The Quantum Frontier of Software Engineering: A Systematic Mapping Study

Manuel De Stefano[a], Fabiano Pecorelli[a], Dario Di Nucci[a], Fabio Palomba[a], Andrea De Lucia[a]

[a] *SeSa Lab - University of Salerno, Italy*

Abstract

Context. Quantum computing is becoming a reality, and quantum software engineering (QSE) is emerging as a new discipline to enable developers to design and develop quantum programs. **Objective.** This paper presents a systematic mapping study of the current state of QSE research, aiming to identify the most investigated topics, the types and number of studies, the main reported results, and the most studied quantum computing tools/frameworks. Additionally, the study aims to explore the research community's interest in QSE, how it has evolved, and any prior contributions to the discipline before its formal introduction through the Talavera Manifesto. **Method.** We searched for relevant articles in several databases and applied inclusion and exclusion criteria to select the most relevant studies. After evaluating the quality of the selected resources, we extracted relevant data from the primary studies and analyzed them. **Results.** We found that QSE research has primarily focused on software testing, with little attention given to other topics, such as software engineering management. The most commonly studied technology for techniques and tools is Qiskit, although, in most studies, either multiple or none specific technologies were employed. The researchers most interested in QSE are interconnected through direct collaborations, and several strong collaboration clusters have been identified. Most articles in QSE have been published in non-thematic venues, with a preference for conferences. **Conclusions.** The study's implications are providing a centralized source of information for researchers and practitioners in the field, facilitating knowledge transfer, and contributing to the advancement and growth of QSE.

Keywords: Quantum Computing; Quantum Software Engineering; Software Engineering for Quantum Programming; Empirical Software Engineering; Systematic Mapping Study.

1. Introduction

Quantum computing will likely become the next concrete asset for researchers and practitioners. Every developer can now access a quantum computer and use quantum computation to solve arbitrary, computationally-intensive problems [1, 2]. This result is possible because of a great effort made by major software companies, like IBM and Google, which are currently investing hundreds of millions of dollars every year to produce novel hardware and software technologies that can support the execution of quantum programs, making a step further in democratizing quantum computing [3, 4, 5, 6, 7, 8].

Quantum programming is promising in resolving problems in various fields, such as machine learning, optimization, cryptography, and chemistry. However, the development of large-scale quantum software is still distant [9, 10, 11, 12]. To address this, researchers have proposed a new discipline, quantum software engineering (QSE), that extends classical software engineering into the quantum domain [13, 14, 15, 16]. This discipline aims to enable developers to design and develop quantum programs with the same confidence as classical programs, providing them with all the methods and tools necessary [17].

*Corresponding author

Email addresses: madestefano@unisa.it (Manuel De Stefano), fpecorelli@unisa.it (Fabiano Pecorelli), ddinucci@unisa.it (Dario Di Nucci), fpalomba@unisa.it (Fabio Palomba), adelucia@unisa.it (Andrea De Lucia)

Since the publication of the Talavera Manifesto [14], which can be considered the pillar of quantum software engineering, many studies have been published, proposing novel approaches, tools, and techniques. Consequently, several secondary studies have been published exploring various aspects of QSE, such as optimization [18, 19], industrial adoption issues [20], testing [21], and architecture [22], to put order among all the published studies, and steer further research. These previous studies, however, focused on specific aspects of the discipline, like testing [21], and have failed to provide a comprehensive overview of the field. As a result, significant research angles like the management and maintenance of quantum-based solutions have been left largely unexplored. To fully understand the potential of this field, and maximize the benefits obtainable from the current evidence [23], a comprehensive synthesis of the latest research is essential.

In this paper, we propose a systematic mapping study on the current status of QSE research to fulfill this gap. This systematic mapping study aims to provide a broad and holistic overview of QSE, determine its achievements, and identify current research gaps. To this end, the study addresses several research questions about QSE research aspects. In particular, our study aims to identify the most investigated topics in QSE, the types and the number of proposed studies, the main reported results, and the most studied quantum computing tools/frameworks. Additionally, the study explores the research community's interest in QSE, how it has evolved, and any prior contributions to the discipline before its formal introduction through the Talavera Manifesto [14]. Moreover, this study aims to understand the main researchers involved in the field, their research groups, their interactions, and their distribution concerning various SE topics. The study also intends to identify which venues will most likely publish QSE articles outside thematic workshops.

The results of our systematic mapping study can provide valuable insights into the development and evolution of the research community, the tools and frameworks being studied, and the distribution of research topics among different software engineering areas. Our final goal is also to facilitate knowledge transfer and provide a centralized source of information for researchers and practitioners in the field, contributing to the advancement and growth of quantum software engineering.

The remainder of the paper is structured as follows: in Section 2 related literature is discussed and analyzed; Section 3 and Section 4 present the research method employed to conduct our study and the achieved results, which are then discussed in Section 5; in Section 6 threats to validity and potential limitations are discussed, while in Section 7 final remarks and future research directions are reported.

2. Related Work

Quantum Software Engineering (QSE) has recently attracted increasing attention. Hence, several secondary studies were published, exploring different aspects of this discipline. This section compares this literature with our secondary study, highlighting the differences and complementary points.

Yarkoni *et al.* [18] provided a literature review of quantum annealing (QA) technology and its applications. The study aims to provide a centralized source of information on the applications of QA and identify the advantages, limitations, and potential of the technology for both researchers and practitioners.

Shi *et al.* [19] published a study that reviews quantum software optimization techniques, focusing on the efficiency of quantum computing (QC) systems. The study argues that greater efficiency of QC systems can be achieved by breaking the abstractions between layers in the quantum software stack.

Awan *et al.* [20] published a study that aims to identify and prioritize the most critical challenges facing the software industry in adopting quantum computing technology. The study implements a fuzzy analytic hierarchy process (F-AHP) to evaluate and rank the identified challenges. The results show that the critical barriers to quantum computing adoption are the lack of technical expertise, information accuracy, organizational interest in adopting the new process, and the lack of standards of secure communication.

Garcìa de la Barrera *et al.* [21] conducted a systematic mapping study on quantum software testing. The research provides a thorough overview of the current state of the art in quantum software testing. It identifies two major trends in testing techniques: statistical approaches based on repeated measurements and the use of Hoare-like logic for reasoning about software correctness. Reversible circuit testing for quantum software unitary testing is also mentioned by the authors, which is partially applicable to this domain.

Ahmad *et al.* [22] published a secondary study that addresses the challenges of architecture-centric implementation of quantum software systems. To achieve this, it explores the role of quantum software architectures (QSA) in supporting quantum software systems' design, development, and maintenance phases. The research focuses on two main aspects: (i) the architectural process with its architecting activities and (ii) the human roles that can leverage available tools to automate and customize the architecture-centric implementation of quantum software. By examining these aspects, the study aims to provide insights into the principles of quantum mechanics and software engineering practices and how they can be integrated into the design and development of quantum software systems.

Reasoning on the current state of the art, we point out that previous systematic literature reviews and mapping studies have focused on specific themes, e.g., testing or architectures, rather than providing a comprehensive view of the research conducted on quantum software engineering. As a consequence, there is still little knowledge on the set of topics more frequently investigated by quantum software engineering researchers, other than of the main results achieved and how these can drive future research. The findings of our study can provide valuable insights into the development and evolution of the QSE research community, the tools and frameworks being studied, and the distribution of research topics among different software engineering areas. In addition, we also identify a gap of knowledge with respect to the venues and research groups that are currently investigating quantum software engineering aspects: these pieces of information are crucial for newcomers, e.g., fresh Ph.D. students, who would like to embrace the quantum software engineering field, other than more senior researchers interested in understanding the dynamics behind the quantum software engineering research community.

3. Research Method

The *goal* of our study was to produce an organic and holistic view of the scientific literature in the field of quantum software engineering, with the *purpose* of understanding what researchers' efforts have focused on, what needs further investigation, what has been achieved so far, and what the research gaps are. The *perspective* is of both researchers and practitioners. The former are interested in having a unique source of information providing a comprehensive view of the current research in quantum software engineering. The latter are interested in understanding the current trends and technologies researchers produce that might be potentially transferred in practice. Our systematic mapping study has been conducted following the guidelines in [24, 25, 26]. In terms of reporting, we followed the *ACM/SIGSOFT Empirical Standards*[1]

3.1. Research Questions

We aimed to answer various research questions to achieve a comprehensive understanding of the current state of quantum software engineering (QSE). Our goal was to identify potential gaps in QSE research, specifically regarding the coverage of different aspects of software engineering and the level of maturity achieved in these areas.

RQ$_1$ Current research trends in QSE

RQ$_{1.1}$ *How many and what kind of studies have been proposed in QSE?*

RQ$_{1.2}$ *Which areas of software engineering have received the most attention in QSE?*

RQ$_{1.3}$ *Which types of studies are most commonly proposed in the different areas of software engineering within QSE?*

Answering RQ$_{1.1}$ will allow us to assess the degree of maturity in quantum software engineering research. Specifically, a prevalence of philosophical papers indicates a different maturity level compared to a prevalence of evaluation research papers, thus highlighting where the field currently stands in its development. The answer to RQ$_{1.2}$ will enable us to identify potential gray areas or unexplored topics that require further

[1] Available at `https://github.com/acmsigsoft/EmpiricalStandards`. Given the nature of our study and the currently available standards, we followed the *"General Standard"* and *"SystematicReviews"* definitions and guidelines.

attention from the research community. Understanding these gaps is crucial for directing future research efforts where they are most needed. By combining insights from RQ$_{1.1}$ and RQ$_{1.2}$, answering RQ$_{1.3}$ will allow us to assess the maturity of research production in each specific area. This will help evaluate the degree of development within those areas, guiding researchers in prioritizing efforts and resources towards the most impactful and underdeveloped aspects of quantum software engineering.

To gain a comprehensive understanding of the research progress in quantum software engineering (QSE) and to identify potential research opportunities, we posed the following research questions:

🔍 **RQ$_{2}$ Achieved results and studied technologies**

RQ$_{2.1}$ What are the primary types of results reported in the analyzed articles?

RQ$_{2.2}$ Which quantum computing tools/frameworks are most being studied?

Answering RQ$_{2.1}$ will allow us to understand the main types of results reported in the literature. This will provide a clear picture of what is currently being studied, what discoveries have been made, and what areas remain unexplored. This insight is crucial for identifying the nature of contributions to the field, distinguishing between practical applications and theoretical advancements. Answering RQ$_{2.2}$ will provide information on the quantum computing tools and frameworks that are most frequently studied. Understanding which tools and frameworks are the focus of research will help us assess the maturity and adoption of different implementations in the field. This knowledge is essential for mapping the current state of quantum computing technologies and guiding future research efforts towards the most impactful and promising areas.

Although discussions on quantum software have been ongoing prior to the introduction of the Talavera Manifesto in 2020, there is a need to analyze the evolving interest and attention directed towards this emerging discipline. To delve into this aspect further, the following research questions are posed to examine the trajectory of interest and engagement with quantum software within the research community:

🔍 **RQ$_{3}$ Evolution of QSE**

RQ$_{3.1}$ Are there contributions to QSE before the discipline was even born?

RQ$_{3.2}$ How has the research community's interest in this new discipline evolved?

Answering these questions can provide valuable insights into the historical and evolutionary context of QSE, which can be beneficial in several ways. First, understanding the contributions to QSE before the discipline was even born can help researchers appreciate the foundational ideas and concepts that underlie the field. Second, examining how the research community's interest in QSE has evolved over time can provide insights into the current state of the field and its trajectory. By understanding the factors that have driven the growth and development of QSE, researchers can identify areas that require further investment and attention to continue advancing the field.

To gain insights into the key figures shaping the landscape of quantum software research and to explore the dynamics of collaboration and interaction within this field, the following research questions are formulated:

🔍 **RQ$_{4}$ Authors and collaborations in QSE**

RQ$_{4.1}$ Who are the researchers most interested in QSE? How are they interconnected?

RQ$_{4.2}$ Given the various SE topics, how are these researchers distributed?

