Quantum Software Engineering Issues and Challenges: Insights from Practitioners

Manuel De Stefano, Fabiano Pecorelli, Fabio Palomba, Davide Taibi, Dario Di Nucci, Andrea De Lucia

Abstract Quantum computing is an emerging field in which theoretical principles are being transformed into practical applications, largely due to the efforts of the developer community. In order to ensure that quantum software engineering continues to advance, it is vital to understand the experiences, challenges, and aspirations of developers. This chapter is a continuation of our previous work, which provided a comprehensive survey exploring the adoption patterns and common challenges in quantum software engineering. In addition to the survey, we conducted in-depth, semi-structured interviews with practitioners in the field to gain a deeper and more detailed understanding of their perspectives. Through the interviews and survey findings, we have gained nuanced insights into the motivations, hurdles, and outlook of developers toward the rapidly evolving quantum computing landscape. We describe the research methodology in detail, including the tools and techniques used, in order to provide a comprehensive understanding of the research process. Furthermore, we present critical insights from both the survey and interviews, enriching the narrative with fresh perspectives obtained from the post-publication interviews. This chapter is a blend of academic investigation and real-world practitioner insights, aiming to provide a comprehensive understanding of the current state of quantum software engineering. By illuminating the path for future research and development in this dynamic field, we hope to guide the way toward continued progress and innovation.

Manuel De Stefano, Fabio Palomba, Dario Di Nucci, Andrea De Lucia
SeSa Lab - University of Salerno, Italy

Fabiano Pecorelli
Jheronimus Academy of Data Science, Netherlands

Davide Taibi
Unversity of Oulu, Finland

1 Introduction

Quantum computing stands at the forefront of technological advancements, with developers serving as the linchpin of this revolution [7]. While the conceptual roots of quantum mechanics are deeply entrenched in theory, the tangible impacts are most discernible in the realm of quantum software engineering, where this theory translates into real-world applications [9, 10]. Thus, gauging developers' experiences and insights is paramount [13]. Our study, initiated in our foundational work [3] and further elaborated in this chapter, seeks to bridge this gap.

Our seminal work [3] embarked on this challenging quest, offering an exploratory analysis of quantum software engineering. Through a thorough survey, we dissected the prevailing state of the field, elucidating adoption strategies, recurrent challenges, and potential avenues necessitating deeper probes. The inferences drawn provided a pragmatic perspective on quantum computing, grounded in the experiences of its primary actors - the developers.

Augmenting the initial insights, this chapter extends our exploration by delving deeper into the experiences of three quantum software field practitioners through semi-structured interviews. This granular approach captures the intricacies of developers' motivations, challenges, and aspirations. Such a comprehensive examination underscores a pivotal realization: while quantum computing is intertwined with intricate physics, its real-world application is unmistakably human-centric.

With its inherent challenges and experiences, the developer community's feedback holds the potential to sculpt the trajectory of quantum technologies [3, 11]. Their shared concerns spotlight the areas needing more refined tools and frameworks, elucidate existing knowledge chasms, and chart out the path for prospective research endeavors.

This chapter embarks on a systematic journey through the terrain of quantum software engineering. An extensive review of the current literature emphasizes practitioners' trials and tribulations, with our foundational work [3] serving as a pivotal reference. This approach encompassed a dual strategy: a macroscopic view of the QSE ecosystem through software repository mining, interspersed with a nuanced, ground-level perspective sourced directly from practitioners via an expansive survey. This congruence of theoretical and practical viewpoints carved a holistic image of QSE's current state. Our approach underscored the indispensability of aligning academic exploration with tangible, on-ground experiences, bridging a crucial literature gap.

Our research quest was anchored in discerning the real-world applications of quantum programming technologies, pinpointing quantum developers' challenges, and assessing software engineering techniques' relevance and applicability. This encompassed insights from researchers, practitioners, and tool vendors, each striving to decode the intricate dance between software engineering and quantum programming.

Our exploration was pivoted on two central research queries:

1. *How and to what purpose are quantum programming frameworks predominantly utilized?*

2. *What predominant hurdles do quantum developers encounter when interfacing with quantum frameworks?*

Our investigation was guided by a series of fundamental questions that aimed to encapsulate the nuances of adopting quantum programming. Our goal was to achieve a deep understanding of these subtleties, as they play a crucial role in uncovering the primary challenges that quantum programming developers face. By shedding light on these challenges, we hoped to empower tool creators and academic researchers to craft innovative strategies that can help overcome these obstacles.

In this book chapter, we take a two-pronged approach to our investigation. We first explore the current literature on quantum programming adoption, seeking to identify trends, patterns, and gaps in existing research. We then provide a comprehensive analysis of both the mining study and the comprehensive survey we conducted, which allowed us to gather detailed insights directly from developers.

Our analysis of the mining study and the comprehensive survey builds on this foundation, providing a more granular view of the challenges that developers face. We examine issues such as debugging, testing, and community issues, as well as the challenges of working with quantum hardware and the need for better documentation and education. By exploring these issues in detail, we aim to provide actionable insights that can help guide the development of new tools and strategies for quantum programming.

