FUDMA Journal of Engineering and Technology
ISSN: 3092-9385
https://fjet.fudutsinma.edu.ng/index.php/fjet/
Volume 2, Issue 1, 2026, 402-412

Quantum Computing Applications in Software Engineering: A Scoping Review

Hillary T. OGBODO[1*], Bilkisu L. MUHAMMAD-BELLO[2], Joshua ABAH[3], Saleh E.-Y. ABDULLAHI[4]

[1*,2,3,4]Software Engineering Department, Faculty of Computing, Nile University of Nigeria, Abuja, Nigeria

[1*]246370027@nileuniversity.edu.ng, [2]bilkisu.muhammad-bello@nileuniversity.edu.ng, [3]joshua.abah@nileuniversity.edu.ng, [4]saleh.abdullahi@nileuniversity.edu.ng

Abstract

The emerging field of quantum computing (QC) uses principles of quantum mechanics, such as superposition and entanglement, to deliver powerful computing capabilities. Through these principles, qubits can exist in a combination of states at the same time, enabling quantum computers to solve complex problems that classical computers cannot. QC promises to be useful for many important tasks in software engineering at different stages of the SDLC. The Software Engineering (SE) field is only beginning to integrate QC practices, and this integration is still at an early stage. This study is a scoping review that investigates the application of QC in SE to understand the techniques, trends, practices, and gaps. Based on the framework developed by Arksey and O'Malley, five databases were screened, yielding 180 initial records. After screening, 20 studies from 2020 to 2025 were identified, consisting of 14 primary studies and 6 secondary works (three SLRs, two surveys, and an edited book), which were kept as contextual references. The results show that testing and operations were largely optimized using quantum-inspired algorithms, while software architecture, economics, and security received less attention. The review also provides a future research platform for quantum software engineering (QSE).

Keywords: *Quantum computing, software engineering, quantum software engineering, quantum optimization, scoping review, post-quantum cryptography, NISQ.*

1. Introduction

As software becomes more embedded in almost everything we do, there is an increasing need for software that is both fast and secure. Quantum computing (QC) offers this and much more, utilizing the principles of quantum mechanics to solve computationally hard problems [1]. QC utilizes qubits to enhance the speed of specific algorithms, such as Shor's algorithm for integer factorization, and provides theoretical speed advantages for NP-hard optimization problems, which are very difficult to compute and consume a lot of computational resources. These problems are commonly seen in software engineering (SE) [2]. For example, within large systems with many test cases and limited resources, it is important to prioritize test cases. This helps in defect identification within a set time, as shown in [11] and [13]. However, quantum computing also has security challenges. An example is Shor's algorithm, discussed in [20], which threatens classical encryption methods such as RSA and ECC.

Quantum software engineering (QSE) combines quantum computing (QC) and software engineering (SE), and it was defined in the official Talavera Manifesto in 2020. Earlier work has shown that QC may accelerate a range of SE processes. However, current quantum hardware is Noisy Intermediate-Scale Quantum (NISQ) [29], meaning that quantum computers are still relatively small and error-prone, making them unsuitable for large-scale, fault-tolerant applications. Zhao [1] and Mandal et al. [6] wrote foundational surveys that mapped the early QSE landscape, while Ali, Yue, and Abreu [4] identified the unique challenges that arise when SE and QC meet. Recent studies have also focused on the security aspect to check whether current cryptographic systems can be broken by quantum computing. If so, post-quantum cryptography (PQC) may help resolve this problem [19], [20]. Earlier overviews are limited in scope, often focusing on a few aspects such as optimization, development, or security, and therefore lack comprehensive coverage.

The literature on QC in SE is fragmented, with insufficient foundational material to support comprehensive analysis. What is needed is a broad scoping review covering all phases of SE up to 2025. This scoping review addresses this gap by mapping the literature to identify QC techniques applied in SE, determine covered SE phases, analyze trends and demographics and highlight gaps for future work.. We use the Arksey and O'Malley scoping review framework [8], which is a good fit for an emerging field because it prioritises breadth over depth.

2. Methodology

We chose a scoping review as the research methodology for this study because it is suited to emerging research areas, where it identifies already published work, the methods used, and the areas that remain unexplored. The

Received: 25/03/2026; Revised: 11/04/2026; Accepted: 17/04/2026; Published: 25/04/2026

review does not heavily assess the quality of the methods used in individual studies or draw conclusions about whether they are right or wrong. This scoping review used the five-stage framework proposed by Arksey and O'Malley to achieve its objectives. The five-stage framework is listed below:

- Identify the research questions
- Identify related studies
- Study the selected related works
- Chart data based on findings
- Compile, analyze and outline findings

2.1 Research Questions

1. RQ1: What QC techniques are applied in SE?
2. RQ2: What are the SE phases that benefit from QC applications?
3. RQ3: What are the trends in publication and research focus?
4. RQ4: What are the existing gaps in the literature?