Answering these questions can be valuable because it can provide a comprehensive understanding of the current landscape of QSE research and the key players in the field. Knowing who the leading researchers are can help identify the most influential and significant work in QSE. Understanding how these researchers and their groups are interconnected can provide insight into how ideas and research are shared and disseminated

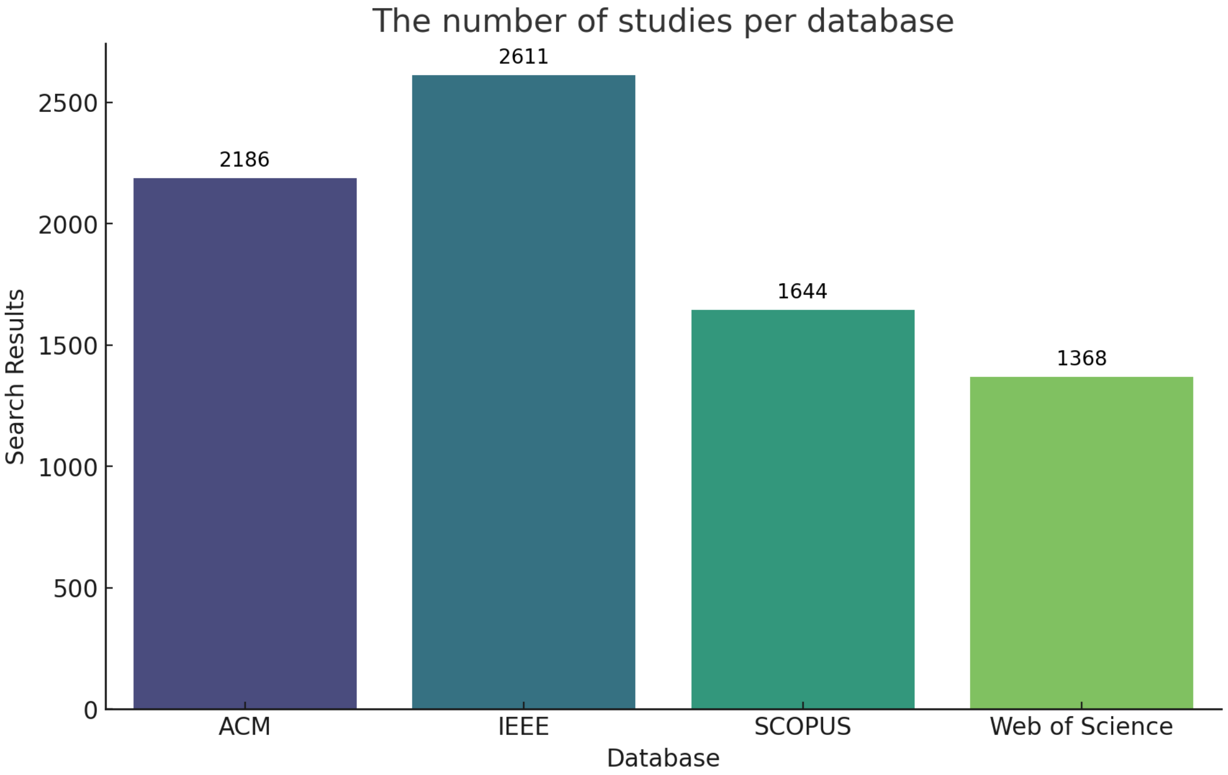


Figure 1: Number of Studies per Database.The values are annotated on top of each bar for clarity.

within the discipline. Understanding the relationships and interactions within the QSE research community can also help identify potential barriers or challenges to collaboration and ways to overcome them.

To enhance our understanding of whether research in quantum software engineering (QSE) is emerging from niche topics and to ascertain which scholarly outlets are particularly appealing to researchers in this field, the following research question is articulated:

RQ$_5$ Publication trends in QSE

RQ$_{5.1}$ Where are studies on quantum software engineering typically published, and do these venues tend to focus on QSE?

Answering this question can be valuable because it can provide insight into the reach and impact of QSE research beyond specialized workshops and conferences. Knowing which venues are most likely to publish QSE articles can help identify the most influential and visible outlets for this type of research and inform strategies for publishing and disseminating work in QSE. Understanding which venues are most attractive to QSE researchers can also help identify trends and patterns in the field and inform decisions about where to submit work for publication. In general, understanding the venues that are most likely to publish QSE articles can help to establish the reach and impact of this type of research and inform strategies for publishing and disseminating work in the field. It must be noted that for *tematic venues*, we intend venues explicitly focused on quantum software engineering or quantum software development.

3.2. Search Criteria

Guidelines [24, 25, 26]suggest that only the variables of population and intervention are necessary to conduct our mapping study. Comparison and outcome are unnecessary for a systematic mapping study

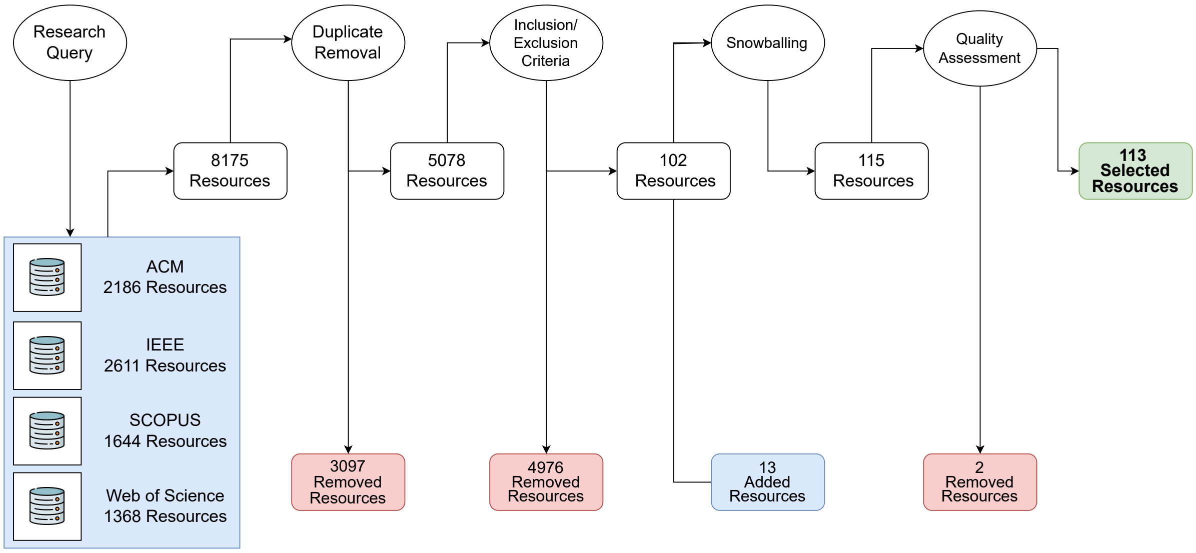


Figure 2: Representation of the study selection process. From the initial set of 5,078 non-duplicated primary studies, 4,975 articles were removed by applying the criteria to titles and abstracts first and to the full texts afterward. By applying the snowballing, 13 articles were added (already filtered with inclusion and exclusion criteria). Finally, two articles were removed by applying the quality assessment criteria. Hence, the final set of included primary studies counts 113 articles.

because it is not a meta-analysis, randomized controlled trial, or systematic review [25, 26]. Instead, it is a preliminary study that provides an overview of the current state of research in a particular area. A systematic mapping study aims to identify gaps in the existing literature and provide a general understanding of the topics, the methods, and the results in a specific field [25]. Therefore, the emphasis is on identifying and categorizing the existing literature rather than comparing or determining outcomes.

We developed the search strings utilizing these two variables (i.e., population and intervention) as the primary keywords to narrow the scope of our research. In software engineering, the term *population* (P) may refer to various subgroups within the field, such as specific software engineering roles, categories of software engineers, application areas, or industry groups. These populations may have unique characteristics or challenges relevant to the research being conducted [24]. In particular, the population in our context refers to papers focusing on quantum software engineering. In the context of software engineering, the term *intervention* (I) refers to any method, tool, technology, or procedure that is implemented or used in the development, maintenance, or optimization of software systems [24]. In our mapping study, we did not limit our research to a single or specific intervention within the field of quantum software engineering. Instead, we sought to broadly examine the various interventions implemented or proposed within this field. This decision allowed us to gain a more comprehensive understanding of the current state of research in quantum software engineering and identify potential trends or gaps in the existing literature. By aggregating the keywords from the P and I and the research questions, we formulated the following research string:

```
("quantum" AND "software" AND ("engineering" OR "development" OR "requirements" OR "quality"
OR "design*" OR "test*" OR "maintenance" OR "management"))
```

The guidelines [24, 25, 26]suggest that employing IEEE and ACM, in addition to two indexing databases, would be sufficient for our mapping study. Therefore, we selected ACM Digital Library, IEEE Xplore, SCOPUS, and Web of Science. We applied our search query to each database, examining all fields. **??** depicts the number of studies obtained by applying the query to each database.

3.3. Study Selection and Quality Assessment

The relevant papers were selected considering the title, abstract, and full-text reading. The inclusion and exclusion criteria were specified and discussed among all the authors [26]. We applied a think-aloud practice describing the process of inclusion/exclusion to one study to align understanding. Based on this discussion, the level of agreement was determined, and the criteria were updated accordingly. After this, the final criteria were applied to the studies by the first author. The third author then reviewed the whole selection process. Articles satisfying at least one of the exclusion criteria listed in Table 1 were excluded. Consequently, articles that were not already excluded by meeting an exclusion criterion and satisfied at least one of the inclusion criteria listed in Table 1 were included.

Applying the whole approach resulted in a total of 115 selected resources. As complementary research approach [26, 27], we performed both forward and backward snowballing —i.e., the selection of publications that cite (*forward snowballing*), or are cited (*backward snowballing*) by each selected study— until no other unseen publication could be added, to guarantee the maximum coverage possible. After filtering them with the same selection criteria (Table 1), 13 resources were added by the end of this procedure.

The final step of the selection phase consists of assessing the quality of the extracted resources to limit the risk of bias and incorrect results [24, 25, 26]. Despite Kitchenam *et al.* [24] do not require systematic mapping studies to conduct a critical appraisal procedure, Petersen *et al.* [26] recommend conducting a quality assessment but not pose too high requirements, not to exclude resources that might be relevant in the scope of a systematic mapping study. Hence, we developed the following quality assessment strategy. We defined a checklist that comprises the following questions:

1. Is the motivation of the paper clearly and explicitly reported?
2. Is the main outcome of the paper clearly and explicitly reported?

The following scores are assigned to each of the above questions. 1 if the answer to the question is *Yes, explicitly* , 0.5 if the answer is *Yes, but not explicitly reported*, 0 if the answer is *Not Reported.* The score of both questions is then summed up. A primary study that achieves a minimum score of 1 is considered acceptable. Applying this quality assessment procedure led to the exclusion of two articles. To conclude, applying the whole procedure resulted in the acceptance of 113 primary studies.

3.4. Data Collection and Analysis

We created an extraction form shown in Table 2 to gather information from the primary studies. This template includes various data items and their corresponding values. It is possible to note that the rationale behind the choice of this gathered information is pretty straightforward: each of the extracted data items corresponds to the phenomenon each research question aims to solve. Table 2 clearly explains this in the description column. The process of extracting this data involved the first author collecting the information and the third author reviewing it for accuracy by comparing it to the original statements in each paper; this was done to mitigate the risk of subjectivity, as depicted in the guidelines [24, 25, 26].

Petersen *et al.* [26] recommend using topic-independent classifications as much as possible. This classification was conducted for the SWEBOK Knowledge Areas [28] (depicted in Section 3.3), the research type (depicted in Table 4) [29, 25, 26], and the venue [26]. Nonetheless, for the result type, a topic-specific classification was needed. Hence, as suggested by Petersen *et al.* [26], we used open coding [30] to organize the articles into categories and then counted the number of articles in each category. We labeled or assigned keywords to concepts found in the text during the open coding process, resulting in several open codes, which were then organized into a larger structure. In this process, we merged or renamed some of the codes representing categories [30]. Then, we used these categories to classify the papers. We applied this process to the abstracts of the articles. However, if the abstracts were not clear enough, we considered other parts of the paper (e.g., introduction and conclusion). The resulting classification is depicted in Table 5.

We employed descriptive statistics tools to analyze the gathered data, as suggested by the guidelines, [24, 25, 26]. In particular, we relied on tables (with bibliographical references) to report absolute frequencies of the categorical variables (e.g., the SWEBOK Knowledge Area of interest of each resource) or bar charts

Table 1: Study selection criteria applied in the study selection phase.

Study Selection Criteria	
Exclusion Criteria	• Article not peer-reviewed. • Peer-reviewed article in English published not earlier than two years than the publication of the Talavera Manifesto (i.e., not earlier than 2018) and not later than 2023. • Article not in English. • Article discussing or proposing SE techniques or practices applied to quantum software development or the quantum software lifecycle.
Inclusion Criteria	• Article published too early than the publication of the Talavera Manifesto (i.e., more than two years earlier) or later than 2023. • Article discussing an empirical investigation on some socio-technical aspects of software engineering applied to quantum software development. • Conference articles extended in journals. • Conference summary. • Secondary or tertiary study on the matter. • Article already selected (i.e., duplicate) or Article whose journal extension was already selected.

(e.g., absolute number of papers per year). To represent co-occurrences of categorical variables, we employed heatmaps and pivot tables depicted as heatmaps (e.g., Knowledge Areas by Research Types). To show the evolution over time of the number of published resources, we employed line graphs (e.g., cumulative number of resources per publication year).

Table 2: Essential data elements collected from the resources, including study ID, article title, research type, research method, SWEBOK focus area, result type, main tool/framework, publication year, authors, and venue/venue type.

Data Item	Description	RQ
Study ID	Integer	
Article Title	Name of the article.	
Research Type	Type of research as reported in the guidelines [25, 26]. A detailed description is given in Table 4	$RQ_{1.2}$, $RQ_{1.3}$
SWEBOK KA	SWEBOK Knowledge Area [28] as representative of a SE topic. A detailed description is given in Section 3.3	$RQ_{1.1}$, $RQ_{1.3}$
Result Type	Type of result achieved by the article. Classification depicted in Table 5.	$RQ_{2.1}$, $RQ_{2.2}$
Tool/Framework	Main quantum tool or framework that was the object of study in the article.	$RQ_{2.2}$
Year	Publication year of the article.	$RQ_{3.1}$, $RQ_{3.2}$
Authors	List of authors of the article.	$RQ_{4.1}$, $RQ_{4.2}$
Venue	Venue of publication of the article.	$RQ_{5.1}$
Venue Type	Type of the publication venue, as depicted in the guidelines (Journal, Conference, Book, Magazine) [26].	$RQ_{5.1}$

4. Analysis of the Results

The following section presents the results of our mapping study. Each of the following paragraphs aims to answer the main topics posed by our research questions.

4.1. RQ1: Main Topics and Studies in QSE

Table 7 illustrates the distribution of analyzed resources categorized by research type. Among these, Solution Proposal papers are the most prevalent, with 41 papers across all focus areas. Afterward, we found Philosophical Papers, comprising 22 resources. Conversely, the Evaluation Research and Validation Research categories, denoting industrial and laboratory empirical studies, are represented with fewer resources (1 and 20, respectively). Experience Papers also make a significant appearance with 11 instances, while Opinion Papers are less common, with a count of five.