2 Bridging the Gap in Quantum Software Engineering

Quantum Software Engineering (QSE) has burgeoned as a pivotal discipline within the quantum computing domain, with the *Talavera Manifesto* marking a significant milestone in its evolution [11, 12, 13]. This manifesto delineated core tenets and principles, laying a roadmap for researchers and developers. However, it inadvertently overlooked the practical challenges practitioners face at the quantum software development forefront.

A recent systematic mapping study by De Stefano et al. [4] delved into the current state of QSE research, aiming to outline the most investigated topics, the types and number of studies, and the primary reported results alongside the most studied quantum computing tools/frameworks. This study also aimed to gauge the research community's interest in QSE, its evolution, and any notable contributions preceding the formal introduction through the Talavera Manifesto.

Employing a meticulous methodology, De Stefano et al. searched for relevant articles across various databases, applying inclusion and exclusion criteria to select the most pertinent studies. Following a quality evaluation of the selected resources, relevant data were extracted and analyzed. The findings underscored a predominant focus on software testing within QSE research, with other crucial topics like software engineering management receiving scant attention. Among the technologies for techniques and tools, Qiskit emerged as the most commonly studied, although many studies either employed multiple technologies or did not specify any. The research community interested in QSE showcased interconnected collaborations, with several strong collaboration clusters identified. Interestingly, most QSE articles were published in non-thematic venues, with a preference for conferences, indicating a burgeoning interest in the domain.

The implications of this study are manifold, serving as a centralized information source for researchers and practitioners, facilitating knowledge transfer, and contributing significantly to QSE's advancement and growth. The study highlighted the nascent stage of QSE research, primarily centered around software testing, leaving other knowledge areas like software engineering management relatively unexplored. A notable uptick in published papers between 2020 and 2021 reflects a growing interest in QSE within the research community. The study also shed light on the most productive authors, the main collaboration clusters, and the distribution of researchers across different Software Engineering (SE) topics, which could catalyze the identification of potential collaborators and foster further research in QSE.

Furthermore, the study accentuated the need for more empirical studies and a better distribution of research efforts across diverse SE topics. It advocated for a broader acceptance of QSE papers in non-thematic publication venues to expand the research community's knowledge and reach. The insights gleaned from this study are instrumental in understanding the development and evolution of the research community, thereby significantly contributing to the advancement and growth of QSE.

The systematic mapping study also illuminated potential avenues for future research in QSE, particularly in the overlooked realms of software engineering management practices and quantum software maintenance. The call to action is for future research to focus on devising effective strategies and tools for managing the software development process and ensuring the reliability and performance of quantum software over time. The unique challenges and opportunities inherent in quantum software engineering warrant a thorough exploration to identify effective strategies for managing the development process and evaluating the efficacy of different software engineering practices and tools.

Despite the academic rigor marking the journey of QSE, the crucial component of practitioners' voices and experiences has often been overlooked. The academic literature has largely remained aloof from the day-to-day challenges and innovative solutions that practitioners often develop. While several studies have delved deep into the theoretical challenges and potential solutions in QSE from a high-level perspective [16, 11], they missed out on the granular details and real-world manifestations of these challenges.

Among the myriad studies in this domain, the work by El Aoun et al. stands out for its empirical approach [6]. By analyzing QSE-related discussions on platforms like Stack Exchange and GitHub, they tapped into a rich vein of practitioner experiences. Their methodology employed automated topic modeling to distill the myriad discussions into coherent themes and challenges, providing a window into the world of quantum developers. However, the limitations of automated topic modeling sometimes missed out on the nuances and subtleties of human communication, and their passive approach did not allow for deeper, engaging discussions with practitioners.

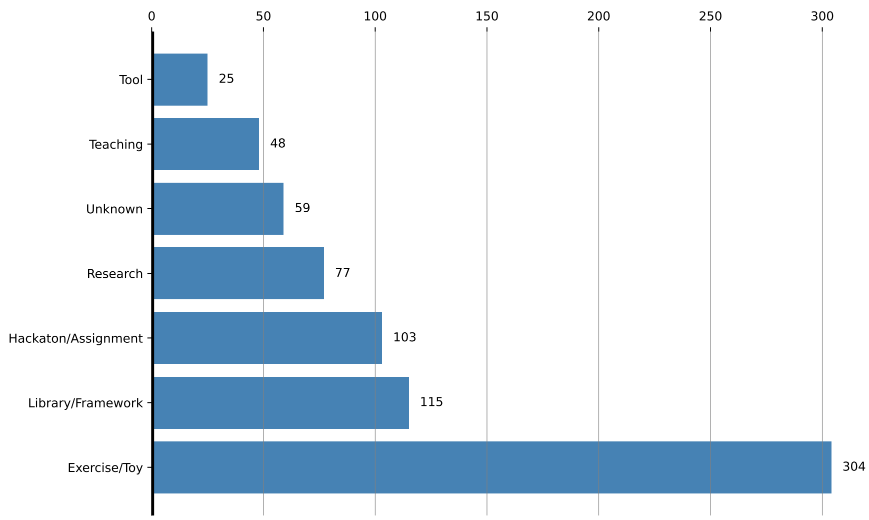

Fig. 1 Main task for which quantum repositories are created [3]

