

# An Exploratory Survey on the Synergy Between Agile Development and Quantum Computing in Engineering Applications

Abid Hussain  
Professor

School of Computer Application & Technology & Dean, Research  
Career Point University

Kota

abid.hussain@cpur.edu.in, dean.research@cpur.edu.in

**Abstract**—Agile development and quantum computing still are very distant entities that have not yet come together, but their convergence, should it be utilized correctly, can deliver not only the fruits of interaction but also the rewards of innovation, performance, and computing capabilities. Scrum, Kanban, and XP are examples of agile methods that emphasize team development iteratively through collaboration and continuous improvement. This helps teams handle complex and dynamic projects. The quantum computing system finds these criteria to be absolutely perfect, as the whole field revolves around exploring and calculating the likelihood of superposition, entanglement, and quantum interference, which are physical phenomena that are foreign to the micro-world. Qiskit, Cirq, and cloud services like IBM Quantum Experience are only a few of the tools that offer various platforms for creators of quantum algorithms, as quantum technology advances. Agile, combined with quantum software engineering, can not only be disaggregated into smaller pieces that can be more easily handled by available staff, but also raise issues such as insufficiently skilled staff, inefficient testing, and hardware constraints. In recap, the two dimensions of up-to-scale and versatile engineering practices are united by this synergy, making them predictable and future-ready.

**Keywords**—Quantum Software Engineering, Agile Methodologies, Quantum Computing, Software Development, Iterative Development, Quantum Programming.

## I. INTRODUCTION

Agile software development has gained significant traction over the last 20 years, becoming as one of the most prominent software paradigms of their time. The goal of adopting an agile development methodology is to make a company more flexible and responsive. Agile principles recognize that consistent software delivery is a key component of agile development [1]. Iteratively improving software is a commonality across all of these methodologies that adhere to these standards. "The software working is the main measure of progress," goes the thinking behind one theory. Communication with clients at all times is another tenet often mentioned in this regard. With today's software development teams, agile approaches are chosen by priority [2]. One example is Scrum, an agile technique that, in contrast to more traditional approaches to software development, relies less on documentation artefacts [3]. Iconix, in contrast, is a semi-agile approach that minimizes artefacts while retaining critical ones, such as the robustness diagram. Enhancing software quality and flexibility to meet evolving customer expectations are the primary objectives of XP. Lastly, Crystal is meant to be an agile or lightweight approach that prioritizes individuals and their ability to communicate.

Quantum computers take advantage of quantum effects to perform calculations that are faster, more efficient, or even impossible to do on regular computers. Quantum computing is not a universal remedy for all problems [4]. The efficiency of quantum computation has been demonstrated; issues that would take an aeon for a conventional computer to solve may be resolved in days by a quantum computer [5]. Nevertheless,

for some other problems, it has been established that quantum computation cannot surpass classical methods, and for another group, the gain is marginal.

The term "Quantum Software Engineering" (QSE) describes the process of applying sound engineering principles to create, maintain, and run quantum programs. One of the biggest problems with QSE is that it requires learning and applying traditional software engineering ideas to the quantum setting, which isn't easy given the specific challenges of quantum computing [6]. The key element in this challenge is the forecasted hybrid quantum software model which will integrate the two worlds of classical and quantum computing [7]. This combined system demands the development of technologies that enable easy communication among the various computing structures. Application of the principles of Service-Oriented Computing (SOC) to the development of interoperable interfaces will be of paramount importance, as will the application of Service Engineering tools to the effective design and monitoring of quantum services.

### A. Structure of the paper

The following is the outline of the paper: Key frameworks and Agile development approaches are described in Section II. The fundamental ideas and methods of quantum computing are presented in Section III. In Section IV, take a look at how engineering firms are using Agile practices with quantum computing. Section V provides a summary of relevant studies, while Section VI offers recommendations for further study.

## II. FUNDAMENTALS OF AGILE DEVELOPMENT

Agile methods are a collection of processes and procedures that adhere to certain ideals and principles. They are similar in that they both focus on incremental, iterative development, which allows for rapid response to changing requirements, happy clients, and top-notch final products. Agile methods emphasize people and their connections rather than processes and technology. They value working software over extensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [8]. Their goal is to enhance company operations and develop novel solutions to address market demands by rapidly redefining resources in response to changing requirements or technological developments, as well as by responding to market shifts or insensitive consumer encounters [9]. Scrum, DSDM, the Crystal Family, and XP are the four most well-known agile methodologies in the business.

### A. Agile Development Framework

This section outlines key Agile development frameworks, Scrum, Kanban, and XP, which are widely used to manage and optimize software development processes. Scrum relies on teamwork, defined roles, and time-boxed sprints. The goal of Kanban is to achieve just-in-time delivery by minimizing work-in-progress and visualizing operations. The XP approach advocates continuous integration, test-driven development, pair programming, and other methods as means to achieve technical excellence. Together, these frameworks provide adaptable and efficient methods for modern software engineering.