2.2 Data Sources and Search Terms

The databases searched were Google Scholar (4 September 2025), Scopus (5 September 2025), Web of Science (5 September 2025), arXiv (6 September 2025) and IEEE Xplore (7 September 2025). Search terms used included various combinations of the terms "quantum computing", "quantum software engineering", "software engineering", "post-quantum cryptography", and "quantum security". Targeted searches didn't include review-type keywords (review, mapping, scoping) to focus on primary studies. The time period covered publications dated 2015-2025.

2.3 Inclusion and Exclusion Criteria

Before screening began, we defined inclusion and exclusion criteria to make sure the studies selected for this scoping review were relevant and appropriate. The same criteria were applied across all five databases.

The criteria for inclusion were: (IC1) publication ranging from 2015 to 2025; (IC2) peer-reviewed journal or conference paper; (IC3) mention of QC techniques, quantum-inspired algorithms or PQC within SE context; (IC4) Studies written in English; and (IC5) originality in the presentation of research findings, including empirical studies, theoretical studies, and proof-of-concept implementations.

The exclusion criteria (EC) were as follows: EC1: We excluded secondary studies, such as systematic literature reviews, mapping studies, and scoping reviews, from our primary list. However, we kept four recent and highly cited reviews to provide broader context. These help us cross-check our findings and compare our work with previous research. These four papers are clearly marked in Table 1 and are not included in the 14 primary studies used for our data analysis. EC2: We excluded non-peer reviewed studies such as blog posts, whitepapers, editorials etc. EC3: We excluded studies that only use quantum physics or quantum mathematics (like quantum algorithm) but do not apply to SE. For instance, research studies which just monitor/discuss competitions/papers on quantum computer without giving original study for software engineering. EC4: Duplicates or extended papers of specification process were excluded. EC5: Papers that only mention Quantum Computing (QC) in passing, or use it as a secondary example while their main contribution lies in a different field (e.g., mentioning QC without actually focusing on testing).

2.4 Study Selection Process

The selection process for the study was conducted in four stages. The first stage was the identification stage, where we searched across all five databases; this yielded about 180 studies. After completing the search, we moved to the next stage, which was the deduplication stage, where we found a total of 62 duplicates, leaving 118 studies. The next stage was the screening stage, where we used the inclusion and exclusion criteria to screen the titles and abstracts, which eliminated another 71 studies and left us with 47 records for full-text review. After the full-text review, we decided to exclude 27 more studies because they were not fully aligned with our objective. This left us with a total of 20 studies: 14 primary studies and 6 secondary studies that were kept as contextual references (see Section 2.3 and Table 1). The complete selection process is shown in Figure 1 as a PRISMA 2020 flow diagram.