3 Current Usage of Quantum Technologies

The mining study [3] focused on identifying and analyzing quantum software repositories to understand the extent and purpose of quantum programming frameworks usage. Our study's scope was primarily defined by the quantum technologies considered. We focused on three state-of-the-practice universal gate quantum programming technologies, namely QISKIT [2], CIRQ [5], and Q♯ [1], which are developed and maintained by IBM, Google, and Microsoft respectively. These frameworks are recognized as more mature and stable, each having unique functionalities and allowing the execution of quantum programs on both local simulators and real quantum devices provided by their vendors. We employed a software repository mining approach to identify projects on GITHUB that use at least one of the considered technologies. This process yielded a total of 731 unique repositories.

The data analysis phase for the mining study aimed at addressing the first research question using information from the repository mining. We employed Straussian Grounded Theory for a systematic approach to constructing theories from the data collected. This methodology involved a cyclical process of open, axial, and selective coding to derive a taxonomy that serves as the foundation for answering our research question.

We provided a data-driven perspective on how quantum technologies are employed in real-world scenarios.

As seen by Figure 1, the mining revealed distinct usage patterns among quantum developers. Many repositories were dedicated to didactic purposes or personal experimentation with quantum technologies. This suggests that many developers are

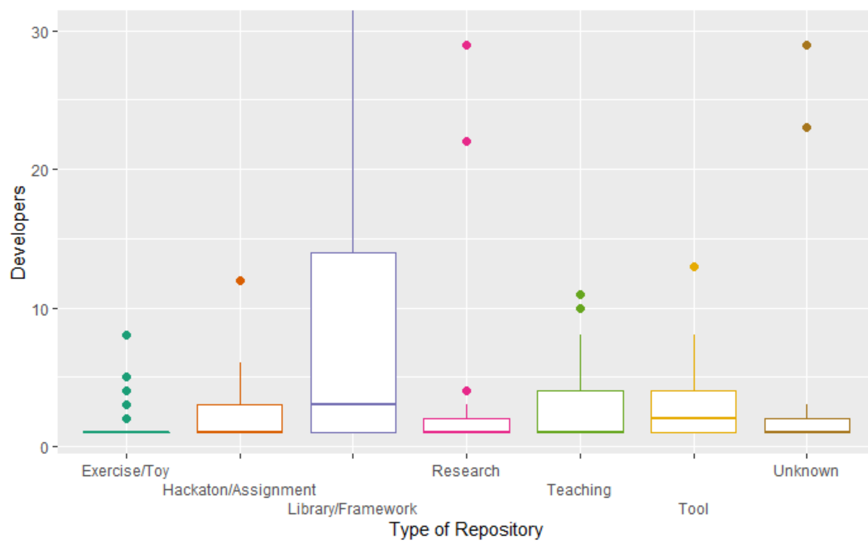

Fig. 2 Distribution of contributors per type of repository [3]

in the early stages of their quantum journey, using repositories as learning tools or platforms for experimentation.

An interesting facet of the mining study was the analysis of contributors to these repositories (Figure 2). The distribution of contributors varied based on the type of repository. For instance, toy projects, which are typically smaller and more experimental, had a distribution skewed towards fewer contributors. In contrast, framework-related repositories, which are more extensive and foundational, had a broader distribution of contributors.

The mining study's results underscore the developing nature of quantum programming. While there is evident enthusiasm and interest in the field, as seen by the proliferation of didactic and experimental repositories, large-scale, collaborative projects still have a long way to go. The contributor analysis further reinforces this, highlighting the need for more collaborative platforms and community-driven initiatives to foster growth in quantum software engineering.

4 The Practitioners' Voice

The survey study aimed to gather insights from quantum developers regarding their challenges and perspectives on the current and potential future adoption of quantum programming technologies.

For engaging with quantum software developers, we utilized the mined repositories to obtain a list of eligible candidates for our survey, ensuring the involvement of developers with real experience in quantum programming. We employed an *opt-in* strategy for recruitment, sending initial emails to gauge interest before providing additional instructions to willing participants. This strategy led to the recruitment of 56 *volunteers*.

