 presents the conclusions and future work.

2. Background

This section explains the basic concepts of QC and RE that support this work. Section 2.1 presents the main concepts of QC. Section 2.2 presents the milestones and current state of QC. We also discuss the main aspects of SE and Quantum SE in Section 2.3. Sections 2.4 and 2.5 present the importance of RE for software development and quality aspects, respectively. RE, the Quantum RE, and the unique aspects of QC on RE are presented in Sections 2.6 and 2.7, respectively. Finally, Section 2.8 presents a couple of case studies that demonstrate the practical application and solutions using QC, considering some aspects of RE.

2.1. Quantum Computing—General Concepts

QC originates from the desire to perform calculations orders of magnitude faster than a conventional computer using the principles of quantum mechanics. The idea of QC emerged in the early 1980s when Paul Benioff created the first model of quantum computation [15] and Tommaso Toffoli presented a quantum gate that further contributed to the idea of universal computation using quantum mechanics [16]. Furthermore, Richard Feynman suggested in a talk that quantum systems could efficiently simulate other quantum systems [17]. David Deutsch developed this idea by proposing a theoretical model of a universal quantum computer in 1985 [18].

QC, based on the principles of quantum mechanics, uses the properties of quantum systems to execute algorithms. Instead of using bits like classical computers that can represent either 0 or 1, it uses qubits that can exist in both states simultaneously, a phenomenon called superposition [19]. Additionally, two or more qubits can be correlated, a phenomenon called entanglement. Quantum algorithms use these features to have an advantage over classical computation, which may range from polynomial to exponential speedups [20].

2.2. Milestones and Current State in Quantum Computing

During the last few years, QC has made several theoretical and practical advances. The Deutsch and Jozsa algorithm [21], Simon's problem [22], Shor's algorithm have been used to find the prime factors of integer numbers [23], which was proven to have "exponential speedup" versus the classical solutions. The Grover's algorithm has been used to perform non-structure searches [24].

The new century began with experimental demonstrations of quantum algorithms and the constructions of working quantum computers: D-Wave [25] and IBM [26]. Arriving at the 2020 decade, efforts have intensified for challenges about scalability, coherence, and bug fixing, fostering academic research.

Nowadays, the situation faces promise and complexity. Some companies such as IBM, Rigetti, Google, and Microsoft continue pushing the limit of the development of hardware and software with milestones such as the QPU "Condor" of IBM, which has 1121 qubits [27]. Some commercialization efforts such as the Q System One of IBM [28] or the Quantum Cloud Computing Service Amazon Bracket [29] have the goal of democratizing access to quantum computer resources. Meanwhile, research focuses on tackling some essential challenges like bug fixing [30] and coherence time [3,31]. Nevertheless, some challenges

remain, like providing robust and fault-tolerant quantum systems and scalable architectures of qubits.

2.3. Software Engineering and Quantum Software Engineering

SE can be defined as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [32]. To meet the growing demands of users and customers, SE is the discipline in charge of considering all aspects related to the development of software products. SE also considers the management of projects and the resources involved [33].

The software is complex by essence [34] and nowadays is still more complex. The software crisis was caused by the rapid increases in computer power and the complexity of the problems that could be tackled; thus, the complexity of the software also increased, and many problems arose because existing methods were inadequate to deal with this complexity. Nowadays, we are facing a new software crisis if you consider the revolutions of technology 4.0, quantum software, and the interoperability of the software applications [35].

QC is a revolutionary computational architecture, requiring adaptations of existing SE concepts for its unique characteristics [36]. Current quantum software development is focused on low abstraction levels, highlighting the need for higher-level engineering support [37].

Quantum Software Engineering (QSE) is emerging to address QC's challenges, such as the complexity of quantum algorithms, qubits' probabilistic nature, and integration with existing technologies [5,10]. QSE involves designing, implementing, and maintaining quantum software, requiring new tools and programming languages tailored to QC [7].

Traditional software development methodologies are not easily applicable to QC, necessitating new approaches that consider quantum phenomena like superposition and entanglement [38]. Reliable quantum systems demand formal techniques for precise algorithm development and analysis. Despite advances in quantum software tools, a comprehensive toolchain for quantum software development is still developing [39].

Understanding the challenges of quantum programming is essential for the growth of QSE [38]. Developing reliable quantum algorithms is crucial given current technological limitations [40]. As QC evolves, new SE models and methods must address QC's unique features and challenges, spanning both technical and socio-technical areas [41].

2.4. Importance of Requirements Engineering in Software Development

Requirement engineering is one of the first stages in software development. Its goal is to identify the needs, wishes, and expectations of the software to be built. If the requirements are incorrect, the software system will be worthless since it will not provide the functionality necessary to satisfy its intended goal. Boehm [42] has estimated that repairing errors made at the requirement engineering stage can cost up to 200 times the initial amount once the software is delivered to the client. Although this stat was calculated more than 20 years ago, it is still true that after eliciting requirements, the software application should be designed, coded, tested, and delivered to the user. Thus, all that work should be redone. Agile methodologies mitigate this issue by proposing an iterative and incremental development process with short development cycles of about one or two weeks, reducing the rework caused by incorrect requirements.

2.5. Requirements Engineering and Software Quality

The importance of software quality in RE cannot be overstated, as it significantly impacts the overall success of software projects. Effective RE practices ensure that user needs are accurately captured and translated into high-quality software products, enhancing user satisfaction and project outcomes [43]. Empirical studies have demonstrated that well-executed RE processes contribute to gaining critical quality attributes (e.g., adequacy, completeness, and consistency) [44]. Moreover, a structured approach to RE, supported by quality models and metrics, facilitates the development of robust software systems [45].

Thus, integrating rigorous RE practices is pivotal for attaining superior software quality and successful project delivery [46].

The ISO/IEC 25010 standard provides a comprehensive framework for evaluating software quality [47]. This standard emphasizes essential attributes such as functionality, reliability, and maintainability, thereby supporting the development of high-quality software systems [48,49].

2.6. Requirements Engineering and Quantum Requirements Engineering

According to [50], the development of specific practices for quantum software testing is required, as well as other SE practices, such as RE, which are of less concern at the current stage due to the lack of real-world applications. Furthermore, the RE for QC will also differ from the classical counterpart, as mentioned in [14,51].

In [52], an initial identification and discussion of the new competencies required for effective RE practices in complex systems is presented. This work highlights how increased complexity imposed by QC impacts RE practices and the identification of critical viewpoints necessary for aligning RE practices with specific design problems.

Traditional requirements engineering processes are related to performing three main activities: eliciting (that is, capturing the knowledge), describing (that is, precisely specifying it), and validating (ensuring the high quality of the description according to user satisfaction and quality) [43]. These processes deal with three different types of information: knowledge about the domain (business rules), functional requirements, and quality attributes (nonfunctional requirements (e.g., adequacy, completeness, and consistency)) [44].

Quantum programming focuses on different types of solutions than traditional programming. Since traditional programming requires a huge effort to elicit knowledge and functional requirements, quantum programs require an effort to elicit quality attributes (nonfunctional requirements) because of the characteristics of the new hardware that support the execution of the software [52]. Thus, the following section describes the quality attributes specific to quantum programming.

2.7. The Unique Aspects of Quantum Computing That Impact Requirements Engineering

The no-go theorems and hardware restrictions impose non-functional requirements. In quantum mechanics, no-go theorems describe physically impossible situations, impacting quantum computer operations. Examples include the no-cloning [53], no-deleting [54], and no-programming theorems [55], which limit copying, deleting, and perfectly discriminating quantum states [56].

Different types of hardware impose limitations on quantum gates and qubit connections. For example, IBM quantum computers have specific connection maps dictating qubit interactions [57]. These constraints affect quantum algorithm development and optimization by determining the efficiency of implementing quantum operations [58,59]. An example of these constraints is the qubit routing problem and the solution proposed by [60].

Quantum error correction mitigates noise and errors in QC hardware. As quantum computers scale, maintaining qubit coherence becomes challenging due to environmental noise and hardware imperfections. Error correction codes are essential to protect quantum information from decoherence and errors. Implementing error correction requires balancing overhead and algorithm performance [61,62].

2.8. Quantum Computing and Requirements Engineering—Case Studies

Next, we present two case studies to show the practical application and solutions to QC's concrete problems and consider the principles of RE. The first case study considers integrating quantum computational primitives into existing software, exploring the use of quantum annealing to solve the Boolean satisfiability problem [63]. The second case study involves the structured application of quantum requirements engineering methodologies

to develop large-scale quantum applications [64]. Table 1 summarizes both case studies, indicating the RE methodologies or techniques used.

These case studies highlight the critical role of RE techniques in QC. By applying structured methodologies to define, model, and analyze requirements, robust and efficient quantum solutions can be developed. These examples provide valuable insights into the challenges and innovative solutions within the field of QRE, emphasizing its importance in advancing QC applications.

Table 1. QC-RE case studies.

Ref.	Summary	Elicitation	Modeling	Analysis
[63]	Integration of quantum annealing for solving Boolean satisfiability problem (SAT).	Identifying specific needs for integrating quantum annealing with SAT solving, including performance metrics (e.g., cost, timing, and constraints).	Transforming the SAT problem into a format suitable for quantum annealers, ensuring the problem's constraints and structure were appropriately represented.	Evaluating different ways of transforming SAT to quantum annealers and assessing their impact on performance.
[64]	Application of QRE methodologies for developing large-scale quantum applications.	Using innovative methodologies to gather and define requirements specific to QC applications.	Analyzing the unique characteristics of quantum software to address challenges in requirement elicitation, modeling, and analysis.	Developing models that capture the complexities of quantum software, ensuring that the requirements align with the intended functionalities and performance metrics.

3. Related Work

The following is a synthesis of secondary studies that discuss the main challenges, considerations and future directions for software development, QC and RE. Table 2 shows the overlapping for our RQs and the related work (✓: fully answered, ~: partially answered). Our RQs (RQ1, RQ2 and RQ3) are fully explained in Section 4.1.2. For details on these related works (goals, RQs, and results), see Appendix A. Notably, we have not identified any work that focuses on the same aspects mentioned in this article. Next, we present a summary of these works.

Table 2. Overlapping RQs for related work.

Ref.	RQ1	RQ2	RQ3
[8]	~	~	
[1]		~	

The study by Junior and de Camargo [8] explores programming infrastructures, fundamental differences with conventional software, and the suitability of quantum software for specific application domains. This study reviewed 24 papers between 2010 and 2021, highlighting the need for specific adaptations in each phase of the quantum software development lifecycle, especially in requirements analysis, where specific tools and methodologies are needed as the field evolves. The authors explain that when conducting a requirements analysis, it is crucial to identify in which application domains and under which non-functional requirements quantum software could bring significant benefits, considering the unique capabilities of QC. This requirements analysis implies a thorough understanding of the application domain and how the inherent features of QC, such as superposition and entanglement, can be exploited to gain advantages over conventional software. The authors highlight the necessity of new tools and methodologies for handling quantum-specific requirements in the requirements analysis phase. The design phase requires new or extended modeling languages like Q-UML and Quantum4BPMN. During implementation, develop-

ers must use quantum programming languages and frameworks like Quipper, Scaffold, Q#, and toolkits like Qiskit and PyQuil while familiar with quantum programming patterns. The testing phase faces significant challenges due to the complexity of quantum algorithms, necessitating new verification and validation approaches. Lastly, maintenance of quantum software involves additional complexities due to potential hardware and software errors, requiring strategies to preserve quality attributes. These adaptations across all phases ensure quantum software applications' efficient and reliable development.

On the other hand, Haghparast et al. [1] explain the challenges from the developers' perspective, proposing a specific workflow model for quantum software development. This study highlights the importance of integrating iterative development practices and formalisms to create reliable quantum applications. Formalisms include taxonomies that determine the superiority of QC. The authors address RE by highlighting the need to understand specific requirements to take profound advantage of QC. They highlight the use of quantum programming languages like Qiskit and Cirq and the extension of classical environments like Python with libraries for quantum development. Model transformations help bridge classical and quantum software engineering. The differences between conventional and quantum software development include the need for initial simulations, unique debugging challenges due to state collapse, and handling errors from both quantum algorithms and hardware, given the stochastic nature of quantum computations. Quantum software is particularly suited for optimization problems, quantum chemistry, material science, cryptography, and machine learning. Finally, the paper advocates for tools supporting experimental prototyping and formal verification to address quantum logic challenges.

In a complementary manner, we considered five papers that account for the challenges presented by QC. These papers cover various topics, from tools and platforms to architectures and engineering challenges. However, they do not focus specifically on RE for QC.

Upama et al. [65] provide a detailed overview of the evolution and current state of QC tools and platforms, highlighting their transition from theory to applicable practices. On the other hand, Gill et al. [66] propose a comprehensive taxonomy that categorizes studies and technologies in the field, identifying open challenges and suggesting future research directions, in areas such as post-quantum cryptography and simulations for complex quantum experiments, energy management, robotics, and cybersecurity. Both works reflect the growing interest and rapid developments in QC, highlighting tools, platforms, and the development of hardware and quantum software/algorithms.

Sood et al. [3] highlight five main research domains in QC: quantum communication, quantum algorithms and simulation, quantum-enhanced analysis methods, cryptography and quantum information security, and quantum computational complexity. Likewise, Almudever et al. [67] focus on the critical engineering challenges for developing QC, such as the need for reliable qubits. Finally, Khan et al. [68] explore the architectural aspects of quantum software, suggesting that existing processes and notations could be adapted for QC.

Despite the depth and breadth of these studies, it is clear that RE specific to QC have not been the primary research focus. This absence highlights a significant opportunity for future work, especially in requirements definition, documentation, and maintenance for quantum software development. Such research will be crucial to successfully transition from quantum theory to practical and functional applications, addressing a significant gap in the existing literature.

4. Methodology

This study has been carried out according to the SLR methodology described by Kitchenham [69,70]. In this paper, we have identified the existing research related to our topic of interest (QC and RE). We have classified the existing research and analyzed trends regarding the pace of publications, the evolution in using different techniques or tools, and the main constraints, challenges, and opportunities.

Next, in Section 4.1, we present the applied SLR protocol. Then, in Section 4.2, we describe the study selection and data extraction process, the outcome of which is the final list of papers included in our SLR. Figure 1 depicts the whole process.

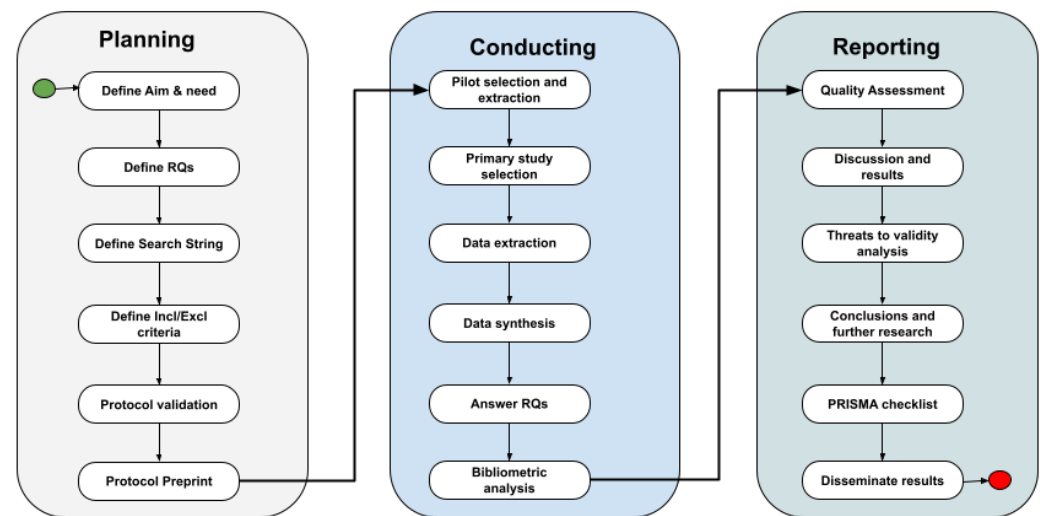


Figure 1. SLR process, phases and steps.

4.1. SLR Protocol

We present the main steps in the protocol definition to conduct this SLR. The first step consisted of determining the aim and need for the SLR (Section 4.1.1). Then, we defined the RQs that drive this SLR (Section 4.1.2). Based on these RQs, we defined the search string to select the primary studies (Section 4.1.3). Finally, we defined the inclusion/exclusion criteria (Section 4.1.4).

We present an initial definition of the SLR protocol in [71]. The protocol validation was performed along with the definition of each step. This validation was based on the expert criteria of two authors who independently assessed each step of the process. The information presented in this paper corresponds to each step's final result (definition plus validation).

4.1.1. Aim and Need

This study aims to identify and analyze the unique challenges of RE for quantum software development. It seeks to evaluate the current methodologies used to address these challenges and propose future perspectives on their evolution to meet the needs of the field.

The expected impact of the SLR is to provide an overview of state-of-the-art RE practices adapted to QC, highlighting the differences of traditional software development and guiding researchers and practitioners to navigate the complexities of this emerging paradigm.

The motivation for this SLR arises from the transformation QC introduced and the challenges it presents in RE compared to conventional computing. It is hoped that this SLR will thoroughly analyze these challenges, evaluate existing methodologies, and propose future directions, thus contributing to theoretical understanding and practical advances in this evolving field.

4.1.2. Research Questions

The context for the RQs revolves around the emergent field of QC, offering capabilities that surpass those of classical computing systems. These capabilities consider parallel processing and significantly enhanced computational speed. This technological leap presents unique challenges in RE, necessitating new models and methodologies distinct from those used in classical SE.

The RQs consider investigating these specific challenges, exploring the current state of methodologies in QC, and identifying future directions for RE within this novel and complex domain.

The RQs arise from the main RQ: *What are the challenges, opportunities, and future directions on RE for QC software?* This RQ is divided into three more detailed RQs focused explicitly on challenges, advances, and future directions of RE for QC. Table 3 shows the RQ identifier, the RQ itself, the aim, and a possible classification schema.

Table 3. Research questions (RQs), aim, and classification schema.

RQ#	Research Question	Aim	Possible Classification Schema
RQ1	What specific challenges are currently faced in RE for QC?	To identify and understand the unique challenges that QC poses to RE.	• Technical Complexity • Evolution and Scalability • Integration Requirements • Human Resources, ...
RQ2	What opportunities does QC present for advancements in RE?	To explore how current RE methodologies are being adapted to suit the needs of QC.	• Methodology Types • Tooling and Technologies • Stakeholders Involvement, ...
RQ3	What future directions or trends are emerging in the field of RE specific to QC?	To identify and forecast potential innovations and future directions in RE specifically tailored for QC.	• Methodological • Tools and Techniques • Collaboration and Communication • Education and Training, ...

4.1.3. Search String

The search string was defined according to Kitchenham’s guidelines:

- We extracted the keywords from the RQs.
- We considered synonyms for the keywords.
- We used the PICOC criterion to build the search string [72].

From the RQs, we derived the keywords: *“quantum computing”*, *“requirements engineering”*, *“challenges”*, and *“future directions”*. Then, we considered synonyms and related terms for these keywords to widen the search scope. Finally, we applied the PICOC criterion. See the details in Table 4.