Regarding the knowledge areas, as shown in Table 6, it is evident that certain focus areas attract more publications than others. For instance, Software Testing has the highest number of publications, with 31 papers across all research types. Conversely, the Software Engineering Process is the least explored knowledge area, represented by only three resources. The other two highly investigated knowledge areas are Software Construction and Software Engineering Models and Methods, with 18 and 20 resources, respectively. Other knowledge areas present 13 resources or less. Notably, several knowledge areas, such as Software Requirements, Configuration Management, and Software Engineering Management, have not been the subject of any published papers.

Figure 3 gives insights into the variation in the distribution of research types across the different focus areas. In the figure, the number of papers is represented by color darkness, with darker colors indicating a higher number. Philosophical papers focus on Software Design, Software Engineering Models and Methods,

Table 3: SWEBOK Knowledge Areas as described in literature [28]

Knowledge Area	Description
Software Requirements	Deals with gathering, analyzing, specifying, and validating software product requirements.
Software Design	Involves defining architecture, components, and behavior of software systems and creating construction plans based on requirements analysis.
Software Construction	Involves detailed software creation through design, coding, testing, debugging, and verification.
Software Testing	Involves evaluating and improving software quality by identifying defects through dynamic verification against expected behavior.
Software Maintenance	Covers enhancing, adapting, and correcting existing software through various maintenance activities.
Software Configuration Management	Deals with identifying and controlling software system configurations throughout the software life cycle.
Software Engineering Management	Involves planning, coordinating, measuring, reporting, and controlling software projects and programs.
Software Engineering Process	Deals with defining, implementing, assessing, measuring, managing, and improving software life cycle processes.
Software Engineering Models and Methods	Covers modeling principles, types, and analysis for software development.
Software Quality	Covers software quality fundamentals, management processes, defect characterization, and measurement.
Software Engineering Professional Practice	Involves professional and ethical practice of software engineering, including professionalism, codes of ethics, group dynamics, and communication skills.

Table 4: Research type facet as depicted in literature [25, 26, 29]

Category	Description
Validation Research	Methods examined are unique and have not yet been used in practice. Experiments, or laboratory work, are an example.
Evaluation Research	Techniques are put into practice, and their effectiveness is assessed. That is, it is demonstrated how the technique is applied in practice (solution implementation) and what the consequences of the application are in terms of benefits and drawbacks (implementation evaluation). This also includes identifying industry problems.
Solution Proposal	A problem solution is proposed; the solution can be either novel or a significant extension of an existing technique. A small example or a good line of argumentation demonstrates the potential benefits and applicability of the solution.
Philosophical Paper	This kind of paper proposes a new way of viewing existing things by organizing the field into a taxonomy or conceptual framework.
Opinion Paper	This kind of paper expresses someone's personal opinion on whether a particular technique is good or bad, or how things should be done. They make no use of related work or research methodologies.
Experience Paper	This kind of paper is written to describe the practical details of what was done and how it was accomplished, based on the personal experience of the author.

Table 5: Classification of the result types.

Result Type	Description
Empirical	These results generally, but not exclusively, come out from an empirical study. The result is not directly exploitable but sheds light on a phenomenon. This kind of result might come from both Validation and Evaluation research and from Experience, Opinion, and Philosophical papers.
Technique	This result comes from a technique to solve a particular issue. The technique is validated through an empirical study.
Tool	This result has the same characteristics of a technique but is released as an available tool.
Dataset	This result is a collection of useful data or similar. It can be exploited as a reference or for empirical studies.
Guidelines	This type of result is a collection of suggestions for a particular scenario, e.g., for developing a cloud-based quantum hybrid application.
Catalog	Similar to a dataset, but more abstract, it is a collection of useful references (e.g., data encoding patterns).
Position	This result concerns publications that provide no explicit result but formulate a position or opinion, such as philosophical papers.

and Software Construction, but they are almost nonexistent in the other knowledge areas. On the other hand, validation research resources are more evenly distributed over all the knowledge areas, with two to three resources each, except for Software Requirements, Software Engineering Models and Methods, and

Table 6: Number of resources per Swebok Knowledge Area. The references are listed in the Bibliography.

Swebok Knowledge Area	Resources	References
Software Testing	31	[R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31]
Software Engineering Models and Methods	20	[R32, R33, R34, R35, R36, R37, R38, R39, R40, R41, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51]
Software Construction	18	[R52, R53, R54, R55, R56, R57, R58, R59, R60, R61, R62, R63, R64, R65, R66, R67, R68, R69]
Software Design	13	[R70, R71, R72, R73, R74, R75, R76, R77, R78, R79, R80, R81, R82]
Software Quality	13	[R83, R84, R85, R86, R87, R88, R89, R90, R91, R92, R93, R94, R95]
Software Maintenance	8	[R96, R97, R98, R99, R100, R101, R102, R103]
Software Engineering Professional Practice	4	[R104, R105, R106, R107]
Software Engineering Process	3	[R108, R109, R110]
Software Requirements	2	[R111, R112]
Software Configuration Management	1	[R113]

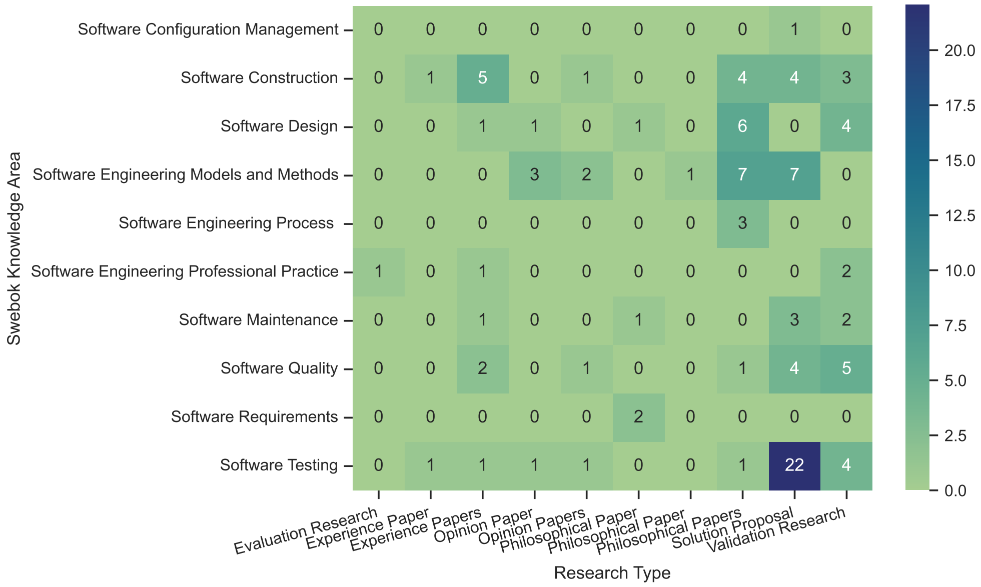


Figure 3: Distribution of published papers by knowledge area and research type. The number of papers is represented by color darkness, with darker colors indicating a higher number of papers. Most papers focus on software testing, followed by software engineering models, methods, and construction. Solution proposals are the most common type of published paper, followed by philosophical papers. Empirical studies, represented by validation and evaluation research categories, are relatively rare.

Table 7: Resources by research type. The references are listed in the bibliography.

Research Type	Resources	References
Solution Proposal	41	[R32, R1, R97, R83, R2, R4, R6, R54, R33, R85, R7, R99, R62, R10, R11, R65, R13, R101, R14, R15, R17, R18, R66, R19, R20, R21, R22, R42, R43, R45, R46, R23, R91, R24, R26, R27, R50, R29, R113, R95, R30]
Philosophical Papers	22	[R55, R34, R35, R58, R36, R37, R60, R38, R61, R12, R73, R108, R109, R110, R41, R75, R90, R76, R44, R77, R78, R79]
Validation Research	20	[R96, R104, R70, R98, R3, R5, R71, R72, R84, R57, R106, R9, R59, R89, R68, R92, R81, R93, R94, R31]
Experience Papers	11	[R52, R53, R8, R87, R100, R107, R63, R64, R88, R74, R67]
Opinion Paper	5	[R47, R49, R28, R51, R82]
Opinion Papers	5	[R56, R86, R39, R16, R40]
Philosophical Paper	4	[R80, R111, R112, R103]
Experience Paper	2	[R25, R69]
Evaluation Research	1	[R105]
Philosophical Paper	1	[R48]

Software Engineering Process, which have none. This phenomenon of some focus areas having a relatively even distribution of papers across the different research types while others have a higher concentration of specific papers can also be observed in other cases. For example, the focus area of Software Engineering Models and Methods has a relatively even distribution of papers across the different research types, with one to seven resources for each research type considered. In contrast, the focus area of Software Testing has a higher concentration of Solution Proposal papers and very few resources in the other research categories.

Main findings for RQ_1

Research in QSE is dominated by Solution Proposal and Philosophical papers, with Empirical studies receiving slightly less attention overall ($\mathbf{RQ}_{1.1}$). While Software Testing, Construction, and Models and Methods receive significant attention, other areas, such as the Software Engineering Process and Management areas have been largely neglected ($\mathbf{RQ}_{1.2}$). The distribution of research types across focus areas is uneven, with some areas having a higher concentration of specific papers than others. For example, Solution Proposal papers are abundant in Software Testing, while Validation Research resources are more evenly distributed across most areas, with some exceptions ($\mathbf{RQ}_{1.3}$).

4.2. RQ2: Main results obtained in QSE

Table 8: Resources per result type.

Result Type	Resources	References
Position	28	[R56, R86, R36, R60, R38, R61, R39, R107, R12, R108, R16, R40, R110, R78, R47, R48, R111, R28, R51, R82, R112, R63, R64, R41, R55, R100, R59, R8]
Technique	26	[R4, R33, R35, R9, R11, R43, R44, R45, R32, R10, R13, R21, R22, R23, R26, R99, R15, R18, R19, R113, R20, R102, R101, R50, R46, R66]
Empirical	25	[R96, R104, R98, R3, R53, R71, R84, R57, R106, R89, R81, R93, R31, R70, R105, R72, R34, R58, R87, R37, R73, R109, R68, R92, R5]
Tool	18	[R1, R2, R6, R62, R14, R17, R91, R95, R54, R7, R24, R30, R52, R83, R65, R29, R97, R42]
Catalog	8	[R85, R74, R90, R76, R77, R79, R75, R103]
Guidelines	6	[R49, R27, R69, R25, R80, R67]
Dataset	2	[R94, R88]

Table 8 depicts the result types achieved in the analyzed resources. The most frequent result is *Position*, with 28 resources. *Technique* results rapidly follow, with 26 resources reporting it as the main achievement. Shortly after, we have *Empirical* as the main result achieved. This result comes from purely empirical studies (i.e., Validation or Evaluation Research) and Experience papers, Opinion Papers, and Philosophical Papers. Mature tools are rare, with only 18 resources reporting it as the main result achieved. *Catalog*, *Dataset*, and *Guidelines* are the least commonly achieved results, with eight, six, and two resources each.

Table 9 shows the quantum technology focused by the analyzed resources. Fifty-four resources did not focus on any specific technology, indicating that many were not interested in exploring a particular quantum technology in depth. On the other hand, 26 resources focused on multiple quantum technologies, indicating an interest in exploring various quantum technologies. Qiskit was the most popular technology, with 18

Table 9: Resources per quantum technology.

Quantum Technology	Resources	References
None	54	[R56, R86, R36, R60, R38, R61, R39, R107, R12, R108, R16, R40, R110, R78, R47, R48, R111, R28, R51, R82, R112, R70, R105, R72, R34, R58, R87, R37, R73, R109, R68, R92, R4, R33, R35, R9, R11, R43, R44, R45, R85, R74, R90, R76, R77, R79, R54, R7, R24, R30, R49, R27, R69, R94]
Multiple	26	[R96, R104, R98, R3, R53, R71, R84, R57, R106, R89, R81, R93, R31, R99, R15, R18, R19, R113, R52, R83, R65, R29, R63, R64, R41, R25]
Qiskit	18	[R1, R2, R6, R62, R14, R17, R91, R95, R32, R10, R13, R21, R22, R23, R26, R55, R88, R80]
Q#	6	[R20, R102, R5, R97, R103, R8]
Dwave	3	[R101, R50, R59]
QASM	3	[R75, R100, R67]
Amazon Braket	1	[R66]
Custom	1	[R42]
QuantME	1	[R46]

resources, followed by Q♯, with six resources. Among the remaining technologies, QASM and Dwave were used by three resources each. In contrast, Amazon Braket, QuantME, and a custom technology were the least popular, each with a single resource focusing on them.

Figure 4 correlates the information carried out by Table 8 and Table 9 to gather further insights. It is also possible to observe that the Result Types Empirical, Technique, Tools, and Position all showed different quantum technologies. Specifically, empirical results mainly focused either on *Multiple* or *None* specific technologies, with the sole exception of a single resource focusing on Q♯. *Position* results, on the other hand, are mainly technology independent, despite focusing on *Multiple*, Q♯, QASM, and Dwawe technologies as well. *Tool* and *Technique* result types show the greatest variety of quantum technologies interested: most of the resources focusing on specific technologies fall into these categories. *Guidelines* results were only reported for QASM, while *Dataset* only for Qiskit. Finally, *Catalogs* results were reported mainly in a technology-agnostic manner, except for a single resource, which focused on QASM.

Main findings for RQ$_2$

Most reported results types are positions, techniques, and empirical, with the other types of results reported in relatively fewer occurrences (**RQ**$_{2.1}$). Most resources focus either on multiple technologies or none specific. Qiskit is the most commonly studied technology for techniques and tools. In contrast, the other technologies were rarely studied on their own (**RQ**$_{2.2}$).