#### 1) Scrum

Scrum is a framework that allows Agile teammates to create and maintain composite items. It is praised for being the approach that prioritizes speed and simplicity. Whatever the situation may be, fully mastering and applying the approach remains a challenge. Members of the team are free to utilize a wide range of strategies and tactics inside Scrum [10]. Decisions that describe the events, components, and artefacts, as well as manage the relationships and interconnections between them, are integral to the Scrum framework. In particular, it highlights the importance of self-association among communities that are not threatened by other people [11]. Also, Agile teams using Scrum need to be small enough to be flexible yet big enough to get a lot done in a short amount of time.

#### 2) Kanban

The Kanban method was first used by David J. Anderson in 2003/2004, when he was trying to help a small Microsoft team that was having trouble. So, in 2010, after six years, Anderson finally transmitted the subtleties of Kanban to software development. In contrast to other Agile methods, where change is virtually never avoided, Kanban aims to safeguard the team from the endless tasks assigned by management while simultaneously recognizing the rapid pace of improvement via adaptation [12]. Particularly when used to software development, the Just-in-Time (JIT) foundation is the bedrock of the Kanban method, which shows that the team shouldn't advance software highlights before their interest.

#### 3) XP

One of the most well-known Agile practices is XP. Many organizations, both large and small, and in various stages of development throughout the globe have recently set it up to be highly effective [13]. XP is a method for minor plans

developed by people like Ward Cunningham and Kent Beck. Following several notable accomplishments, XP has generated significant interest and controversy. The XP approach to improvement is realistic, which adds to the mystery [14]. Combination programming, writing tests directly, site refactoring, and revamping continual joining and testing are key methods. Important principles include taking on code ownership, using the simplest arrangement possible, eliminating unnecessary documentation, and adopting a gradual, iterative approach.

### B. Agile in Software and Systems Engineering

Software and systems engineers using the Agile methodology combine iterative development with state-of-the-art engineering techniques, such as MBSE, within a software lifecycle approach in which models serve as the primary artefacts. The shortcomings of conventional DBSE were the primary impetus for the development of MBSE, which was implemented in response to the increasing complexity of information-intensive and cyber-physical systems [15]. An improved system design, traceability, and development over time can be achieved through the use of MBSE, which emphasizes architecture defined by rules, structure, and behavior.

In an Agile environment, MBSE enables rapid prototyping, model simulation, and continuous validation, helping teams adapt to changing needs and reduce risk early. It works in tandem with methods like Domain-Driven Design as it encourages code reuse and stronger business-technology alignment. Despite the many advantages, there are a few drawbacks to combining MBSE with Agile techniques. These include the complexity of the toolchain, the requirement for specialized modelling knowledge, and the risk of organizational resistance to change [16]. The flexibility that Agile strives for might be compromised if problems are not handled appropriately, thus it's important to find a happy medium between model integrity and iterative adaptability.

### C. Benefits and Limitations of Agile Practices

The software and systems engineering process benefits greatly from agile approaches, which enable the efficient, rapid delivery of high-quality solutions, even in domains where requirements are continually evolving. The agile methodology enables companies to adapt quickly to shifting customer demands and market circumstances through its focus on iterative development, close cooperation among diverse teams, and emphasis on adaptability. By embracing change, an agile methodology ensures the end result meets or exceeds customer expectations. Early and continuous feedback is made possible through the iterative process, which aids decision making and enables for the early detection of faults. Agile also makes sharing of knowledge and communication much easier, as big and complicated projects are dismantled into smaller and simpler parts, which also help the teams to keep things clear, increase their efficiency, and speed up the process of delivery. The constant focus on technical superiority through practices such as refactoring also leads to improved system design quality, resulting in software that is more reliable, scalable, and easier to maintain.

However, Agile practices also impose some limitations which organizations must take into account. The environments of Agile depend on very close-knit teams, open and honest communications, and a culture that accepts and supports changes, and this may be hard to achieve for

companies that are used to working in traditional rigid hierarchies [17]. A major problem can arise from excessive reliance on consumer engagement if the relevant parties are either unavailable or inconsistent in their input. The iterative nature of Agile could embrace the scope of the project if alterations are not properly controlled, resulting in an increase in the project duration or its costs. In addition, it can prove difficult to maintain thorough documentation when using the Agile method since the focus is often on speed rather than on details that can be recorded. Technical debt can grow when teams choose to be fast rather than thorough in their design planning, and scaling Agile across large, diverse, or multidisciplinary teams can cause synchronization problems.

### III. QUANTUM COMPUTING: CONCEPTS AND ENGINEERING RELEVANCE

The foundation of quantum computing lies in the fundamental principles of quantum mechanics, a branch of theoretical physics that describes the universe at the atomic and subatomic levels.

