Research Paper

Quantum-ready software engineering: principles, frameworks, and hybrid development pipelines

Hasanain Hazim Azeez*
*Department of Computer Science and IT, Wasit University, Kut, Iraq.

Article Info

Article History:
Received: 10 May 2025
Revised: 19 July 2025
Accepted: 27 July 2025
Published: 13 September 2025

Keywords:
Quantum Software Engineering
Hybrid Quantum Classical Systems
Quantum Ready Framework
Qiskit Programming
Software Design Patterns
NISQ Era Computing

ABSTRACT

Quantum computing is quickly becoming a potentially disruptive technology that can tackle the problems inaccessible to classical computers, but the use of quantum computing in real software systems has yet to be fully explored. Present day quantum studies are largely focused on the development of algorithms and hardware, whereas engineering workflows to be used to enable smooth hybrid quantum-classical operations attract limited attention. The paper suggests a concept of quantum-ready software engineering- a systematic approach to applying established principles of software engineering to hybrid systems. In order to meet maintainability, scalability, and reliability objectives in hybrid applications, propose the adoption of a multilayered design, such as abstraction, integration, orchestration, and verification. An example concrete programming template, written in Python in Qiskit, demonstrates how quantum functions being used as modular services can be simulated with classical software. Evaluation measures (maintainability index, latency, and modularity) demonstrate the feasibility of the framework. Results indicate that the hybrid pipelines can be designed in a systematic way using quantum-classical-adapted design patterns. The study introduces a repeatable methodology, program templates, and design pattern engineering principles, which open the door to additional development of quantum DevOps and design pattern formalization. This project, by bridging the gap between classical software engineering and quantum integration, gives a platform upon which to build extraordinarily reliable, scalable systems in the Noisy Intermediate-Scale Quantum and beyond.

Corresponding Author:
Hasanain Hazim Azeez
Department of Computer Science and IT, Wasit University, Kut, Iraq.
Email: hbashagha@uowasit.edu.iq

1. INTRODUCTION

The field of quantum computing is no longer just theorizing but is now being experimented with in application, with firms such as IBM, Google, and Rigetti now making quantum hardware accessible through cloud computing [1], [2]. These systems are already useful in fields such as optimization, cryptography, and machine learning even though they are currently constrained by the number of qubits and error rates [3], [4]. Yet strategies to realize quantum modules as the interaction with more complex systems remain immature [5].

Conservative software engineering is concerned with modularity, testability, maintainability and scalability. In comparison most quantum programs nowadays are monolithic and experimental, run on small-scale proofs-of-concept. This incompatibility poses adoption challenges: unless engineering is thought about in terms of abstraction, interoperability, error control and long-term maintainability, organizations cannot merely add a quantum algorithm to existing infrastructure. The next generation of quantum technology is hybrid quantum-classical systems, which are unavoidable in the NISQ era [6].

The classical computers will never be replaced by the quantum computers regarding the processing of the data and its control as well as post processing; the latter are only used as the special processors [7]. The difficulty is making the quantum and classical elements to live together within the software systems. That gap is met with this article where quantum-ready principles of software engineering are proposed. In particular: (1) introduce a framework of hybrid development pipelines, (2) show programming templates in Python and Qiskit, (3) test the framework by engineering metrics such as modularity and latency, and (4) reflect on challenges and implications of a quantum future to software engineering.

2. RELATED WORK

The classical software engineering is grounded in modular design patterns [8], structured testing, continuous integration and layer architecture [9]. Those values ensure long-run scalability and flexibility.
In recent recent years quantum programming has witnessed a strong rise, and various different frameworks support hybrid development. The most researched are Qiskit [1], Cirq, Amazon Braket, and PennyLane, with each focusing on approaches to quantum computing being varied. For example, Qiskit can be written to support end-to-end quantum hardware-based designs and applications (albeit provided by IBM), and Cirq can be used to provide fine-grained control over near-term algorithms, and PennyLane to target quantum machine learning Table 1.

The materials of recent papers, [5] explain the challenges of the design of quantum software and [10] discuss financial applications. [11] Discuss the factoring in the context of software engineering, whereas van Meter [12] considers quantum networking integration. These works are educative, yet does not constitute a comprehensive means to hybrid software engineering. The innovation of the paper is that it offers a syntactic framework, which specifically seeks to have the engineering integration of quantum and classical components [13].

Table 1. Comparison of Approaches to Development

Approach	Focus	Strengths	Weaknesses
Classical	General-Purpose SW	Mature Engineering Practices	Cannot Solve Quantum-Hard Problems

Quantum-Only	Circuit Development	Explores Novel Algorithms	Lacks Interoperability
Hybrid (Proposed)	Quantum + Classical	Leverages Strengths of Both	Requires New Engineering Methods

3. METHODOLOGY

The method proposed implements a holistic and layered approach to building a hybrid quantum classical system. At its base, the Abstraction Layer encapsulates the quantum algorithm into modular services that can be reused, so that details of qubit manipulation are imparted no higher than the application level. The Integration Layer functions just like middleware by providing reliable communication protocols (e.g., REST, GRPC APIs, etc.) for classical software components to interact with quantum backends.

The Pipeline Layer manages the end-to-end execution, mixing together classical pre-processing, quantum computation, and classical post-processing so as to afford dynamically orchestrated scalable workflows. The last layer, Testing and Verification assures correctness, robustness, and reproducibility through hybrid test cases, benchmarking of quantum computations with classical approximations, and anomaly detection introduced by noise. These layers, together, build a structured framework for reliable, extendable, and production-ready hybrid systems. Figure 1 presents the improved UML-inspired architecture diagram, highlighting the layered interactions between classical applications, middleware, quantum services, and verification mechanisms [14], [15].