4.3. RQ$_3$: Evolution of the discipline over time

Figure 5a depicts the number of publications over the years, while Figure 5b shows the cumulative number over the years. These figures indicate that the number of papers published in this discipline increased steadily from its inception, with the first publications (2) dating back to 2018. There were six papers in 2019, 15 in

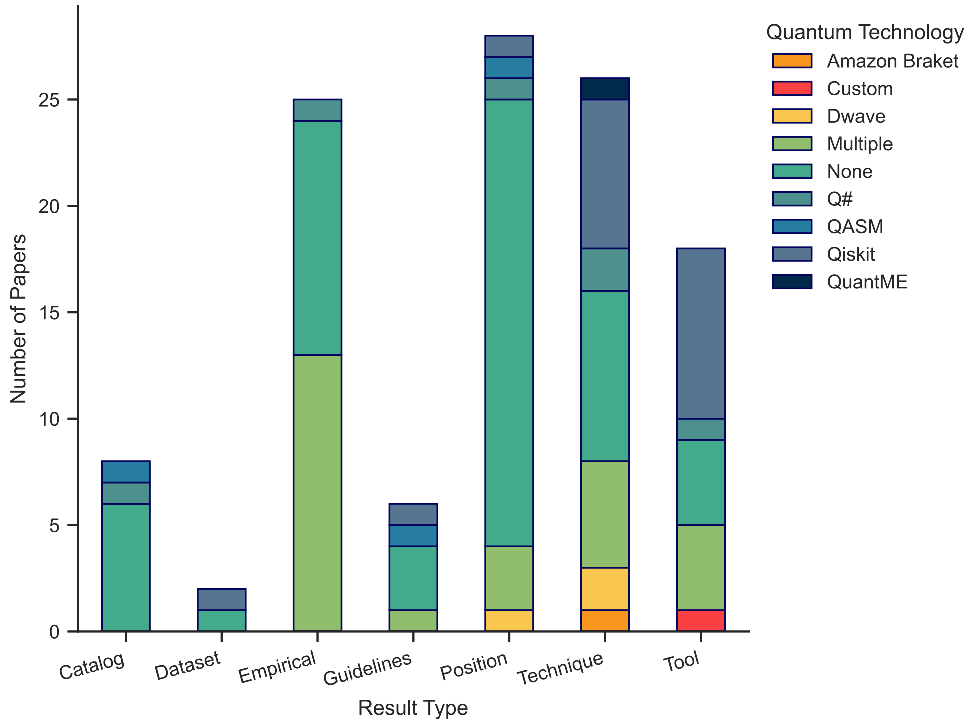
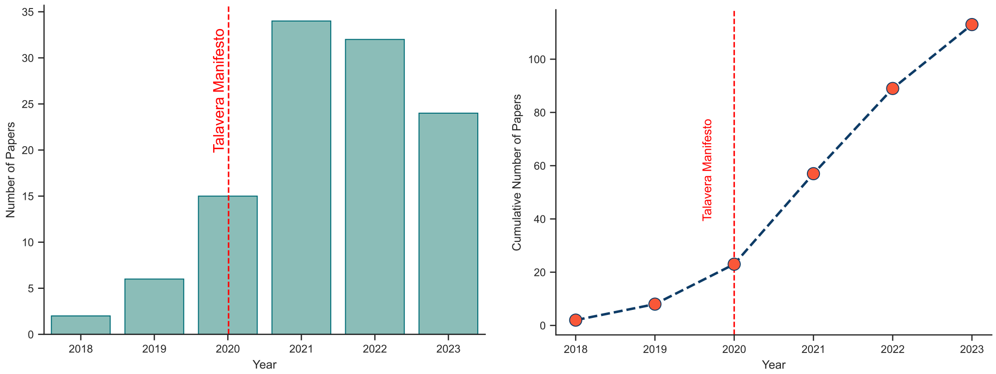


Figure 4: Distribution of the main reported result types alongside the quantum technology interested. The most frequently reported Result Type is *Position*. *Qiskit* is the most commonly mentioned quantum technology, after *None* and *Multiple*.



(a) Publication over years

(b) Cumulative number of publications over the years

Figure 5: In both graphs, the dashed line indicates the year of publication of the Talavera Manifesto, which is considered to have given birth to QSE. Both graphs show a history of contributions to the quantum software engineering field before the discipline's formal establishment in 2020 and that researchers' interest in this area grew significantly, hitting the maximum number of published articles in 2021.

2020, and 34 in 2021. The cumulative number of papers published in this field has also been rising, with two papers published by the end of 2018, eight papers by the end of 2019, 23 papers by the end of 2020, and 57 papers by the end of 2021, reaching a total of 87 by November 2022.

It is worth noting that the Talavera Manifesto, which established the foundations of this discipline, was published in 2020, indicating that the field of quantum software engineering was officially recognized that year. Interestingly, some papers were published in this field before its official establishment. Following 2020, there was a rapid increase in published papers, nearly doubling from 15 in 2020 to 34 in 2021.

However, we observe a decrease in the number of publications in 2023 and a progressive decline in the growth rate of publications in this field: 34 in 2021, 32 in 2022, and 24 in 2023.

Main findings for RQ$_3$

Some papers (8) were published in the field of quantum software engineering before the official establishment of the discipline in 2020 (**RQ**$_{3.1}$). The research community's interest in this discipline grew significantly, with a rapid increase in published papers between 2020 and 2021 (**RQ**$_{3.2}$). However, there was a decreased number of new publications in 2023, suggesting a potential stabilization or temporary slowdown in research activity.

4.4. RQ$_4$: Authors

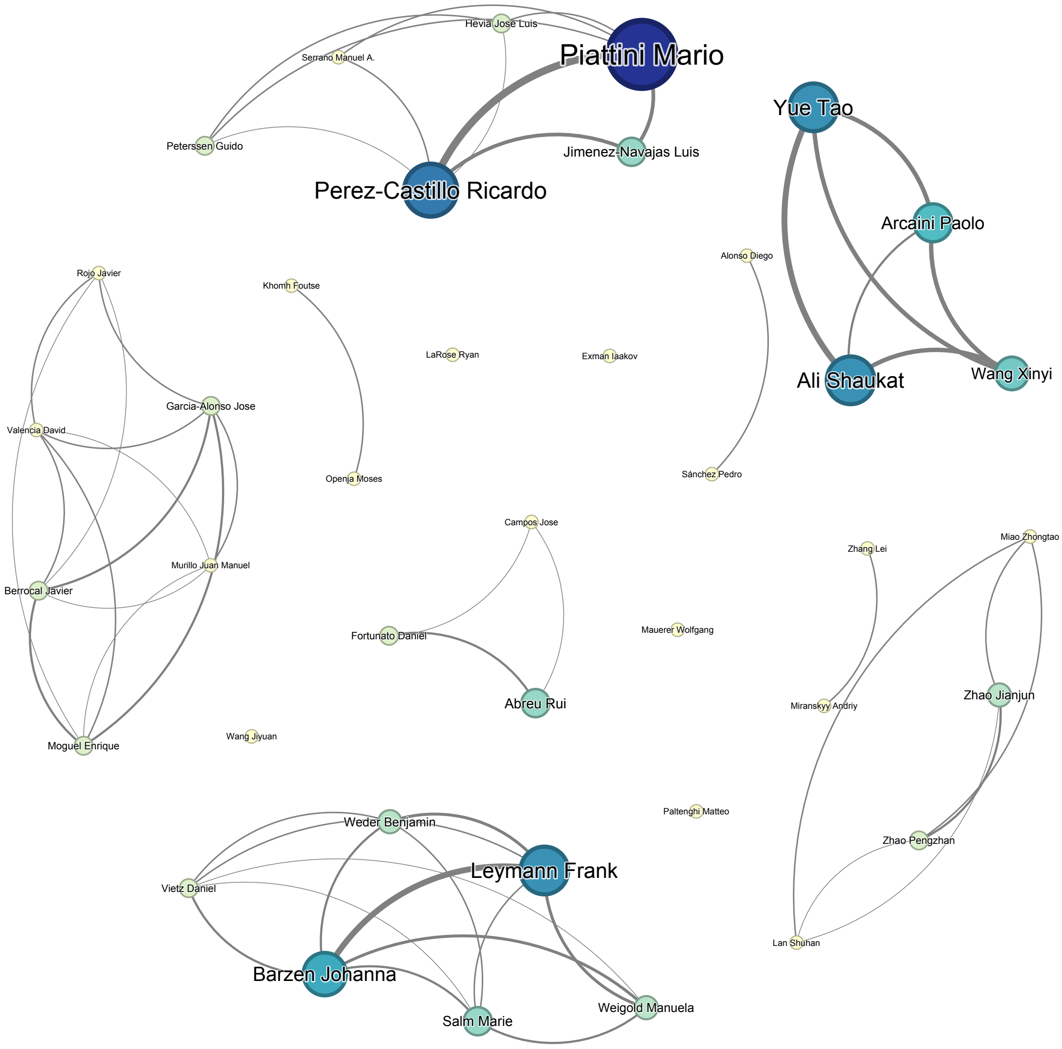


Figure 6: Social network among the most productive authors in the field of Quantum Software Engineering, showing the collaborations among the authors and the number of papers they have published. The nodes' size and the edges' thickness represent the number of papers published by each author and the strength of the collaborations, respectively.

Table 10: Authors by the number of published papers. Mario Piattini stands out as the most productive author, followed by Ricardo Perez-Castillo, Shaukat Ali, and Tao Yue.

Author	Resources	References
Piattini Mario	14	[R52, R97, R33, R86, R36, R99, R38, R101, R68, R102, R43, R90, R45, R50]
Perez-Castillo Ricardo	11	[R97, R33, R36, R99, R38, R101, R68, R102, R43, R45, R50]
Ali Shaukat	11	[R6, R13, R73, R14, R15, R17, R21, R22, R41, R48, R112]
Yue Tao	11	[R6, R13, R73, R14, R15, R17, R21, R22, R41, R48, R112]
Leymann Frank	9	[R58, R109, R74, R110, R75, R76, R77, R46, R79]
Arcaini Paolo	8	[R6, R13, R14, R15, R17, R21, R22, R112]
Barzen Johanna	8	[R58, R109, R74, R110, R76, R77, R46, R79]
Abreu Rui	6	[R1, R2, R4, R5, R61, R23]
Wang Xinyi	6	[R6, R13, R15, R17, R21, R22]
Salm Marie	5	[R74, R110, R77, R46, R79]
Jimenez-Navajas Luis	5	[R33, R38, R101, R102, R43]
Weigold Manuela	4	[R74, R76, R77, R79]
Weder Benjamin	4	[R58, R109, R110, R46]

Table 10 depicts the authors that published more than three papers in QSE. fig. 6 depicts a social network among the most productive authors for the analyzed literature. The network nodes represent authors: the larger the node, the more papers the author has published. The network edges represent collaborations among authors; the edge's thickness represents the number of collaborations between two authors.

Through the network and the table, it is possible to pinpoint the authors who are considered the most productive in the field. Mario Piattini has published 14 papers, followed by Perez-Castillo Ricardo, Shaukat Ali, and Tao Yue, who authored 11 papers each.

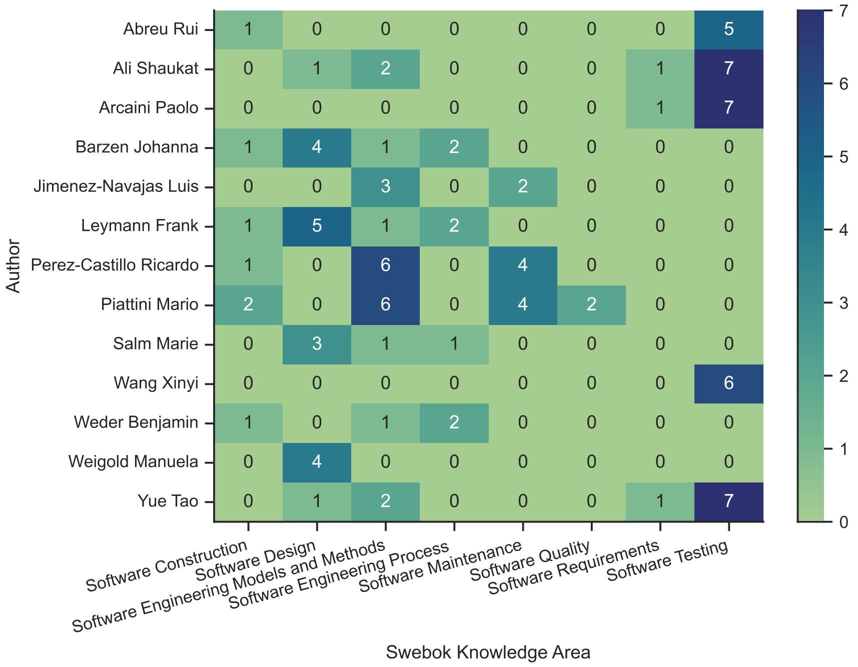
The most substantial collaboration cluster has been identified as the one formed by Mario Piattini and his collaborators Ricardo Perez-Castillo, Luis Jimenez-Navajas, Guido Peterssen, and Manuel Serrano. These authors have a strong collaboration history and the highest number of published papers.

The second strongest collaboration cluster comprises Shaukat Ali, Paolo Arcaini, Tao Yue, and Xinyi Wang. Although it is evident that the cluster size is smaller than other clusters, the strength of the collaborations is higher than any other identified cluster.