- **Superposition:** There is no possible value for a classical bit other than zero or one. Quantum bits, or qubits, in contrast, take advantage of quantum superposition. A qubit may so represent both 0 and 1 at the same time.
- **Entanglement:** Entangled quantum bits (qubits) depend on one another's states, independent of each other's location. Prior to this, Einstein had characterized the phenomenon as "spooky action at a distance."
- **Quantum Measurement:** Quantum bits, in contrast to traditional bits, reduce their states to a binary 0 or 1 when measured. An important factor in this assessment is the predicted amplitude of the qubit's state, which indicates the likelihood that the qubit will collapse to a specific value.

#### A. Agile Roles and Processes in Quantum Engineering

The quantum software engineering debate is heating up, and as developers gradually abandon classical programming techniques in favor of a new computational model predicated on superposition, entanglement, and probability, the use of adaptive and iterative processes is becoming more acceptable [18]. Quantum software engineering will very likely follow a similar path to the one where conventional programming shifted from hardware-centered 1950s practices to today's Agile fostering development ecosystem. But, the tools, roles, and practices that offer agility in classical environments have to be re-evaluated for uniqueness with the characteristics of quantum systems. The developers dealing with quantum programs have to change their mental models, for even simple tasks like qubit state manipulations, issuing quantum commands, and interpreting measurement outcomes, are already pervaded with difficulties unheard of in classical development. Testing and debugging would, also, need new strategies since it would be particularly difficult to define the test oracles for quantum states, create quantum test data, and handle hardware-imposed errors, thus primarily presenting challenges different than those in traditional software engineering [19]. Fig. 1 illustrates the different stages of quantum software engineering, spanning from quantum domain engineering to algorithm design, coding, compilation, and, eventually, quantum hardware execution. Additionally, it explicates the process by which classically-written quantum

programs are transformed into quantum circuits that operate on qubits by virtue of aspects like entanglement and superposition.

The foundational concepts of quantum physics provide the basis of quantum computers. These are:

- **Superposition:** A qubit can be in more than one state at once. A qubit may have two states 0 and 1, at the same time, in contrast to traditional bits, which can only have one of these states [20]. Because of this, quantum computers can speed up certain computations by evaluating multiple possibilities concurrently.
- **Entanglement:** There is a connection between the states of two qubits when they are intertwined. Determining the status of one qubit requires knowledge of the other qubit's state, which is obviously not achievable. Quantum communication and algorithms can benefit from the characteristic.
- **Quantum Interference:** Quantum algorithms can be implemented using quantum interference to improve the success rate and reduce the failure rate. Quantum computing boosts algorithm performance by utilizing this property.
- **Quantum Measurement:** A qubit may only take on one of two possible states during a measurement: 0 or 1. Quantum computing is fundamentally uncertain due to measurement, yet it may also yield valuable insights.

#### B. Current Technologies of Quantum

A survey of the underlying technologies of quantum computing will make up this section. Platforms for quantum computing hardware covered as well are the foundations of quantum computation [21]. They are audited quantum APIs and software development kits, the gateway to quantum algorithm development, and investigate quantum cloud services, which allows the provision of quantum-based services. Lastly, quantum sensors, which will introduce a new age in measurements of precision and quantum networking which will trigger the process of information exchange between parties are explored.

##### 1) Quantum Software Development Kits

Integrated development kits (IDKs) for computers, such as IBM's Qiskit and Microsoft's Quantum Development Kit, allow for the creation, testing, and optimization of quantum algorithms and circuits [22]. They are software development kits (SDKs) offering toolkit and libraries of quantum algorithm development. These kits provide a set of features, including simulators, specialized programming languages, and a number of resources to ease the generation and implementation of quantum code. Microsoft Quantum Development Kit, Google Cirq SDK, and IBM Qiskit are the most famous quantum software development kits (SDKs).

##### 2) Quantum Software Libraries

Quantum Libraries include ready-to-use code and functions and resources that facilitate the creation of quantum algorithms and applications. These generalized sets of tools, algorithms, and resources are programmed to use the colossal potential of quantum computers [23]. An example is QuTip, which is particularly good at quantum optics and quantum information science, and has quantum system simulation and analysis tools [24]. Cirq, a project of Google, is specialized in the creation of quantum circuits and noise-aware simulation, whereas Open Fermion is specialized in quantum chemistry simulation. The leading tech firms and research

establishments are leading the way in development of quantum software libraries. In this field, IBM, Google, Microsoft and Righetti are appreciated participants.

### 3) Quantum Cloud Services

The spread of quantum computing resources is made possible by cloud computing with quantum capabilities. With these services, customers may run quantum algorithms and apps without investing in and managing their own quantum computer [25]. Google, IBM, and Righetti are just a few of the cloud-based providers that provide remote access to quantum computers [26]. Cloud-based services like IBM Quantum Experience and Amazon Braket make quantum computing resources available. IBM's quantum processors and simulators are part of IBM, while AWS's quantum computing capabilities are part of Amazon Web Services. Through the use of these gadgets, the possibilities of quantum computing, as well as the study of quantum algorithms, may be investigated. Fig. 1 shows a simplified version of the software for quantum cloud computing.