Table 1 Questions asked in the survey

Question Text	Answer Type	Possible Answers
Part 1 - Background		
What is your current employment status?	Multiple Choice	B.Sc. Student; M.Sc. Student; Ph.D. Student; Researcher; Open Source Developer; Industrial Developer; Other
What is your educational background?	Single Choice	Computer Science; Chemistry; Physics; Other
What is your age range?	Single Choice	18-24; 25-34; 35-44; 45-54; 55+
What is your gender?	Free Text	-
Please indicate your expertise (in years) in Software Development.	Single Choice	None; 0-3; 3-5; 5-10; 10+
Please indicate your expertise (in years) in Industrial Development.	Single Choice	None; 0-3; 3-5; 5-10; 10+
Please indicate your expertise (in years) in Quantum Programming.	Single Choice	None; 0-3; 3-5; 5-10; 10+
What is your Country?	Free Text	-
Part 2 - Current Adoption		
Which quantum technology are you most confident with?	Single Choice	QISKIT; CIRQ; Q♯; Other
Which other quantum technology do you use?	Multiple Choice	QISKIT; CIRQ; Q♯; Other
In which context are you using quantum computing?	Multiple Choice	Academic Study; Hackaton; Industry; OSS; Personal Study; Research; Other
Could you please tell me more about the tasks you perform with quantum computing?	Long Free Text	-
Part 3 - Potential Adoption and Challenges		
Consider the technology you are most confident with. What were the top 3 challenges that you have faced?	Multiple Free Text	-
Based on your experience, have you ever solved (or tried to solve) a problem using quantum programming that has no "traditional" solution (or the solution is intractable)?	Single Choice	Yes; No
If yes, could you please elaborate on the problem and why you have to use quantum computing?	Long Free Text	-
Based on your experience, have you ever solved (or tried to solve) a problem that has a "traditional" solution using quantum programming?	Single Choice	Yes; No
If yes, could you please elaborate on what it was and explain why you chose to use quantum computing?	Long Free Text	-

The survey was structured into three main sections: gathering background information, understanding the current use of quantum technologies, and assessing their longer-term adoption and challenges.

The data analysis phase for the survey study aimed at addressing our second question by using the responses provided in the third part of the survey. Similar to the mining study, we employed Straussian Grounded Theory for a systematic approach to constructing theories from the data collected. This methodology involved a cyclical process of open, axial, and selective coding to derive a taxonomy that serves as the foundation for answering our research question.

Based on the practitioner feedback, we built a detailed taxonomy of their main challenges while working with quantum computing. This taxonomy, represented in Figure 3, was developed through a rigorous Straussian Grounded Theory exercise. Some challenges are independent, while others lead to more specific sub-challenges.

4.1 The Quantum Environment: Hardware and Software

The quantum environment, encompassing both hardware and software, presents its own set of unique challenges. Software infrastructure issues can be related to frameworks, integration, and execution.

- **Framework.** Developers often grapple with the ever-changing API designs of quantum technologies. A significant number of our interviewees, 15 to be precise, lamented the frequent and unpredictable changes in API. Others highlighted the lack of support for certain operations, like those in Qiskit, and the absence of standardization across frameworks.
- **Integration.** Integrating quantum systems with traditional ones is no easy feat. Some developers mentioned the complexities of integrating classical algorithms into their quantum counterparts or connecting quantum computers to blockchain networks.
- **Execution.** Setting up execution environments, simulators, or interfacing with classical systems can be daunting, as 11 of our participants reported.

Hardware infrastructure issues attain the developmental nature and availability of the hardware and the related performance.

- **Hardware.** The specialized nature of quantum hardware, which is still in its developmental phase, poses challenges. Developers often find themselves restricted by the limited number of qubits available in quantum computers.
- **Performance.** Emulating quantum programs on classical computers brings forth several performance issues. Emulators can be resource-intensive, and running programs on actual quantum devices can be time-consuming due to vendor-imposed job queues.

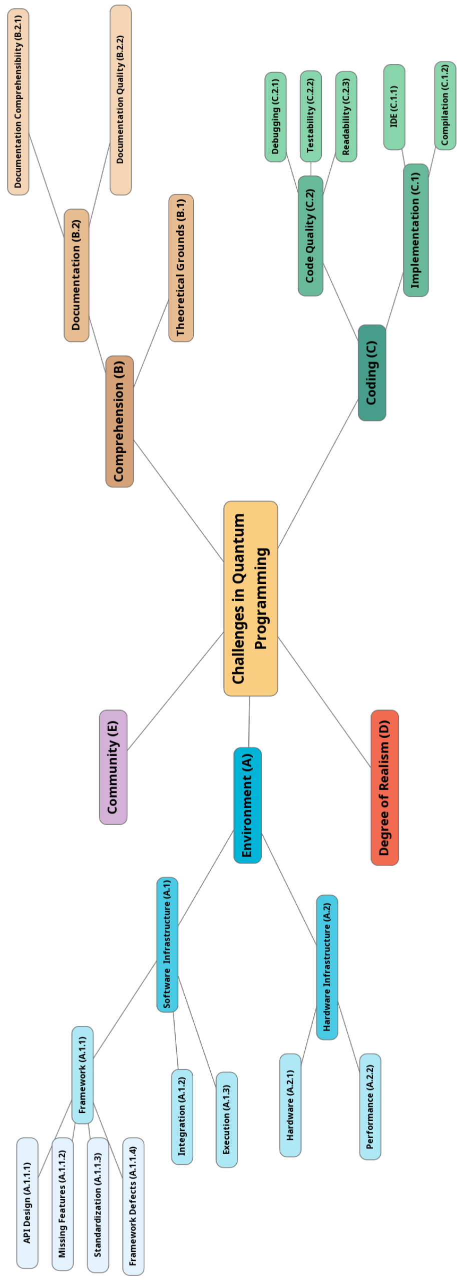

Fig. 3 Taxonomy of quantum programming challenges identified by practitioners [3]

4.2 Comprehending the Quantum Realm

Understanding quantum programs is a challenge in itself.