The initial search string was:

(“quantum computing” OR “quantum software”) AND “requirements engineering” AND (“challenges” OR “opportunities” OR “future directions”)

After several tests, we modified the resulting string. For the tests, we considered using Scholar Google and Scopus as data sources, and Publish or Perish (<https://harzing.com/resources/publish-or-perish>) as a tool for recovering and analyzing the data. The final result considered includes the results of the tests, obtaining the search string:

(“quantum computing” OR “quantum software”) AND (“requirements engineering” OR “software requirements”)

Then, the string was entered sequentially into each data source, adapting it accordingly. These sources are recognized among the most relevant in the scientific community, and in particular in the SE research community [6,69,70]. Complementarily, we considered the two most recognized scientific data sources, Web of Science and Scopus [73,74]. For the whole list, see Table 5.

Table 4. Details of PICOC criterion.

Criteria	Scope	Details in SE Domain	Application in Our Case
Population	Who?/What?	For SE, it should correspond to one of the following: (1) specific role, (2) a category of software engineer, (3) an application area or (4) an industry group.	QC software projects
Intervention	How?	Methodology, tool, technology or procedure that addresses a specific issue.	RE methodologies and approaches
Comparison	Compare to...?	N/A	Not applicable, the study does not compare interventions.
Outcomes	What ... to accomplish?/effect?	Outcomes should relate to factors of importance to practitioners such as improved reliability, reduced production costs, and reduced time to market.	Identified challenges, opportunities, and directions in RE for QC.
Context	Under what circumstances ?	This is the context in which the comparison takes place, the participants taking part in the study, and the tasks being performed.	The application in both academic research and industry practice.

Table 5. Selected data sources.

Source	Link
ACM Digital Library	https://dl.acm.org/
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
Science Direct	https://www.sciencedirect.com/
Springer Link	https://www.springer.com/
WoS	https://www.webofscience.com
Scopus	https://www.scopus.com

4.1.4. Inclusion and Exclusion Criteria

This study defined two types of inclusion and exclusion criteria: format-related and content-related. The format-related criteria involved publication language and publication type. English was selected as the language for the search, considering that the trending journals and conferences publish their works in this language. Furthermore, grey literature (i.e., technical reports, white papers, and work in progress), secondary studies, and short papers were excluded. Table 6 depicts these criteria.

Table 6. Exclusion criteria.

Criteria#	Description
EC1	The paper is not written in English.
EC2	The paper is not peer-reviewed (posters, tutorials, slides, PhD or master thesis and any piece of work considered as grey literature).
EC3	The paper is a secondary study (eventually considered in the related work section).
EC4	The paper is a short paper (less than four pages).
EC5	The focus of the paper is not on proposals treating SE, RE on QC.

4.2. Study Selection and Data Extraction Process

Next, we present the main steps in the study selection and data extraction process to conduct this SLR. The first step consisted of deciding the search process (Section 4.2.1).

Then, we conducted a pilot selection to ensure the protocol's reliability (Section 4.2.2). Finally, we defined the data extraction protocol to select the primary studies (Section 4.2.3).

We aimed to compile a complete list of papers on RE and QC. According to the recency of the subject and initial searches, the search considers the time lapse from 2017 to 2024. The search was conducted between March and April 2024.

4.2.1. Search Process

We ran the search in the selected data sources (See Table 5). We considered a search process consisting of two phases:

- Automatic search applying the search string on selected data sources, considering:
 - a. different versions of the same paper (we keep the last one),
 - b. eliminate duplicates.
- Snowballing process, using the list of selected papers as seed [75].

The remaining papers go through the screening phase; the researchers must decide if they are relevant for the SLR. We review the title, abstract, and keywords to do this. The papers superimposing this phase were filtered using the exclusion criteria (See Table 6).

4.2.2. Pilot Selection

Once the search string, exclusion criteria, and the data sources had been defined, we performed a pilot selection and extraction to further ensure the protocol's reliability.

To avoid bias from a particular researcher examining each paper, we verified that applying and understanding the criteria was similar for the researchers involved in the SLR (inter-rater agreement). This validation was achieved by two researchers individually deciding on the inclusion/exclusion of a set of 45 papers randomly chosen from those retrieved by this pilot selection.

A test of concordance based on the Fleiss' Kappa statistic was performed as a means of validation [76]. The first attempt failed ($Kappa = 0.61$). Following this attempt, the research team conducted a set of meetings and discussions to clarify the issues, all due to differences of opinion regarding the meaning of the content-related criteria. We rewrote the criteria accordingly. We held a meeting to adjust and clarify the considerations and scope of the papers, and we selected those that mentioned some aspects of QC, SE, and software requirements (functional or non-functional). Another set of twenty papers was selected, and the protocol was applied independently again. This second time, we obtained a $Kappa = 0.78$, which suggests that the criteria were clear enough for the researchers to apply the inclusion and exclusion criteria consistently [77].

4.2.3. Data Extraction Protocol

Once the pilot was tested and validated, we launched the primary study retrieval and the data extraction phase.

First, we ran the search string in the selected data sources. This process produced 1742 results. After eliminating duplicates, 1678 results remained on the list.

Next, we applied the format-related inclusion/exclusion criteria and discarded papers that were not in English, grey literature, and short papers. Also, we discarded papers that presented two or more versions of the same proposal. When the latter was the case, we retained the most current proposal version in the selection. The result was a set of 1251 papers. Initially, we considered excluding papers from pre-print platforms such as arXiv, but considering the novelty and interdisciplinary nature of the paper, we finally considered this kind of paper.

Then, we looked through the title, abstract, keywords, and introduction to obtain an initial impression of their thematic relevance. In this step, 287 papers proceeded to the next step; all the rest were rejected.

Last, we divided the final list among two researchers, and each one applied the content-related exclusion criteria. The final list included 97 papers. The main reason for excluding the other 190 papers was that, although they referenced QC and RE, they did not

specifically focus on challenges, opportunities, trends, or future directions about the subject. In addition, 11 articles to which we did not have access were also discarded. Attempts were made to find other versions of the paper and to contact the authors, but only in one case was there a positive response.

To avoid excluding relevant papers, we conducted a first snowballing search review according to the guidelines proposed by Wohlin [75]. The search range will be extended by reviewing the references of each selected article (backward snowballing) and the citations obtained by the articles (forward snowballing). This snowballing search added 8 papers (3 back ward and 5 forward). The whole process is in Figure 2.

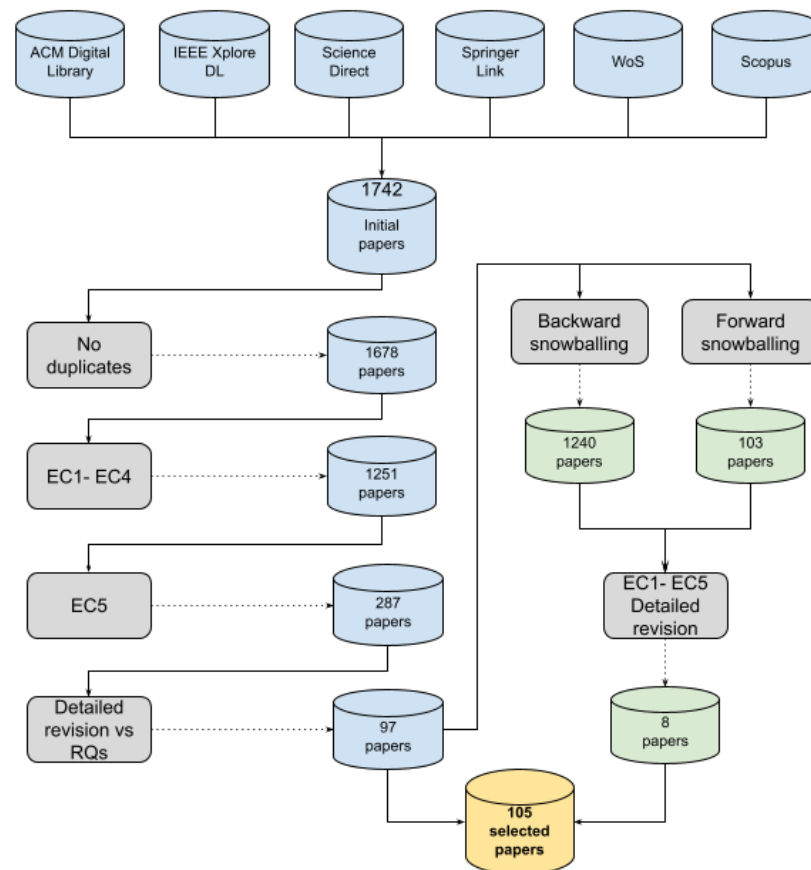


Figure 2. SLR selection process.

This information was then jointly reviewed to accept the final list of 106 papers collaboratively. Table A2 presents the whole list of the selected papers for this SLR, including paper ID, title, authors and publication year. The ID of each paper reflects the number given in the first list of possible papers selected. In addition, to maintain the traceability of the process, papers added at the snowballing stage were prefixed with *bw* or *fw* (from the backward or forward snowballing stage) [75].

Finally, for each of the papers that met the selection criteria, we proceeded with their reading to extract relevant data to answer the RQs. The extracted data for each paper and the assessment strategy were as follows: (i) title, authors, year, (ii) reason why the paper was initially included, (iii) type of publication journal or conference and the affiliated editor, (iv) challenges, opportunities, trends, or future directions reported, and (v) results and future work.

4.3. SLR Tool Support

In order to facilitate finding, selecting, documenting, and analyzing the information gathered, we considered the following support tool. As a shared repository of resources,

we used Google Docs (<https://docs.google.com/>). We used Publish or Perish (<https://harzing.com/resources/publish-or-perish/>) for the initial validation of the search string and automatic spreadsheet creation. To store, read, and annotate reviews for selected papers and to automatically create .bib files, we used Scholarcy (<https://www.scholarcy.com/>). To write the paper, edit the text files, and manage and version control the files repository, we used OverLeaf (<https://www.overleaf.com/>). We used VOSViewer (<https://www.vosviewer.com/>), TagCrowd (<https://tagcrowd.com/>), and Termine (<https://www.nactem.ac.uk/software/termine/>) to display and summarize the results.

5. Results

In the following, we present the answers to the RQs raised by this SLR. We report the results for RQ1, showing the challenges that QC poses to RE (Section 5.1). The results for RQ2 show the methodologies adapted to suit the needs of QC (Section 5.2). Answering RQ3 is based on identifying potential innovations and future directions in RE specifically tailored for QC (Section 5.3). We present the research trends and cataloging (Section 5.4). Finally, we present the strength of the evidence collected (Section 5.5).

Based on the relevance of software quality attributes, we present an analysis of the elements identified for each of the RQs and how they relate to the different characteristics and sub-characteristics of the framework defined by the ISO/IEC 25010 standard [47]. This standard ISO/IEC 25010 provides a framework for evaluating software quality characteristics and sub-characteristics.

5.1. RQ1: What Specific Challenges Are Currently Faced in RE for QC?

The analysis of selected papers reveals the main challenges faced in RE for QC. These findings include issues with the definition of quantum-specific requirements, hybrid system requirements, and continuous update of requirements, among others.

5.1.1. Specific RE Challenges on QC

After conducting a detailed analysis, we focused exclusively on challenges related to RE. We identified eight main categories, which the following section examines in detail. Table 7 presents these specific challenges, their frequency of occurrence, and the ID of selected papers where the challenge was identified.

- *Defining Quantum-Specific Requirements:* Quantum-specific requirements are challenging due to the unique properties of quantum mechanics, such as superposition and entanglement, which complicate the definition of clear and actionable requirements.
- *Hybrid System Requirements:* Integrating quantum and classical components is particularly challenging. Careful balancing is required to ensure seamless interaction between both components. Requirements must specify data transfer and handle computational paradigm differences.
- *Absence of Established Standards:* A significant gap in standardized requirement engineering processes tailored for QC complicates the gathering, analysis, and validation of requirements.
- *Continuous Update of Requirements:* The rapid evolution of quantum technology necessitates continuous updates to requirement specifications to keep them relevant and accurate.
- *Knowledge and Awareness:* There needs to be an educational gap among requirements engineers regarding QC principles, limiting their ability to translate quantum capabilities into software requirements effectively.
- *Requirement Specification Challenges:* Defining and managing requirements for quantum systems is complex due to the unique properties of QC, such as superposition and entanglement, requiring new approaches to RE.
- *Quantum-Specific Security Requirements:* Addressing new security threats QC poses, such as vulnerabilities in classical cryptographic methods, requires specifying quantum-resistant cryptographic protocols.

- *Testing and Verification*: The fundamental differences between quantum and classical computing make developing effective testing and verification processes for quantum software requirements difficult. The difficulty in developing effective testing and verification processes can be attributed to the unique characteristics of QC requirements.

Table 7. Frequency and file IDs of identified RE challenges.

Identified RE Challenge	Frequency	File IDs
Requirements Specification	16	3, 41, 43, 45, 46, 47, 52, 75, 85, 89, 109, 118, 160, 277, 278, 279
Hybrid System Requirements	12	3, 41, 43, 45, 46, 52, 75, 85, 109, 160, 278, 279
Absence of Established Standards	11	43, 46, 47, 75, 85, 109, 118, 160, 277, 278, 279
Defining Quantum-Specific Requirements	10	13, 24, 34, 56, 87, 152, 161, 162, 227, 231
Continuous Update of Requirements	9	18, 47, 52, 89, 109, 160, 277, 278, 279
Knowledge and Awareness	9	3, 43, 85, 109, 118, 160, 277, 278, 279
Quantum-Specific Security Requirements	9	13, 24, 56, 87, 152, 161, 162, 227, 231
Testing and Verification	6	159, 160, 214, 223, 258, bw1

5.1.2. RE Challenges vs. ISO25010

Identifying specific challenges in RE for QC is crucial to ensuring software quality in this emerging domain. Figure 3 maps the identified challenges in RE for QC to the relevant characteristics and sub-characteristics of the ISO/IEC 25010 software quality standards.

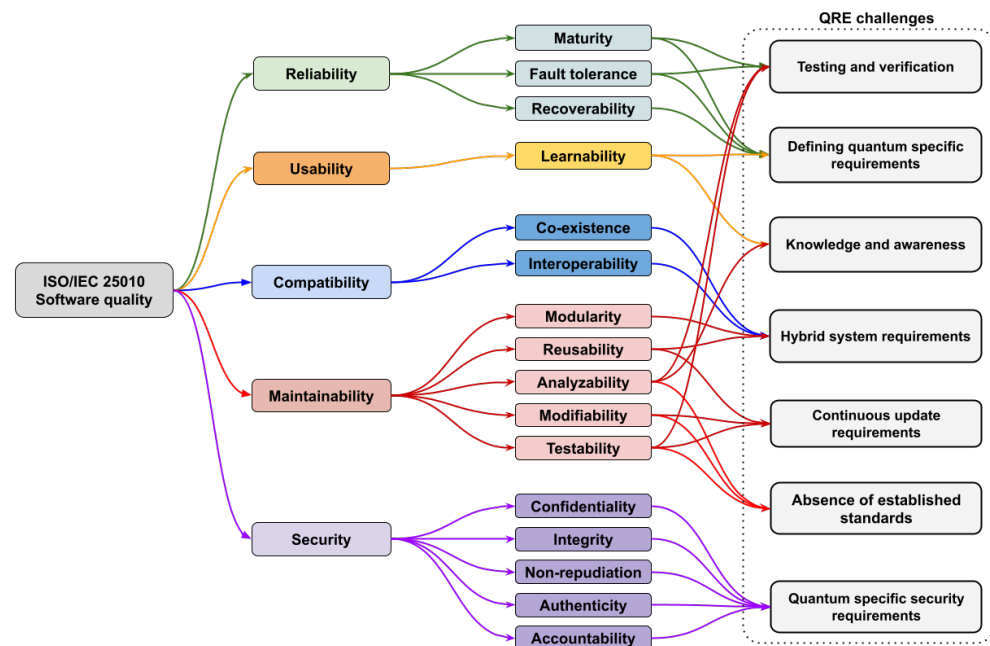


Figure 3. ISO/IEC 25010 characteristics and sub-characteristics vs. RE challenges identified.

5.1.3. Other Challenges

Next, we present an analysis of other QC software development challenges. These challenges are not directly related to RE but can have an impact on developing quantum-classical software. We grouped the challenges into five areas and the corresponding sub-areas identified. Table 8 presents the challenges, their frequency of occurrence, and the ID of selected papers where the challenge was identified.

- *Hardware Challenges:* The availability and stability of current quantum devices, such as NISQ systems, are inherently unstable, with high error and decoherence rates, which significantly limit the performance of quantum computations. Scaling quantum hardware to support fault-tolerant execution of quantum algorithms remains a critical challenge, including implementing robust error correction methods to handle the high error rates.
- *Algorithm and Programming Challenges:* Developing and optimizing efficient quantum algorithms is complex due to the need to handle quantum properties like superposition and entanglement. The lack of standardized and accessible quantum programming languages and the complexity of existing ones like Qiskit and Cirq pose significant obstacles for developers. Additionally, quantum programming requires developers to acquire new skills and knowledge, which can be lengthy and challenging due to the intrinsic complexity of quantum mechanics.
- *Security Challenges:* QC threatens classical cryptographic algorithms like RSA and ECC, which could become vulnerable to quantum attacks using algorithms like Shor's. It is necessary to advance the creation of quantum-resistant cryptographic methods that can withstand potential quantum attacks. Integrating quantum-safe protocols into existing infrastructures is complex and requires considerations of compatibility and efficiency.
- *Integration Challenges:* Integrating quantum systems with classical computing infrastructures is a significant challenge due to fundamental differences in computational paradigms and the need to manage hybrid interfaces efficiently. Ensuring compatibility and effective integration between different quantum software platforms and quantum hardware is complex, especially given the rapid evolution of quantum technologies. The fast-paced evolution of quantum technology means that requirement specifications need continuous updates, presenting a significant challenge to maintain consistency and relevance.
- *Quality and Maintenance Challenges:* Ensuring the confidentiality and integrity of data in QC systems presents significant challenges. Testing and verifying quantum software is complex due to the fundamental differences between quantum and classical computing, necessitating new approaches and tools.

Table 8. Frequency and file IDs of other challenges.