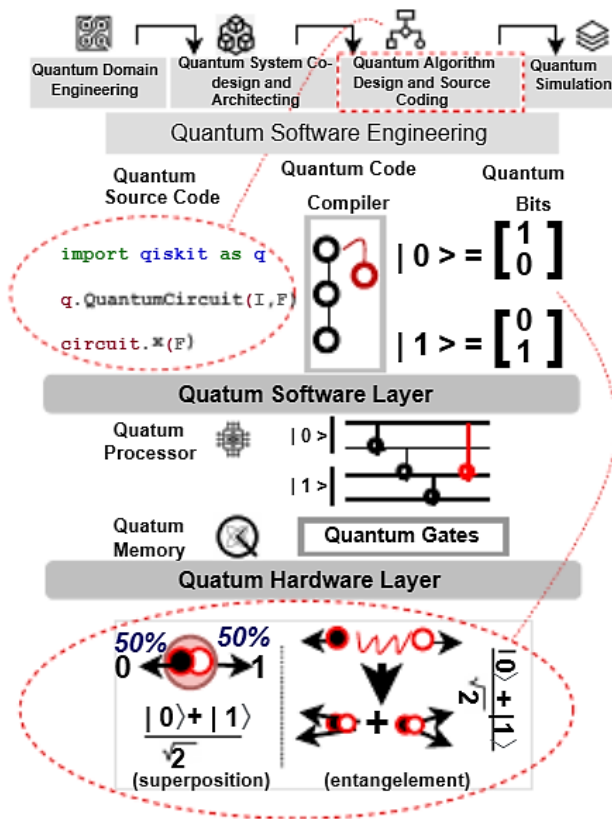


Fig. 1. Quantum Computing

## IV. INTERSECTION OF AGILE DEVELOPMENT AND QUANTUM COMPUTING

Using Agile and Quantum computing gives a flexible and iterative structure on how quantum applications can be developed. The focus of Agile on the short development processes, cross-functional teams, and a continuous feedback approach is rather suitable in terms of the experimental character of the quantum computing business, as the progress can be rather incremental and can be easily altered by the technological advancements [27]. This means that Agile gives teams a way to react to advances in both technology and theory related to quantum computing by breaking down big, hard quantum algorithms into smaller, easier tasks that can be completed. Through the implementation of a notion of

continuous development, quantum solutions are enhanced for optimization and machine learning applications in business, with the goal of decreasing risk and delivering meaningful improvements.

- **Iterative Development:** Agile sprints allow for the sequential construction of quantum algorithms and the rapid adaptation to new situations.
- **Collaboration:** Efficient problem resolution and innovation are the results of functional group collaboration, for example, between software engineering and quantum physics.
- **Flexibility:** The ability to quickly adjust to changes in quantum hardware and new discoveries is what makes agility so competitive; it allows for quick realignment when needed.
- **Test-Driven Development:** The provided quantum algorithms may be regularly evaluated on simulators to make sure they work before they are executed on the quantum hardware. Striving for Perfection: Enhancing quantum algorithms and raising the performance and feedback thresholds evaluation of past appreciations and comments.

### A. Adapting Agile Practices to the Unique Nature of Quantum Software Engineering

The QSE eventually follows in the footsteps of conventional programming's agile development practices. Both approaches have their roots in the 1950s with hardware-focused, hard-wired techniques and have since evolved into the modern standards for agile system development. But, the characteristics of quantum software development need a reevaluation of the methods and instruments used to achieve agility in conventional software engineering [28]. Software engineers meet paradoxical quantum concepts and must adopt a completely new way of thinking about programming while creating quantum programs, which presents unique hurdles in and of itself [29]. For example, determining the values of qubits when they are under quantum instructions. Quantum programs must be tested by developing answers to the following problems: (1) creating test oracles because it may be difficult to get precise results from a quantum program's state being in superposition; (3) managing false positives or negatives due to hardware flaws; and (2) effectively producing quantum test data because quantum variables may be exponentially higher than classical variables.

### B. Agile Methodologies for Quantum Software Development

Agile approaches provide the iterative and flexible structure needed to manage uncertainty, changing tools, and hardware limits, making them a crucial tool in quantum software development [30]. The continuous cycles of development, experimentation, and fast feedback characterize the Agile approach to quantum projects, which also permit the teams to develop the circuits and algorithms of the quantum systems as the technologies become more mature. Some of the main Agile contributions are: (1) Iterative development which involves testing very often on hardware and simulators; (2) Incremental delivery which makes it possible to gradually improve hybrid quantum-classical systems; (3) Cross-functional collaboration which gathers physicists, software engineers, and domain experts; (4) Adaptive planning which is compulsory for qubit availability to change, for updates in the toolchain, and for research insights to be altered; and (5)

Continuous integration and improvement which keeps the quantum software in step with the upcoming platforms[31]. Agile methodologies with these roles and practices will be the ones to give a structured but flexible foundation for the creation of reliable and inventive quantum applications.