Frank Leymann also stands out as having many collaborations, co-authoring papers with Daniel Vietz, Joanna Barzen, Benjamin Weder, Manuela Weigold, and Marie Salm. These authors form a solid cluster regarding published papers and the strength of the collaborations.

Another collaboration cluster comprises Enrique Moguel, Javier Rojo, David Valencia, Jose Garcia-Alonso, Javier Berrocal, and Juan Manuel Murillo. However, this cluster shows a less strong relationship between the authors and fewer papers. Another smaller but noticeable cluster comprises Jianjun Zhao, Zhongtao Miao, Pengzhan Zhao, and Shuhan Lan.

In addition to the strong collaboration clusters identified, it is worth noting that there are many tiny



Author	Software Construction	Software Design	Software Engineering Models and Methods	Software Engineering Process	Software Maintenance	Software Quality	Software Requirements	Software Testing
Abreu Rui	1	0	0	0	0	0	0	5
Ali Shaukat	0	1	2	0	0	0	1	7
Arcaini Paolo	0	0	0	0	0	0	1	7
Barzen Johanna	1	4	1	2	0	0	0	0
Jimenez-Navajas Luis	0	0	3	0	2	0	0	0
Leymann Frank	1	5	1	2	0	0	0	0
Perez-Castillo Ricardo	1	0	6	0	4	0	0	0
Piattini Mario	2	0	6	0	4	2	0	0
Salm Marie	0	3	1	1	0	0	0	0
Wang Xinyi	0	0	0	0	0	0	0	6
Weder Benjamin	1	0	1	2	0	0	0	0
Weigold Manuela	0	4	0	0	0	0	0	0
Yue Tao	0	1	2	0	0	0	1	7

Figure 7: Top authors for each considered Knowledge Area. Mario Piattini is the leading author with the most publications covering many knowledge areas. Four authors have published in multiple areas, while most focus on one or two.

other groups and other one-time collaborations among the authors in the data, which may not be visible in the provided figures, but they likely occur between authors who have published occasionally together.

Figure 7 presents a ranking of top authors in the field of quantum software engineering based on the number of publications related to the considered knowledge areas of the SWEBOK.

The top author in the table, with the highest number of publications overall, is Mario Piattini, as previously shown, who has published a total of 14 papers in the field. Piattini has contributed to a wide range of knowledge areas, with the most publications in Software Engineering Models and Methods (five papers) and Software Quality (two papers). Other notable authors in the table include Yue Tao, with seven publications in the field of software testing, alongside Paolo Arcaini and Shaukat Ali. Interestingly, all the authors have publications focused on a very restricted number of knowledge areas. It is possible to note that there is no author that have published in more than 4 areas, and most of them focused on just one or two areas.

Main findings for RQ$_4$

Mario Piattini, Ricardo Perez-Castillo, Shaukat Ali, and Tao Yue are the researchers most interested in quantum software engineering (QSE), with Piattini being the most productive author in the field. We identified several collaboration clusters, with the most robust one being led by Piattini and his collaborators, Perez-Castillo, Jimenez-Navajas, Peterssen, and Serrano (**RQ**$_{4.1}$). Most researchers have focused on specific knowledge areas, not exceeding their interest in more than four SE topics (**RQ**$_{4.2}$).

4.5. RQ5: Venues

Figure 8 depicts the number of papers published in thematic and non-thematic venues over the years, while Figure 9 depicts the total number of publications by venue type. Table 11 shows the top venues by number of papers. Several insights can be made. First, the field is active and diverse, with research published in various venues. The data shows that papers have been published in conferences (53 papers),

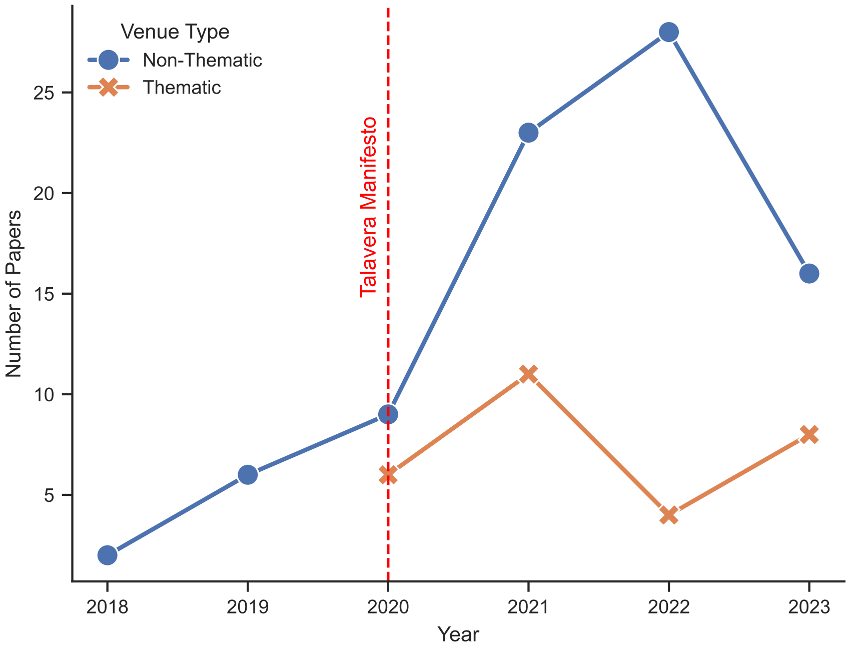


Figure 8: Number of papers published in both thematic and non-thematic venues over the years.

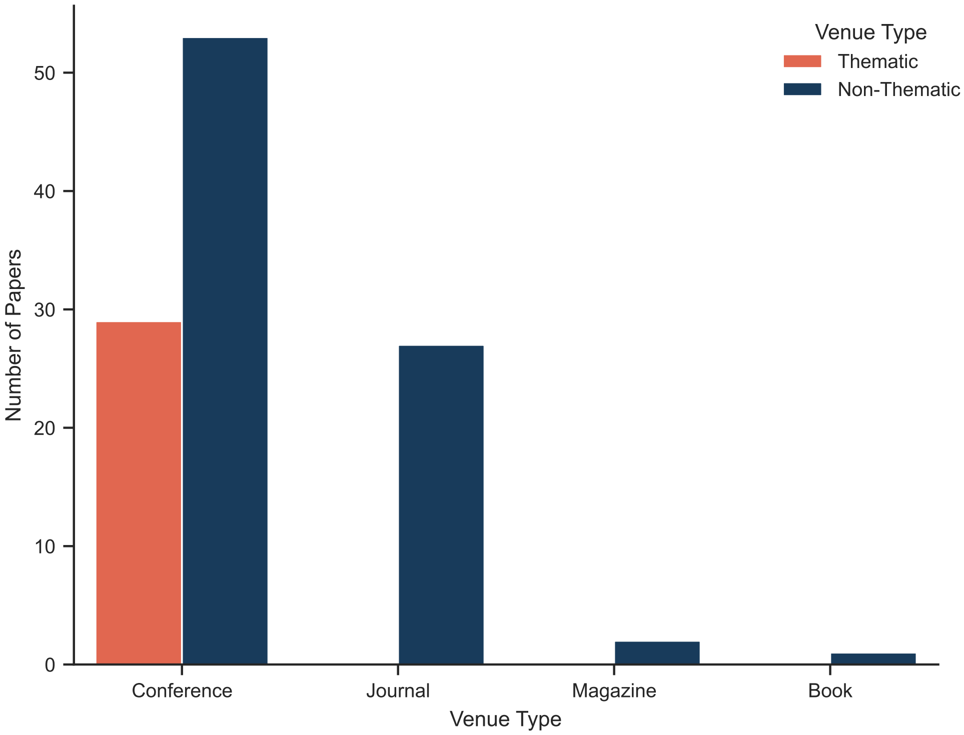


Figure 9: Papers by venue type.

journals (27 papers), magazines (two papers), and books (one paper). This result suggests that the field is well-established and has a strong presence in multiple publication outlets.

Table 11: Top venues by number of papers. The venues are divided into thematic and non-thematic workshops. All thematic venues are reported, whilst non-thematic venues with more than 2 papers are reported.

Venue	Thematic	Articles	References
International Workshop on Quantum Software Engineering (Q-SE)	Yes	12	[R3, R4, R72, R33, R84, R85, R34, R73, R88, R23, R25, R95]
Quantum Software Engineering and Technology Workshop (QSET)	Yes	7	[R35, R8, R86, R60, R38, R61, R39]
QCE	No	5	[R58, R87, R29, R112, R103]
International Workshop on Quantum Software (QSW)	Yes	5	[R101, R49, R80, R27, R93]
ICSE Workshops (ICSEW)	No	4	[R37, R59, R20, R40]
Automated Software Engineering (ASE)	No	4	[R14, R17, R18, R30]
Quantum	No	3	[R83, R63, R65]
The International Conference on the Quality of Information and Communications Technology (QUATIC)	Yes	3	[R102, R43, R90]
International Workshop on the Quantum Software Engineering and Programming (QANSWER)	Yes	1	[R107]
International Workshop on Quantum Programming for Software Engineering	Yes	1	[R48]

Conferences are the most preferred venue type and the only type showing thematic venues. Among conference venues, ICSEW and ASE have published most of the papers in this field, with four papers each. In general, many venues comprised papers about quantum software engineering, indicating that the field is active and diverse, with research published in many different outlets.

Journals have also published many papers in this field, with 27 papers. Quantum Journal, Journal of Systems and Software, Advances in Engineering Software, APEQS, Software Quality Journal, and SummerSOC are among the venues that have published most papers in this field. Several venues, including top-tier journals and conferences, such as IEEE Transactions on Software Engineering, have published only one paper in this field. The remaining non-thematic venues are represented by magazines and books, which have published a relatively small number of papers in this field, i.e., two and one respectively.

Thematic workshops are essential for publishing new ideas in the field. Twenty-nine papers were published in thematic workshop conferences, representing a significant portion of the published papers. Q-SE and QSET have published the most papers, with 12 and seven, respectively.

Figure 8 shows a peak of publications in non-thematic venues in 2022 and a decrease in 2023. Thematic venues show a more swinging trend, with a decrease in 2022 and a subsequent increase in 2023. It is worth mentioning that thematic venues were inaugurated in 2020 after the publication of the Talavera Manifesto.

Main findings for RQ$_5$

Since thematic venues represent a significant proportion of the publication venues, articles in QSE have been published in many other non-thematic venues, with a preference for conferences over others. The preferred non-thematic publication venues are ICSE Workshops and ASE, both having four publications each ($\mathbf{RQ}_{5.1}$).

5. Discussion and Insights for Future Research

5.1. On the current research trends in QSE

The resource distribution analysis reveals several key insights. First, the high prevalence of Solution Proposal papers (33 resources across all focus areas) suggests a strong demand for practical solutions. This result implies that researchers are actively working to address the challenges practitioners face and develop innovative approaches that can help improve software engineering practices. The most challenging aspect is indeed related to testing and debugging quantum programs, for the intrinsic issues of the practice [21].

Second, the few resources focusing on the software engineering processes (only three resources) highlight the need for further research. It suggests that there is still much to be discovered regarding best practices, processes, and methodologies for software engineering. More research is needed to advance our understanding of this area. Moreover, we saw that some knowledge areas in software engineering, such as software requirements, configuration management, and software engineering management, have been neglected and have not been the subject of any published papers.

In particular, the lack of software project management can have significant implications for the socio-technical aspects of quantum software development, such as communication and collaboration among developers, which we can assume come from different backgrounds [17, 13, 31]. Effective management is critical to ensure that resources are effectively utilized, goals are aligned, and stakeholders are effectively engaged.

Such areas could have been neglected because they are perceived as less technically challenging or trending. However, the issue could also be a prioritization choice of the researchers, who focus on solving more tangible and concrete problems, such as testing and debugging [21]. Nevertheless, more research is necessary to address the issues and ensure all aspects of software engineering are adequately covered.

Third, Validation research resources are distributed relatively evenly across focus areas, with two to three resources each, highlighting its significance in software engineering. This type of research is important for providing evidence and supporting effective practices and solutions. At the time of writing, however, empirical research in QSE is limited compared to other activities like solution proposals. Additionally, the discipline almost completely lacks industrial empirical studies.

Finally, the unequal distribution of research types in different focus areas (e.g., a higher concentration of Solution Proposals in Software Testing) underscores the importance of considering each area's unique characteristics and needs when planning research activities to help ensure that research efforts are directed toward the areas that will have the most significant impact.

Take Away Message 1. Several areas require **further research.** These include software engineering processes, software project management, neglected knowledge areas such as software requirements and configuration management, and the need for more academic and industrial empirical studies in quantum software engineering. Additionally, it is important to **consider the unique characteristics and needs of each area** when planning research activities to ensure that efforts are directed toward those that will have the most significant impact.

5.2. On the achieved results and studied technologies

In Section 4, we presented the main results achieved by the analyzed resources and the main quantum technologies interested. The results provide insights into the state of research in quantum software engineering and highlight the diversity of research in this field.

The technique was the most commonly achieved result type, with 22 resources reporting it. As a newly born discipline, the field of quantum software engineering is still in its early stages of development. Thus, it is unsurprising that techniques are the most prominent result types reported in the resources analyzed. In the early stages of a field's development, researchers tend to focus on developing methods and frameworks that can be used to tackle specific challenges because the field is still in its formative years, and it is essential to establish a solid foundation of techniques and tools that can be used to advance the field further. In quantum software engineering, the development of new techniques has been driven by the need to address various challenges in quantum software development [17, 13, 15].