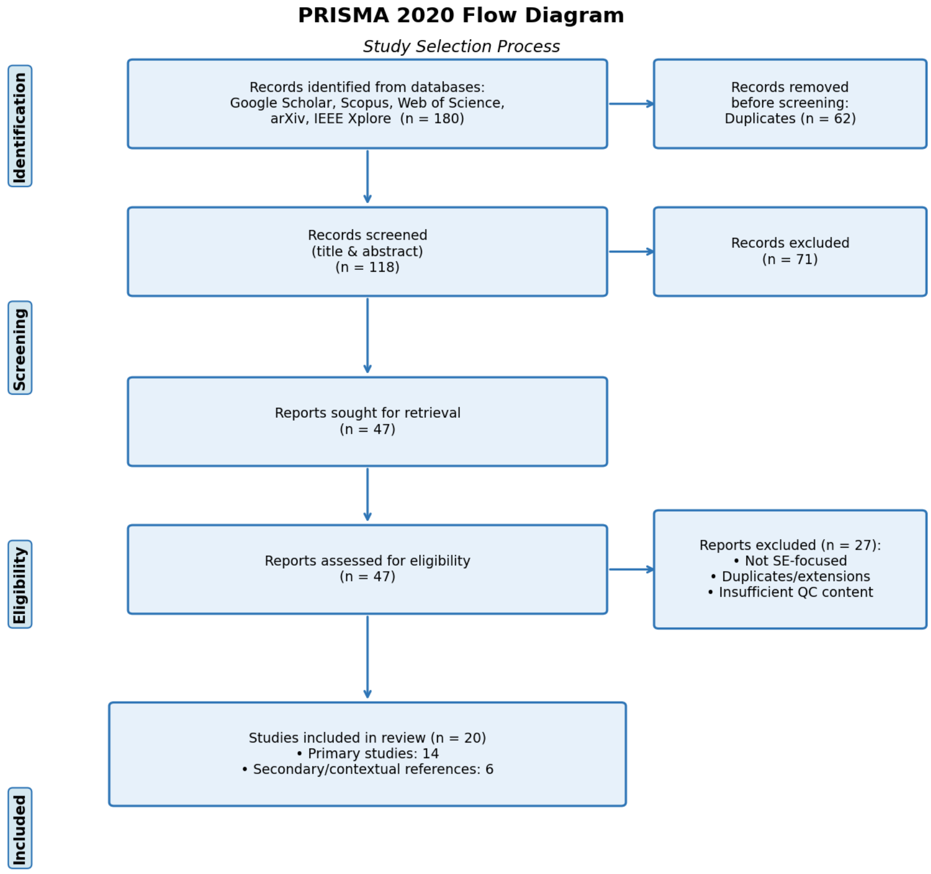

Fig. 1. PRISMA 2020 flow diagram showing the study selection process

2.5 Data Charting

We filled a structured form for each of the 20 studies (14 primary and 6 secondary) with their bibliographic details, the type of QC technique used (pure quantum, quantum-inspired, or hybrid), the SE phase being addressed, the type of validation (theoretical, simulation, or empirical), and their main findings. This form made it relatively easier to see the patterns and combine their individual results. All counts, distributions, and trends reported in this review used only the 14 primary studies, while the 6 contextual references were used in the Discussion (Section 4) to compare and cross-check the findings.

3. Results

This section presents the findings from the scoping review in relation to each research question.

3.1 Overview of Primary Studies

These studies were published between 2020 and 2025 and were found in journals like IEEE TSE, ACM TOSEM, and JSS, as well as in conference proceedings from ICSE, ICST, and QCE.

Table 1. List of Primary Studies Included in the Review

No.	Author(s), Year	QC Technique	SE Phase	Validation	Ref.
1	Khan et al., 2023	N/A (SLR)	Architecture	SLR	[9]
2	Pérez-Delgado & Perez-Gonzalez, 2020	Pure Quantum	Design	Theoretical	[10]
3	Wang et al., 2024a	QAOA	Testing	Empirical	[11]
4	Wang et al., 2024b	Quantum Annealing	Testing	Empirical	[12]
5	Bajaj et al., 2022	Quantum-Inspired PSO	Testing	Empirical	[13]

No.	Author(s), Year	QC Technique	SE Phase	Validation	Ref.
6	Trovato et al., 2024	Quantum Annealing	Testing	Empirical	[14]
7	Ali et al., 2021	Pure Quantum	Testing	Empirical	[15]
8	Hevia et al., 2022	Hybrid	Operations	Empirical	[16]
9	Zhang et al., 2025	Multiple	Cross-cutting	SLR	[17]
10	Moguel et al., 2020	Hybrid	Operations	Theoretical	[18]
11	NIST, 2024	PQC Standards	Security	Standard	[19]
12	Mavroeidis et al., 2018	N/A	Security	Theoretical	[20]
13	De Stefano et al., 2022	N/A (Survey)	Cross-cutting	Empirical	[21]
14	Openja et al., 2022	N/A	Maintenance	Empirical	[22]
15	Sepúlveda & Cravero, 2024	Multiple	Requirements	SLR	[23]
16	Yue et al., 2023	Quantum-Inspired	Requirements	Theoretical	[24]
17	Pérez-Castillo et al., 2021	Hybrid	Maintenance	Theoretical	[25]
18	Awan et al., 2022	N/A	Cross-cutting	Empirical	[26]
19	Serrano et al., 2022	Multiple	Cross-cutting	Book	[27]
20	Khan et al., 2024	Multiple	Cross-cutting	Survey	[28]

Note. Rows 1 (Khan et al., 2023 [9]), 9 (Zhang et al., 2025 [17]), 13 (De Stefano et al., 2022 [21]), 15 (Sepúlveda & Cravero, 2024 [23]), 19 (Serrano et al., 2022 [27]), and 20 (Khan et al., 2024 [28]) are secondary works (SLRs, survey, edited volume) retained as contextual references only; they are excluded from the quantitative analyses, which use only the 14 primary studies.

Table 1 lists all 20 works. The 14 primary studies (used for our counts) are clearly separated from the 6 contextual references ([9], [17], [21], [23], [27], [28]). Across the primary studies, testing is the most addressed SE phase with 5 studies. Operations, security, requirements and maintenance each have 2 studies, and architecture and design have 1 each. One primary study [26] cuts across phases. So, testing has clearly dominated the QSE research agenda so far. This is the same pattern reported by Zhang et al. [17] and Mandal et al. [6].

Table 3 (Section 3.3) and Fig. 3 both present the distribution of the 14 primary studies across SE phases, with each study assigned to its dominant phase. The testing phase clearly dominates the field, followed at a distance by operations, security, and maintenance.

3.2 RQ1: QC Techniques in SE

Three types of QC techniques were identified. Quantum techniques that are executed on quantum hardware comprise Quantum Annealing (QA) for optimization on D-Wave systems [12], [14], the Quantum Approximate Optimization Algorithm (QAOA) for combinatorial optimization [11], and Grover's Search for quadratic speedup of unstructured search [1]. Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are examples of quantum-inspired solutions that simulate quantum principles on classical hardware [13]. Hybrid approaches combine classical machine learning with quantum circuits [16], [18].