- **Theoretical Grounds.** A significant number of our respondents (20) emphasized the steep learning curve associated with quantum programming, especially the need for a strong foundation in linear algebra.
- **Documentation.** These issues concern the comprehensibility and quality of the documentation related to quantum frameworks and code.
  - **Comprehensibility.** Inconsistent tutorials and documentation can hinder the learning process, a sentiment echoed by three participants.
  - **Quality.** Sixteen participants highlighted issues with outdated, incomplete, or missing documentation.

4.3 Quantum Coding Challenges

Coding in the quantum domain presents its own set of unique challenges related to implementation and code quality.

- **Implementation.** These issues are related to integrated development environments and compilation.
  - **IDE.** A good Integrated Development Environment (IDE) can be a game-changer. However, some developers found existing quantum IDEs lacking, especially when working with environments like Q♯.
  - **Compilation.** Translating quantum circuits into executable code for quantum computers is a complex process, with developers often struggling to adapt ideal quantum circuits to available device architectures.
- **Code Quality.** Code quality issues include problems related to debugging, testing, and readability of quantum code.
  - **Debugging.** Deciphering error messages and debugging quantum programs can be particularly challenging due to the unique nature of quantum programming.
  - **Testability.** Ensuring that a quantum program functions as intended is not straightforward. Some developers found it challenging to verify the correctness of their circuits.
  - **Readability.** With quantum code primarily defining qubit registers and applying gates, creating readable code becomes challenging.

4.4 The Realism Quotient

While quantum computers promise groundbreaking solutions, their practical application remains a challenge.

- **Degree of Realism.** Developers often find it challenging to design quantum programs that can address real-world problems. The limitations of current quantum applications make it difficult to find problems that quantum solutions can address better than traditional technologies.

4.5 Building a Quantum Community

The early stage of quantum programming means a small community of developers.

- **Community.** Many developers desired a more robust community for peer support and collaboration. Slow code reviews and the effort required to understand quantum programs further compound the challenges.

5 Deepening the Practictioners' Insights

This book chapter comprehensively analyzes the challenges of using quantum software development technologies in various fields. To achieve this, we took a unique approach of directly interviewing experts specializing in quantum technologies. These experts come from diverse backgrounds and working ecosystems.

During our interviews, we discussed practitioners' particular difficulties when working with quantum technologies. We covered a range of topics from the taxonomy of challenges we established [3], such as the shortcomings of existing quantum technologies, the requisite for more sophisticated hardware and software, and the struggles in creating quantum algorithms.

The insights provided by these experts were priceless, as they offered a wealth of knowledge and experience from their respective fields. We present the extracts of each interview in the following sections, accompanied by detailed analysis and commentary on the valuable insights provided by these practitioners. By doing so, we hope to shed more light on the challenges of working with quantum technologies and provide a better understanding of the field.

5.1 First Interview

In the first of our series of interviews, we engaged with an expert deeply involved in the practical application of quantum mechanics. This individual, currently affiliated with a consultancy company, has been actively working on the lattice Boltzmann method for quantum computing. Their experience primarily revolves around using platforms like Qiskit and Penny Lane, and they are keenly interested in the challenges and opportunities of quantum hardware and software. Throughout our discussion, they shared insights on the evolution of quantum computing, its challenges, and the real-world implications of integrating quantum solutions with classical systems.

Our first expert acknowledged the ever-evolving nature of APIs in quantum computing, emphasizing their appreciation for the consistent updates. They lauded the community's supportive character, particularly around a specific platform, which they described as responsive and invaluable.

Much of the discussion revolved around the practical application of quantum programming. While some in the field find it challenging to harness quantum programming for tangible tasks, this expert has successfully navigated these waters in their projects.

When discussing software infrastructure, the expert highlighted the seamless integration of quantum software with traditional software, primarily through Python. However, they also brought to light a theoretical challenge: the intricate process of mapping classical input to quantum.

The conversation then veered towards hardware challenges. The inherent noise in quantum mechanics was identified as a natural obstacle in quantum computing. Despite this, the expert recognized the commendable progress made in recent years, especially by a leading tech company, in mitigating this noise. But they also pointed out that this noise currently restricts the depth of circuits on real devices, leading them to use simulators often.

On the coding front in quantum computing, the expert found the process straightforward, especially with the support for various gates in specific platforms. They also shared their unique approach to testing, which involves juxtaposing quantum results with classical methods or analytical solutions.

The expert addressed the prevailing hype around quantum computing and the inherent challenge of achieving a quantum advantage. They proudly mentioned their company's significant strides, especially with specific models.