Hernández González and Paradela (2020) have done a study that focuses on the connection of agile methodologies with quantum technology where they suggest a quantum agile development framework, their study reveals that agile principles can be easily adapted to quantum software projects thus; project teams get to know the way they can use quantum computing for iterative development and continuous improvement. Agile principles have been recognized and applied to quantum software projects. Thus, the authors point to the adaptability of project teams in improving and developing quantum software regularly, i.e. by using the computer for a constructive and iterative process. They mention that in project management a new digital era has begun and using the right methodologies for the technological advancements is very important and thus, they would suggest adapting from the previous ones [32]. The acknowledgment of the inevitable transition in project management practices in the face of emerging technologies is one of the reasons the authors wrote the paper, and quantum computing is among those technologies.

### C. Impacts of Quantum Computing on Software Engineering

Software engineering is being transformed by quantum computing, which is driving new algorithms, greater complexity, and hybrid systems. It also drives advances in optimization and demands a shift to post-quantum cryptographic methods.

- **Algorithm Development:** The development of quantum computing has spawned new approaches to old problems, such factoring big numbers and searching unsorted databases, respectively. Software developers might also need to modify or develop new algorithms to take advantage of quantum speedup.
- **Increased Complexity:** Quantum computers are complex by nature and thus developing software that can run on them is more difficult. Software engineers have to deal with special defects and performance problems in quantum.
- **Hybrid Systems:** Integration of both classical and quantum computing is going to be required in many applications. Software programmers will be required to devise new integration strategies to fully utilize both forms of processing.
- **Optimization Problems:** Quantum computing is capable of reliably solving some optimization problems, including finance and logistics. Software engineers will need to re-evaluate these issues to maximize the use of quantum algorithms.
- **Security and Cryptography:** A new kind of cryptography known as post-quantum cryptography has emerged in reaction to the danger that quantum computing poses to classical methods of encryption. Software engineers would need to adopt and implement these new standards to secure them.

## V. LITERATURE OF REVIEW

The next section presents a literature review of modern engineering projects that incorporate quantum computing and

agile development practices; Table I provides a concise overview of this review.

Shameem et al. (2025). The suggested measurements and categories were confirmed by empirical research conducted through a survey. To further establish the relative weight of each measurement, they employed the fuzzy AHP method. They conclude with a taxonomy framework that provides a prioritized list of approaches for handling estimates in quantum software development. The results indicate that the subsequent improvement in general project management in the field of quantum software engineering can be achieved to a considerable extent through the implementation of the suggested framework. Quantum computing is a paradigm of change in computing, but quantum software development is considerably more complicated than a conventional software development [33].

Valdés-Rodríguez et al. (2024) there is an urgent need for research into how small and medium-sized businesses (SMEs) can better incorporate security measures into their software development processes, as this is an issue that is gaining traction quickly. Specifically, studies examining the role of security in agile software development have focused on SMEs (20.2% of the total). Additionally, being a significant obstacle in software development, the agile environment evaluates and categorizes methods that have successfully integrated security measures into the SDLC. Small businesses also need to have effective strategies that would guarantee protection of applications and sustainability [34].

Sohail et al. (2024) suggest innovative frameworks and methods to overcome this shortcoming, which would guarantee sustainability during the Agile development process. This research can be of significant value to both researchers and practitioners, as it enables them to identify and analyze applicable procedures and methods that enhance sustainability within the Agile framework by leveraging the long-term viability and adaptability of software products. The study adds to the accumulating rule of sustainable software development and provides effective tips on how to introduce sustainability considerations into the Agile systems. The results show how crucial it is to make software more resilient and flexible in a world where technology is changing at a rapid pace [35].

Khan et al. (2023) A new subfield of software engineering called "quantum software engineering" is emerging, and its practitioners are already finding better ways to tackle difficult computational problems by drawing on concepts of quantum bits and quantum gates. Although the interviewees expressed optimism that agile software development practices could address many of the issues plaguing quantum software development, the results showed that new obstacles remain to be overcome before these practices can be fully implemented. quantum software engineering (QSE) that seamlessly integrates agile approaches to design next-generation quantum software systems and applications [36]

Ali et al. (2023) Reported That Customer satisfaction is high and that the product is delivered quickly and affordably through agile approaches. Based on the results of a SLR, agile methodologies are desirable from a management perspective for large-scale agile projects due to their cheap cost and great customer satisfaction. A grand total of 72 practices were uncovered during the evaluation process. SLR's analytical hierarchical methodology has uncovered and prioritized de-

motivator variables. One famous technique for making judgements based on several factors is analytical hierarchical processing (AHP). The study used AHP to rank the many types of de-motivators and the relevance of each. Software project managers and practitioners have solid grounding in the prioritized de-motivators and their categories, which helps them embrace Agile Software Development for large-scale projects [37]

Marinho et al. (2022) Software development using agile principles has grown in popularity as a means for businesses to boost efficiency and effectiveness, especially for large-scale, geographically dispersed software projects. According to some research, Communities of Practice (CoPs) play a crucial role in helping big organizations successfully implement agile techniques. Thirteen categories—such as "coordination between teams," "knowledge sharing," and "open community"—that support best practices for using CoPs to scale agile methodologies were synthesized, along with three unique themes related to the application and use of CoPs in large-scale projects [38].