Table 2. Summary of QC Technique Types

Technique Type	Studies	Examples
Pure Quantum	5	QA for test minimization, Grover's for search
Quantum-Inspired	2	QPSO for test prioritization, GA for allocation

Technique Type	Studies	Examples
Hybrid	3	QAOA + classical decomposition, QuantumPath
Standards / Non-algorithmic	4	NIST PQC standards, threat analyses, empirical surveys

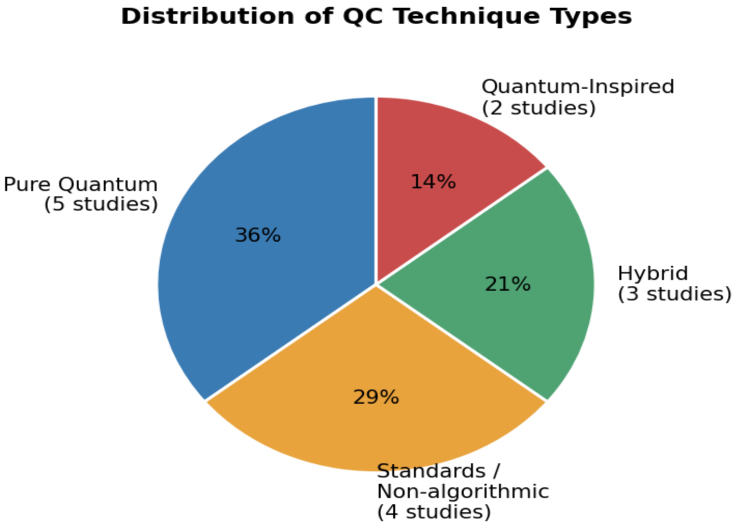

Fig. 2. Distribution of QC techniques across the primary studies

Fig. 2 illustrates the distribution of technique types across the primary studies. Pure quantum techniques account for the largest share (5 studies), followed by Standards / Non-algorithmic methods (4 studies), hybrid approaches (3 studies) and Quantum-inspired methods (2). Wang et al. [11] applied QAOA on IBM quantum hardware for test case optimization, while Bajaj et al. [13] demonstrated that a quantum-behaved PSO outperformed classical metaheuristic algorithms for test case prioritization on standard benchmarks.

3.3 RQ2: SE Phases Covered

Table 3. Distribution of Studies Across SE Phases

SE Phase	Studies	Key QC Application
Testing	5	Test case prioritization [11]–[15]
Operations	2	Platform development, scheduling [16], [18]
Security	2	PQC standardization, threat analysis [19], [20]
Requirements	1	Prioritization, RE challenges [24]
Architecture	1	Architecture patterns, modeling [9]
Maintenance	2	Technical debt, modernization [22], [25]
Cross-cutting	1	Challenges in QC adoption [26]

Received: 25/03/2026; Revised: 11/04/2026; Accepted: 17/04/2026; Published: 25/04/2026

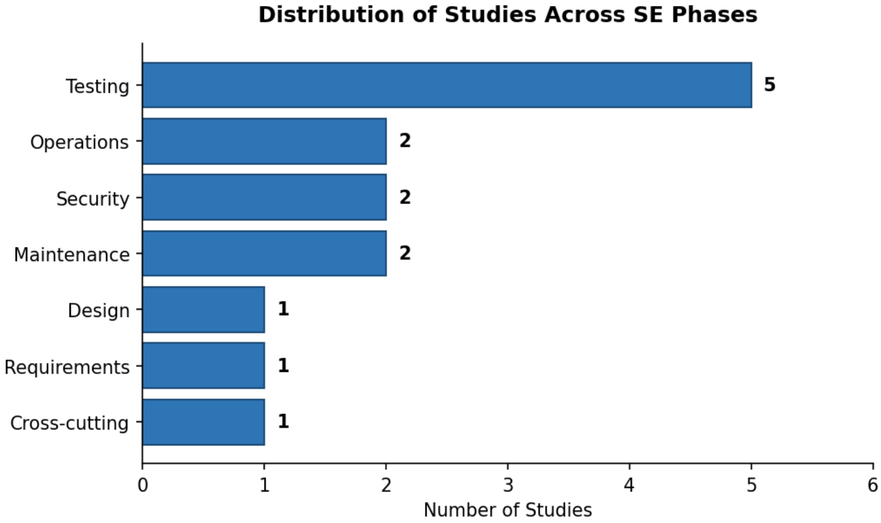

Fig. 3. Distribution of studies across SE phases

Testing is the most researched software engineering phase, with five primary studies [11]–[15] addressing tasks such as test case prioritization, minimization, and optimization. Operations was addressed by Hevia et al. [16] through QuantumPath and Moguel et al. [18] through a quantum software development roadmap. In security, NIST's PQC standardization [19] and the threat analysis by Mavroeidis et al. [20] represent the key contributions. Requirements engineering was addressed by Yue et al. [24] and Sepúlveda and Cravero [23], while Khan et al. [9] conducted the only SLR on quantum software architecture, identifying 34 primary studies in the area.