5.2 Second Interview

The individual shared their experiences and insights into quantum computing during the interview. They emphasized the importance of grasping the fundamentals before delving into more complex frameworks and languages. They preferred a bottom-up approach, utilizing more straightforward tools like NumPy to build a foundational understanding. They believed this mitigated the challenges posed by the steep learning curve associated with quantum computing, especially when compared to the more mature field of machine learning.

The interviewee found that existing frameworks like Qiskit were more geared towards professional deployment rather than aiding in learning or debugging. They mentioned the difficulty in debugging in quantum computing, attributing it to the complexity added by these frameworks. They advocated a more straightforward approach to coding and debugging to understand and learn quantum computing.

Drawing parallels between the evolution of machine learning and quantum computing, the interviewee noted hype and venture capital involvement similarities. They observed that while machine learning has found broad applications and has become integral to many fields, quantum computing might find its niche in more specific areas like chemistry and physics, particularly in quantum simulation. They believed this could lead to significant advancements, such as discovering new pharmaceuticals.

The interviewee also touched on the potential for quantum computing to become a significant part of data center infrastructure in the future. However, they expressed skepticism regarding the timeline for such developments, likening the anticipation around quantum computing to the long-standing expectation around fusion energy.

Exploring various quantum computing languages, the interviewee found that understanding the underlying mechanics was crucial for making sense of these languages. They mentioned having examined various quantum computing languages and found that having a foundational understanding aided in making sense of how these languages and compilers were implemented.

Looking towards the future, the interviewee foresees a potential hype cycle for quantum computing, similar to what machine learning experienced. They anticipated initial excitement, followed by a period of disillusionment and, eventually, the emergence of practical applications as the field matures. They stressed the need for real impact or significant advancements in quantum computing to sustain momentum in the area, expressing cautious optimism for the potential of quantum computing to contribute to specific scientific and technological advances.

5.3 Third Interview

The interviewee is a professional in the field of quantum computing with several years of industry experience. They specialize in the interface aspect of quantum computing projects and have actively created educational resources for the community. Their work primarily focuses on software engineering within quantum computing, and they have a dedicated team to ensure code quality.

During the discussion, the interviewee highlighted several challenges and considerations in the quantum computing field. One primary concern is the lack of adequate documentation, which can be a barrier for newcomers or those not deeply versed in the science behind quantum computing. They emphasize the importance of understanding the input and output of specific modules in quantum computing projects, which is crucial for effective implementation and debugging.

The conversation also touches on the accessibility of the quantum computing community. The interviewee believes that while the community is robust, it may not be well-advertised or easy to find for newcomers. They suggest better communication and information distribution could help bridge this gap, making the community more accessible to those interested in quantum computing.

Regarding realism and expectations, the interviewee acknowledges that the field is not yet usable, and much of the work is about paving the way for future usability. They compare the hype around quantum computing to machine learning in its early stages, indicating that the field might face similar challenges in meeting high expectations.

Regarding code quality and debugging, the interviewee mentions the challenges when code is written by scientists who might not have strong coding practices. They highlight the difference in code quality when a dedicated software engineering team is involved versus when researchers or academics write the code.

Regarding the accessibility to quantum computers, the interviewee mentions that they have never actually run anything on a real quantum computer but have used simulators instead. They speculate about the future accessibility of quantum computers to the public, comparing it to the current accessibility of supercomputers.

The interviewee also mentions creating educational resources, such as blog posts and podcasts, to help others in the quantum computing community. They are willing to share these resources, indicating a collaborative spirit within the community.

Lastly, the interviewer expresses interest in the open-source software the interviewee is working on, indicating a willingness to share and collaborate within the community, further emphasizing the collaborative nature of quantum computing.

6 Synthesizing Academic Findings and Practical Insights

Exploring the Quantum Software Engineering landscape involved conducting a mining study, a practitioner survey, and expert interviews. The mining study involved analyzing vast amounts of data to identify patterns and trends in quantum programming. The practitioner survey was conducted to gather feedback from professionals working in the quantum programming domain. Lastly, expert interviews were conducted to gain insights from knowledgeable individuals well-versed in the field.

The findings from these diverse sources provide a rich tapestry of insights into the current state and challenges of the quantum programming domain. These insights include the tools and technologies currently being used, the challenges practitioners face, and the potential for future advancements.

This section synthesizes these findings to understand the quantum programming domain fully. By doing so, we hope to provide a comprehensive overview of the current state of the field as well as its potential for future growth and development.

Early Adoption and Experiments The study on quantum programming revealed that the field is still in its early stages, with many repositories dedicated to didactic purposes or personal experimentation. This early experimentation is further supported by the first interviewee, who discussed the challenges in practical application, and the second interviewee, who emphasized the need for a solid foundation before tackling complex frameworks. Moreover, the survey results of 20 participants, along with the second interviewee's comments, highlighted the steep learning curve in the current quantum programming landscape, which further underlines the experimental nature of this field. The findings indicate that, at this stage, quantum programming is primarily used for educational and exploratory purposes, with few practical applications. However, with further research and development, quantum programming could have significant implications for various industries.