Identified Challenge	Frequency	File IDs
Availability and Stability of Hardware	27	3, 7, 8, 12, 16, 17, 18, 22, 33, 34, 38, 39, 63, 109, 118, 143, 214, 215, 220, 222, 266, 275, 276, bw3, fw1, fw3, fw4
Integration of Quantum and Classical Systems	16	18, 22, 23, 26, 33, 40, 45, 52, 62, 89, 109, 118, 222, 258, 264, fw4
Security and Privacy Management in Quantum Systems	15	13, 14, 24, 56, 87, 152, 161, 162, 183, 193, 203, 204, 227, 231, 281
Scalability of Quantum Hardware	14	7, 12, 16, 18, 33, 38, 63, 143, 214, 215, 220, 264, 275, bw3
Compatibility and Efficiency	14	18, 23, 26, 62, 87, 118, 161, 162, 216, 264, 270, 275, fw6, fw7
Threats to Classical Cryptography	10	13, 24, 34, 56, 87, 152, 173, 198, 203, 280
Development of Quantum Cryptography	9	13, 24, 34, 87, 152, 173, 198, 203, 280
Quantum Programming Languages	9	18, 23, 28, 29, 40, 85, 109, 224, 235
Cost and Technical Complexity	8	13, 24, 26, 56, 62, 87, 196, fw1
Development and Optimization of Quantum Algorithms	8	7, 12, 17, 18, 33, 214, 256, bw2
Rapid Technological Evolution	8	18, 29, 47, 52, 89, 224, 265, fw7
Learning Curve	6	18, 28, 29, 85, 224, 283
Testing and Verifying Quantum Software	6	159, 160, 214, 223, 258, bw1
Integration of Quantum-Safe Protocols	5	24, 56, 87, 161, 162

5.1.4. Answering the RQ1: What Specific Challenges Are Currently Faced in RE for QC?

Based on the analysis of the identified challenges in QC software development, the specific challenges in RE for QC can be outlined as follows.

- *Complexity in Requirement Specification:* Defining quantum-specific requirements is challenging due to the unique properties of quantum mechanics, such as superposition and entanglement. These properties make it difficult to define clear and actionable requirements for quantum software. Understanding how quantum operations differ fundamentally from classical ones is essential. For example, specifying how quantum algorithms should handle entangled states and superpositions in a way that aligns with expected outcomes is particularly complex.
- *Integration with Classical Systems:* Developing requirements for systems integrating both quantum and classical components presents significant challenges. These hybrid systems require a balance to ensure seamless interaction between both components. Requirements must specify how data will be transferred between quantum and classical systems and how to handle the differences in computational paradigms.
- *Lack of Standardized Methodologies:* A significant gap exists in standardized RE processes tailored for QC. This lack of established standards complicates the gathering, analyzing, and validating requirements. Developing standardized methods for eliciting requirements that consider quantum computational limitations and error rates is crucial.
- *Rapid Technological Evolution:* The fast-paced evolution of quantum technology necessitates continuous updates to requirement specifications to remain relevant and accurate. It is essential to keep requirements documents up-to-date with the latest quantum hardware capabilities and software improvements.
- *Educational Gap:* There is a significant educational gap among requirements engineers regarding QC principles. This gap limits their ability to translate quantum capabilities into software requirements effectively. Requirements engineers need training in quantum mechanics and QC concepts to develop accurate and feasible requirements.
- *Security and Privacy Considerations:* Quantum-specific security requirements are essential to address new security threats posed by QC, such as vulnerabilities in classical cryptographic methods. This challenge involves specifying requirements for integrating quantum-resistant cryptographic protocols to protect sensitive data within QC applications. The unique properties of QC necessitate that RE explicitly addresses these security concerns to ensure robust protection against quantum threats.
- *Complexity of Testing and Validation:* The fundamental differences between quantum and classical computing make developing effective testing and verification processes for quantum software requirements difficult. Defining test cases that can verify the correct implementation of quantum algorithms under the constraints of current quantum hardware is particularly challenging.

5.2. RQ2: What Opportunities Does QC Present for Advancements in RE?

Developing software for QC is a critical area of research, as it bridges the gap between theoretical quantum algorithms and practical applications. This analysis aims to explore the advancements in QC software development. The information has been synthesized into several key categories, providing a comprehensive overview of progress in this dynamic field. Below, the content is organized by topics highlighting specific advancements in quantum algorithms, programming languages, simulation tools, hardware development, quantum information theory, practical applications, and interdisciplinary collaborations.

5.2.1. Specific RE Advances on QC

After conducting a more detailed analysis, we focused on the advancements that correspond exclusively to software RE for developing quantum software. We identified four primary categories, which the following section examines in detail. Table 9 presents these specific advances, their frequency of occurrence, and the ID of selected papers where the advance was identified.

- *Specific Requirements Techniques for QC*: Emerging techniques aim to adapt classical requirement engineering methods to the quantum context, defining quantum-specific functionalities more effectively. Advanced modeling tools are being developed to represent quantum algorithms and their integration with classical systems, aiding requirement analysts.
- *Frameworks and Tools for Quantum Software RE*: Developments in frameworks and tools facilitate the RE process for quantum software, addressing its unique needs. Integrating quantum and classical computing elements within software applications creates hybrid systems leveraging both paradigms' strengths. Hybrid approaches combining classical and quantum methodologies bridge gaps between these realms. Standardization efforts and educational programs prepare requirement engineers for quantum-specific needs. New tools and frameworks enhance the precision and feasibility of modeling and specifying quantum software requirements.
- *Hybrid RE Methods*: Hybrid RE approaches combine classical and quantum methodologies, bridging gaps between these realms. Interdisciplinary collaborations lead to more robust strategies by incorporating insights from quantum physicists, software engineers, and blockchain experts, enhancing the overall requirement engineering process for quantum software.
- *Requirements Modeling Techniques*: Advances in modeling tools simulate quantum behaviors and integrate them with classical systems, providing new ways to specify quantum software requirements. Standardization efforts aim to create a common framework for these requirements. Educational programs prepare engineers with the necessary skills for handling quantum-specific needs. New tools and frameworks are being developed to enhance the precision and feasibility of specifying quantum software requirements.

Table 9. Frequency and file IDs of identified RE advances.

Identified RE Advance	Frequency	File IDs
Specific requirements techniques for QC	4	41, 45, 46, 47
Frameworks and tools for quantum software RE	4	3, 43, 46, 47
Requirements modeling techniques	3	45, 46, 47
Hybrid RE methods	1	43

5.2.2. RE Advances vs. ISO25010

Identifying specific advances in RE for QC is crucial for ensuring software quality and setting the guidelines for further research. Figure 4 maps the identified challenges in RE for QC to the relevant characteristics and sub-characteristics of the ISO/IEC 25010 software quality standards.

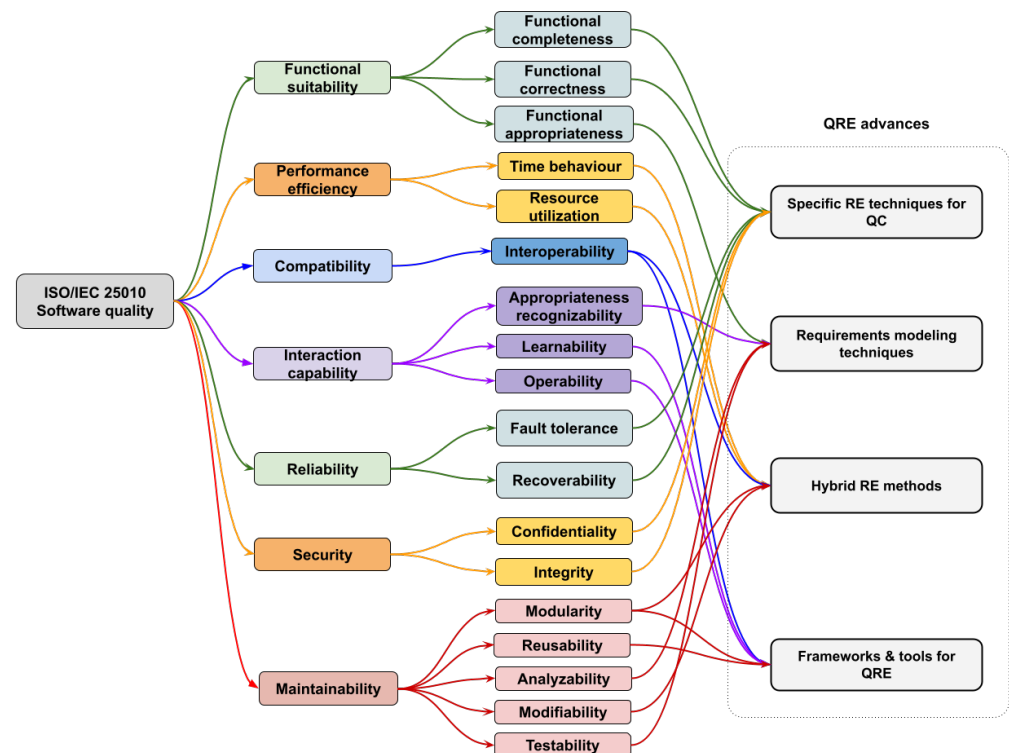


Figure 4. ISO/IEC 25010 characteristics and sub-characteristics vs. RE advances identified.

5.2.3. Other Advances

Next, we present a detailed analysis of other QC software development advances. These advances are not directly related to RE but can have an impact in the development of quantum-classical software. We grouped the advances into seven identified areas. Table 10 presents the advances, their frequency of occurrence, and the ID of selected papers where the advance was identified.

- *Development of new quantum algorithms:* Algorithms utilizing quantum superposition, entanglement, and interference have been introduced to enhance ensemble methods, allowing for exponential expansion of ensemble size with only a linear increase in quantum circuit depth. However, the real game-changer has been the quantum search algorithms. These algorithms, explicitly optimized for structured datasets, have improved search efficiency and reduced quantum resource consumption.
- *Development of specific Quantum Programming Languages:* Quantum programming languages such as Q#, Qiskit, and Silq have emerged as powerful tools, simplifying the implementation of quantum algorithms. These languages are specifically designed to handle the intricacies of quantum logic and computation, empowering engineers and developers to delve into the world of QC with confidence. Moreover, quantum software development frameworks and SDKs provide a wealth of resources, enabling engineers and developers to experiment with and implement quantum algorithms more efficiently and effectively.
- *Quantum simulation tools and techniques:* These advances include the development of parametric compilation and active qubit reset methods, which allow rapid adjustments of quantum program parameters without complete recompilation and reduce latency by quickly resetting qubits to their ground state between computations. Additionally, modeling and simulation tools that integrate quantum behaviors with classical systems have progressed, offering new ways to visualize and specify requirements for quantum software.

- *Quantum Hardware Development:* Advances in quantum hardware design, such as increased qubit coherence times and improved qubit connectivity, allow for overcoming current limitations and enabling more efficient quantum computations. Additionally, innovations in the architectural design of QRAM enable more scalable and efficient designs, overcoming some of the current physical limitations in QC hardware.
- *Quantum Information Theory:* Quantum-resistant cryptographic algorithms have been developed to withstand attacks from both quantum and classical computers. These include lattice-based, hash-based, multivariate, and code-based cryptographic schemes. Many initiatives seek to standardize quantum-resistant cryptographic protocols, ensuring a robust and unified approach to protecting data against quantum threats.
- *Practical Applications of QC in Machine Learning and Security:* Developing specialized architectures for quantum machine learning supports the entire lifecycle of development, from data handling to model deployment. Implementations such as quantum reinforcement learning and quantum neural networks demonstrate the practical applicability of these architectures. Furthermore, quantum key distribution (QKD) and quantum-resistant algorithms safeguard IoT systems against emerging quantum threats.
- *Interdisciplinary Collaborations and Research Projects:* Collaboration among computer scientists, physicists, and engineers is leading to more comprehensive approaches in quantum requirements engineering (QRE), integrating knowledge from various disciplines to improve requirements gathering and analysis processes. Additionally, the integration of quantum cryptography and blockchain technology offers a dual layer of security, leveraging the unconditional security of quantum key distribution and the immutability of blockchain.

Table 10. Frequency and file IDs of other advances.

Identified Advance	Frequency	File IDs
Quantum-resistant cryptography	7	56, 87, 152, 161, 162, 198, 280
Development of quantum programming languages	4	220, 223, 224, 287
Quantum algorithms	3	bw3, fw1, fw4
Quantum key distribution	2	34, 183
Quantum DevOps software development	2	166, 202
Hybrid quantum-classical algorithms	2	fw3, fw4
Agile-Quantum software project success models	2	28, 29
Improvements in quantum hardware	2	bw3, fw1
Categorization and taxonomy of technologies	1	fw7
Strategic business approaches	1	265
Development of the quantum ecosystem	1	265
Introduction of the quantum software lifecycle	1	266

5.2.4. Answering RQ2: What Opportunities Does QC Present for Advancements in RE?

The unique capabilities and challenges of QC necessitate advancements in RE methods to ensure that software systems fully leverage the power of QC while maintaining high standards of quality, security, and performance.

Upon the challenges identified in QC software development, we outline the specific challenges in RE for QC as follows.

- *Enhanced Specification Techniques:* QC requires new RE techniques to address unique properties such as qubits and entanglement. Developing quantum-specific requirements ensures these aspects are accurately captured and managed. Adapting classical methods to define quantum functionalities helps align software with quantum operational paradigms.

The following is an overview of the techniques used for RE within quantum software development projects based on the authors' claims in the selected papers. To elicit information, the authors propose practical techniques such as interviews and questionnaires, literature reviews, and the analysis of repositories such as Stack Exchange and GitHub. Taxonomies, UML, and Q-UML can be mentioned for modeling and representing the problem domain. User stories are specifically mentioned for the system's functionalities. In the particular case of non-functional requirements, the use of ISO-9126 [78] and ISO-25010 [47] quality frameworks is mentioned. Finally, agile QC practices and techniques, the Quingo framework, and the Talavera Manifesto, among others, are mentioned as reference frameworks that guide not only the RE but also the SE of the entire project.

- *Improved Modeling and Simulation Tools:* The complexity of quantum algorithms necessitates advanced modeling tools to represent quantum behaviors and their integration with classical systems. These tools enhance understanding and specification of requirements, facilitating the development of robust quantum software. Integrating quantum behaviors into modeling tools improves precision in requirement specification.
- *Hybrid Requirement Engineering Approaches:* Combining classical and quantum methodologies bridges the gap between the two paradigms, enabling comprehensive requirement engineering processes. Hybrid approaches leverage the strengths of both computing types, ensuring effective and integrated solutions. These methods address the unique challenges of both quantum and classical systems.
- *Interdisciplinary Collaboration:* QC is a field that necessitates collaboration among computer scientists, physicists, and engineers. Diverse expertises and perspectives are crucial in these interdisciplinary efforts, leading to robust requirement engineering strategies. This collaboration enhances the quality and completeness of requirement specifications.
- *Standardization Efforts:* Standardizing quantum software development practices, including requirement engineering processes, is a crucial step. It creates common frameworks for specifying and validating requirements, ensuring consistency and reliability across quantum software projects. These standardized practices foster high quality and interoperability in quantum-classical applications, providing a solid foundation for our work.
- *Educational Programs:* Educational programs and resources in QC equip requirement engineers with the knowledge to handle quantum-specific requirements. This training bridges the educational gap and fosters a skilled workforce capable of advancing RE in quantum contexts. Enhanced education ensures accurate specification of quantum software requirements.
- *Security and Cryptography:* Developing quantum-resistant cryptographic algorithms ensures robust security requirements against QC threats. This advancement is crucial for specifying quantum-safe security measures. Quantum-resistant schemes provide a foundation for secure requirements, protecting sensitive data from quantum threats.

5.3. RQ3: What Future Directions or Trends Are Emerging in the Field of RE Specific to QC?

After analyzing and synthesizing the selected papers, we identified future research directions in QC software development and grouped them into relevant thematic categories.

5.3.1. Specific RE Future Directions on QC

After conducting a more detailed analysis, we focused on the future directions that correspond exclusively to software RE for developing quantum software. We identified twelve primary categories, which the following section examines in detail. Table 11 presents these specific advances, their frequency of occurrence, and the ID of selected papers where the advance was identified.

Table 11. Frequency and file IDs of RE future directions.

RE Future Directions	Frequency	File IDs
Establishing educational programs for quantum requirement engineering	10	41, 43, 45, 46, 47, 85, 89, 277, 278, 279
Formalizing quantum requirement engineering processes	8	41, 43, 45, 46, 47, 52, 85, 89
Developing tools and frameworks for quantum requirement validation	7	41, 43, 45, 46, 47, 52, 85
Enhancing tools for quantum requirement validation and verification	7	41, 43, 45, 46, 47, 85
Creating quantum-specific requirement engineering methodologies	4	46, 47, 277, 278
Standardizing architectural practices and methodologies for quantum software	3	45, 85, 89
Developing sophisticated modeling tools for quantum requirements	3	89, 147, 277
Developing methodologies for dynamic requirement updates and interdisciplinary collaboration	2	46, 47
Developing automated tools for quantum requirement engineering	1	278
Innovating hybrid modeling techniques that integrate quantum and classical computing elements	1	277
Establishing defined software development lifecycles for quantum applications	1	279
Improving methodological support for RE in QC	1	279

5.3.2. RE Future Directions vs. ISO25010

Identifying specific research future directions in RE for QC is crucial to ensuring software quality in this emerging domain. Figure 5 maps the future research directions to the relevant characteristics and sub-characteristics of the ISO/IEC 25010 software quality standards.

5.3.3. Other Future Research Directions

Next, we present the details of these categories. Table 12 presents the other future research directions, their frequency of occurrence, and the ID of selected papers where it was identified.

- *Quantum Algorithms*: Future research in quantum algorithms focuses on several key areas. Firstly, there is significant interest in optimizing algorithms to reduce quantum resource requirements and enhance efficiency, particularly for NP-complete problems and molecular simulation. Additionally, quantum algorithms are pushed to be extended to more complex and diverse problems, such as network optimization and energy systems. Integrating quantum algorithms with classical systems to maximize efficiency and applicability is also a crucial research direction.
- *Quantum Hardware*: Future research in quantum hardware highlights the need to improve qubits' stability and coherence times, essential for reducing errors during quantum operations. There is also a focus on developing robust quantum hardware that can operate under real-world conditions and exploring new materials to construct more stable qubit systems. Additionally, integrating quantum hardware with classical systems to create scalable quantum-classical hybrid systems is a crucial area of interest.
- *Error Correction*: Error correction is a critical aspect of QC, and future research aims to develop more efficient and less resource-intensive error correction techniques. These techniques include exploring advanced quantum error correction methods and integrating them into the quantum software development lifecycle to manage inherent quantum errors more effectively.
- *Quantum Security*: Future research in quantum security focuses on optimizing quantum-resistant algorithms to enhance their efficiency and scalability. There is also a significant emphasis on developing comprehensive security frameworks that integrate

quantum capabilities into existing IoT systems and other platforms, ensuring robust protection against classical and quantum threats.

- *Applications in Physics and Chemistry:* Future research directions for applications in physics and chemistry include developing quantum algorithms tailored for specific domains like molecular simulation and material science. These efforts aim to broaden the scope of QC applications and improve the accuracy and efficiency of simulations in these fields.
- *Optimization and Logistics:* Research in optimization and logistics focuses on enhancing the capabilities of quantum algorithms to solve complex scheduling and optimization problems. This optimization includes the development of hybrid algorithms that dynamically allocate tasks between quantum and classical systems to leverage their respective strengths.
- *User Interface and Software Tools:* Future research in user interfaces and software tools involves creating more sophisticated virtual instrument interfaces and enhancing real-time data processing capabilities. Developing robust tools seamlessly integrating quantum and classical computing elements is also a significant research focus.
- *Education and Training in QC:* Addressing the growing need for skilled professionals in QC is crucial. Future research emphasizes the development of comprehensive educational programs and training materials. These programs aim to equip engineers and developers with the necessary skills to navigate the complexities of QSE and integrate QC technologies effectively.