3.4 RQ3: Research Trends

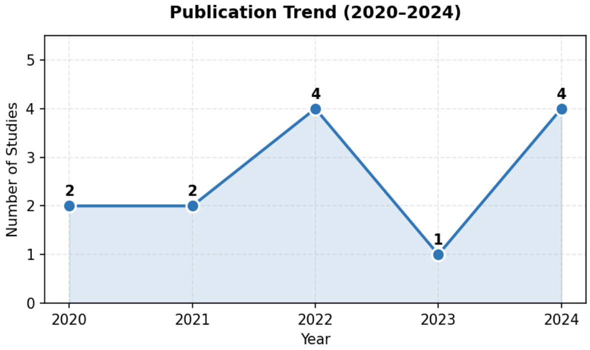

Fig. 4. Publication trend of primary studies (2020–2024)

Figure 4 shows the 2020–2024 publication trend of primary studies. The research published on QC in SE has grown unevenly from 2020 through 2024. These studies can be classified into three phases.

The first stage is the foundational stage (2018 - 2020), when the QSE field emerged. In this period, Zhao [1] published the first complete survey on the quantum software lifecycle, and the first International Workshop on Quantum Software Engineering and Programming (QANSWER) was held, where the Talavera Manifesto [2] was released. Around the same time, the NISQ paradigm was introduced by Preskill [29], which gave a realistic picture of the hardware limits the field would need to work around.

During the expansion phase (2021–2023), there was an increase in empirical research activities. During this time, studies began to show that classical concepts from SE could be used in QC. Ali et al. [15] proposed some of the first input-output coverage criteria for testing quantum programs. In a 2022 study by Bajaj et al. [13], it was shown that quantum-behaved PSO outperformed up to five popular metaheuristic algorithms on real software benchmarks. Nonetheless, this phase also revealed practical challenges facing the industry. A survey of quantum programmers conducted in 2023 by De Stefano et al. [21] documented 56 quantum programmers reporting that many developers find debugging difficult. In addition, due to the design of frameworks such as Qiskit and Cirq, steep learning curves exist for a significant number of developers. Based on their software repository analysis, research currently outpaces practice, as most projects are educational and not production-ready.

We are now in the maturing stage (2024), which is characterized mainly by a decisive shift toward hardware-based quantum systems. The recent literature clearly reflects this shift; for instance, impactful papers such as Wang et al.'s QAOA-based optimization on IBM systems [11] and their work on D-Wave annealers [12] have appeared

Received: 25/03/2026; Revised: 11/04/2026; Accepted: 17/04/2026; Published: 25/04/2026

in high-impact venues, including IEEE TSE and ACM TOSEM. There has also been a rise in studies in this field; Zhang et al. [17] recently delivered the most comprehensive SLR to date, covering 77 primary studies across 32 distinct SE problems. Although the "aQuantum" consortium in Europe, as well as major centres in East Asia (Kyushu and Beihang Universities), still dominate, there is increasing influence from Nordic groups at the University of Oulu and Simula Research Laboratory [4], [7]. From a tooling perspective, IBM's Qiskit remains the industry standard, followed by D-Wave Ocean and Google's Cirq [6], [21].

3.5 RQ4: Research Gaps

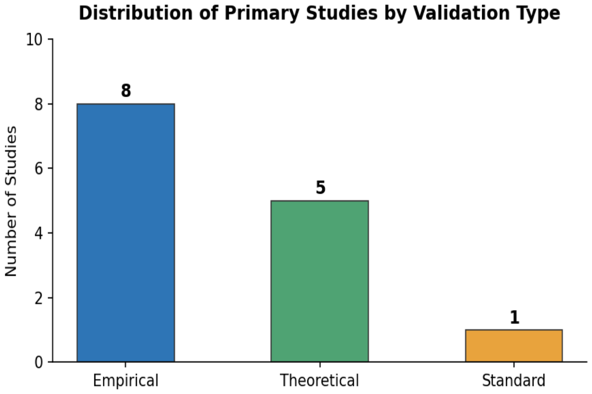

Fig. 5. Summary of identified research gaps in QC applications in SE

1. **Limited multi-objective optimization**: The application of multi-objective optimization techniques is generally limited in the literature; it is a major limitation of the current literature. While most of the optimization research focuses on a single objective, the software engineering (SE) tasks in the real world require multiple objectives to be considered simultaneously. One common requirement is the maximization of fault detection and the minimization of execution time [13, 14]
2. **Hardware constraints:** The current NISQ hardware does not have enough qubit and suffers from noise, decoherence etc. Wang et al. research shows that when the Quantum Approximate Optimization Algorithm (QAOA) is run on IBM quantum hardware, performance severely deteriorates in the presence of noise. Thus, error mitigation is necessary.
3. **Lack of industrial validation:** A majority of quantum software subjects are educational rather than industrial, observed De Stefano et al. [21]. Awan et al. found that accessibility and expertise are the main challenges for quantum adoption in software.
4. **The security imperative:** Shor's algorithm can break the RSA and ECC encryption which raises a security issue [20]. Despite the NIST's finalization of PQC standards in 2024 [19], the techniques for updating existing software systems are still under development [25].
5. **Underexplored SE areas:** Software engineering (SE) areas such as software architecture [9], requirements engineering [23], [24], maintenance [22], [25], and deployment remain underexplored, despite their potential suitability for quantum optimization [17].