Community and Collaborative Initiatives. The importance of building a stronger and more collaborative community platform was a recurring theme that emerged from all sources. The mining study revealed a skewed distribution of project contributors towards a limited number of toy projects, denoting a lack of collaboration on a larger scale. This finding underscores the need for better community-building and collaboration platforms. Practitioners also expressed their desire for a more robust community to provide peer support, facilitate collaboration, and improve communication within the quantum computing field. The third interviewee emphasized the importance of information distribution and better communication within the quantum computing community. The first interviewee appreciated the supportive nature of the community around specific platforms, which further highlights the significance of a collaborative ecosystem in advancing the field. In summary, building a collaborative and supportive community is crucial for advancing the quantum computing field, and there is a need for better platforms to facilitate communication, collaboration, and information sharing.

Software and Hardware Infrastructure Challenges The survey and interviews with practitioners in the quantum computing field have brought to light several key challenges in the software and hardware infrastructure. A prominent concern among practitioners is the frequent and unpredictable changes in quantum computing APIs, as highlighted by 15 survey participants. This issue complicates the process of keeping software up-to-date with the latest developments. Another significant challenge is the integration of quantum systems with traditional computing systems. This was particularly emphasized by the first interviewee, who stressed the importance of seamless integration of quantum software with conventional software. For effective integration, software developers must comprehensively understand both quantum and traditional software systems, along with their distinct characteristics. The hardware challenges in quantum computing were also addressed, particularly the limited availability of qubits and the inherent noise in quantum mechanics. These factors restrict the complexity of calculations that can be performed and limit the depth of circuits on real quantum devices, posing a hurdle for practitioners undertaking complex quantum computations. A possible future direction for the hardware support might be seen in codesign [14]. Codesign has been a foundational element in the evolution of computer architecture since the inception of the first systems. This concept, where end-user applications influence the design and capabilities of the hardware and vice versa, is crucial in quantum computing (QC). Especially in its resource-constrained early stages, QC heavily relies on codesign strategies. This approach involves tailoring the quantum hardware and software to optimize performance and functionality. The article explores the significance of codesign in the QC context, illustrating its benefits and proposing essential attributes for effective QC codesign strategies moving forward. This perspective suggests a future direction where addressing the current challenges in quantum computing infrastructure could involve a more integrated and co-evolutionary approach between software and hardware, aligning with the principles of quantum codesign.

Real-world Application and Quantum Advantage. The concept of realism quotient, explored in a survey conducted among practitioners, seems to align with expert opinions on the practical implementation of quantum programming. Introducing the concept of 'quantum utility,' which measures the effectiveness and practicality of quantum computers in various applications, provides a more holistic view of the field's progress. This new metric, focusing on achieving a quantum advantage in terms of speed, accuracy, or energy efficiency compared to classical machines of similar size, weight, and cost, enhances the realism quotient by considering the physical footprint and industrial value of quantum processors [8, 15]. The first interviewee's insights on achieving quantum advantage echo the goals set forth in the quantum utility concept [8, 15] and the second interviewee's doubts regarding the timeline for quantum computing to become a significant part of data center infrastructure indicate that the field is still struggling to establish a strong foothold in real-world applications. Moreover, the proposed application readiness levels (ARLs) and extended classification labels further refine the criteria for assessing quantum computing's practical applications in fields like quantum chemistry and machine learning [8, 15]. The second interviewee's expectation of a hype cycle similar to that experienced by the machine learning industry reflects a cautious optimism toward the potential of quantum computing to contribute to specific scientific and technological advancements. Overall, the survey and interviews highlight the ongoing challenges and possibilities that quantum programming presents for the future of computing and underscore the importance of structured analysis and tooling, as emphasized in the concept of Quantum Computing Optimization Middleware (QCOM)[8, 15].

Educational Resources and Code Quality. During the third interview, the interviewee's efforts in creating educational resources and ensuring code quality were discussed at length. It was noted that these efforts resonate with the concerns of practitioners in the quantum programming community regarding the comprehensibility and quality of documentation. Interestingly, when written by scientists, as opposed to a dedicated software engineering team, the mention of challenges in code quality reflects a broader concern in the community regarding the accessibility and readability of code. This highlights the need for collaborations between scientists and software engineering teams in the quantum programming community to ensure the development of high-quality, understandable, and readable code.

Future Usability and Accessibility. During the third interview, the interviewee expressed their opinion on the field's current state, highlighting its unusability and speculating on the future accessibility of quantum computers to the public. This encapsulates the overall sentiment of cautious optimism prevalent in the field. While the field is full of potential and possibilities, it faces substantial challenges that need addressing before it can transition from a stage of experimentation to one of significant real-world impact. These challenges include infrastructure, community collaboration, and real-world application. Addressing these challenges is crucial for the field to realize its potential and significantly impact the real world.

7 Conclusion and Future Directions

The field of quantum software engineering is still in its nascent stages, facing a range of potential and significant obstacles that pose challenges to the development and application of quantum computing. However, combining the results of a comprehensive mining study, practitioner survey, and expert interviews provides a detailed and thorough understanding of the current state of the field, as well as its trajectory.