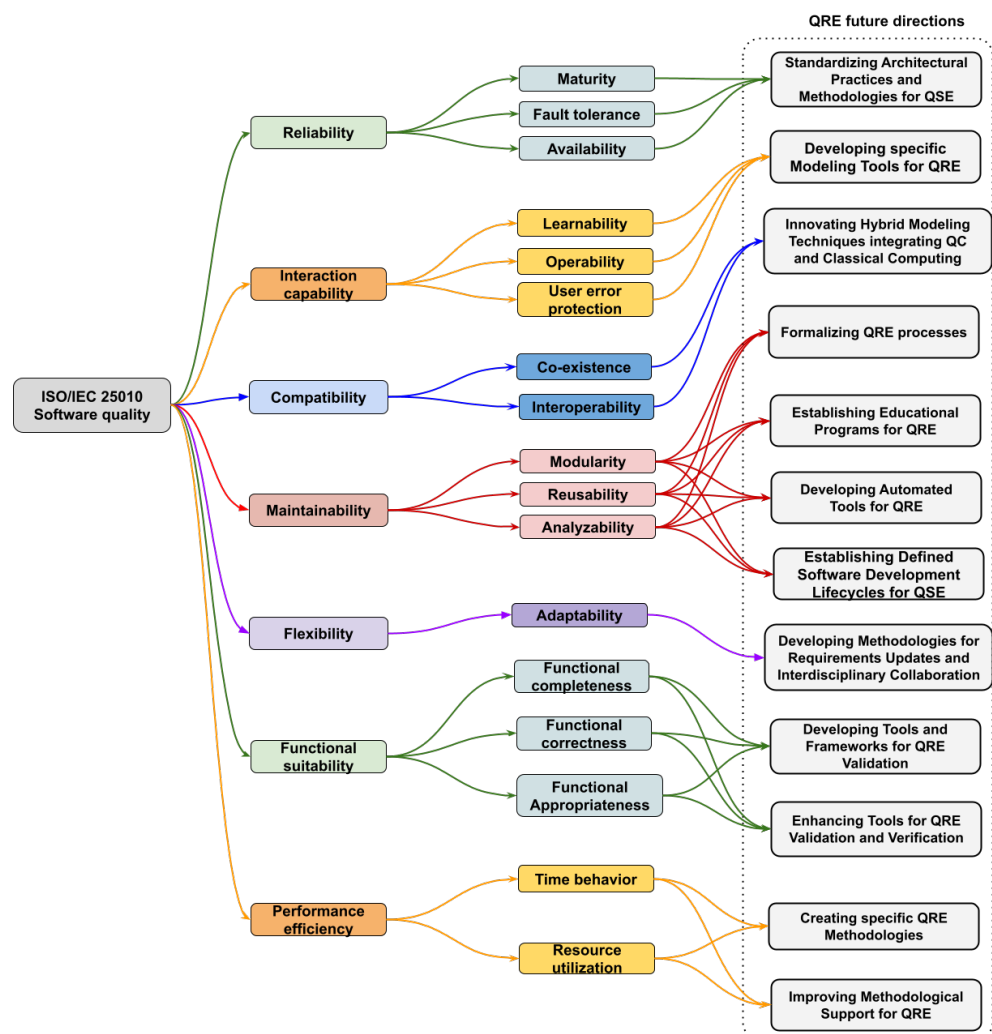


Figure 5. ISO/IEC 25010 characteristics and sub-characteristics vs. RE future directions identified.

Table 12. Frequency and file IDs of identified research future directions.

Identified Future Direction	Frequency	File IDs
Improving tools and methods for quantum software development	15	3, 30, 31, 41, 43, 45, 46, 47, 85, 89, 202, 222, 277, 278, 279, 283
Developing educational programs for QC	14	41, 43, 45, 46, 47, 52, 85, 89, 166, 169, 222, 223, 253, 270
Optimizing quantum-resistant algorithms for IoT devices	9	8, 24, 152, 162, 223, 241, 244, 280, 281
Optimizing algorithms for NP-complete problems and molecular simulation	8	3, 12, 33, 39, 143, 147, 284, 285
Enhancing algorithms for broader applicability	8	16, 17, 34, 262, 270, 275, 283
Developing comprehensive security frameworks	6	13, 24, 89, 152, 173, 241
Developing tools and platforms for quantum-classical integration	6	18, 23, 25, 28, 62, 214
Developing advanced quantum error correction techniques	6	12, 13, 14, 16, 256, 276
Developing educational programs for QC	6	41, 43, 45, 46, 47, 85, 89
Increasing qubit stability and scalability	5	2, 3, 255, 256, 276
Developing hybrid algorithms for broader applications	4	9, 18, 118, 228
Enhancing efficiency and scalability of quantum-inspired algorithms	4	17, 28, 29, 263
Improving coherence times and developing new materials	4	10, 11, 143, 215
Refining quantum algorithms for larger datasets	4	12, 22, 33, 214
Exploring quantum-inspired approaches for optimization problems	4	8, 28, 29, 197
Creating robust QSE practices	3	277, 278, 279
Expanding training initiatives and community collaboration	3	253, 270, 275
Integrating quantum capabilities for IoT security	3	87, 194, 198
Developing automated testing tools and integration strategies for hybrid systems	2	159, 160
Enhancing QEC codes for broader use	2	38, 276
Continuous experimental verification of theoretical models	2	16, 276
Enhancing compiler efficiency and extending software frameworks	2	23, 32
Integrating quantum hardware into scalable systems	2	18, 194
Expanding QC applications in manufacturing	1	234
Expanding dynamic programming techniques for quantum circuit designs	1	7
Refining approximation techniques and error-resilient schemes	1	39

5.3.4. Answering the RQ3: What Future Directions or Trends Are Emerging in the Field of RE Specific to QC?

Based on the analysis of the identified future directions in QC software development, these future directions can be outlined as follows.

- *Formalizing QRE Processes:* Future research emphasizes formalizing the processes specific to quantum requirement engineering. This formalization includes developing standardized methodologies for specifying, modeling, and analyzing quantum software requirements. These methodologies aim to address the unique characteristics of QC, such as superposition and entanglement, which are not present in classical computing. The critical research focus involves developing comprehensive frameworks for quantum requirement engineering, creating standardized practices to ensure consistency and quality across projects and employing formal methods to handle quantum-specific phenomena.

- *Development of Validation and Verification Tools:* Another emerging trend is the development of sophisticated tools and frameworks for validating and verifying quantum software requirements. These tools ensure that the requirements specified are correct, complete, and feasible given the current state of quantum hardware and algorithms. Critical research efforts are directed towards building tools to validate quantum software requirements, ensuring alignment with intended functionalities, and leveraging simulations to test requirements against quantum phenomena.
- *Standardizing Architectural Practices:* There is a push towards standardizing architectural practices and methodologies specific to quantum software. This helps create mature, reliable, and fault-tolerant quantum systems, ensuring their stability and performance. The primary research objectives include developing best practices and standardized architectural methodologies and promoting consistency and integration efficiency across different quantum software projects.
- *Methodologies for Dynamic Requirement Updates and Interdisciplinary Collaboration:* QC evolves rapidly, necessitating methodologies that support dynamic requirements updates and encourage interdisciplinary collaboration. This trend focuses on creating adaptive tools and processes to handle the fast-paced advancements in quantum technologies. Key focus areas include developing tools for dynamic requirement updates and encouraging collaboration between quantum physicists, software engineers, and requirement engineers.
- *Creation of specific QRE Methodologies:* It is crucial to develop methodologies tailored specifically to QC. These methodologies must address QC's probabilistic nature and rapid technological evolution. The primary research focus involves creating specific methodologies for quantum requirements and continuously refining approaches to keep pace with technological advancements.
- *Sophisticated Modeling Tools:* The development of advanced modeling tools for quantum requirements is essential. These tools should provide accurate modeling capabilities that account for the complex behaviors of quantum systems. Key research areas include enhancing existing modeling tools, creating new domain-specific languages, and providing tools that can effectively model and test quantum requirements.
- *Educational Programs for QRE:* There is a significant need for educational programs that equip engineers with the necessary skills and knowledge to handle quantum requirements. This trend focuses on developing curricula, workshops, and certification programs to bridge the knowledge gap. Critical efforts involve establishing comprehensive educational and training programs and equipping engineers with QC principles and practices.
- *Automated Tools for RE:* Another emerging trend is the development of automated tools to support the RE process. These tools aim to reduce manual effort and increase accuracy. Research focuses on creating AI-driven tools for requirement engineering and incorporating simulations and testing frameworks.
- *Hybrid Modeling Techniques:* Innovating hybrid modeling techniques that integrate quantum and classical computing elements is essential for seamless requirement modeling. This trend focuses on bridging the gap between different computational paradigms. Key research areas include developing hybrid models integrating quantum and classical computing and refining integration techniques for practical co-existence.
- *Defined Software Development Lifecycles:* Establishing well-defined software development lifecycles specific to quantum applications is crucial. This trend focuses on creating structured frameworks for developing, testing, and maintaining quantum software. The key focus areas include defining lifecycle phases for quantum software and integrating requirement engineering, testing, and maintenance processes.
- *Improved Methodological Support:* Enhancing methodological support for RE in QC is necessary to handle the complexities of quantum systems. This improvement involves refining existing methodologies and creating new approaches tailored to QC.

The research aims to develop robust methodologies for quantum RE and create new approaches to address quantum-specific challenges.

5.4. Trends and Cataloging

Next, we describe the main trends and cataloging derived from conducting the SLR. These findings considered the origin and content of the papers.

An initial classification considered the category of the paper. These categories included papers published as journal articles (70, 67%), conference papers (18, 17%), book chapters (2, 2%), and pre-prints (15, 14%). Figure 6 illustrates the number of papers by category. It is worth noting that initially, we considered excluding papers published in “pre-print” sources (e.g., arXiv). However, due to the dynamic and recent nature of the topic, we were compelled to include them.

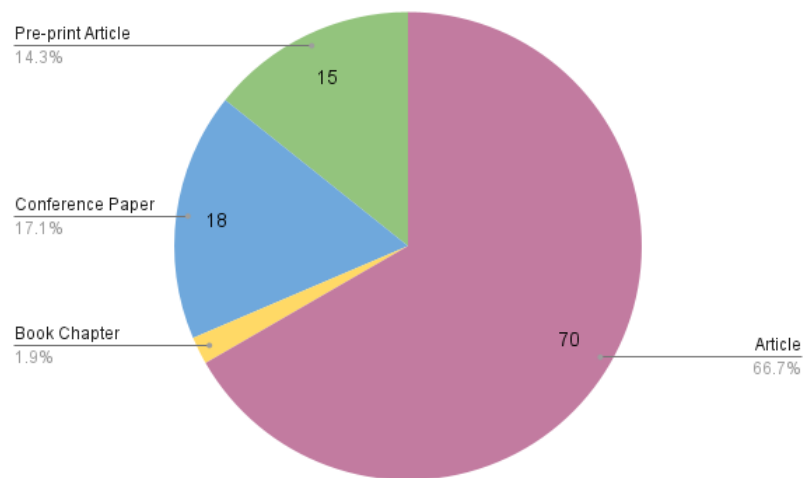


Figure 6. Number of papers by category.

Journals

Of the 105 selected papers, 70 (67%) were indexed in WoS-JCR Journals, Scopus, or other sources. If we conduct a detailed analysis of the published papers in journals, we can note that 59 (84%) were published in WoS-JCR journals, 5 (7%) were published in Scopus, and 6 (9%) were published in other sources. Figure 7 shows the details of this analysis.

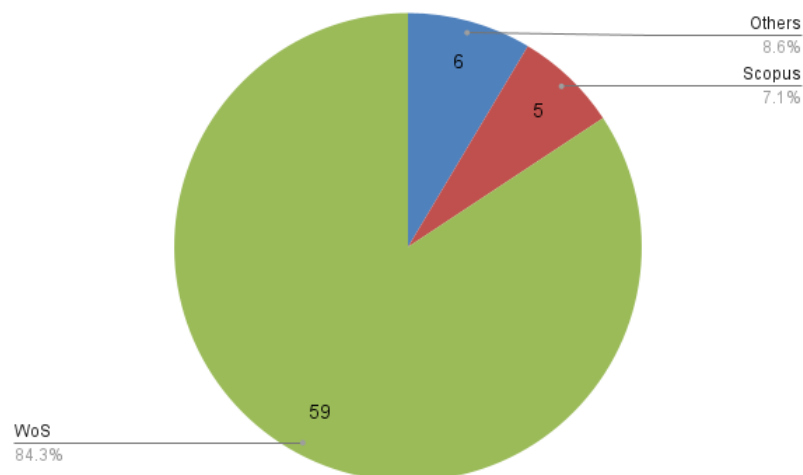


Figure 7. Number of papers by type of journal.

For the 59 papers published in WoS-JCR-indexed journals, we performed a quartile analysis: 18 papers (31%) are in the first quartile (Q1), 28 (48%) in the second quartile (Q2), 9 (15%) in the third quartile (Q3), and 4 (7%) in the fourth quartile (Q4). Figure 8 shows the details of this analysis.

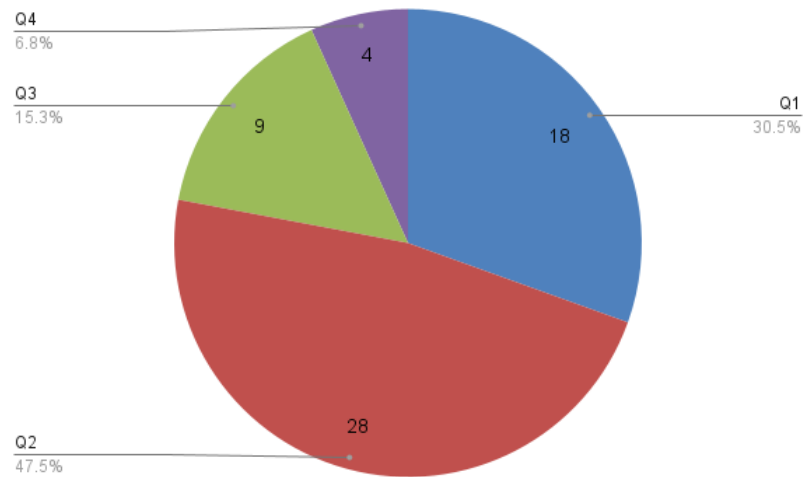


Figure 8. Number of papers by JCR quartile.

Publishers and Sources

Of the 105 selected papers, 24 (23%) were published in Springer, 23 (22%) in IEEE, and 10 (9.5%) in ACM. Thus, we might initially consider Springer, IEEE, and ACM as reference sources. Figure 9 shows a detailed list of papers by publisher.

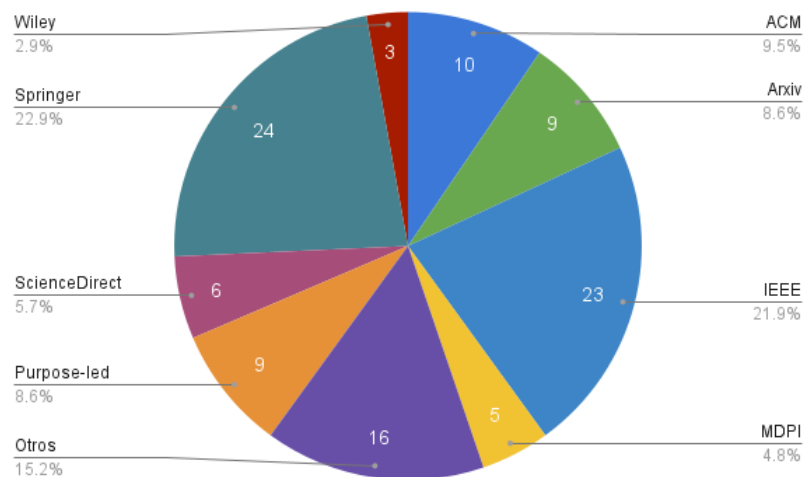


Figure 9. Number of papers by publisher.

Temporal Analysis

We can observe a sustained increase in articles published between 2017 and 2023. Specifically, in 2024, despite the incomplete search and selection of articles, eight articles have already been published. This indicates that the topic continues to be of significant interest to the scientific community. Figure 10 shows the annual evolution of the number of published documents for the period 2017–2024.

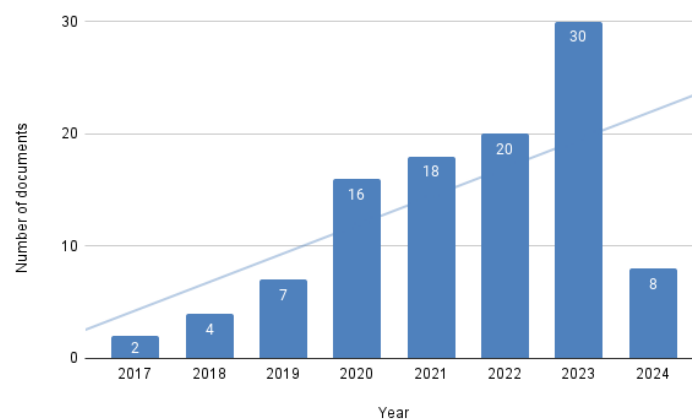


Figure 10. Annual evolution of the number of published documents (2017–2024).

5.5. Strength of Evidence

As a research methodology, an SLR is the optimal method to objectively evaluate and synthesize research evidence for specific RQs. However, a critical question arises: how much can we trust the findings and generalizations presented in an SLR? A reliable approach is to assess the quality of the primary studies [79].

This quality assessment (QA) establishes a solid foundation of confidence in the strength of the evidence and the credibility of the conclusions [80]. This process involves critically evaluating the included studies to identify potential biases and methodological weaknesses. By assessing the quality of studies, researchers can distinguish between studies that provide robust evidence and those with methodological flaws that might compromise them [81].

This QA strengthens the review’s integrity and guides the interpretation and application of the results in evidence-based practices. Therefore, omitting this step could lead to erroneous conclusions and recommendations for policies or practices not adequately supported by robust evidence.

To perform the QA, we used the criteria defined by Dybå and Dingsøyr [82]. This study proposes a detailed approach to assessing the quality of empirical studies, using specific criteria covering study design, execution, analysis, and the credibility of the findings. The authors used this quality data to select relevant literature and support conclusions, ensuring that only high-quality studies influence the results. The article posed several crucial questions for assessing the quality of the included empirical studies. These questions focus on assessing the studies’ validity, reliability, and applicability.

During the data extraction process, the principal author checked the questions in Table 13 for each selected paper. Every question can be answered with *Yes*, *Partially*, or *No*. Other authors have adopted similar QA criteria [83–85]. Figure 11 shows how the papers answer each QA question.

Table 13. QA questions.