4. Discussion

In this section, we interpret the results presented in Section 3 and provide novel insights from this scoping review. In addition, our findings are compared with those of other systematic reviews and surveys in QSE.

4.1 Interpretation of Key Findings

Based on the analysis of the 14 primary studies, it was observed that QC research focuses more on the testing phases of SE compared to other SDLC phases. Out of the 14 relevant studies, 5 were centered around various software testing tasks, such as test-case prioritization [13], minimization [12], and optimization [11]. This means that 36% of the primary studies (5 of 14) focused on a single phase in the SDLC. It is no coincidence that quantum techniques dominate the testing phase, as testing tasks are highly combinatorial in nature. However, this causes an imbalance in research, with 36% of the primary studies (5 of 14) focused on just one SDLC phase.

Mandal et al. [6] conducted a systematic review of 77 studies and discovered that the testing stage, along with the optimization phase, largely dominates the QSE literature. Moreover, most life cycle stages remain quite neglected or unexplored. Our results confirm this observation and extend it to more recent studies (2024–2025), which also show the imblance despite increasing calls for coverage in the Talavera Manifesto [2].

Table 2 shows that pure quantum techniques (5 studies) are the most common approach, followed by standards and non-algorithmic contributions (4 studies), hybrid approaches (3 studies), and quantum-inspired methods (2 studies). Zhang et al. [17] showed that more than 50% of quantum studies on SE make use of simulators, classical approximations, and hybrid ideas, instead of real quantum hardware. Quantum-inspired ideas are useful based on this review, but they do have drawbacks. Quantum principles are already beneficial to SE tasks even in the absence of fault-tolerant quantum computers, since quantum-behaved PSO outperforms classical algorithms. On the other hand, this raises the question of whether quantum effects are responsible for these enhancements or simply the result of novel optimization formulations. Subsequent research should include studies that isolate the quantum component of such algorithms.

Researchers need to focus more on the security aspect of QC in SE. Only two studies focused on security directly [19], [20]. This is significant problem because Shor's algorithm can break classical public-key cryptography. The finalization of post-quantum cryptographic (PQC) standards by NIST in 2024 [19] was an important milestone for the field. Mavroeidis et al. [20] noted that there are many theoretical works regarding the PQC threat landscape. However, there is insufficient software engineering literature that addresses challenges in real-world systems. He states: "There is also a growing concern about the migration of legacy systems to PQC-compliant architectures."

An initial effort to modernize software to become quantum-ready was made by Pérez-Castillo et al. [25]. However, a complete roadmap for migration that includes dependency analysis, regression testing, and backward compatibility has not yet been achieved. As outlined in this review, this remains one of the key gaps.

Table 4. Comparative matrix of the 14 primary studies by QC technique, SE phase, and validation

#	**Study**	**Technique**	**SE Phase**	**Validation**
1	Pérez-Delgado & Perez-Gonzalez, 2020	Pure Quantum	Design	Theoretical
2	Wang et al., 2024a	QAOA	Testing	Empirical
3	Wang et al., 2024b	Quantum Annealing	Testing	Empirical
4	Bajaj et al., 2022	Quantum-Inspired PSO	Testing	Empirical
5	Trovato et al., 2024	Quantum Annealing	Testing	Empirical
6	Ali et al., 2021	Pure Quantum	Testing	Empirical
7	Hevia et al., 2022	Hybrid	Operations	Empirical
8	Moguel et al., 2020	Hybrid	Operations	Theoretical
9	NIST, 2024	PQC Standards	Security	Standard
10	Mavroeidis et al., 2018	N/A	Security	Theoretical
11	Openja et al., 2022	N/A	Maintenance	Empirical
12	Yue et al., 2023	Quantum-Inspired	Requirements	Theoretical
13	Pérez-Castillo et al., 2021	Hybrid	Maintenance	Theoretical
14	Awan et al., 2022	N/A	Cross-cutting	Empirical

In Table 4, we compare the 14 primary studies using a matrix across three dimensions: QC technique type, SE phase addressed, and validation approach. This table provides an analytical view of the scoping review to complement its descriptive nature. The matrix reveals relationships that are not visible when the three dimensions are considered independently. The table shows that empirical validation is concentrated in the testing phase, where 5 of the 8 empirical studies (63%) address testing, and every study outside the testing phase relies on theoretical or standards-based validation. Pure quantum techniques are applied only in testing and design, never in operations, security, maintenance, requirements, or cross-cutting work. Hybrid approaches, in contrast, appear only in operations and maintenance. Taken together, these patterns suggest that the apparent diversity of QC techniques in the QSE literature is, in fact, tightly coupled to a limited set of phase–technique combinations, indicating that the field has not yet explored large parts of the QSE design space.

4.2 Comparison with Related Reviews

In three key ways, our results align with prior reviews, but also add to them. In the year 2020, Zhao proposed the first complete taxonomy of the quantum software lifecycle, however, this work predates most of the empirical research we discuss, such as the hardware focused work by Wang et al. [11, 12. This shift in the field is reflected in our review, which notes the move from simulations to hardware on IBM and D-Wave, which Zhao could not have accounted for at the time.