The finding that a considerable number of resources focus on multiple or no specific quantum technologies can be attributed to the interdisciplinary nature of the field and the need to consider multiple technologies and perspectives. As said, QSE is a relatively new field, and researchers and practitioners are still exploring the potential of different quantum technologies and their integration with classical computing systems [14, 17]. The use of multiple or no specific technologies may aim to develop technology-agnostic methods that can be used with various quantum technologies, as advocated by the Talavera Manifesto [14].

The resources that propose tools as the primary result in quantum software engineering are more focused on specific technologies because there is a practical need for concrete implementations in this field. As a newly-born discipline, there is still much room for exploration and experimentation. Tools provide a tangible means for researchers and practitioners to test and validate their ideas and for users to put the theories into practice. The concentration of resources focused on specific technologies like Qiskit indicates that this technology has become popular among the community, possibly due to its ease of use, user-friendly interface, and compatibility with various quantum computing hardware. This popularity will likely drive the development of many tools and techniques, further increasing its popularity and usability. In addition to tools addressing developers' needs, tools represent a fundamental requirement for conducting experiments on codebases, which explains the significant effort devoted to developing tools early in a newly-born discipline. In conclusion, the prominence of the tool result type and the concentration of resources focused on specific technologies highlights the practical and tangible nature of quantum software engineering and the importance of concrete implementations for advancing the field.

The low representation of result types such as Datasets, Guidelines, and Catalogs in the analyzed resources could be due to several reasons. Firstly, the field of quantum software engineering is relatively new and still in its early stages of development. As a result, there might not be a strong emphasis on building extensive datasets, guidelines, or catalogs to support quantum software development. Instead, the focus is on developing techniques and tools necessary for software engineering practices in the quantum domain. Additionally, collecting and preparing large datasets, guidelines, or catalogs in the quantum software engineering domain could be challenging because quantum technologies are still being developed and standardized. The associated data and information are constantly evolving, making it challenging to build robust and comprehensive datasets, guidelines, or catalogs that can be widely adopted and used by the research community. Finally, the low representation of Datasets, Guidelines, and Catalogs could also be attributed to the current state of research in quantum software engineering. Many researchers in this field might focus more on exploring new techniques and tools rather than investing time and resources into building datasets, guidelines, and catalogs. As the field of quantum software development grows, it's crucial to have comprehensive and current resources to support it. However, the lack of such resources shows a significant research gap in quantum software engineering. As the discipline matures, the need for these resources is likely to become more evident, and more researchers may push in this direction.

Take Away Message 2. Further research in quantum software engineering should focus on **filling the research gap** highlighted by the low representation of result types such as Datasets, Guidelines, and Catalogs in the analyzed resources. Building **comprehensive and up-to-date resources** to support quantum software development could be essential in advancing the field further. Additionally, **exploring new techniques and tools** to tackle specific challenges and investigating the potential of different quantum technologies and their integration with classical computing systems should remain a priority for researchers and practitioners.

5.3. On the evolution of QSE

The field of quantum software engineering has experienced rapid growth in recent years, as evidenced by the significant increase in publications in the discipline. The establishment of the discipline, through the publication of the Talavera Manifesto in 2020, has provided a clear definition and set of guidelines for future research and development in this field. This milestone has been crucial in recognizing quantum software engineering as a distinct discipline and will continue to play an essential role in shaping its future.

Early publications in quantum software engineering indicate that researchers and practitioners have already explored and developed the concept of quantum software engineering, even before its formal recognition. This result highlights the growing importance and recognition of the field and its potential to impact the future significantly. The rapid increase in the number of publications, doubling from 2020 to 2021, suggests that there has been a significant increase in research and development in this field, and it is likely to continue to proliferate. This increase in publications also suggests that quantum software engineering attracted more attention from researchers, practitioners, and investors, who recognize its potential to drive innovation and impact in various industries and domains, despite the sligtly decrease of absolute number of publications in 2022 and 2023.

Based on the rapid growth and increasing interest in the field of quantum software engineering, it is likely that the discipline will continue to experience significant development in the near future. The growing recognition and importance of the field, the support and demand from various industries and domains, and the potential of quantum computing and quantum software to revolutionize the way we process and store data, optimize complex systems, and solve problems in various fields, will drive further investment and research in quantum software engineering. This discipline is expected to have significant growth and development in the near future and will play a crucial role in shaping technology and innovation.

> **Take Away Message 3.** The rapid growth and increasing interest in the field of quantum software engineering suggest that it will continue to experience significant development in the near future. To further advance the discipline, future research could explore the **development of best practices, standards, and tools** for quantum software engineering and investigate the **potential applications** of quantum software in various industries and domains.

5.4. On authors and collaborations in QSE

In Section 4, we pointed out which are the most interested authors in QSE in the software engineering community, how they are related, and how they are distributed among the SWEBOK knowledge areas.

The social network analysis conducted on QSE authors has revealed several collaborative clusters with varying degrees of strength and cohesion. These clusters suggest that QSE researchers tend to form closely-knit communities, which could hinder the exchange of knowledge and ideas between different groups. To address this issue, initiatives to promote interdisciplinary collaboration among researchers in different collaboration clusters may be advantageous. These initiatives may include hosting joint conferences or workshops to facilitate the exchange of knowledge and ideas, potentially leading to novel insights and discoveries in the field of QSE. Such efforts may also aid in building more robust and diverse research communities, enabling the development of innovative approaches to QSE problems.

The analysis of the knowledge areas of QSE authors reveals that most researchers focus on specific areas rather than the entire spectrum of QSE knowledge. While some authors have published many papers, they mostly focus on a few knowledge areas, such as software design and engineering processes. In contrast, other areas, such as software quality and software maintenance, are less well-represented, which suggests that there is a need for more research on these underrepresented areas to create a more holistic understanding of quantum software engineering. In addition, very few authors publish in multiple areas suggesting that there may be a lack of interdisciplinary collaboration between quantum computing and software engineering experts. Encouraging more collaboration between these groups could help to create a more balanced research focus in quantum software engineering.

Take Away Message 4. The social network analysis of QSE authors highlights the need for initiatives to **promote interdisciplinary collaboration between closely-knit research communities,** which could lead to novel insights and discoveries in QSE and help build more robust and diverse research communities. Additionally, the analysis of QSE authors' coverage of SWEBOK knowledge areas reveals a need for **more research on underrepresented areas** such as software testing and maintenance. Encouraging collaboration between quantum computing and software engineering experts could help create a more balanced research focus in quantum software engineering.

5.5. On the publication trends in QSE

The field of quantum software engineering is a growing and dynamic field that is seeing an increase in the number of publications in various venues. Despite the significant share represented by thematic venues, more recently, non-thematic venues were preferred. The preference for non-thematic venues over thematic ones may suggest the field is leaving its niche discipline status and permeating into well-established software engineering publication outlets. This result can be seen as a positive trend, indicating that quantum software engineering is gaining recognition and attention from the broader software engineering community. Reaching a broader audience through non-thematic venues may help to promote the field and increase its visibility, leading to more collaboration and exchange of ideas with other software engineering sub-fields. Furthermore, publishing in non-thematic venues may also increase the impact of the research, as it can reach a more significant number of readers who may not be familiar with the specific field. Moreover, publishing in non-thematic venues also allows QSE researchers to showcase their work to a wider audience and demonstrate their research's relevance and potential impact in the broader context of software engineering. It may also help attract new researchers and provide a platform for interdisciplinary collaboration. In conclusion, such evolution could give birth to new research directions and the advancement of existing ones and could further support the establishment of the field.

It is important to note that the data presented covers up to December 2022 (excluded) and may not accurately represent the trends in the field over the entire year. It could be beneficial to gather data on other factors that might influence the publication trends in this field, such as funding or resource availability.

Take Away Message 5. The increasing preference for non-thematic publication venues in quantum software engineering suggests that the field is gaining recognition and attention from the broader software engineering community, which could lead to new research directions and interdisciplinary collaborations. However, it would be beneficial to **investigate other factors influencing publication trends,** such as funding or resource availability. Furthermore, more research is needed to understand how the **integration** of quantum software engineering with the broader software engineering community can be leveraged further to support the growth and recognition of the field.

6. Threats to Validity

The following discusses potential threats and limitations to the study's validity.

6.1. Descriptive Validity

The extent to which observations are accurately and objectively described is descriptive validity. Threats to descriptive validity are prevalent in qualitative and quantitative studies. As reported by the guidelines [24, 25, 26], a data collection form has been designed to support data recording to reduce this threat. As with the primary studies, the form objectified the data extraction process and allowed it to be revisited regularly. Moreover, as depicted by the guidelines [26], not only we extracted the research type and method (which were also fundamental for answering our research questions), but we also mainly applied topic-independent classifications. For topic-specific classification, i.e., for the main results of the primary studies, we applied open coding [30], as the guidelines recommend [26]. As a result, this threat is considered to be under control.

6.2. Theoretical Validity

The ability to capture what we intend to capture determines theoretical validity. Furthermore, confounding factors such as biases and subject selection play a significant role. In our case, this refers to the study identification (or sampling) and the data extraction and classification.

As Wohlin *et al.* [27] pointed out, the selection of the studies could have threatened the validity of the study. To mitigate this risk, we supplemented the search with backward and forward snowball sampling [32] of all studies included by full-text reading. In addition, to increase the reliability of the inclusion and exclusion criteria, we applied the think-aloud protocol to the criteria defined and discussed among all the authors. Then the first author conducted the selection process that was reviewed by the third afterward.

Researcher bias is a threat regarding data extraction and classification. Following the guidelines [24, 25, 26], the third author assessed all extractions made by the first author to mitigate this threat. Nonetheless, since this step involves human judgment, the threat cannot be entirely excluded.

7. Conclusions

In this systematic mapping study, we aimed to provide a comprehensive overview of the current state of research in Quantum Software Engineering (QSE). Through an extensive analysis of 113 studies, our main contributions are: (i) a comprehensive synthesis and analysis of the research conducted in the field of quantum software engineering, which might be useful to researchers and practitioners to learn how software engineering and quantum computing have been combined so far; (ii) a systematic mapping of the venues and research groups that are currently focusing on quantum software engineering, which may be useful to newcomers and researchers interested in starting their path in this research field to discover relevant venues and potential collaborators; (iii) A research roadmap that highlights the major achievements that further research should pursue; (iv) An online appendix reporting all the material used to conduct our systematic mapping study, which researchers might use to build on top of our research and extend our findings[2].

Our results indicate that QSE research is still in its early stages, focusing on software testing and neglecting some knowledge areas, such as software engineering management. The most reported results types are techniques, empirical, and positions, with Qiskit being the most commonly studied technology. We also observed a growing interest in QSE within the research community, with a speedy increase in published papers between 2020 and 2021. Regarding researchers, we identified the most productive authors, the main collaboration clusters, and the distribution of researchers across different SE topics. This information can help to identify potential collaborators and promote further research in QSE. Finally, our study highlights the need for more empirical studies and a better distribution of research efforts across different SE topics. We also encourage more non-thematic publication venues to consider QSE papers to broaden the research community's knowledge and reach. Our study provides valuable insights into the development and evolution of the research community, contributing to the advancement and growth of QSE.

This systematic mapping study highlights potential future directions for research in Quantum Software Engineering (QSE), especially in the neglected areas of software engineering management practices and quantum software maintenance. Future research could focus on developing effective strategies and tools for managing the software development process and maintaining reliable and performing quantum software over time. Additionally, there is a need for more empirical studies and the development of appropriate metrics to provide a more rigorous empirical basis for future research in QSE. Future research could investigate effective strategies for managing the software development process in the context of quantum computing to address the neglect of software engineering management practices in QSE. Such efforts could involve exploring the unique challenges and opportunities of quantum software engineering, identifying effective strategies for managing the development process, and evaluating the effectiveness of different software engineering practices and tools. Research on quantum software maintenance could also provide new insights and discoveries. Future studies could explore the challenges and opportunities of maintaining quantum software over time, including

[2] Appendix available at `https://figshare.com/articles/online_resource/The_Quantum_Frontier_of_Software_Engineering_A_Systematic_Mapping_Study/22263448`

issues such as version control, bug fixes, and updates. Developing appropriate metrics could also help provide a more rigorous empirical basis for future research in QSE, allowing for the evaluation of code quality, performance, and reliability.

Appendix

Our online appendix is avalilable at the following link `https://figshare.com/articles/online_resource/The_Quantum_Frontier_of_Software_Engineering_A_Systematic_Mapping_Study/22263448`.

Acknowledgement

This work has been partially supported by project "QUASAR: QUAntum software engineering for Secure, Affordable, and Reliable systems", grant 2022T2E39C, under the PRIN 2022 MUR program funded by the EU - NGEU

References

[1] W. Knight, Serious quantum computers are finally here. what are we going to do with them, MIT Technology Review. Retrieved on October 30 (2018) 2018.

[2] T. Hoare, R. Milner, Grand challenges for computing research, The Computer Journal 48 (2005) 49–52.

[3] B. Ömer, Qcl-a programming language for quantum computers, Software available on-line at http://tph. tuwien. ac. at/˜ oemer/qcl. html (2003).

[4] Q#: A quantum programming language, `https://qsharp.community`, 2021. Accessed: 2021-09-21.

[5] T. Altenkirch, J. Grattage, Qml: Quantum data and control, Submitted for publication, Febuary (2005).

[6] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, et al., Qiskit: An open-source framework for quantum computing, Accessed on: Mar 16 (2019).

[7] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, et al., Tensorflow quantum: A software framework for quantum machine learning, arXiv preprint arXiv:2003.02989 (2020).

[8] D. S. Steiger, T. Häner, M. Troyer, Projectq: an open source software framework for quantum computing, Quantum 2 (2018) 49.

[9] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (2017) 195–202.

[10] G. G. Guerreschi, M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, arXiv preprint arXiv:1701.01450 (2017).