ID	QA Question	Yes # (%)	Partially # (%)	No # (%)
QA1	Is the aim of the research sufficiently explained?	105 (100%)	0	0
QA2	Is the paper based on research methodology?	86 (82%)	5 (5%)	14 (13%)
QA3	Is there an adequate description of the context in which the research was carried out?	105 (100%)	0	0
QA4	Are threats to validity taken into consideration?	28 (27%)	48 (46%)	29 (27%)
QA5	Is there a clear statement of findings?	102 (97%)	3 (3%)	0

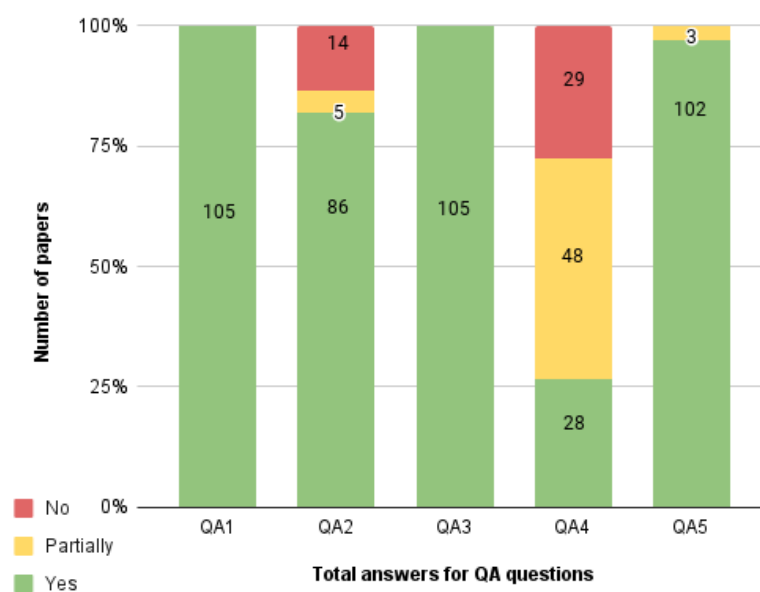


Figure 11. Quality assessment questions.

With QA1, we meticulously assessed whether the authors clearly stated the research's aims and objectives. We reviewed each paper's abstract, introduction, primary findings, and discussions. All the reviewed publications answered this question positively, affirming the robustness of our review process.

With QA2, we asked if the study was created using a research methodology. To do this, we meticulously reviewed the structure of the paper, looking for the methods and models used. A total of 91 reviewed publications answered this question positively ('yes' or 'partially'), underscoring the importance of this criterion. This crucial aspect distinguishes the study from a 'lessons learned' report based on expert opinion (14 papers).

For QA3, we ensured that the study provided ample information to establish the research's context and background, directly or through references to relevant literature. This task encompassed elements such as the problem statement, research gap, and theoretical framework, which we diligently sought in the introduction, related work, or general concepts sections. All publications satisfactorily addressed this crucial aspect, fostering trust in the research's solid foundation.

QA4 was marked 'Yes' if the study explicitly discussed validity threats. However, if the study merely mentioned them without adequately explaining how they were identified or addressed, the answer was 'Partially'. We meticulously sought validity threats, such as selection bias, measurement error, or confounding variables, in the paper's threats to validity, discussion, or limitations sections. Then, 73% of the papers did not address the potential threats to validity ('no' or 'partially'), highlighting the need for more comprehensive discussions in empirical studies.

With QA5, we ensured that the research's outcome was adequately documented. We thoroughly examined the discussion, main findings, conclusions, and future work sections to do this. This question was answered positively ('yes' or 'partially') for all the studies, providing strong reassurance about the quality of the research papers under review.

The high rates in all the QA questions but one were not surprising, given the rigorous selection process we undertook to ensure the validity and reliability of our research.

Once we had answered the QA questions for all the selected papers, we employed the comprehensive GRADE system to evaluate the strength of evidence provided by the selected papers [86]. This system, known for its thoroughness and broad usage in other disciplines, such as medicine, ensures a robust evaluation process [87–89]. It evaluates four critical criteria: study design, study quality, study consistency, and directness. Each criterion is rated high, moderate, low, and very low, ensuring a robust evaluation process.

We can establish the value for this criterion as high concerning study design and according to QA1, QA2, and QA3 in Table 13. For the study quality, the data collected for QA4 and QA5 suggests that the value for this criterion is low. The consistency could not be assessed because we considered that the studies in our review were not comparable.

Finally, in terms of directness, the selected papers did not present substantial evidence of adoption in the industry; none of the proposals originated in the industry. Therefore, it is likely that these studies have had an insignificant effect on industrial practices, although they have contributed relevant data to the research community. Directness can thus be rated as low.

Given the limitations we encountered in the primary studies we selected, we have rated the strength of evidence as moderate. This rating underscores the need for further efforts to consolidate the quality and maturity of primary studies and increase the adoption of RE for QC. Only through these efforts could the QRE be considered a mature discipline with a significant impact on industrial practices.

6. Discussion

In the following discussion, we examine the findings detailed previously. Section 6.1 provides insights and interpretations concerning the RQs posed in the study. Section 6.2 is dedicated to presenting a bibliometric analysis of the data. Lastly, Section 6.3 addresses potential validity threats to this systematic review study.

6.1. Interpreting the Answers to RQs

Next, we present a detailed analysis of the findings for each RQ. This analysis is crucial to understanding how to create, adapt, and evolve the RE practices within the emerging context of QC and quantum software development.

6.1.1. Interpreting the Answer to RQ1

According to the evidence gathered to answer RQ1, we can state that our study addresses the primary challenges and approaches in RE for QC.

RE in QC faces unique challenges that differ significantly from those in classical SE. The main challenges identified in the literature include:

- Inherent complexity of quantum systems,
- Lack of specific tools and methodologies,
- Uncertainty in the behavior of quantum systems,
- Standardized approaches for quantum systems.

A recurring challenge is managing complexity in quantum systems. Unlike classical systems, quantum systems operate under the principles of quantum mechanics, introducing a significantly higher level of complexity and abstraction. This challenge reflects the need to develop new modeling and specification tools to capture quantum phenomena adequately.

Another challenge is the need for more specific tools and methodologies. Currently, the tools and methodologies used in RE are not designed to handle the peculiarities of quantum systems. For example, traditional modeling and analysis techniques cannot fully capture quantum states' probabilistic and superposition nature.

The uncertainty in the behavior of quantum systems is another significant challenge. Quantum systems are inherently probabilistic, meaning their behavior can be unpredictable compared to deterministic classical systems. This uncertainty complicates the task of defining and verifying system requirements.

One of the most significant challenges identified is more standardization in RE approaches for quantum systems. This gap makes it difficult to compare and synthesize results from different studies. Furthermore, the emerging nature of the field means many studies still need to be in conceptual stages, limiting the immediate applicability of some findings. Figure 12 shows the main RE challenges identified vs. RE stages.

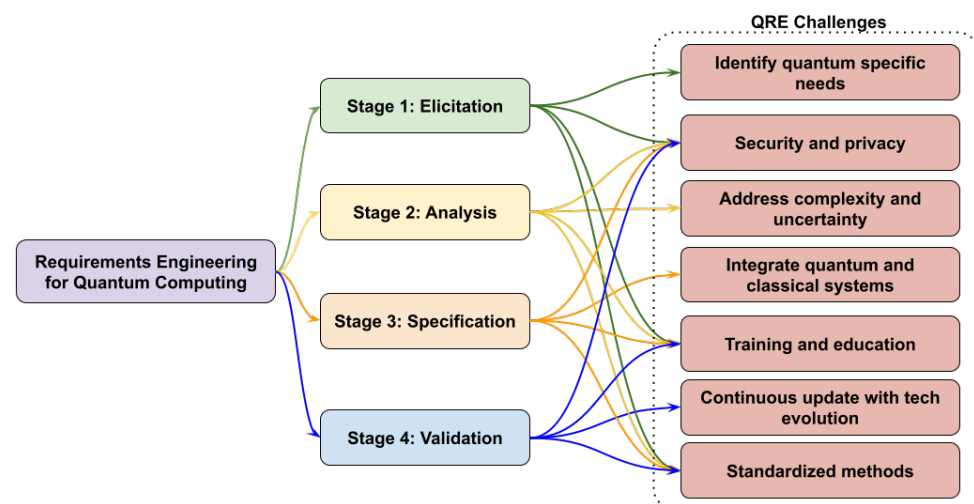


Figure 12. RE stages vs. main identified challenges.

Some implications for researchers, from the answer to RQ1 findings, extend the existing theoretical framework by introducing new paradigms of modeling and specifications unique to the quantum domain. These include the need for probabilistic models and formal techniques to capture quantum behavior. Additionally, it highlights the importance of adapting RE principles to address the peculiarities of quantum systems.

For practitioners, these findings imply the need for continuous education and the development of specialized tools that can be integrated with agile development methodologies tailored to the peculiarities of quantum systems. Software engineers must acquire knowledge of quantum physics and advanced mathematics to work effectively in this emerging field.

6.1.2. Interpreting the Answer to RQ2

According to the evidence gathered to answer RQ2, this SLR addresses the primary advances and opportunities in RE for QC.

Developing quantum-specific requirements techniques is crucial for enhancing quantum software specifications. Standardization efforts are essential to establish consistent and reliable practices and ensure developers follow common frameworks. These efforts help ensure quantum software meets high quality and performance standards, facilitating broader adoption and integration into existing systems.

Improved modeling and simulation tools are necessary to support quantum software development. Developing simulation tools that integrate quantum and classical systems is vital for testing and validating hybrid applications, ensuring robustness and efficiency. Interdisciplinary collaborations can enhance requirement engineering strategies, leveraging expertise from various fields to create comprehensive solutions for quantum software.

Effective quantum software development requires collaboration among computer scientists, physicists, and engineers. Such interdisciplinary efforts are essential for developing robust, scalable quantum applications. Specialized training programs are needed to equip requirement engineers with quantum-specific skills, ensuring they can address the unique challenges of QC.

Finally, developing quantum-resistant cryptographic algorithms and protocols is critical as QC advances. Ensuring quantum software incorporates strong security measures is essential for safeguarding sensitive information and maintaining trust in quantum technologies. Figure 13 shows the main RE advances identified vs. RE stages.

The need to develop new methodologies for QRE offers significant research opportunities, bridging QC and SE. This field promotes interdisciplinary collaborations among computational scientists, physicists, and engineers, leading to comprehensive studies that address both the theoretical and practical challenges of QRE. Adapting modeling and specification techniques to quantum contexts enriches theoretical models applicable to

various quantum domains, significantly advancing academic knowledge and practical developments in reliable quantum software systems.

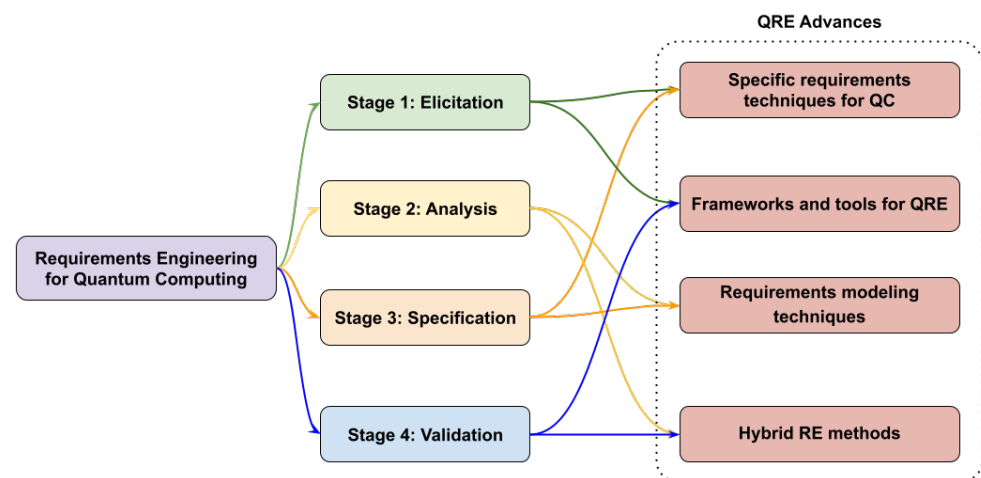


Figure 13. RE stages vs. main identified advances.

Practitioners must continually update their knowledge of new techniques and tools for managing quantum requirements, integrating advanced quantum physics and specific modeling techniques. Specialized frameworks and tools will facilitate the transition to quantum software development, enhancing the efficiency and reliability of the RE process. Validation and verification of quantum requirements will ensure the quality and robustness of quantum systems, which are essential for their proper functioning in practical applications and meeting high standards for critical deployments across industries.

6.1.3. Interpreting the Answer to RQ3

According to the evidence gathered to answer RQ3, this SLR addresses future research directions in RE for QC.

QC presents unique challenges and opportunities in RE. This discussion explores emerging trends and future directions identified in recent research, highlighting how these developments can advance RE specifically for QC software.

A critical future direction is formalizing RE processes specific to QC. This formalization involves developing standardized methodologies to specify, model, and analyze quantum software requirements. These methodologies must address the unique features of QC, such as superposition and entanglement, which are absent in classical computing.

Another emerging trend is the creation of sophisticated tools for validating and verifying quantum software requirements. Given the current state of quantum hardware and algorithms, these tools are essential to ensure that the specified requirements are correct, complete, and feasible.

Standardizing architectural practices and methodologies for quantum software is also crucial. This standardization helps create mature, reliable, fault-tolerant quantum systems, ensuring their stability and performance. Due to the rapid evolution of quantum technologies, there is a need for methodologies that support dynamic updates to requirements and foster interdisciplinary collaboration.

Adapting RE methodologies specifically for QC is essential. These methodologies must address QC's probabilistic nature and rapid technological advancements. Advanced modeling tools that accurately represent quantum systems' complex behaviors are also essential. These tools should provide precise modeling and testing capabilities for quantum requirements.

Educational programs and resources are necessary to equip engineers with the skills and knowledge required for quantum RE. These initiatives help bridge the knowledge gap in this emerging field. Another promising trend is the development of AI-driven

automated tools to support the RE process. These tools aim to reduce manual effort and increase accuracy.

Hybrid modeling techniques that integrate quantum and classical computing elements are essential for seamless requirements modeling. Additionally, creating well-defined software development lifecycles specifically for quantum applications is crucial. This software process involves structured frameworks for developing, testing, and maintaining quantum software.

Enhancing methodological support for RE in QC is necessary to manage the complexities of quantum systems. These efforts collectively contribute to advancing the field of quantum RE, ensuring the development of reliable and efficient quantum software systems. Figure 14 shows the main RE trends and future research identified vs. RE stages.

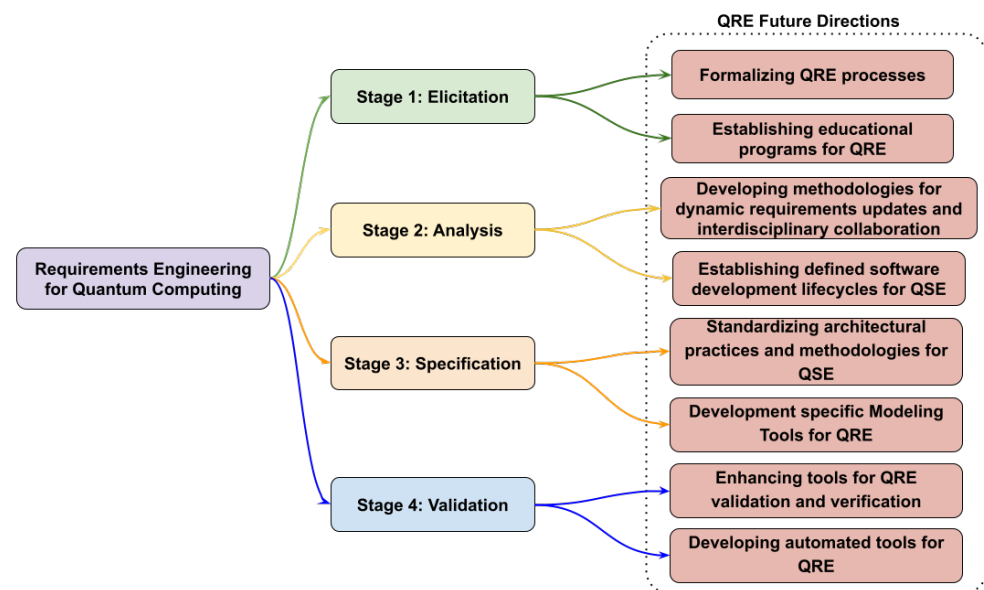


Figure 14. RE stages vs. trends and future research.

Finally, we identify challenges and opportunities for progress in the RE for QC by analyzing the selected papers. These challenges and opportunities give rise to new research questions, the answer to which is beyond the scope of this paper. Other RQs to be addressed in the future include: what adaptations need to be made to classical RE methodologies for use in QC? What techniques and tools are needed to specify and validate requirements in hybrid quantum-classical systems? How can quantum security principles be integrated into the RE process, and how can quantum security principles be integrated into the RE process?

Answering these questions will require simulations and case studies conducted in collaboration with the industry to validate the proposed methods and techniques and control experiments to assess the effectiveness of new methodologies, techniques, or tools.

As we conclude, it is clear that promoting research in QRE will require collaboration between academia and industry. This collaboration is not just about sharing knowledge and resources, but also about applying academic advances in industrial environments. Establishing agreements and joint projects that facilitate technology transfer and the practical application of academic advances will be crucial in this endeavor.

6.1.4. Quality Attributes on Quantum Computing

The identification of quality attributes is a critical aspect of software development. The development of software for QC is no exception, and it must accomplish these attributes to ensure the reliability, efficiency, and scalability of quantum software systems.

Considering the nascent and complex nature of QC, understanding and addressing the quality attributes becomes critical. These attributes influence the software's operational

effectiveness and its adoption and integration within existing technological infrastructures. Figure 15 summarizes the quality attributes identified from the selected papers. Two authors reviewed each paper and documented the quality attributes explicitly mentioned. It is important to note that the quality attributes shown in the figure correspond to those shown in the RQ responses in Figures 3–5.



Figure 15. Weighted word cloud for quality attributes from selected papers.

6.1.5. Security and QRE

The advent of QC introduces significant security challenges, necessitating new approaches in QRE. Integrating quantum cryptography and post-quantum cryptography is essential to protect data against quantum computers' advanced capabilities. These cryptographic methods offer robust security solutions that can withstand quantum attacks, ensuring the privacy and integrity of digital communications.

Quantum security scenarios are critical for developing secure protocols [90]. Hybrid encryption approaches enhance security and efficiency, particularly for resource-constrained devices like those on the IoT [91]. Transitioning to post-quantum cryptography requires significant standardization efforts [92]. Integrating post-quantum cryptography into industrial control and cyber-physical systems ensures long-term protection against quantum threats, with hybrid solutions offering a practical approach for current implementations [93].

Quantum-specific security requirements are crucial to address the new security threats posed by quantum computing, such as vulnerabilities in classical cryptographic methods. This challenge involves specifying requirements for integrating quantum-resistant cryptographic protocols to protect sensitive data within QC applications. The unique properties of QC necessitate that RE explicitly addresses these security concerns to ensure robust protection against quantum threats. Future research in quantum security focuses on optimizing quantum-resistant algorithms to enhance their efficiency and scalability. Significant emphasis is also placed on developing comprehensive security frameworks that integrate quantum capabilities into existing IoT systems and other platforms, ensuring robust protection against classical and quantum threats.