Tanaka and Aoyama (2021) Despite Scrum's simpler principles and ease of implementation, large-scale implementations of Scrum-based agile frameworks need extensive expertise. A large company can produce many items at once using the Multi-items Agile Method, which is based on Less Huge. The approach that has been suggested for three Japanese goods. They had been monitoring the speeds of around 10 teams for a full year. They found that the suggested approach works well for scaling agile organizations to handle many products. Another benefit of the approach is that it facilitates drawing on the expertise of Less or Less Huge. The authors arrive at the conclusion that the Multi-Product Agile Method is a more extensive framework of Small and Minimalist [39].

The literature on Understanding the Use of Agile Development Practices Alongside Quantum Computing in Modern Engineering Projects is summarized in Table I, including the approach, major findings, problems, and future directions.

TABLE I. LITERATURE OF REVIEW ON UNDERSTANDING THE USE OF AGILE DEVELOPMENT PRACTICES ALONGSIDE QUANTUM COMPUTING IN MODERN ENGINEERING PROJECTS

Author	Study On	Approach	Key Findings	Challenges	Future Directions
Shameem et al. (2025)	Validation of quantum software development metrics and creation of a prioritized taxonomical framework	Survey-based empirical study; Fuzzy-AHP method for determining relative significance of each metric	Created a ranked, organised taxonomy of metrics for controlling estimates of quantum software development; Improved framework for quantum software engineering project management	Complexity of quantum software development; Limited standardized metrics; Difficulty in evaluating quantum-specific attributes compared to classical software	Expand the framework with more real-world quantum project datasets; Develop tool support for automated metric evaluation; Explore integration with agile and hybrid development models
Valdés-Rodríguez et al. (2024)	Security integration in agile software development for SMEs	Literature analysis of strategies	Identified effective practices for embedding security into the SDLC in SMEs using agile methods.	Security implementation remains a key challenge, especially for small businesses.	Develop lightweight, cost-effective security integration frameworks for SMEs using agile.
Sohail et al. (2024)	Sustainability in Agile frameworks (e.g., GLUX)	Framework proposal and analysis	Agile practices often neglect long-term software viability; introduced new approaches for sustainable agile development.	Lack of long-term sustainability focus limits software adaptability.	Design agile frameworks that emphasize software longevity and adaptability.
Khan et al. (2023)	Agile in Quantum Software Engineering (QSE)	Interview-based exploratory study	Agile practices can address unique QSE challenges; however, integration is still complex and evolving.	Difficulty in aligning quantum principles with iterative agile cycles.	Build hybrid models for Agile-QSE integration; enhance practitioner training.
Ali et al. (2023)	Agile adoption in large-scale projects	Systematic Literature Review (SLR) + AHP	Identified 72 agile practices and ranked de-motivator factors using AHP.	De-motivators hinder Agile scalability in large environments.	Provide prioritized solutions to mitigate de-motivators in large-scale agile projects.
Marinho et al. (2022)	Role of Communities of Practice (CoPs) in Agile scaling	Thematic synthesis of prior studies	Synthesized 13 categories promoting CoPs to scale Agile, such as "team coordination" and "knowledge sharing."	Applying CoPs effectively across distributed teams remains difficult.	Foster open community practices to improve CoPs for agile scaling.
Tanaka & Aoyama (2021)	Multi-Products Agile Method in large Japanese firms	Case study on 3 products and 10 teams	Proposed method based on Less Huge proved effective for scaling agile across multiple products.	Initial implementation requires substantial experience and expertise.	Develop onboarding and training strategies to adopt Multi-Product Agile at scale

## VI. CONCLUSION AND FUTURE WORK

This review addresses the increasing interconnection between Agile practices and quantum software engineering (QSE), in which the Agile iterative and flexible paradigms, such as Scrum, XP, and Kanban, prove highly responsive to the emerging problems quantum computing is putting on the agenda. The flip side of the coin, Agile practices, has inflicted enormous changes on the traditional software development process where working software, collaboration and responsiveness to changes are the priorities but their combination to QSE requires a lot of accommodation due to the peculiarities of quantum mechanics and hybrid classical-quantum models. Agile and quantum computing together can

create a more structured and at the same time a more adaptable creation of the quantum software systems. However, the difficulties in the form of the lack of standardization of quantum programming languages as well as the low number of quantum-literate developers prevent this integration.

It is suggested that future research should be focused on adapting the Agile techniques to the quantum software development cycle by developing quantum-focused Agile architectures that will consider the quantum properties of superposition and entanglement. Some of the required measures will include the creation of hybrid devices to make the interaction between classical and quantum systems more

efficient, creation of educational courses to train the new generation quantum software developers, and establishment of the most efficient means of testing and maintenance of quantum software. The academic and industrial community will play a vital role in creating standardized protocols, addressing the issues of scalability, and also improving the Agile practices to address the accelerating pace of quantum hardware and algorithms.