A closer look at the opportunities and challenges identified in this study reveals a range of factors shaping the field of quantum software engineering. For instance, there is great enthusiasm and interest in quantum programming, with abundant educational and experimental repositories indicating a fertile ground for innovation. The potential applications of quantum computing, especially in fields like chemistry, physics, and cryptography, are promising, and this has led to a growing community of developers and researchers eager to explore and contribute to this emerging field.

However, many challenges must be overcome before the full potential of quantum software engineering can be realized. These challenges include a steep learning curve, a lack of standardized frameworks, hardware limitations, and a nascent stage of community collaboration. Developers face significant hurdles in integrating quantum systems with traditional ones, frequent API changes, and complexities. The lack of large-scale collaborative projects and robust community support further exacerbates the challenges in advancing quantum software engineering. Moreover, the struggle in harnessing quantum programming for tangible real-world tasks remains a significant concern.

Addressing the identified challenges requires a concerted effort from academia, industry, and the quantum computing community. Standardizing frameworks, improving documentation quality, and fostering a collaborative ecosystem are essential for nurturing the growth of quantum software engineering. Investments in educational initiatives to lower the entry barrier and nurture a new generation of quantum programmers are crucial. Creating platforms facilitating large-scale collaborative projects can accelerate the transition from experimentation to substantial real-world impact.

Moreover, continued research and development are vital in overcoming hardware limitations and enhancing the software infrastructure. Establishing partnerships between academia and industry can expedite the translation of academic findings into practical solutions, driving the field closer to achieving quantum advantage. In addition, managing the hype around quantum computing and setting realistic expectations can help navigate the hype cycle, ensuring sustained momentum in the field. The cautious optimism expressed by the interviewees and survey participants reflects a collective acknowledgment of the long yet promising journey ahead.

In conclusion, the quantum software engineering landscape presents a frontier of opportunities awaiting exploration and innovation. The insights garnered from the current state of the field provide a compass for navigating the uncharted waters of quantum software engineering, steering toward a future where quantum computing realizes its transformative potential.

Acknowledgements

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 by the EU - NGEU.

References

1. Quantum development kit. https://azure.microsoft.com/it-it/resources/development-kit/quantum-computing/ (2021). Accessed: 2021-06-05
2. Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernández, F.J., Carballo-Franquis, J., Chen, A., Chen, C.F., et al.: Qiskit: An open-source framework for quantum computing. Accessed on: Mar **16** (2019)
3. De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., De Lucia, A.: Software engineering for quantum programming: How far are we? Journal of Systems and Software **190**, 111326 (2022)
4. De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., De Lucia, A.: The quantum frontier of software engineering:\\a systematic mapping study. arXiv preprint arXiv:2305.19683 (2023)
5. Developers, C.: Cirq (2021). DOI 10.5281/zenodo.4750446. URL https://doi.org/10.5281/zenodo.4750446. See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/contributors
6. El aoun, M.R., Li, H., Khomh, F., Openja, M.: Understanding quantum software engineering challenges: An empirical study on stack exchange forums and github issues. In: 37th International Conference on Software Maintenance and Evolution (ICSME) (2021)
7. Galitski, V.: Quantum computing hype is bad for science. https://www.linkedin.com/pulse/quantum-computing-hype-bad-science-victor-galitski-1c. Accessed: 2021-07-21
8. Herrmann, N., Arya, D., Doherty, M.W., Mingare, A., Pillay, J.C., Preis, F., Prestel, S.: Quantum utility–definition and assessment of a practical quantum advantage. arXiv preprint arXiv:2303.02138 (2023)
9. Hoare, T., Milner, R.: Grand challenges for computing research. The Computer Journal **48**(1), 49–52 (2005)
10. Knight, W.: Serious quantum computers are finally here. what are we going to do with them. MIT Technology Review. Retrieved on October **30**, 2018 (2018)
11. Piattini, M., Peterssen, G., Pérez-Castillo, R.: Quantum computing: A new software engineering golden age. ACM SIGSOFT Software Engineering Notes **45**(3), 12–14 (2020)
12. Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J.L., Serrano, M.A., Hernández, G., de Guzmán, I.G.R., Paradela, C.A., Polo, M., Murina, E., et al.: The talavera manifesto for quantum software engineering and programming. In: QANSWER, pp. 1–5 (2020)
13. Piattini, M., Serrano, M., Perez-Castillo, R., Petersen, G., Hevia, J.L.: Toward a quantum software engineering. IT Professional **23**(1), 62–66 (2021)
14. Tomesh, T., Martonosi, M.: Quantum codesign. IEEE Micro **41**(5), 33–40 (2021)
15. Tsymbalista, M., Maksymenko, M., Katernyak, I.: Approaching quantum utility by leveraging quantum software stack. In: 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), pp. 210–215. IEEE (2023)
16. Zhao, J.: Quantum software engineering: Landscapes and horizons. arXiv preprint arXiv:2007.07047 (2020)