Finally, incorporating quantum and post-quantum cryptography into QRE and hybrid system development is crucial for future-proofing security infrastructures. These technologies provide robust, quantum-resistant solutions that address existing and emerging security challenges. By adopting these advanced cryptographic methods, organizations can ensure the long-term security of their data and systems in the evolving quantum landscape.

6.1.6. Summary

The answer to RQ1 has revealed that RE for QC faces unique and complex challenges. The lack of standardization, specific tools, and quantum systems' inherent uncertainty underscores the need for methodological and educational innovations in this field. Integrat-

ing knowledge of quantum physics with advanced SE practices is crucial to overcoming these challenges.

The response to RQ2 indicates that RE for QC show various advancements. These advancements include developing techniques specific to quantum specifications, enhancing modeling tools, adopting hybrid approaches, fostering interdisciplinary collaborations, standardizing practices, and improving educational programs.

The answer to RQ3 has revealed emerging trends in QRE, including formalizing RE processes, developing sophisticated validation tools, and standardizing architectural practices. It is crucial to create methodologies for dynamic updates and interdisciplinary collaboration, along with advanced modeling tools and educational programs. These trends aim to enhance the reliability and efficiency of quantum software systems.

Finally, the evolution of QC promises to transform various industrial sectors, but its realization largely depends on our ability to address these RE challenges, advances, and further research. Therefore, there is a call for interdisciplinary collaboration, encouraging research that develops new methodologies and adapts existing ones to meet the demands of quantum software development. This holistic approach will facilitate theoretical advancement and have a significant practical impact, enabling more robust and effective quantum applications.

6.2. Bibliometric Analysis

We conducted an initial bibliometric analysis of the selected papers. This analysis aims to identify the most relevant terms and authors and their relationships.

6.2.1. Keywords and Relevant Concepts

Figure 16 provides a first impression of the scope of the selected papers by showing a simple weighted word cloud. This word cloud was generated from the titles and abstracts of the selected papers, considering the fifty most relevant terms. The results align with this study's scope (e.g., quantum, software, requirements, engineering, challenges, and computing among others).



Figure 16. Weighted word cloud from selected papers.

In addition, the most relevant concepts were extracted from the titles and abstracts of the selected papers. We used a web service called Termine, which is freely available from the academic domain of the University of Manchester. This service is based on the C-/NC-value method [94]. A total of 1347 relevant concepts were obtained. We present a top 40 list of terms in Figure 17, ordered by relevance (score) in descending order. Through the analysis of the obtained concepts, we visualize topics associated with QC, SE, RE, quantum software development, and their subsequent adaptations. The results support the appropriateness of the inclusion of the selected papers in our study.

Rank	Term	Score
1	quantum computing	131.785721
2	quantum software	118.273682
3	quantum computer	66.897438
4	quantum software engineering	44.167622
5	software engineering	44.029411
6	quantum software development	33.140121
7	software development	30.111111
8	quantum algorithm	27
9	quantum circuit	22.692308
10	quantum programme	21
11	quantum gate	11.5
12	quantum technology	11
13	quantum computation	10.9
14	quantum state	10.8
15	noisy intermediate-scale quantum	10.302257
16	quantum programming	10.285714
17	classical computer	10
17	programming language	10
19	quantum hardware	8.888889
20	life cycle	8.777778
21	signature generation	8
21	software engineer	8
23	quantum software system	7.924812
23	quantum programming language	7.924812
25	quantum machine	7.333333
26	classical algorithm	7
26	quantum advantage	7
28	software application	6.571428
29	machine learning	6.555555
30	quantum machine learning	6.33985
30	quantum circuit simulation	6.33985
30	quantum computing system	6.33985
30	quantum computing technology	6.33985
34	physical qubits	6
34	architecture decision	6
34	classical computing	6
37	requirement engineering	5.888889
38	software architecture	5.857143
39	mobile payment	5.8
40	sealed-bid auction	5.666667

Figure 17. Forty most relevant concepts.

6.2.2. Relationship—Most Relevant Terms

Figure 18 shows the relationship between the most relevant terms for RE in the quantum computing domain for the selected papers. The size and color of each circle correspond to the relevance and the evolution of terms over time, respectively. In the figure, it is possible to observe 46 clusters, including 131 terms. Then, we built a thesaurus to focus on more specific terms, and also, we unified the terms and all their variants under a single term (e.g., terms such as QC, quantum computer, quantum computation, and quantum restrictions). Finally, we discarded the less frequent and less relevant terms to focus exclusively on the most relevant ones.

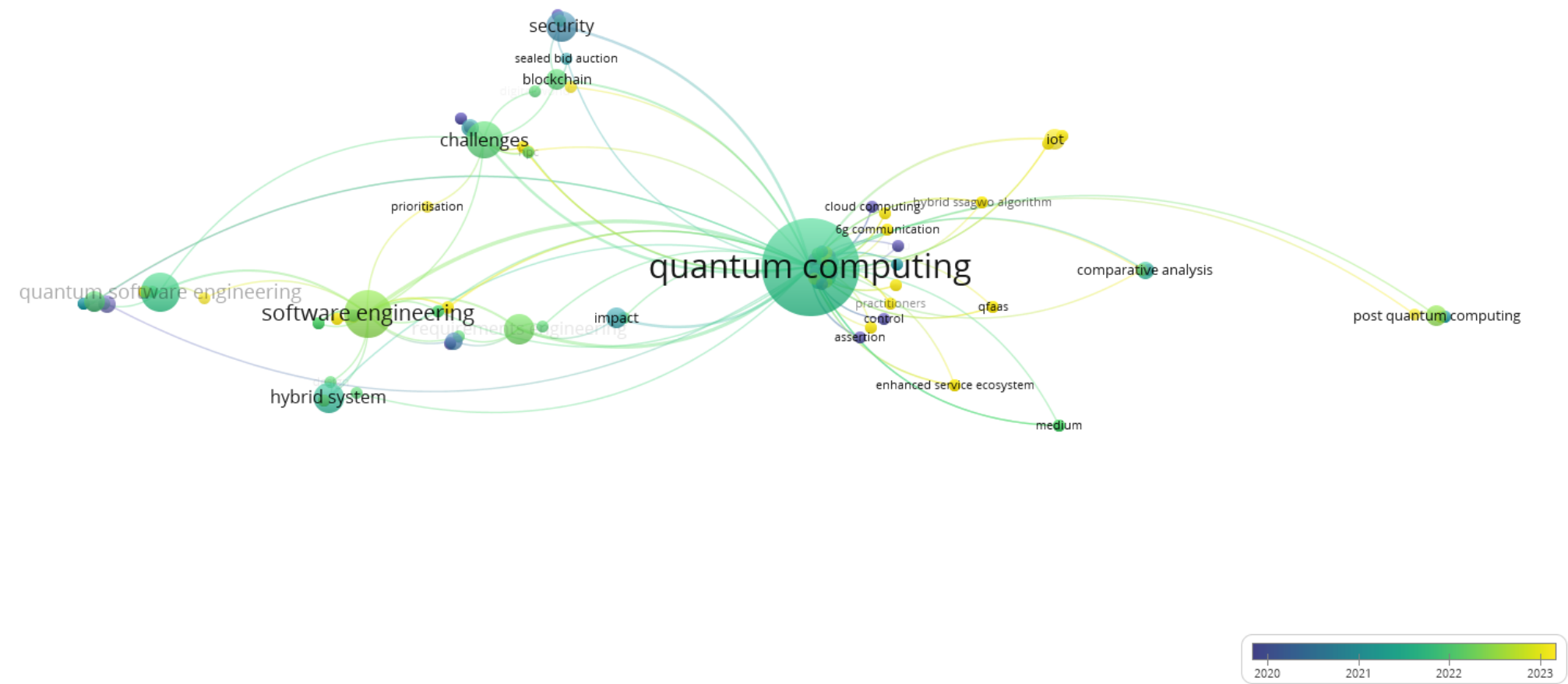


Figure 18. The 100 most relevant authors.

Through the clusters, we can visualize a connection between keywords and titles, and keeping QC and SE as the central axis, there is a solid connection between RE, quantum software engineering, hybrid systems and security concerns. Between 2020 and 2021, we observed a connection between physics and QC, observing topics such as constraint programming or formal languages. Between 2021 and 2022, a boom is visualized in SE and RE proposals, focusing on QRE. Between 2022 and 2024, QC was connected with IoT and cloud-based systems. Also, the post-QC concept appears, generating an evolution of QC application areas.

6.2.3. Relationship—Most Relevant Authors

Figure 19 shows the relationships between the most relevant authors and teams. The figure includes the 100 most relevant authors in the RE and QC domains. The size and color of each circle correspond to the relevance and the evolution of these collaborations over time, respectively. The lines connecting the circles show the groups of authors working together.

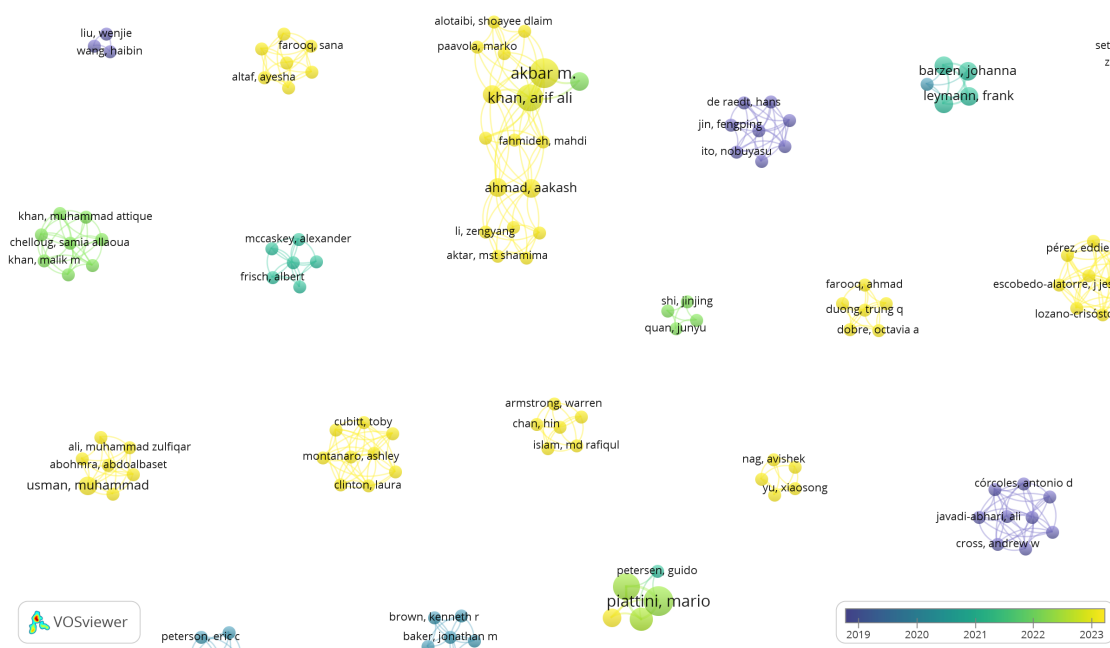


Figure 19. The 100 most relevant authors.

6.3. Threats to Validity

Despite the care taken in the definition and execution of an SLR protocol, SLRs suffer from some well-known limitations and threats to validity [95,96]. These limitations include the potential need for more exhaustiveness in the literature search due to the emerging nature of the field. The number of empirical studies available is also limited, which may affect the generalizability of the findings.

Next, we present the proactive measures taken to mitigate the effects of these limitations and threats to the validity of the secondary studies [97]. By addressing these issues head-on, we aim to enhance the reliability and credibility of our research.

6.3.1. Descriptive Validity

This validity criterion ensures that observations are described objectively and accurately independent of the researcher. The mitigation actions considered:

- The information to be collected was structured using various forms of data extraction through a Google Sheets data spreadsheet to support uniform data recording and ensure the objectivity of the data extraction process.
- We held weekly meetings to unify critical concepts with the research and classification criteria, answer any questions, and demonstrate how to carry out the process.

6.3.2. Theoretical Validity

This validity criterion is associated with obtaining the information to be captured. The associated mitigation actions considered:

- We built a search string and adapted it to the data sources defined.
- We defined exclusion and inclusion criteria to guarantee objectivity in the selection process.
- We performed cross-checks among researchers to visualize the criteria's applicability.
- Including articles written in English and discarding studies in other languages could have a minimal impact on this criterion.
- We expanded the scope of the study with a first snowballing search review, according to Wohlin's guidelines [75], obtaining eight additional papers for the study.

6.3.3. Generalizability

This validity criterion refers to generalizing the results to the entire domain. The associated mitigation actions considered:

- We ensured that the scope of RQs was broad enough to identify and classify results on different QC and RE approaches, regardless of specific cases and domains, among others.

6.3.4. Interpretive Validity

Given the data, this validity criterion is met when the study's conclusions are reasonable. The associated mitigation actions considered:

- All researchers reviewed and validated the conclusions of the study.
- A researcher with expertise in the RE area assisted us in interpreting the data.

6.3.5. Repeatability

This validity criterion ensures that the research approach is well-detailed and its results can be replicated thoroughly. The associated mitigation actions considered:

- We previously published an initial protocol through the arXiv platform [71].
- We designed a detailed protocol (see Section 4) so other researchers can repeat the process and corroborate the results.
- We validated the structure of our report using the 2020 PRISMA Statement (Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA), <https://www.prisma-statement.org/> (Preferred Reporting Items for Systematic Reviews and Meta-Analyses). This statement is a checklist that contains the minimum set of items that should be reported in SLR and meta-analyses [98]. We accomplished all the items except those assessing quantitative survey data and comparing them (items n° 13a-13f, 19, 20a-20d, and 24a).

The whole PRISMA Checklist (Supplementary Materials) is published in the Zenodo platform at <https://zenodo.org/records/12827582>.

7. Conclusions

This paper presented a novel systematic literature review (SLR) in the QC domain from 2017 to 2024. A total of one hundred and five papers were selected, each proposing, adapting, discussing, or utilizing methods, tools, or techniques for various RE activities. Three specific research questions (RQs) were formulated to delve into the challenges, advancements, and future directions of RE in QC. A well-defined data extraction procedure was implemented, and the synthesized data allowed us to provide comprehensive answers to each of the RQs.

The work has demonstrated the necessity of adapting and developing new methodologies that address the unique characteristics of quantum systems. The results obtained reveal that QC offers significant opportunities to advance RE, such as formalizing processes specific to QC, developing advanced modeling techniques, and standardizing architectural practices.

The response to RQs highlighted how traditional techniques can be adapted, identifying unique opportunities, such as integrating quantum phenomena into RE processes. Also,

emerging trends and future directions emerged, emphasizing the importance of interdisciplinary collaboration and the need for innovative theoretical and practical frameworks.

From the answer to RQ1, we observe that Quantum Computing (QC) presents unique challenges in RE. These challenges include the complexity of specifying requirements due to quantum mechanics properties like superposition and entanglement, integrating quantum and classical systems, and the need for standardized methodologies. The rapid evolution of QC technology demands continuous updates to requirements, and the educational gap among engineers hampers the effective translation of quantum capabilities into software requirements. Additionally, quantum-specific security requirements and the complexity of testing and validation add to the challenges.

Answering RQ2, QC offers significant opportunities for advancements in RE. Enhanced specification techniques and improved modeling tools are needed to address quantum properties. Hybrid RE approaches combining classical and quantum methodologies can bridge gaps between the two paradigms. Interdisciplinary collaboration is crucial for robust RE strategies; standardization efforts can create consistent frameworks. Educational programs are essential to equip engineers with QC-specific knowledge, and advancements in cryptography are necessary for developing quantum-resistant algorithms.

According to the answer for RQ3, future directions in RE for QC include formalizing QRE processes, developing advanced validation tools, and standardizing architectural practices. Dynamic methodologies and interdisciplinary collaboration are essential to keep up with rapid technological advances. Tailored methodologies, advanced modeling tools, and educational programs are critical for bridging knowledge gaps. Automated tools can increase accuracy, and hybrid modeling techniques are needed for seamless requirement modeling. Well-defined development lifecycles and improved methodological support are necessary for handling quantum system complexities. Addressing these challenges through interdisciplinary collaboration and new methodologies will enable more robust and practical quantum applications.

The impact and potential of these findings are profound, not only for researchers but also for professionals in the field. For researchers, these findings open up new areas of study and opportunities for collaboration, paving the way for the development of robust theories and tools that can support the evolution of quantum technologies. For professionals, the adoption of these methodologies and tools will not only enhance the efficiency and reliability of quantum software development but also ensure its quality and robustness for practical applications. In essence, this work lays the foundation for a comprehensive RE-QC framework, facilitating the transition to an era of advanced computing and promoting the development of reliable and effective quantum systems.

Future research should focus on developing standardized methodological frameworks and support tools that address the needs of RE in QC. Robust empirical research is also needed to validate and refine these approaches in practical contexts. Initiatives like creating a repository of case studies and practical examples could serve as a reference for future work.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/electronics13152989/s1>, PRISMA checklist.

Author Contributions: Conceptualization, S.S., G.F., and A.C.; methodology, S.S.; validation, S.S. and A.C.; formal analysis, S.S., G.F., L.A., and A.C.; investigation, S.S., A.C., G.F., and L.A.; resources, S.S.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, S.S., A.C., G.F., and L.A.; visualization, S.S. and A.C.; supervision, S.S. and A.C.; project administration, S.S. and A.C.; funding acquisition, S.S. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: Agencia Nacional de Investigación y Desarrollo ANID, Fondecyt de Iniciación 2024, research project 11240702.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data are contained within the article.

Acknowledgments: Samuel Sepúlveda thanks to ANID—Fondecyt de Iniciación, research project 11240702. Ania Cravero thanks to Universidad de La Frontera, Vicerrectoría de Investigación y Postgrado, research project GI23-0012. Special thanks to Fernanda Gutiérrez for her helpful technical support in this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACM	Association for Computing Machinery
CORE	Computing Research and Education Association of Australasia
GRADE	Grading of Recommendations, Assessment, Development, and Evaluations
IEEE	Institute of Electrical and Electronics Engineers
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
PICOC	Population, Intervention, Comparison, Outcomes, Context
QA	Quality Assessment
QC	Quantum Computing
QRE	Quantum Requirements Engineering
QSE	Quantum Software Engineering
RE	Requirements Engineering
RQ	Research Question
SCI-JCR	Science Citation Index—Journal Citation Report
SE	Software Engineering
SLR	Systematic Literature Review
WoS	Web of Science

Appendix A. Detailed RQs and Answers from Related Work

Table A1 shows the main characteristics of each related work. These characteristics include goal, RQs, number of papers (#) and time span, and the main results.

Table A1. Summary—related work.