REFERENCES

- [1] P. Chandrashekar and M. Kari, "A Study on Artificial Intelligence in Software Engineering with Methodologies, Applications, and Effects on SDLC," *TIJER – Int. Res. J.*, vol. 11, no. 12, pp. 932–937, 2024.
- [2] M. A. Quintana, R. R. Palacio, G. B. Soto, and S. González-López, "Agile Development Methodologies and Natural Language Processing: A Mapping Review," *Computers*, vol. 11, no. 12, pp. 1–22, 2022, doi: 10.3390/computers11120179.
- [3] V. Prajapati, "Advances in Software Development Life Cycle Models: Trends and Innovations for Modern Applications," *J. Glob. Res. Electron. Commun.*, vol. 1, no. 4, pp. 1–6, 2025.
- [4] I. Ray, "Quantum Computing," in *IEEE-ICAER*, 2011. doi: 10.13140/2.1.1021.7286.
- [5] S. Name1 and S. D. A. M. T. A. Chintala, "AI-Driven Decision Support Systems in Management: Enhancing Strategic Planning and Execution," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 12, no. 1, 2024.
- [6] P. Chandrashekar, "Advancements in Automated Incident Management: A Survey within Cloud-Native SRE ( Site Reliability Engineering ) Practices," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 601–609, 2023.
- [7] J. M. Murillo *et al.*, "Quantum Software Engineering: Roadmap and Challenges Ahead," *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 5, pp. 1–48, Jun. 2025, doi: 10.1145/3712002.
- [8] A. M. M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," *Proc. - 2013 Int. Conf. Comput. Electr. Electron. Eng. 'Research Makes a Differ. ICCEEE 2013*, no. February, pp. 160–166, 2013, doi: 10.1109/ICCEEE.2013.6633925.
- [9] G. Maddali and S. J. Wawge, *Site Reliability Engineering*. 2025.
- [10] K. Schwaber and J. Sutherland, "Scrum Guide V7," *Agil. Metrics Agil. Heal. Metrics Predict.*, no. November, pp. 133–152, 2020.
- [11] S. K. Tiwari, "Quality Assurance Strategies in Developing High-Performance Financial Technology Solutions," *Int. J. data Sci. Mach. Learn.*, vol. 05, no. 01, pp. 323–335, 2025, doi: 10.55640/ijdsml-05-01-26.
- [12] M. Alqudah and R. Razali, "A comparison of scrum and Kanban for identifying their selection factors," *Proc. 2017 6th Int. Conf. Electr. Eng. Informatics Sustain. Soc. Through Digit. Innov. ICEEI 2017*, vol. 2017-Novem, no. April, pp. 1–6, 2017, doi: 10.1109/ICEEI.2017.8312434.
- [13] S. M. Saleh, M. A. Rahman, and K. A. Asgor, "Comparative Study on the Software Methodologies for Effective Software Development," *Int. J. Sci. Eng. Res.*, vol. 8, no. 4, pp. 1018–1025, 2017.
- [14] S. P. Kalava, "Enhancing Software Development with AI-Driven Code Reviews," *North Am. J. Eng. Res.*, vol. 5, no. 2, pp. 1–7, 2024.
- [15] L. S. Awasthi, A. K. Rai, and K. S. Awasthi, "Agile Software Development Issues: A Review," vol. 3, no. 10, pp. 2000–2011, 2016.
- [16] M. Huss, D. R. Herber, and J. M. Borky, "An Agile Model-Based Software Engineering Approach Illustrated through the Development of a Health Technology System," *Software*, vol. 2, no. 2, pp. 234–257, Apr. 2023, doi: 10.3390/software2020011.
- [17] P. Lutwama, M. Dzulane, T. Pillay, F. S. Hassan, and S. Grobbelaar, "Agile: Advantages, Disadvantages, Enablers, and Barriers," *South African J. Ind. Eng.*, vol. 35, no. 4, 2024, doi: 10.7166/35-4-3058.
- [18] T. M. Forcer, A. J. G. Hey, D. A. Ross, and P. G. R. Smith, "Superposition, entanglement and quantum computation," *Quantum Inf. Comput.*, vol. 2, no. 2, pp. 97–116, 2002, doi: 10.26421/qic2.2-1.
- [19] A. Khan *et al.*, "Agile Practices for Quantum Software Development: Practitioners' Perspectives," 2022. doi: 10.48550/arXiv.2210.09825.
- [20] D. Kaur, "A Review Paper on Quantum Cryptography," vol. 11, no. 11, pp. 189–193, 2025.
- [21] A. Acín *et al.*, "The quantum technologies roadmap: A European community view," *New J. Phys.*, vol. 20, no. 8, 2018, doi: 10.1088/1367-2630/aad1ea.
- [22] R. LaRose, "Overview and comparison of gate level quantum software platforms," *Quantum*, vol. 3, 2019, doi: 10.22331/q-2019-03-25-130.