[11] L. O. Mailloux, C. D. Lewis II, C. Riggs, M. R. Grimaila, Post-quantum cryptography: what advancements in quantum computing mean for it professionals, IT Professional 18 (2016) 42–47.

[12] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, M. Troyer, Elucidating reaction mechanisms on quantum computers, Proceedings of the National Academy of Sciences 114 (2017) 7555–7560.

[13] M. Piattini, M. Serrano, R. Perez-Castillo, G. Petersen, J. L. Hevia, Toward a quantum software engineering, IT Professional 23 (2021) 62–66.

[14] M. Piattini, G. Peterssen, R. Pérez-Castillo, J. L. Hevia, M. A. Serrano, G. Hernández, I. G. R. de Guzmán, C. A. Paradela, M. Polo, E. Murina, et al., The talavera manifesto for quantum software engineering and programming., in: QANSWER, 2020, pp. 1–5.

[15] M. Piattini, G. Peterssen, R. Pérez-Castillo, Quantum computing: A new software engineering golden age, ACM SIGSOFT Software Engineering Notes 45 (2020) 12–14.

[16] E. Moguel, J. Berrocal, J. García-Alonso, J. M. Murillo, A roadmap for quantum software engineering: Applying the lessons learned from the classics., in: Q-SET@ QCE, 2020, pp. 5–13.

[17] J. Zhao, Quantum software engineering: Landscapes and horizons, arXiv preprint arXiv:2007.07047 (2020).

[18] S. Yarkoni, E. Raponi, T. Bäck, S. Schmitt, Quantum annealing for industry applications: Introduction and review, Reports on Progress in Physics (2022).

[19] Y. Shi, P. Gokhale, P. Murali, J. M. Baker, C. Duckering, Y. Ding, N. C. Brown, C. Chamberland, A. Javadi-Abhari, A. W. Cross, et al., Resource-efficient quantum computing by breaking abstractions, Proceedings of the IEEE 108 (2020) 1353–1370.

[20] U. Awan, L. Hannola, A. Tandon, R. K. Goyal, A. Dhir, Quantum computing challenges in the software industry. a fuzzy ahp-based approach, Information and Software Technology 147 (2022) 106896.

[21] A. García de la Barrera, I. García-Rodríguez de Guzmán, M. Polo, M. Piattini, Quantum software testing: State of the art, Journal of Software: Evolution and Process (2021) e2419.

[22] A. Ahmad, A. A. Khan, M. Waseem, M. Fahmideh, T. Mikkonen, Towards process centered architecting for quantum software systems, in: 2022 IEEE International Conference on Quantum Software (QSW), IEEE, 2022, pp. 26–31.

[23] B. A. Kitchenham, T. Dyba, M. Jorgensen, Evidence-based software engineering, in: Proceedings. 26th International Conference on Software Engineering, IEEE, 2004, pp. 273–281.

[24] S. Keele, et al., Guidelines for performing systematic literature reviews in software engineering, Technical Report, Technical report, ver. 2.3 ebse technical report. ebse, 2007.

[25] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, 2008, pp. 1–10.

[26] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and software technology 64 (2015) 1–18.

[27] C. Wohlin, P. Runeson, P. A. d. M. S. Neto, E. Engström, I. do Carmo Machado, E. S. De Almeida, On the reliability of mapping studies in software engineering, Journal of Systems and Software 86 (2013) 2594–2610.

[28] P. Bourque, R. Fairley, Swebok, Nd: IEEE Computer society (2004).

[29] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements engineering 11 (2006) 102–107.

[30] A. Strauss, J. Corbin, Grounded theory methodology: An overview. (1994).

[31] S. Ali, T. Yue, R. Abreu, When software engineering meets quantum computing, Communications of the ACM 65 (2022) 84–88.

[32] S. Jalali, C. Wohlin, Systematic literature studies: database searches vs. backward snowballing, in: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, 2012, pp. 29–38.

Selected Studies

[R1] D. Fortunato, J. Campos, R. Abreu, Qmutpy: A mutation testing tool for quantum algorithms and applications in qiskit, in: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, 2022, pp. 797–800.

[R2] D. Fortunato, J. Campos, R. Abreu, Mutation testing of quantum programs: A case study with qiskit, IEEE Transactions on Quantum Engineering 3 (2022).

[R3] G. Pontolillo, M. R. Mousavi, A Multi-Lingual benchmark for Property-Based testing of quantum programs (2022).

[R4] R. Abreu, J. P. Fernandes, L. Llana, G. Tavares, Metamorphic testing of oracle quantum programs, in: 2022 IEEE/ACM 3rd International Workshop on Quantum Software Engineering (Q-SE), IEEE, 2022, pp. 16–23.

[R5] M. Trinca, J. F. Ferreira, R. Abreu, A preliminary study on generating Well-Formed q# quantum programs for fuzz testing (2022).

[R6] X. Wang, P. Arcaini, T. Yue, S. Ali, QuSBT: Search-Based testing of quantum programs (2022).

[R7] L. Burgholzer, R. Wille, QCEC: A JKQ tool for quantum circuit equivalence checking[formula presented], Software Impacts 7 (2021).

[R8] M. Mykhailova, M. Soeken, Testing quantum programs using q# and microsoft quantum development kit., in: Q-SET@ QCE, 2021, pp. 81–88.

[R9] G. Li, L. Zhou, N. Yu, Y. Ding, M. Ying, Y. Xie, Projection-based runtime assertions for testing and debugging quantum programs, Proceedings of the ACM on Programming Languages 4 (2020).

[R10] H. Zhou, G. T. Byrd, Quantum circuits for dynamic runtime assertions in quantum computation, IEEE Computer Architecture Letters 18 (2019) 111–114.

[R11] Y. Huang, M. Martonosi, Statistical assertions for validating patterns and finding bugs in quantum programs, in: Proceedings of the 46th International Symposium on Computer Architecture, 2019, pp. 541–553.

[R12] M. Paltenghi, Cross-platform testing of quantum computing platforms, in: Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, 2022, pp. 269–271.

[R13] X. Wang, P. Arcaini, T. Yue, S. Ali, Application of combinatorial testing to quantum programs, in: 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2021, pp. 179–188.

[R14] E. Mendiluze, S. Ali, P. Arcaini, T. Yue, Muskit: A mutation analysis tool for quantum software testing, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2021, pp. 1266–1270.

[R15] S. Ali, P. Arcaini, X. Wang, T. Yue, Assessing the effectiveness of input and output coverage criteria for testing quantum programs, in: 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST), IEEE, 2021, pp. 13–23.

[R16] A. Miranskyy, L. Zhang, On testing quantum programs, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), IEEE, 2019, pp. 57–60.

[R17] X. Wang, P. Arcaini, T. Yue, S. Ali, Quito: a coverage-guided test generator for quantum programs, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2021, pp. 1237–1241.

[R18] J. Wang, Q. Zhang, G. H. Xu, M. Kim, Qdiff: Differential testing of quantum software stacks, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2021, pp. 692–704.

[R19] J. Wang, F. Ma, Y. Jiang, Poster: Fuzz testing of quantum program (2021).

[R20] S. Honarvar, M. R. Mousavi, R. Nagarajan, Property-Based testing of quantum programs in q#, in: Proceedings of the IEEE/ACM 42nd International Conference on Software, Association for Computing Machinery, 2020, pp. 430–435.

[R21] X. Wang, T. Yu, P. Arcaini, T. Yue, S. Ali, Mutation-Based test generation for quantum programs with Multi-Objective, in: Proceedings of the Genetic and Evolutionary Computation Conference, Association for Computing Machinery, 2022, pp. 1345–1353.

[R22] X. Wang, P. Arcaini, T. Yue, S. Ali, Generating failing test suites for quantum programs with search, in: Search-Based Software Engineering: 13th International Symposium, SSBSE 2021, Bari, Italy, October 11–12, 2021, Proceedings, Springer, 2021, pp. 9–25.

[R23] R. Abreu, Fern, J. a. P. es, L. Llana, G. Tavares, Metamorphic testing of oracle quantum programs - proceedings of the 3rd international workshop on quantum software engineering (2023) 16–23. URL: https://doi.org/10.1145/3528230.3529189. doi:10.1145/3528230.3529189.

[R24] Y.-F. Chen, K.-M. Chung, O. Lengál, J.-A. Lin, W.-L. Tsai, D.-D. Yen, An automata-based framework for verification and bug hunting in quantum circuits, Proc. ACM Program. Lang. 7 (2023). URL: https://doi.org/10.1145/3591270. doi:10.1145/3591270.

[R25] G. Pontolillo, M. R. Mousavi, A multi-lingual benchmark for property-based testing of quantum programs, in: Proceedings of the 3rd International Workshop on Quantum Software Engineering, 2022, pp. 1–7.

[R26] M. Paltenghi, M. Pradel, Morphq: Metamorphic testing of the qiskit quantum computing platform - proceedings of the 45th international conference on software engineering (2023) 2413–2424. URL: https://doi.org/10.1109/ICSE48619.2023.00202. doi:10.1109/ICSE48619.2023.00202.

[R27] C. K.-U. Becker, I.-D. Gheorghe-Pop, N. Tcholtchev, A testing pipeline for quantum computing applications - 2023 ieee international conference on quantum software, qsw (2023) 54–59. doi:10.1109/QSW59989.2023.00016, 2nd IEEE International Conference on Quantum Software (IEEE QSW), Chicago, IL, JUL 02-08, 2023.

[R28] A. Kumar, Formalization of structural test cases coverage criteria for quantum software testing, INTERNATIONAL JOURNAL OF THEORETICAL PHYSICS 62 (2023). doi:10.1007/s10773-022-05271-y.

[R29] S. A. Metwalli, R. Van Meter, A tool for debugging quantum circuits - 2022 ieee international conference on quantum computing and engineering (qce 2022) (2022) 624–634. doi:10.1109/QCE53715.2022.00085, 3rd IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, SEP 18-23, 2022.

[R30] X. Wang, P. Arcaini, T. Yue, S. Ali, Qucat: A combinatorial testing tool for quantum software, Proceedings - 2023 38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023 (2023) 2066–2069. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85179004129&doi=10.1109%2fASE56229.2023.00062&partnerID=40&md5=3dbf70fd53870302f8fd2a7dea3d5e06. doi:10.1109/ASE56229.2023.00062.

[R31] L. Zhang, M. Radnejad, A. Miranskyy, Identifying flakiness in quantum programs, International Symposium on Empirical Software Engineering and Measurement (2023). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85178666869&doi=10.1109%2fESEM56168.2023.10304850&partnerID=40&md5=fab7dda40cb5f476e3cd16a8f88c839b. doi:10.1109/ESEM56168.2023.10304850.

[R32] D. Alonso, P. Sánchez, F. Sánchez-Rubio, Engineering the development of quantum programs: Application to the boolean satisfiability problem, Advances in Engineering Software 173 (2022).

[R33] R. Pérez-Castillo, L. Jiménez-Navajas, M. Piattini, Modelling quantum circuits with UML (2021).

[R34] F. Gemeinhardt, A. Garmendia, M. Wimmer, Towards Model-Driven quantum software engineering (2021).

[R35] M. Scheerer, J. Klamroth, O. Denninger, Engineering reliable hybrid quantum software: An architectural-driven approach 3008 (2021).

[R36] M. Piattini, M. Serrano, R. Perez-Castillo, G. Petersen, J. L. Hevia, Toward a quantum software engineering, IT Professional 23 (2021) 62–66.

[R37] C. A. Pérez-Delgado, H. G. Perez-Gonzalez, Towards a quantum software modeling language (2020).

[R38] L. Jiménez-Navajas, R. Pérez-Del Castillo, M. Piattinia, A tool for quantum software evolution 2705 (2020).

[R39] E. Moguel, J. Berrocal, J. García-Alonso, J. M. Murillo, A roadmap for quantum software engineering: Applying the lessons learned from the classics 2705 (2020).

[R40] L. S. Barbosa, Software engineering for 'quantum advantage', in: Proceedings of the IEEE/ACM 42nd International Conference on Software, Association for Computing Machinery, 2020, pp. 427–429.

[R41] S. Ali, T. Yue, Modeling quantum programs: Challenges, initial results, and research, in: Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures, Association for Computing Machinery, 2020, pp. 14–21.

[R42] X. Fu, J. Yu, X. Su, H. Jiang, H. Wu, F. Cheng, X. Deng, J. Zhang, L. Jin, Y. Yang, L. Xu, C. Hu, A. Huang, G. Huang, X. Qiang, M. Deng, P. Xu, W. Xu, W. Liu, Y. Zhang, Y. Deng, J. Wu, Y. Feng, Quingo: A programming framework for heterogeneous Quantum-Classical, ACM Transactions on Quantum Computing 2 (2021).

[R43] L. Jiménez-Navajas, R. Pérez-Castillo, M. Piattini, KDM to UML model transformation for quantum software modernization (2021).

[R44] I. Kumara, W.-J. Van Den Heuvel, D. A. Tamburri, QSOC: Quantum Service-Oriented computing (2021).

[R45] R. Pérez-Castillo, M. Piattini, Design of classical-quantum systems with UML, Computing 104 (2022) 2375–2403.

[R46] B. Weder, J. Barzen, F. Leymann, M. Salm, Automated quantum hardware selection for quantum workflows, Electronics 10 (2021).

[R47] J. A. Miszczak, Symbolic quantum programming for supporting applications of quantum computing technologies - companion proceedings of the 7th international conference on the art, science, and engineering of programming (2023) 101–108. URL: https://doi.org/10.1145/3594671.3594688. doi:10.1145/3594671.3594688.