Ref.	Goal	RQs	# Papers and Time Span	Results
[8]	Present a wide review on the particularities and characteristics of software that are developed for QC.	- What are the main ‘programming infrastructures’ for developing quantum software? - What is the difference between developing conventional software and quantum software? - Is quantum software more suited to a particular application domain?	24 2010–21	Identifying the main existing programming infrastructures for QC, some differences that QC can bring to software development, and presenting some application domains to which QC is more suited.
[1]	Bridge the gap, starting with the QC workflow and by mapping existing SE research to this workflow	How can current SE practices be adapted to efficiently and reliably develop quantum software applications?	35 2015–23	Identification of directions for SE research for QC.

Appendix B. Selected Papers—Final List

Next, in Table A2, we present the final list of selected papers for this SLR. This list includes: paper id, title, authors and publication year. In addition to this table, we prepared a list of the details of each selected paper. Also, we included a file with all references

in BibTex format. All this additional material is available on the Zenodo platform at <https://zenodo.org/records/12760928>.

Table A2. Final list of papers included in this SLR.

ID	Title	Authors	Year
3	A comparative analysis of quantum-based approaches for scalable and efficient data mining in cloud environments	Sudharson, K.; Alekhya, B.	2023
7	A dynamic programming approach to multi-objective logic synthesis of quantum circuits	Rajaei, A.; Houshmand, M.; Hosseini, S. A.	2023
8	A Generic IoT Quantum-Safe Watchdog Timer Protocol	Eckel, M.; Gutsche, T.; Lauer, H.; Rein, A.	2023
12	A New Heuristic for N-Dimensional Nearest Neighbor Realization of a Quantum Circuit	Kole, A.; Datta, K.; Sengupta, I.	2018
13	A new post-quantum voting protocol based on physical laws	Sun, Z.; Gao, W.; Dong, H.; Xie, H.; Yang, L.	2022
14	A Novel Hierarchical Security Solution for Controller-Area-Network-Based 3D Printing in a Post-Quantum World	Cultice, T.; Clark, J.; Yang, W.; Thapliyal, H.	2023
16	A quantum deep convolutional neural network for image recognition	Li, Y.; Zhou, R.; Xu, R.; Luo, J.; Hu, W.	2020
17	A quantum inspired hybrid SSA-GWO algorithm for SLA based task scheduling to improve QoS parameter in cloud computing	Jain, R.; Sharma, N.	2023
18	A quantum-classical cloud platform optimized for variational hybrid algorithms	Karalekas, P. J.; Tezak, N. A.; Peterson, E. C.; Ryan, C. A.; da Silva, M. P.; Smith, R. S.	2020
22	A Software Architecting for Quantum Machine Learning Platform in Noisy Intermediate-Scale Quantum Era	Wenbin, Y.; Yuhao, C.; Chengjun, Z.; Yadang, C.; Hongyu, W.; Zongyuan, C.; Yifan, Z.	2023
23	A software methodology for compiling quantum programs	Häner, T.; Steiger, D. S.; Svore, K.; Troyer, M.	2018
24	A Third-Party Mobile Payment Scheme Based on NTRU Against Quantum Attacks	Xia, Y.; Ying, C.; Lin, G.; Sun, Z.	2019
26	Accelerating HPC With Quantum Computing: It Is a Software Challenge Too	Schulz, M.; Ruefenacht, M.; Kranzlmüller, D.; Schulz, L. B.	2022
28	Agile Meets Quantum: A Novel Genetic Algorithm Model for Predicting the Success of Quantum Software Development Project	Khan, A. A.; Akbar, M. A.; Lahtinen, V.; Paavola, M.; Niazi, M.; Alatawi, M. N.; Alotaibi, S. D.	2024
29	Agile practices for quantum software development: practitioners' perspectives	Khan, A. A.; Akbar, M. A.; Ahmad, A.; Fahmideh, M.; Shameem, M.; Lahtinen, V.; Waseem, M.; Mikkonen, T.	2023
33	An efficient quantum algorithm for ensemble classification using bagging	Macaluso, A.; Clissa, L.; Lodi, S.; Sartori, C.	2024
34	An enhanced architecture to resolve public-key cryptographic issues in the internet of things (IoT), Employing quantum computing supremacy	Shamshad, S.; Riaz, F.; Riaz, R.; Rizvi, S. S.; Abdulla, S.	2022
38	Analysis of physical requirements for simple three-qubit and nine-qubit quantum error correction on quantum-dot and superconductor qubits	Sohn, I.; Tarucha, S.; Choi, B.	2017

Table A2. Cont.

ID	Title	Authors	Year
39	Approximating Decision Diagrams for Quantum Circuit Simulation	Hillmich, S.; Zulehner, A.; Kueng, R.; Markov, I. L.; Wille, R.	2022
40	Architecture Decisions in Quantum Software Systems: An Empirical Study on Stack Exchange and GitHub	Aktar, M. S.; Liang, P.; Waseem, M.; Tahir, A.; Ahmad, A.; Zhang, B.; Li, Z.	2023
41	Assertion-Based Optimization of Quantum Programs	Haener, T.; Hoefler, T.; Troyer, M.	2020
43	Barriers of adopting quantum technology in blockchain: a prioritization-based framework	Alahmari, M.	2023
45	Challenges and Opportunities in Quantum Software Architecture	Yue, T.; Mauerer, W.; Ali, S.; Taibi, D.	2023
46	Challenges and Opportunities of Near-Term Quantum Computing Systems	Corcoles, A. D.; Kandala, A.; Javadi-Abhari, A.; McClure, D. T.; Cross, A. W.; Temme, K.; Nation, P. D.; Steffen, M.; Gambetta, J. M.	2020
47	Challenges in making blockchain privacy compliant for the digital world: some measures	Bansod, S.; Ragha, L.	2022
52	Classical to quantum software migration journey begins: a conceptual readiness model	Akbar, M. A.; Rafi, S.; Khan, A. A.	2022
56	Comparative analysis of classical and post-quantum digital signature algorithms used in Bitcoin transactions	Noel, M. D.; Waziri, O. V.; Abdulhamid, M. S.; Ojeniyi, A. J.; Okoro, M. U.	2020
62	Continuous-Variable Deep Quantum Neural Networks for Flexible Learning of Structured Classical Information	Basani, J. R.; Bhattacharjee, A.	2021
63	Control and Readout Software for Superconducting Quantum Computing	Guo, C.; Liang, F.; Lin, J.; Xu, Y.; Sun, L.; Liu, W.; Liao, S.; Peng, C.	2019
75	Design of classical-quantum systems with UML	Pérez-Castillo, R.; Piattini, M.	2022
85	Engineering the development of quantum programs: Application to the Boolean satisfiability problem	Alonso, D.; Sánchez, P.; Sánchez-Rubio, F.	2022
87	Enhancing IoT Security: Quantum-Level Resilience against Threats	Alhakami, H.	2024
89	Epoque: practical end-to-end verifiable post-quantum-secure e-voting	Boyen, X.; Haines, T.; Müller, J.	2021
93	Experimental study on the quantum search algorithm over structured datasets using IBMQ experience	Das, K.; Sadhu, A.	2022
95	Extending the Frontier of Quantum Computers With Qutrits	Gokhale, P.; Baker, J. M.; Duckering, C.; Chong, F. T.; Brown, K. R.; Brown, N. C.	2020
109	Guidelines to use the incremental commitment spiral model for developing quantum-classical systems	Pérez-Castillo, R.; Serrano, M. A.; Cruz-Lemus, J. A.; Piattini, M.	2024
118	Hybrid Quantum-Classical Computing for Future Network Optimization	Fan, L.; Han, Z.	2022
140	Massively parallel quantum computer simulator, eleven years later	De Raedt, H.; Jin, F.; Willsch, D.; Willsch, M.; Yoshioka, N.; Ito, N.; Yuan, S.; Michielsen, K.	2019
143	Minimum hardware requirements for hybrid quantum-classical DMFT	Jaderberg, B.; Agarwal, A.; Leonhardt, K.; Kiffner, M.; Jaksch, D.	2020
147	Modeling Quantum programs: challenges, initial results, and research directions	Ali, S.; Yue, T.	2020
152	Navigating the Quantum Threat Landscape: Addressing Classical Cybersecurity Challenges	Sokol, S.	2023
156	Non-Functional Requirements for Quantum Programs	Saraiva, L.; Haeusler, E. H.; Costa, V.; Kalinowski, M.	2021

Table A2. Cont.

ID	Title	Authors	Year
159	On testing and debugging quantum software	Miranskyy, A.; Zhang, L.; Doliskani, J.	2021
160	On the definition of quantum programming modules	Sánchez-Palma, P.; Alonso-Cáceres, D.	2021
161	On the Development of a Protection Profile Module for Encryption Key Management Components	Sun, N.; Li, C.; Chan, H.; Islam, M. Z.; Islam, M. R.; Armstrong, W.	2023
162	On the importance of cryptographic agility for industrial automation	Paul, S.; Niethammer, M.	2019
166	Optimizing DevOps Enablers for Quantum Software Development	Al-Sanad, A.; Akbar, M.	2023
169	Overview and Comparison of Gate Level Quantum Software Platforms	LaRose, R.	2019
173	Password authentication key exchange based on key consensus for IoT security	Zhao, Z.; Ma, S.; Qin, P.	2023
179	Prioritisation of research challenges in software technologies: A multi-factor approach [version 1; peer review: awaiting	Alonso, J.; Ostolaza, E.; Sanchez, B.	2023
182	QFaaS: A Serverless Function-as-a-Service framework for Quantum computing	Nguyen, H. T.; Usman, M.; Buyya, R.	2024
183	Quantitative Assessment of Software Security by Quantum Technique Using Fuzzy TOPSIS	Nadeem, M.; Ahmad, M.; Ansar, S. A.; Pathak, P. C.; Khan, R. A.	2023
193	Quantum Computers and the Risks They Pose to Small and Medium-Sized Enterprises	Schindler, P.	2022
194	Quantum Computers for High-Performance Computing	Humble, T. S.; McCaskey, A.; Lyakh, D. I.; Gowrishankar, M.; Frisch, A.; Monz, T.	2021
195	Quantum computing for financial risk measurement	Wilkens, S.; Moorhouse, J.	2023
196	Quantum computing for social business optimization: a practitioner's perspective	Aljaafari, M.	2023
197	Quantum computing platforms: assessing the impact on quality attributes and sdlc activities	Sodhi, B.; Kapur, R.	2021
198	Quantum computing threat modelling on a generic cps setup	Lee, C. C.; Tan, T. G.; Sharma, V.; Zhou, J.	2021
202	Quantum devops: Towards reliable and applicable nisq quantum computing	Gheorgue-Pop, I. D.; Tcholtchev, N.; Ritter, T.; Hauswirth, M.	2020
203	Quantum for 6G communication: A perspective	Ali, M. Z.; Abohmra, A.; Usman, M.; Zahid, A.; Heidari, H.; Imran, M. A.; Abbasi, Q. H.	2023
204	Quantum healthcare analysis based on smart IoT and mobile edge computing: way into network study	Zhang, J.	2024
213	Quantum power flows: From theory to practice	Liu, J.; Zheng, H.; Hanada, M.; Setia, K.; Wu, D.	2022
214	Quantum Program Synthesis Through Operator Learning and Selection	Lee, S.; Nam, S. Y.	2023
215	Quantum Random Access Memory for Dummies	Phalak, K.; Chatterjee, A.; Ghosh, S.	2023
216	Quantum Searchable Encryption for Cloud Data Based on Full-Blind Quantum Computation	Liu, W.; Xu, Y.; Liu, W.; Wang, H.; Lei, Z.	2019
220	Quantum Software Components and Platforms: Overview and Quality Assessment	Serrano, M. A.; Cruz-Lemus, J. A.; Perez-Castillo, R.; Piattini, M.	2023
222	Quantum software engineering landscape and challenges	Piattini, M.; Murillo, J. M.	2022
223	Quantum software engineering: a new genre of computing	Akbar, M. A.; Khan, A. A.; Mahmood, S.; Rafi, S.	2022
224	Quantum software engineering: Landscapes and horizons	Zhao, J.	2020

Table A2. Cont.

ID	Title	Authors	Year
227	Quantum-based privacy-preserving sealed-bid auction on the blockchain	Abulkasim, H.; Mashatan, A.; Ghose, S.	2021
228	Quantum-Inspired Differential Evolution for Resource-Constrained Project-Scheduling: Preliminary Study	Saad, H. M.H.; Chakraborty, R. K.; Elsayed, S.	2021
231	Quantum2FA: Efficient Quantum-Resistant Two-Factor Authentication Scheme for Mobile Devices	Wang, Q; Wang, D.; Cheng, C.; He, D.	2023
234	QUASIM: Quantum computing enhanced service ecosystem for simulation in manufacturing	Agrawal, A.; Stein, H.; Xu, S.; Janzen, S.; Maass, W.	2023
235	Quingo: A Programming Framework for Heterogeneous Quantum-Classical Computing with NISQ Features	Fu, X.; Yu, J.; Su, X.; Jiang, H.; Wu, H.; Cheng, F.; Deng, X.; Zhang, J.; Jin, L.; Yang, Y.; Xu, L.; Hu, C.; Huang, A.; Huang, G.; Qiang, X.; Deng, M.; Xu, P.; Xu, W.; Liu, W.; Zhang, Y.; Deng, Y.; Wu, J.; Feng, Y.	2021
241	Resilience Optimization of Post-Quantum Cryptography Key Encapsulation Algorithms	Farooq, S.; Altaf, A.; Iqbal, F.; Thompson, E. B.; Vargas, D. L.; Diez, I.; Ashraf, I.	2023
244	Review and analysis of classical algorithms and hash-based post-quantum algorithm	Noel, M. D.; Waziri, V. O.; Abdulhamid, S. M.; Ojeniyi, J. A.	2021
253	Society 5.0 and the future of work skills for software engineers and developers	Smuts, S.; Smuts, H.	2022
255	Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks	Serret, M. F.; Marchand, B.; Ayrat, T.	2020
256	Space and Time-Efficient Quantum Multiplier in Post Quantum Cryptography Era	Putranto, D. S. C.; Wardhani, R. W.; Larasati, H. T.; Kim, H.	2023
258	Studying efficacy of traditional software quality parameters in quantum software engineering	Faryal, M.; Rubab, S.; Khan, M. M.; Khan, M. A.; Shebab, A.; Tariq, U.; Chelloug, S. A.; Osman, L.	2022
261	Technical debts and faults in open-source quantum software systems: An empirical study	Openja, M.; Morovati, M. M.; An, L.; Khomh, F.; Abidi, M.	2022
262	TensorFlow Quantum: Impacts of Quantum State Preparation on Quantum Machine Learning Performance	Sierra-Sosa, D.; Telahun, M.; Elmaghraby, A.	2020
264	The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime	Webber, M.; Elfving, V.; Weidt, S.; Hensinger, W. K.	2022
265	The quantum computing business ecosystem and firm strategies	Jenkins, J.; Berente, N.; Angst, C.	2022
266	The Quantum software lifecycle	Weder, B.; Barzen, J.; Leymann, F.; Salm, M.; Vietz, D.	2020
270	Toward a quantum software engineering	Piattini, M.; Serrano, M.; Perez-Castillo, R.; Petersen, G.; Hevia, J. L.	2021
275	Towards near-term quantum simulation of materials	Clinton, L.; Cubitt, T.; Flynn, B.; Gambetta, F. M.; Klassen, J.; Montanaro, A.; Piddock, S.; Santos, R. A.; Sheridan, E.	2024
276	Towards Physical Implementation of Quantum Computation	Ugwuishi, C. H.; Ayegbusi, O. A.; Eneh, A. H.; Ujah, J.	2020
277	Towards Quantum Requirements Engineering	Spoletini, P.	2023
278	Towards Quantum Software Requirements Engineering	Yue, T.; Ali, S.; Arcaini, P.	2023
279	Towards requirements engineering for quantum computing applications in manufacturing	Stein, H.; Schröder, S.; Kienast, P.; Kuling, M.	2024

Table A2. Cont.

ID	Title	Authors	Year
280	Towards security recommendations for public-key infrastructures for production environments in the post-quantum era	Yunakovsky, S. E.; Kot, M.; Pozhar, N.; Nabokov, D.; Kudinov, M.; Guglya, A.; Kiktenko, E. O.; Kolycheva, E.; Borisov, A.; Fedorov, A. K.	2021
281	Two-factor authentication using biometric based quantum operations	Sharma, M. K.; Nene, M. J.	2020
283	Unleashing quantum algorithms with Qinterpreter: bridging the gap between theory and practice across leading quantum computing platforms	Contreras-Sepúlveda, W.; Torres-Palencia, A. D.; Sánchez-Mondragón, J. J.; Villegas-Martínez, B. M.; Escobedo-Alatorre, J. J.; Gesing, S.; Lozano-Crisóstomo, N.; García-Melgarejo, J. C.; Sánchez-Pérez, J. C.; Palacios-Pérez, E. N.; Palillero-Sandoval, O.	2023
284	Using quantum annealers to calculate ground state properties of molecules	Copenhaver, J.; Wasserman, A.; Wehefritz-Kaufmann, B.	2021
285	Variational quantum compiling with double Q-learning	He, Z.; Li, L.; Zheng, S.; Li, Y.; Situ, H.	2021
287	When software engineering meets quantum computing	Ali, S.; Yue, T.; Abreu, R.	2022
fw1	Quantum Computing: An Overview Across the System Stack	Resch, S.; Karpuzcu, U. R.	2019
fw3	Quantum in the Cloud: Application Potentials and Research Opportunities	Leymann, F.; Barzen, J.; Falkenthal, M.; Vietz, D.; Weder, B.; Wild, K.	2020
fw4	Patterns For Hybrid Quantum Algorithms	Weigold, M.; Barzen, J.; Leymann, F.; Vietz, D.	2021
fw6	A systematic decision-making framework for tackling quantum software engineering challenges	Akbar, M. A.; Khan, A. A.; Rafi, S.	2023
fw7	On decision support for quantum application developers: categorization, comparison, and analysis of existing technologies	Vietz, D.; Barzen, J.; Leymann, F.; Wild, K.	2021
bw1	Open source software in quantum computing	Fingerhuth, M.; Babej, T.; Wittek, P.	2018
bw2	Programming languages and compiler design for realistic quantum hardware	Chong, F. T.; Franklin, D.; Martonosi, M.	2017
bw3	Quantum Computing in the NISQ era and beyond	Preskill, J.	2018

References

- Haghparsat, M.; Mikkonen, T.; Nurminen, J.K.; Stirbu, V. Quantum Software Engineering Challenges from Developers' Perspective: Mapping Research Challenges to the Proposed Workflow Model. In Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, WA, USA, 17–22 September 2023; IEEE: Piscataway, NJ, USA, 2023; Volume 2, pp. 173–176.
- Sun, Z.; Gao, W.; Dong, H.; Xie, H.; Yang, L. A new post-quantum voting protocol based on physical laws. *Quantum Inf. Process.* **2022**, *21*, 289. [\[CrossRef\]](#)
- Sood, S.K.; Pooja. Quantum computing review: A decade of research. *IEEE Trans. Eng. Manag.* **2023**, *71*, 6662–6676. [\[CrossRef\]](#)
- Yue, T.; Maunder, W.; Ali, S.; Taibi, D. Challenges and Opportunities in Quantum Software Architecture. In *Software Architecture: Research Roadmaps from the Community*; Springer: Cham, Switzerland, 2023; pp. 1–23.
- Piattini, M.; Serrano, M.; Perez-Castillo, R.; Petersen, G.; Hevia, J.L. Toward a quantum software engineering. *IT Prof.* **2021**, *23*, 62–66. [\[CrossRef\]](#)
- Ali, O.; Abdelbaki, W.; Shrestha, A.; Elbasi, E.; Alryalat, M.A.A.; Dwivedi, Y.K. A systematic literature review of artificial intelligence in the healthcare sector: Benefits, challenges, methodologies, and functionalities. *J. Innov. Knowl.* **2023**, *8*, 100333. [\[CrossRef\]](#)
- Zhao, J. Quantum software engineering: Landscapes and horizons. *arXiv* **2020**, arXiv:2007.07047.
- Zanni Junior, P.E.; Vieira de Camargo, V. A systematic mapping on quantum software development in the context of software engineering. *arXiv* **2021**, arXiv:2106.00926.
- Cartiere, C.R. Formal Methods for Quantum Software Engineering. In *Quantum Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 85–101.