- [23] J. R. Johansson, P. D. Nation, and F. Nori, "QuTiP: An open-source Python framework for the dynamics of open quantum systems," *Comput. Phys. Commun.*, vol. 183, no. 8, pp. 1760–1772, 2012, doi: 10.1016/j.cpc.2012.02.021.
- [24] C. M. Penugonda, V. K. Devalla, C. Rami Reddy, and S. B. Karri, "Quantum Computing for Nanomaterials: Accelerating Material Discovery," in *Volume 414: Advancements in Smart Innovations, Intelligent Systems, and Technologies*, 2025. doi: 10.3233/FAIA251501.
- [25] D. Subhi and L. Bacsardi, "Using Quantum Nodes Connected via the Quantum Cloud to Perform IoT Quantum Network," *Condens. Matter*, vol. 8, no. 1, p. 24, Feb. 2023, doi: 10.3390/condmat8010024.
- [26] N. K. Prajapati, "Quantum Computing and Its Impact on Cryptographic Security," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 5, pp. 1–13, 2025.
- [27] A. Goyal, "Scaling Agile Practices with Quantum Computing for Multi-Vendor Engineering Solutions in Global Markets," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 06, Jun. 2022, doi: 10.14741/ijcet/v.12.6.10.
- [28] A. I. S. Als Salman, "Quantum Software Engineering: Best Practices from Classical to Quantum Approaches," *J. Quantum Inf. Sci.*, vol. 14, no. 04, pp. 234–258, 2024, doi: 10.4236/jqis.2024.144010.
- [29] M. R. R. Deva, *DevOps and Continuous Delivery Adoption: Trends, Challenges, and Best Practices in Modern Software Development Life Cycle*. 2025.
- [30] M. Salam and M. Ilyas, "Quantum computing challenges and solutions in software industry—A multivocal literature review," *IET Quantum Commun.*, vol. 5, no. 4, pp. 462–485, Dec. 2024, doi: 10.1049/qtc2.12096.
- [31] B. R. Cherukuri, "Quantum machine learning: Transforming cloud-based AI solutions," *Int. J. Sci. Res. Arch.*, vol. 1, no. 01, 2025.
- [32] K. C. Lakshminarasimham, "Quantum Computing in Project Management: Unlocking New Frontiers for Optimization and Decision-Making," vol. 4, no. 4, pp. 61–67, 2024, doi: 10.56472/25832646/JETA-V4I4P108.
- [33] M. Shameem, M. Nadeem, M. Niazi, S. Mahmood, and A. Kumar, "Taxonomy of metrics for effectively estimating quantum software projects: A fuzzy-AHP based analysis," *Appl. Soft Comput.*, vol. 172, p. 112816, Mar. 2025, doi: 10.1016/j.asoc.2025.112816.
- [34] Y. Valdés-Rodríguez, J. Hochstetter-Diez, M. Diéguez-Rebolledo, A. Bustamante-Mora, and R. Cadena-Martínez, "Analysis of Strategies for the Integration of Security Practices in Agile Software Development: A Sustainable SME Approach," *IEEE Access*, vol. 12, pp. 35204–35230, 2024, doi: 10.1109/ACCESS.2024.3372385.
- [35] M. T. Sohail, S. Mughal, U. Abiha, A. Abbas, M. Ali, and S. U. Khan, "An Empirical Study to Investigate the Impact of Sustainable Practices in Agile Software Development," in *2024 International Conference on Frontiers of Information Technology (FIT)*, 2024, pp. 1–6. doi: 10.1109/FIT63703.2024.10838460.
- [36] A. A. Khan *et al.*, "Agile Practices for Quantum Software Development: Practitioners' Perspectives," in *2023 IEEE International Conference on Quantum Software (QSW)*, 2023, pp. 9–20. doi: 10.1109/QSW59989.2023.00012.
- [37] F. Ali, M. Usman, M. F. Abrar, S. U. Rahman, I. Khan, and B. Niazi, "Practices of De-Motivators in Adopting Agile Software

- Development Methods at Large Scale Development Teams From Management Perspective,” *IEEE Access*, vol. 11, pp. 130368–130390, 2023, doi: 10.1109/ACCESS.2023.3331759.
- [38] I. Monte, L. Lins, and M. Marinho, “Communities of Practice in Large-Scale Agile Development: A Systematic Literature Mapping,” in *2022 XLVIII Latin American Computer Conference (CLEI)*, 2022, pp. 1–10. doi: 10.1109/CLEI56649.2022.9959929.
- [39] M. Tanaka and M. Aoyama, “A Distributed Large-Scale Agile Software Development for Multiple Products and Its Practical Evaluation,” in *2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2021, pp. 66–72. doi: 10.1109/SERA51205.2021.9509268.