[R48] S. Ali, T. Yue, On the need of quantum-oriented paradigm - proceedings of the 2nd international workshop on quantum programming for software engineering (2023) 17–20. URL: https://doi.org/10.1145/3617570.3617868. doi:10.1145/3617570.3617868.

[R49] A. A. Khan, M. A. Akbar, A. Ahmad, M. Fahmideh, M. Shameem, V. Lahtinen, M. Waseem, T. Mikkonen, Agile practices for quantum software development: Practitioners' perspectives - 2023 ieee international conference on quantum software, qsw (2023) 9–20. doi:10.1109/QSW59989.2023.00012, 2nd IEEE International Conference on Quantum Software (IEEE QSW), Chicago, IL, JUL 02-08, 2023.

[R50] R. Perez-Castillo, L. Jimenez-Navajas, M. Piattini, Dynamic analysis of quantum annealing programs?, JOURNAL OF SYSTEMS AND SOFTWARE 201 (2023). doi:10.1016/j.jss.2023.111683.

[R51] A. Cobb, J.-G. Schneider, K. Lee, Towards higher-level abstractions for quantum computing - 2022 australian computer science week (acsw 2022) (2022) 115–124. doi:10.1145/3511616.3513106, australian Computer Science Week (ACSW), ELECTR NETWORK, FEB 14-17, 2022.

[R52] J. L. Hevia, G. Peterssen, M. Piattini, QuantumPath: A quantum software development platform, Software - Practice and Experience 52 (2022) 1517–1530.

[R53] E. Moguel, J. Rojo, D. Valencia, J. Berrocal, J. Garcia-Alonso, J. M. Murillo, Quantum service-oriented computing: current landscape and challenges, Software Quality Journal (2022).

[R54] E. C. R. Da Rosa, R. De Santiago, Ket quantum programming, ACM Journal on Emerging Technologies in Computing Systems 18 (2022).

[R55] I.-D. Gheorghe-Pop, N. Tcholtchev, T. Ritter, M. Hauswirth, Computer Scientist's and Programmer's View on Quantum Algorithms: Mapping Functions' APIs and Inputs to Oracles, volume 283, 2022, pp. 188–203.

[R56] T. S. Humble, A. McCaskey, D. I. Lyakh, M. Gowrishankar, A. Frisch, T. Monz, Quantum computers for High-Performance computing, IEEE Micro 41 (2021) 15–23.

[R57] B. Sodhi, R. Kapur, Quantum computing platforms: Assessing the impact on quality attributes and SDLC activities (2021).

[R58] D. Vietz, J. Barzen, F. Leymann, B. Weder, V. Yussupov, An exploratory study on the challenges of engineering quantum applications in the cloud 3008 (2021).

[R59] T. Krüger, W. Mauerer, Quantum Annealing-Based software components: An experimental case study with SAT solving (2020).

[R60] J. L. Hevia Oliver, Requirements for quantum software platforms 2705 (2020).

[R61] C. Gomes, D. Fortunato, J. P. Fernandes, R. Abreu, Off-the-shelf components for quantum programming and testing 2705 (2020).

[R62] P. Dreher, M. Ramasami, Prototype container-based platform for extreme quantum computing algorithm development (2019).

[R63] R. LaRose, Overview and comparison of gate level quantum software platforms, Quantum 3 (2019).

[R64] M. Fingerhuth, T. Babej, P. Wittek, Open source software in quantum computing, PLoS ONE 13 (2018).

[R65] D. S. Steiger, T. Häner, M. Troyer, ProjectQ: An open source software framework for quantum computing, Quantum 2 (2018).

[R66] J. Garcia-Alonso, J. Rojo, D. Valencia, E. Moguel, J. Berrocal, J. M. Murillo, Quantum software as a service through a quantum API gateway, IEEE Internet Computing 26 (2022) 34–41.

[R67] A. Cobb, J.-G. Schneider, K. Lee, Towards Higher-Level abstractions for quantum computing, in: Australasian Computer Science Week 2022, Association for Computing Machinery, 2022, pp. 115–124.

[R68] M. A. Serrano, J. A. Cruz-Lemus, R. Pérez-Castillo, M. Piattini, Quantum software components and platforms: Overview and quality assessment, ACM Comput. Surv. (2022).

[R69] M. Scheerer, J. Klamroth, S. Garhofer, F. Knable, O. Denninger, Experiences in quantum software engineering - 2023 ieee international parallel and distributed processing symposium workshops, ipdpsw (2023) 552–559. doi:10.1109/IPDPSW59300.2023.00095, 37th IEEE International Parallel and Distributed Processing Symposium (IPDPS), St Petersburg, FL, MAY 15-19, 2023.

[R70] I. Exman, A. Nechaev, Quantum software models: Software density matrix is a perfect direct sum of module matrices (2022).

[R71] M. Schonberger, M. Franz, S. Scherzinger, W. Mauerer, Peel | pile? Cross-Framework portability of quantum software (2022).

[R72] I. Exman, A. T. Shmilovich, Quantum software models: The density matrix for classical and quantum software systems design (2021).

[R73] N. Oldfield, T. Yue, S. Ali, Investigating quantum Cause-Effect graphs (2022).

[R74] M. Weigold, J. Barzen, F. Leymann, M. Salm, Data encoding patterns for quantum computing, in: Proceedings of the 27th Conference on Pattern Languages of Programs, The Hillside Group, 2022.

[R75] F. Leymann, Towards a pattern language for quantum algorithms (2019).

[R76] M. Weigold, J. Barzen, F. Leymann, D. Vietz, Patterns for hybrid quantum algorithms (2021).

[R77] M. Weigold, J. Barzen, F. Leymann, M. Salm, Expanding data encoding patterns for quantum algorithms (2021).

[R78] P. Sánchez, D. Alonso, On the definition of quantum programming modules, Applied Sciences 11 (2021).

[R79] M. Weigold, J. Barzen, F. Leymann, M. Salm, Encoding patterns for quantum algorithms, IET Quantum Communication 2 (2021) 141–152.

[R80] F. Greiwe, T. Krueger, W. Mauerer, Effects of imperfections on quantum algorithms: A software engineering perspective - 2023 ieee international conference on quantum software, qsw (2023) 31–42. doi:10.1109/QSW59989.2023.00014, 2nd IEEE International Conference on Quantum Software (IEEE QSW), Chicago, IL, JUL 02-08, 2023.

[R81] Z. Pan, Y. Feng, Z. Li, Y. Liu, Y. Li, Understanding the impact of quantum noise on quantum programs - 2023 ieee international conference on software analysis, evolution and reengineering, saner (2023) 426–437. doi:10.1109/SANER56733.2023.00047, 30th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Macao, PEOPLES R CHINA, MAR 21-24, 2023.

[R82] J. Romero-Alvarez, J. Alvarado-Valiente, E. Moguel, J. Garcia-Alonso, J. M. Murillo, Using open api for the development of hybrid classical-quantum services - service-oriented computing - icsoc 2022 workshops 13821 (2023) 364–368. doi:10.1007/978-3-031-26507-5_34, 20th International Conference on Service-Oriented Computing (ICSOC), Sevilla, SPAIN, NOV 29-DEC 02, 2022.

[R83] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, P. Czarnik, M. El Mandouh, M. H. Gordon, Y. Hindy, A. Robertson, P. Thakre, M. Wahl, D. Samuel, R. Mistri, M. Tremblay, N. Gardner, N. T. Stemen, N. Shammah, W. J. Zeng, Mitiq: A software package for error mitigation on noisy quantum computers, Quantum 6 (2022).

[R84] J. Campos, A. Souto, QBugs: A collection of reproducible bugs in quantum algorithms and a supporting infrastructure to enable controlled quantum software testing and debugging experiments (2021).

[R85] J. Zhao, Some size and structure metrics for quantum software (2021).

[R86] J. Verduro, M. Rodríguez, M. Piattini, Software quality issues in quantum information systems 3008 (2021).

[R87] L. Saraiva, E. H. Haeusler, V. Costa, M. Kalinowski, Non-functional requirements for quantum programs 3008 (2021).

[R88] P. Zhao, J. Zhao, L. Ma, Identifying bug patterns in quantum programs (2021).

[R89] M. Paltenghi, M. Pradel, Bugs in quantum computing platforms: An empirical study, Proc. ACM Program. Lang. 6 (2022).

[R90] J. A. Cruz-Lemus, L. A. Marcelo, M. Piattini, Towards a set of metrics for quantum circuits understandability (2021).

[R91] Q. Chen, R. Câmara, J. Campos, A. Souto, I. Ahmed, The smelly eight: An empirical study on the prevalence of code smells in quantum computing - proceedings of the 45th international conference on software engineering (2023) 358–370. URL: https://doi.org/10.1109/ICSE48619.2023.00041. doi:10.1109/ICSE48619.2023.00041.

[R92] M. Openja, M. M. Morovati, L. An, F. Khomh, M. Abidi, Technical debts and faults in open-source quantum software systems: An empirical study, JOURNAL OF SYSTEMS AND SOFTWARE 193 (2022). doi:10.1016/j.jss.2022.111458.

[R93] P. Zhao, X. Wu, J. Luo, Z. Li, J. Zhao, An empirical study of bugs in quantum machine learning frameworks - 2023 ieee international conference on quantum software, qsw (2023) 68–75. doi:10.1109/QSW59989.2023.00018, 2nd IEEE International Conference on Quantum Software (IEEE QSW), Chicago, IL, JUL 02-08, 2023.

[R94] P. Zhao, Z. Miao, S. Lan, J. Zhao, Bugs4q: A benchmark of existing bugs to enable controlled testing and debugging studies for quantum programs*, JOURNAL OF SYSTEMS AND SOFTWARE 205 (2023). doi:10.1016/j.jss.2023.111805.

[R95] P. Zhao, X. Wu, Z. Li, J. Zhao, Qchecker: Detecting bugs in quantum programs via static analysis, 2023 IEEE/ACM 4th International Workshop on Quantum Software Engineering (Q-SE) (2023) 50–57. doi:10.1109/Q-SE59154.2023.00014.

[R96] M. Openja, M. M. Morovati, L. An, F. Khomh, M. Abidi, Technical debts and faults in open-source quantum software systems: An empirical study, Journal of Systems and Software 193 (2022).

[R97] R. Pérez-Castillo, L. Jiménez-Navajas, M. Piattini, QRev: migrating quantum code towards hybrid information systems, Software Quality Journal 30 (2022) 551–580.

[R98] J. Luo, P. Zhao, Z. Miao, S. Lan, J. Zhao, A comprehensive study of bug fixes in quantum programs (2022).

[R99] R. Pérez-Castillo, M. A. Serrano, M. Piattini, Software modernization to embrace quantum technology, Advances in Engineering Software 151 (2021).

[R100] A. Miranskyy, L. Zhang, J. Doliskani, Is your quantum program Bug-Free? (2020).

[R101] R. Pérez-Castillo, L. Jiménez-Navajas, M. Piattini, Reverse engineering of hamiltonian expressions from D-Wave programs (2022).

[R102] L. Jiménez-Navajas, R. Pérez-Castillo, M. Piattini, Reverse engineering of quantum programs toward KDM models (2020).

[R103] J. Zhao, On refactoring quantum programs in q#, Proceedings - 2023 IEEE International Conference on Quantum Computing and Engineering, QCE 2023 2 (2023) 169–172. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85180004597&doi=10.1109%2fQCE57702.2023.10203&partnerID=40&md5=843e78a216cb4d3d8facff89405a0dea. doi:10.1109/QCE57702.2023.10203.

[R104] M. De Stefano, F. Pecorelli, D. Di Nucci, F. Palomba, A. De Lucia, Software engineering for quantum programming: How far are we?, Journal of Systems and Software 190 (2022).

[R105] C. Hughes, D. Finke, D. German, C. Merzbacher, P. M. Vora, H. J. Lewandowski, Assessing the needs of the quantum industry, IEEE Transactions on Education (2022).

[R106] M. R. El Aoun, H. Li, F. Khomh, M. Openja, Understanding quantum software engineering challenges an empirical study on stack exchange forums and GitHub issues (2021).

[R107] G. Peterssen, Quantum technology impact: The necessary workforce for developing quantum software 2561 (2020).

[R108] L. Nallamothula, Selection of quantum computing architecture using a decision tree approach (2020).

[R109] B. Weder, J. Barzen, F. Leymann, M. Zimmermann, Hybrid quantum applications need two orchestrations in superposition: A software architecture perspective (2021).

[R110] B. Weder, J. Barzen, F. Leymann, M. Salm, D. Vietz, The quantum software lifecycle, in: Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures, Association for Computing Machinery, 2020, pp. 2–9.

[R111] P. Spoletini, Towards quantum requirements engineering - 2023 ieee 31st international requirements engineering conference workshops, rew (2023) 371–374. doi:10.1109/REW57809.2023.00072, 31st IEEE International Requirements Engineering Conference (RE), Hannover, GERMANY, SEP 04-05, 2023.

[R112] T. Yue, S. Ali, P. Arcaini, Towards quantum software requirements engineering, Proceedings - 2023 IEEE International Conference on Quantum Computing and Engineering, QCE 2023 2 (2023) 161–164. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85180012819&doi=10.1109%2fQCE57702.2023.10201&partnerID=40&md5=68dce11956f47d5601664140ad1b55a8. doi:10.1109/QCE57702.2023.10201.

[R113] J. Romero-Alvarez, J. Alvarado-Valiente, E. Moguel, J. Garcia-Alonso, J. M. Murillo, A workflow for the continuous deployment of quantum services - 2023 ieee international conference on software services engineering, sse (2023) 27–34. doi:10.1109/SSE60056.2023.00015, 1st IEEE International Conference on Software Services Engineering (IEEE SSE), Chicago, IL, JUL 02-08, 2023.