10. Barbosa, L.S. Software engineering for ‘quantum advantage’. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, Seoul, Republic of Korea, 27 June–19 July 2020; pp. 427–429.
11. Weder, B.; Barzen, J.; Leymann, F.; Vietz, D. Quantum software development lifecycle. In *Quantum Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 61–83.
12. Pérez-Castillo, R.; Serrano, M.A.; Piattini, M. Software modernization to embrace quantum technology. *Adv. Eng. Softw.* **2021**, *151*, 102933. [\[CrossRef\]](#)
13. Akbar, M.A.; Khan, A.A.; Mahmood, S.; Rafi, S. Quantum software engineering: A new genre of computing. *arXiv* **2022**, arXiv:2211.13990.
14. Saraiva, L.; Haeusler, E.H.; Costa, V.G.; Kalinowski, M. Non-Functional Requirements for Quantum Programs. In Proceedings of the Q-SET@ QCE, Broomfield, CO, USA, 17–22 October 2021; pp. 89–73.
15. Benioff, P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J. Stat. Phys.* **1980**, *22*, 563–591. [\[CrossRef\]](#)
16. Toffoli, T. Reversible computing. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Noordwijkerhout, The Netherlands, 14–18 July 1980; Springer: Berlin/Heidelberg, Germany, 1980; pp. 632–644.
17. Feynman, R.P. Simulating physics with computers. *Int. J. Theor. Phys.* **1982**, *21*, 467–488. [\[CrossRef\]](#)
18. Deutsch, D.; Penrose, R. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A Math. Phys. Sci.* **1985**, *400*, 97–117. [\[CrossRef\]](#)
19. Vedral, V. *Introduction to Quantum Information Science*; Oxford University Press: Oxford, UK, 2006.
20. Horodecki, R.; Horodecki, P.; Horodecki, M.; Horodecki, K. Quantum entanglement. *Rev. Mod. Phys.* **2009**, *81*, 865. [\[CrossRef\]](#)
21. Deutsch, D.; Jozsa, R. Rapid Solution of Problems by Quantum Computation. *Proc. R. Soc. Lond. Ser. A* **1992**, *439*, 553–558. [\[CrossRef\]](#)
22. Simon, D.R. On the Power of Quantum Computation. *SIAM J. Comput.* **1997**, *26*, 1474–1483. [\[CrossRef\]](#)
23. Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. [\[CrossRef\]](#)
24. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the STOC ’96: Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219. [\[CrossRef\]](#)
25. Quantum Computing | D-Wave—dwavesys.com. Available online: <https://www.dwavesys.com/learn/quantum-computing/> (accessed on 2 June 2024).
26. IBM Quantum Computing—ibm.com. Available online: <https://www.ibm.com/quantum> (accessed on 2 June 2024).
27. IBM Quantum System Two: The Era of Quantum Utility is Here | IBM Quantum Computing Blog—ibm.com. Available online: <https://www.ibm.com/quantum/blog/quantum-roadmap-2033> (accessed on 2 June 2024).
28. Russell, J. IBM Quantum Update: Q System One Launch, New Collaborators, and QC Center Plans—hpcwire.com. Available online: <https://www.hpcwire.com/2019/01/10/ibm-quantum-update-q-system-one-launch-new-collaborators-and-qc-center-plans/> (accessed on 2 June 2024).
29. Servicio de Computación Cuántica en la Nube—Amazon Braket—AWS—aws.amazon.com. Available online: <https://aws.amazon.com/es/braket/> (accessed on 2 June 2024).
30. Luo, J.; Zhao, P.; Miao, Z.; Lan, S.; Zhao, J. A Comprehensive Study of Bug Fixes in Quantum Programs. In Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 15–18 March 2022; pp. 1239–1246. [\[CrossRef\]](#)
31. Yang, Z.; Zolanvari, M.; Jain, R. A survey of important issues in quantum computing and communications. *IEEE Commun. Surv. Tutorials* **2023**, *25*, 1059–1094. [\[CrossRef\]](#)
32. *IEEE Std 610.12-1990*; IEEE Standard Glossary of Software Engineering Terminology; IEEE: Piscataway, NJ, USA, 1990.
33. Bourque, P.; Fairley, R.E. *Guide to the Software Engineering Body of Knowledge-SWEBOK: v3.0*; IEEE Computer Society Press: Washington, DC, USA, 2014.
34. Brooks, F.P., Jr. No Silver Bullet Essence and Accidents of Software Engineering. *Computer* **1987**, *20*, 10–19. [\[CrossRef\]](#)
35. Vukšić, V.B.; Ivančić, L.; Vugec, D.S. A Preliminary Literature Review of Digital Transformation Case Studies. *Int. J. Comput. Inf. Eng.* **2018**, *12*, 737–742.
36. Felderer, M.; Taibi, D.; Palomba, F.; Epping, M.; Lochau, M.; Weder, B. Software Engineering Challenges for Quantum Computing: Report from the First Working Seminar on Quantum Software Engineering (WSQSE 22). *ACM SIGSOFT Softw. Eng. Notes* **2023**, *48*, 29–32. [\[CrossRef\]](#)
37. Carleton, A.; Klein, M.; Robert, J. and Harper, E. *Architecting the Future of Software Engineering. A National Agenda for Software Engineering Research & Development*; Technical Report; Carnegie Mellon University, Software Engineering Institute: Pittsburgh, PA, USA, 2021.
38. De Stefano, M.; Pecorelli, F.; Di Nucci, D.; Palomba, F.; De Lucia, A. Software engineering for quantum programming: How far are we? *J. Syst. Softw.* **2022**, *190*, 111326. [\[CrossRef\]](#)
39. Alexeev, Y.; McCaskey, A.; De Jong, W. Introduction to the Special Issue on Software Tools for Quantum Computing: Part 1. *ACM Trans. Quantum Comput.* **2022**, *3*, 1–3. [\[CrossRef\]](#)

40. Greiwe, F.; Krüger, T.; Mauerer, W. Effects of imperfections on quantum algorithms: A software engineering perspective. In Proceedings of the 2023 IEEE International Conference on Quantum Software (QSW), Chicago, IL, USA, 2–8 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 31–42.
41. Serrano, M.A.; Pérez-Castillo, R.; Piattini, M. *Quantum Software Engineering*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
42. Boehm, B. *Software Engineering Economics*; Prentice Hall: Upper Saddle River, NJ, USA, 1981.
43. Daneva, M.; Pastor, O. *Requirements Engineering: Foundation for Software Quality*; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9619. [\[CrossRef\]](#)
44. Noorin, U.; Sirshar, M. Quality Assurance in Requirement Engineering. *Glob. J. Comput. Sci. Technol.* **2017**, *17*, V1.
45. Srivastava, P.R.; Kumar, K. An Approach towards Software Quality Assessment. In *Information Systems, Technology and Management*; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2009; pp. 150–160. [\[CrossRef\]](#)
46. Hofmann, H.F.; Lehner, F. Requirements Engineering as a Success Factor in Software Projects. *IEEE Softw.* **2001**, *18*, 58–66. [\[CrossRef\]](#)
47. *Standard ISO/IEC 25010:2023*; Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Product Quality Model. International Organization for Standardization: Geneva, Switzerland, 2023.
48. Marir, T.; Mokhati, F.; Seridi-Bouchelaghem, H. Do we need specific quality models for multi-agent systems? Toward using the ISO/IEC 25010 quality model for MAS. In Proceedings of the 2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA), Vienna, Austria, 29–31 August 2014; pp. 363–368. [\[CrossRef\]](#)
49. Falcó, M.; Robiolo, G. Building a Catalogue of ISO/IEC 25010 Quality Measures Applied in an Industrial Context. *J. Phys. Conf. Ser.* **2021**, *1828*, 012077. [\[CrossRef\]](#)
50. Murillo, J.M.; Garcia-Alonso, J.; Moguel, E.; Barzen, J.; Leymann, F.; Ali, S.; Yue, T.; Arcaini, P.; Pérez, R.; de Guzmán, I.G.R.; et al. Challenges of Quantum Software Engineering for the Next Decade: The Road Ahead. *arXiv* **2024**, arXiv:2404.06825.
51. Yue, T.; Ali, S.; Arcaini, P. Towards Quantum Software Requirements Engineering. In Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, WA, USA, 17–22 September 2023; IEEE: Piscataway, NJ, USA, 2023; Volume 2, pp. 161–164.
52. Jantunen, S.; Dumdum, R.; Gause, D. Towards new requirements engineering competencies. In Proceedings of the 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Montreal, QC, Canada, 27 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 131–134.
53. Park, J.L. The Concept of Transition in Quantum Mechanics. *Found. Phys.* **1970**, *1*, 23–33. [\[CrossRef\]](#)
54. Kumar Pati, A.; Braunstein, S.L. Impossibility of deleting an unknown quantum state. *Nature* **2000**, *404*, 164–165. [\[CrossRef\]](#)
55. Nielsen, M.A.; Chuang, I.L. Programmable Quantum Gate Arrays. *Phys. Rev. Lett.* **1997**, *79*, 321–324. [\[CrossRef\]](#)
56. Bae, J.; Kwek, L.C. Quantum state discrimination and its applications. *J. Phys. A Math. Theor.* **2015**, *48*, 083001. [\[CrossRef\]](#)
57. System information | IBM Quantum Documentation—docs.quantum.ibm.com. Available online: <https://docs.quantum.ibm.com/run/system-information> (accessed on 2 June 2024).
58. Holmes, A.; Johri, S.; Guerreschi, G.G.; Clarke, J.S.; Matsuura, A.Y. Impact of qubit connectivity on quantum algorithm performance. *Quantum Sci. Technol.* **2020**, *5*, 025009. [\[CrossRef\]](#)
59. Yuan, P.; Allcock, J.; Zhang, S. Does Qubit Connectivity Impact Quantum Circuit Complexity? *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 520–533. [\[CrossRef\]](#)
60. Cowtan, A.; Dilkes, S.; Duncan, R.; Krajenbrink, A.; Simmons, W.; Sivarajah, S. On the Qubit Routing Problem. *arXiv* **2019**, [\[CrossRef\]](#)
61. Shor, P. Fault-tolerant quantum computation. In Proceedings of the 37th Conference on Foundations of Computer Science, Burlington, VT, USA, 14–16 October 1996; pp. 56–65. [\[CrossRef\]](#)
62. Devitt, S.J.; Munro, W.J.; Nemoto, K. Quantum error correction for beginners. *Rep. Prog. Phys.* **2013**, *76*, 076001. [\[CrossRef\]](#)
63. Krüger, T.; Mauerer, W. Quantum Annealing-Based Software Components: An Experimental Case Study with SAT Solving. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, Seoul, Republic of Korea, 27 June 2020–19 July 2020; pp. 445–450. [\[CrossRef\]](#)
64. Spoletini, P. Towards Quantum Requirements Engineering. In Proceedings of the 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW), Hannover, Germany, 4–5 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 371–374. [\[CrossRef\]](#)
65. Upama, P.B.; Faruk, M.J.H.; Nazim, M.; Masum, M.; Shahriar, H.; Uddin, G.; Barzanjeh, S.; Ahamed, S.I.; Rahman, A. Evolution of quantum computing: A systematic survey on the use of quantum computing tools. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, 27 June–1 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 520–529.
66. Gill, S.S.; Kumar, A.; Singh, H.; Singh, M.; Kaur, K.; Usman, M.; Buyya, R. Quantum computing: A taxonomy, systematic review and future directions. *Softw. Pract. Exp.* **2022**, *52*, 66–114. [\[CrossRef\]](#)
67. Almudever, C.G.; Lao, L.; Fu, X.; Khammassi, N.; Ashraf, I.; Iorga, D.; Varsamopoulos, S.; Eichler, C.; Wallraff, A.; Geck, L.; et al. The engineering challenges in quantum computing. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, 27–31 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 836–845.
68. Khan, A.A.; Ahmad, A.; Waseem, M.; Liang, P.; Fahmideh, M.; Mikkonen, T.; Abrahamsson, P. Software architecture for quantum computing systems—A systematic review. *J. Syst. Softw.* **2023**, *201*, 111682. [\[CrossRef\]](#)

69. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report; Software Engineering Group School of Computer Science and Mathematics Keele University Keele: Staffs, UK, 2007.
70. Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* **2007**, *80*, 571–583. [\[CrossRef\]](#)
71. Sepúlveda, S.; Cravero, A. Systematic Review Protocol: Requirements Engineering in Quantum Computing. *arXiv* **2023**, arXiv:2311.12998.
72. Petticrew, M.; Roberts, H. *Systematic Reviews in the Social Sciences: A Practical Guide*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
73. Chadegani, A.; Salehi, H.; Yunus, M.; Farhadi, H.; Fooladi, M.; Farhadi, M.; Ebrahim, N.A. A Comparison between Two Main Academic Literature Collections: Web of Science and Scopus Databases. *Political Econ.—Dev. Public Serv. Deliv.* **2013**. [\[CrossRef\]](#)
74. Zhu, J.; Liu, W. A tale of two databases: The use of Web of Science and Scopus in academic papers. *Scientometrics* **2020**, *123*, 321–335. [\[CrossRef\]](#)
75. Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th international Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 1–10.
76. Gwet, K. Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity. *Stat. Methods Inter-Rater Reliab. Assess. Ser.* **2002**, *2*, 9.
77. Fleiss, J.L.; Levin, B.; Paik, M.C. *Statistical Methods for Rates and Proportions*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
78. ISO/IEC 9126:2001; Software Engineering—Product Quality. International Organization for Standardization: Geneva, Switzerland, 2001.
79. Yang, L.; Zhang, H.; Shen, H.; Huang, X.; Zhou, X.; Rong, G.; Shao, D. Quality assessment in systematic literature reviews: A software engineering perspective. *Inf. Softw. Technol.* **2021**, *130*, 106397. [\[CrossRef\]](#)
80. Kitchenham, B.A.; Sjøberg, D.I.; Dybå, T.; Pfahl, D.; Brereton, P.; Budgen, D.; Höst, M.; Runeson, P. Three empirical studies on the agreement of reviewers about the quality of software engineering experiments. *Inf. Softw. Technol.* **2012**, *54*, 804–819. [\[CrossRef\]](#)
81. Budgen, D.; Brereton, P. Performing systematic literature reviews in software engineering. In Proceedings of the 28th International Conference on Software Engineering, Shanghai, China, 20–28 May 2006; pp. 1051–1052.
82. Dybå, T.; Dingsøyr, T. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.* **2008**, *50*, 833–859. [\[CrossRef\]](#)
83. Zhou, Y.; Zhang, H.; Huang, X.; Yang, S.; Babar, M.A.; Tang, H. Quality assessment of systematic reviews in software engineering: A tertiary study. In Proceedings of the EASE '15: 19th International Conference on Evaluation and Assessment in Software Engineering, Nanjing, China, 27–29 April 2015. [\[CrossRef\]](#)
84. Sepúlveda, S.; Cravero, A.; Cachero, C. Requirements modeling languages for software product lines: A systematic literature review. *Inf. Softw. Technol.* **2016**, *69*, 16–36. [\[CrossRef\]](#)
85. Usman, M.; Bin Ali, N.; Wohlin, C. A Quality Assessment Instrument for Systematic Literature Reviews in Software Engineering. *arXiv* **2021**, arXiv:2109.10134.
86. Dybå, T.; Dingsøyr, T. Strength of evidence in systematic reviews in software engineering. In Proceedings of the ESEM '08: Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden, 19–20 September 2008; pp. 178–187. [\[CrossRef\]](#)
87. Chen, Y.; Zhang, Q.; Peng, W.; Liu, D.; You, Y.; Liu, X.; Tang, S.; Zhang, T.E. Efficacy and safety of mesenchymal stem cells for the treatment of patients infected with COVID-19: A systematic review and meta-analysis protocol. *BMJ Open* **2020**, *10*, e042085. [\[CrossRef\]](#) [\[PubMed\]](#)
88. Darley, S.; Coulson, T.; Peek, N.; Moschogiannis, S.; van der Veer, S.N.; Wong, D.C.; Brown, B. Understanding How the Design and Implementation of Online Consultations Affect Primary Care Quality: Systematic Review of Evidence With Recommendations for Designers, Providers, and Researchers. *J. Med. Internet Res.* **2022**, *24*, e37436. [\[CrossRef\]](#) [\[PubMed\]](#)
89. Filho, C.A.T.T.; Junior, E.P.; Vendrame, J.W.; Silva, G.M.D.; Cavina, A.P.S.; Lemos, L.K.; Vanderlei, F. Effect of aerobic training with blood flow restriction on strength and hypertrophy: A meta-analysis. *Int. J. Sport. Med.* **2024**. [\[CrossRef\]](#)
90. Gagliardoni, T. Quantum Security of Cryptographic Primitives. *arXiv* **2017**, arXiv:1705.02417.
91. V, M.C.; N, N.A. A Hybrid Double Encryption Approach for Enhanced Cloud Data Security in Post-Quantum Cryptography. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 242–248. [\[CrossRef\]](#)
92. Takagi, T.; Peyrin, T. *Advances in Cryptology—ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017*; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10625.
93. Paul, S.; Scheible, P. Towards Post-Quantum Security for Cyber-Physical Systems: Integrating PQC into Industrial M2M Communication. *IACR Cryptol. EPrint Arch.* **2020**, *2021*, 1563. [\[CrossRef\]](#)
94. Frantzi, K.; Ananiadou, S.; Mima, H. Automatic recognition of multi-word terms: the c-value/nc-value method. *Int. J. Digit. Libr.* **2000**, *3*, 115–130. [\[CrossRef\]](#)
95. Zhou, X.; Jin, Y.; Zhang, H.; Li, S.; Huang, X. A map of threats to validity of systematic literature reviews in software engineering. In Proceedings of the 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Hamilton, New Zealand, 6–9 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 153–160.
96. Ampatzoglou, A.; Bibi, S.; Avgeriou, P.; Verbeek, M.; Chatzigeorgiou, A. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf. Softw. Technol.* **2019**, *106*, 201–230. [\[CrossRef\]](#)

97. Petersen, K.; Gencel, C. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In Proceedings of the 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, 23–26 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 81–89.
98. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.