Our scope also differs from that of Zhang et al. [17], whose SLR on quantum optimization is much larger than ours, but was focused on optimization problems. By adopting a broader perspective on the entire SE life cycle, we can detect gaps in certain areas such as requirements engineering [23], [24] and software architecture [9] that Zhang was not highlighting.

Finally, our data points to a persistent Industrial Validation Challenge. According to De Stefano et al. [21] many of the projects found in GitHub are mostly theoretical, i.e based on simulation. We also find that this is true for GitHub projects that use Qiskit or Cirq. Most programmers face a steep learning curve with these libraries. All 14 primary studies suffer from an absence of industrial validation. This is not merely a research gap; it is more a symptomatic representation of the systemic impediment with respect to the adoption of quantum computing in software engineering (SE) practice.

4.3 Limitations of This Review

This scoping review had some limitations that are clearly stated below:

1. We only searched five major databases, and all the papers we considered were written in the English language; this means that we may have excluded relevant works published in other languages or indexed in different databases.
2. This was a scoping review, so we didn't we did not thoroughly assess the methodology used in all the studies we considered; we focused on mapping the literature.
3. QC is a fast-paced emerging field, which means new innovative studies could have been released since our last search which could contradict and dispute the results of this review.
4. The categorization of studies into different SDLC phases is subject to judgment, particularly for cross-cutting papers that covered more than one SDLC phase.

4.4 Future Research Directions

Some of the gaps identified in Section 3.5 provide useful insights for future research. These future research directions are presented below:

1. Multi-objective quantum optimization: Recent studies have based their research on single-objective optimization. We gave an example with test case optimization, where these studies focus on a single objective, such as minimizing execution time only or fault detection only. But in reality, SE problems are more complex, so future works should focus on multi-objective optimization using quantum techniques, striking a balance between various objectives such as, maximizing fault detection while minimizing execution time and resource usage.
2. Quantum hardware validation: Recent studies show that QC is gradually moving past the Noisy Intermediate-Scale Quantum (NISQ) era, where readily available quantum devices are error-prone and not production-ready. This should also be reflected in academia, where there should be more focus on real quantum hardware validation as opposed to theoretical and simulator-based evaluation. They should use Wang et al. [11, 12]'s work as a point of reference. They should also test the use of quantum techniques to solve complex problems across the entire SDLC, not just the testing phase.
3. Software engineering exploration: SE is a very broad field and consists of more than the SDLC. There are areas of SE that are largely underexplored in QC studies, such as requirements engineering and software architecture design. There are many ways QC can contribute to these areas; for example, quantum algorithms can be used to prioritize requirements in a large agile project and also help model the design of software architecture.
4. Real-word deployable: NIST's finalization of PQC standards [19] is an indication that industries should begin migrating and transitioning their systems to post-quantum cryptography. The migration of these legacy systems is a major problem, and the SE community needs to help address it by developing automated tools and migration frameworks that will aid in this transition. In order to do this effectively, researchers and industry practitioners have to collaborate to ensure that the solutions are real-world deployable and usable.

5. Conclusion

The findings of our review point to the fact that most studies in QSE focus more on the testing and operations phases of the SDLC, focusing on using quantum techniques to optimize various tasks [11–14], while ignoring other SDLC phases such as architecture, requirements engineering, and maintenance. Another area that has received significant attention is the security aspect of SE; this increase in attention is due to the fact that some quantum algorithms, such as Shor's, can break the encryption of classical algorithms, which also led to NIST finalizing the PQC standards [19]. Due to this, there is an urgent need for frameworks that would aid institutions and companies in migrating their systems. Future work should not only develop these frameworks, but also ensure they incorporate multi-objective optimization, are suitable for NISQ hardware, and are validated in real industrial settings; this will help close the gap between academia and industry.

Acknowledgements

This scoping review did not receive any public grant from a non-profit organisation or any funding from a commercial entity. The author(s) would like to acknowledge the Software Engineering Department at Nile University, Abuja, Nigeria, for their support throughout the course of this review; they provided us with unlimited access to their digital libraries and academic databases, including Google Scholar, Scopus, Web of Science, arXiv, and IEEE Xplore.

References

[1] J. Zhao, "Quantum software engineering: Landscapes and horizons," arXiv preprint arXiv:2007.07047, 2020.
[2] M. Piattini, G. Peterssen, R. Pérez-Castillo, J. L. Hevia, M. A. Serrano, and J. M. Murillo, "The Talavera Manifesto for quantum software engineering and programming," in Proc. 1st Int. Workshop QANSWER, 2020, pp. 1–5.
[3] M. Piattini, M. A. Serrano, R. Pérez-Castillo, G. Peterssen, and J. L. Hevia, "Toward a quantum software engineering," IT Professional, vol. 23, no. 1, pp. 62–66, 2021.
[4] S. Ali, T. Yue, and R. Abreu, "When software engineering meets quantum computing," Commun. ACM, vol. 65, no. 4, pp. 84–88, 2022.
[5] J. L. Hevia, G. Peterssen, C. Ebert, and M. Piattini, "Quantum computing," IEEE Softw., vol. 38, no. 5, pp. 7–15, 2021.
[6] A. K. Mandal, M. Nadim, C. K. Roy, B. Roy, and K. A. Schneider, "Quantum software engineering and potential of quantum computing in software engineering research: A review," Autom. Softw. Eng., vol. 32, Art. 27, 2025.
[7] J. M. Murillo, J. L. Hevia, E. Moguel, J. Berrocal, J. García-Alonso, M. Piattini, R. Pérez-Castillo, G. Peterssen, and M. A. Serrano, "Quantum software engineering: Roadmap and challenges ahead," ACM Trans. Softw. Eng. Methodol., 2024.
[8] H. Arksey and L. O'Malley, "Scoping studies: Towards a methodological framework," Int. J. Social Research Methodology, vol. 8, no. 1, pp. 19–32, 2005.
[9] A. A. Khan, M. A. Akbar, A. Waseem, M. Fahmideh, A. Ahmad, P. Liang, M. A. Niazi, and P. Abrahamsson, "Software architecture for quantum computing systems — A systematic review," J. Syst. Softw., vol. 201, Art. 111682, 2023.
[10] C. A. Pérez-Delgado and H. G. Perez-Gonzalez, "Towards a quantum software modeling language," in Proc. ICSEW, 2020, pp. 442–444.
[11] X. Wang, S. Ali, T. Yue, and P. Arcaini, "Quantum approximate optimization algorithm for test case optimization," IEEE Trans. Softw. Eng., vol. 50, no. 12, pp. 3249–3264, 2024.
[12] X. Wang, A. Muqeet, T. Yue, S. Ali, and P. Arcaini, "Test case minimization with quantum annealers," ACM Trans. Softw. Eng. Methodol., vol. 34, no. 1, Art. 5, 2024.
[13] A. Bajaj, A. Abraham, S. Ratnoo, and L. A. Gabralla, "Test case prioritization, selection, and reduction using improved quantum-behaved particle swarm optimization," Sensors, vol. 22, no. 12, Art. 4374, 2022.
[14] A. Trovato, S. Ali, T. Yue, and P. Arcaini, "Reformulating regression test suite optimization using quantum annealing," Int. J. Softw. Tools Technol. Transfer, vol. 26, pp. 767–780, 2024.
[15] S. Ali, P. Arcaini, X. Wang, and T. Yue, "Assessing the effectiveness of input and output coverage criteria for testing quantum programs," in Proc. ICST, 2021, pp. 13–23.
[16] J. L. Hevia, G. Peterssen, and M. Piattini, "QuantumPath: A quantum software development platform," Softw.: Pract. Exper., vol. 52, no. 6, pp. 1517–1530, 2022.
[17] M. Zhang, Y. Li, T. Yue, and K.-Y. Cai, "Quantum optimization for software engineering: A survey," arXiv preprint arXiv:2506.16878, 2025.
[18] E. Moguel, J. Berrocal, J. García-Alonso, and J. M. Murillo, "A roadmap for quantum software engineering," in Proc. Q-SET Workshop, 2020.
[19] NIST, "Post-quantum cryptography standardization," 2024. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography
[20] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The impact of quantum computing on present cryptography," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 3, pp. 405–414, 2018.
[21] M. De Stefano, F. Pecorelli, D. Di Nucci, F. Palomba, and A. De Lucia, "Software engineering for quantum programming: How far are we?," J. Syst. Softw., vol. 190, Art. 111326, 2022.

[22] M. Openja, A. Majidi, F. Khomh, Z. Heng, and H. Li, "An empirical study of quantum computing technical debt," in Proc. Q-SE Workshop, 2022, pp. 14–21.

[23] S. Sepúlveda and A. Cravero, "Systematic review on requirements engineering in quantum computing," Electronics, vol. 13, no. 15, Art. 2989, 2024.

[24] T. Yue, S. Ali, and P. Arcaini, "Towards quantum software requirements engineering," in Proc. QCE, vol. 02, 2023, pp. 161–164.

[25] R. Pérez-Castillo, M. A. Serrano, and M. Piattini, "Software modernization to embrace quantum technology," Adv. Eng. Softw., vol. 151, Art. 102933, 2021.

[26] U. Awan, S. Hannola, A. Tandon, R. K. Goyal, and A. Dhir, "Quantum computing challenges in the software industry: A fuzzy AHP-based approach," Inf. Softw. Technol., vol. 147, Art. 106896, 2022.

[27] M. A. Serrano, R. Pérez-Castillo, and M. Piattini, Eds., Quantum Software Engineering. Springer, 2022.

[28] S. A. Khan, S. Awan, and S. S. Gill, "Quantum software engineering and quantum software development lifecycle: A survey," Cluster Comput., vol. 27, pp. 7451–7468, 2024.

[29] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, 2018.

[30] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandão, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guber, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.