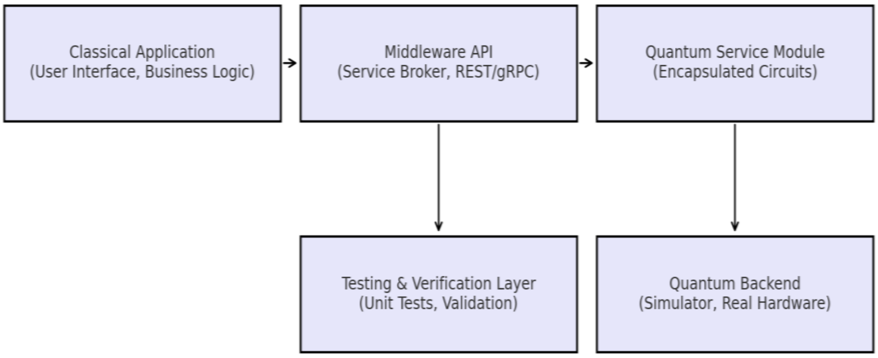

Figure 1. Improved Uml Architecture of Hybrid Quantum-Classical System

Programming Templates and Examples

Listing 1. Classical-to-Quantum Integration (Python + Qiskit):

```
from qiskit import Quantum Circuit, Aer, execute
def quantum_rng(num_qubits=3):
qc = Quantum Circuit (num_qubits, num_qubits)
qc.h(range(num_qubits))
qc.measure(range(num_qubits), range(num_qubits)
backend = Aer.get_backend('qasm_simulator')
job = execute (qc, backend, shots=1)
result = job.result(). get_counts()
return list(result.keys()) [0]
def hybrid_workflow ():
classical_part = "Hello, Quantum!"
quantum_part = quantum_rng ()
return f"{classical_part} → Random: {quantum_part}"
print(hybrid_workflow())
```

This example demonstrates service encapsulation: the quantum RNG is modular, testable, and callable from classical code.

Evaluation

The proposed framework is evaluated on three dimensions that are interrelated and include: maintainability, performance and adaptability of classical design patterns to hybrid environment. In terms of maintainability, quantum functionality in separate modules improves modularity and lowers connection, as well as, increasing long-term maintainability indices [16], [17], [18]. The performance analysis shows that quantum backbends have increased latency relative to completely classical implementations, but this overhead is manageable in carefully coordinated pipelines.

The comparisons of latency between various workloads in Figure 2 show that quantum computation can be optimally integrated with any existing system in real-world scenarios, as quantum-based benefits are not only unique but also frequently significant. Moreover, popular software engineering design patterns, including Strategy, Observer, Adapter and Factory, have effectively been applied to hybrid settings, showing that traditional engineering wisdom can inform quantum integration without needing to re-discover the underlying practices [19], [20], [21], [22].

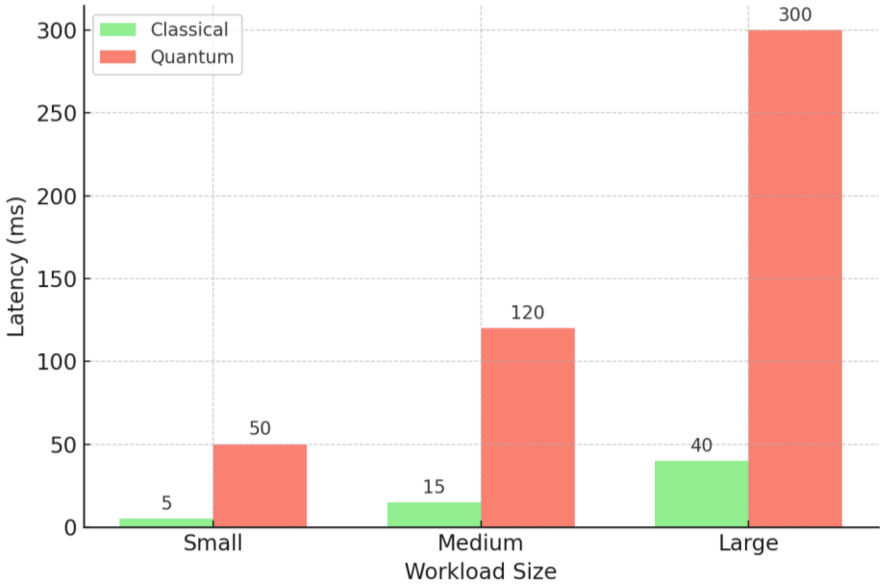

Figure 2. Latency Comparison Between Classical and Quantum

Table 2. Mapping of SE Patterns to Hybrid Context

Pattern	Hybrid Application Example
Strategy	Switch between quantum/classical algorithm
Observer	Monitor job states via API callbacks
Adapter	Wrap quantum circuits for classical calls
Factory	Instantiate backend-specific circuits

4. RESULTS AND DISCUSSION

Design of quantum-ready systems poses both experimental and theoretical challenges. Hardware limits the depth of the circuit. The versioning issue is that SDKs evolve at a high rate. Quantum non-determinism makes testing difficult. There are cryptography and data protection problems that are related to controllable ethics. However, the proposed methodology becomes practical when it comes to hybrid systems [23], [24], [25].

5. CONCLUSION

We thus introduce a new paradigm of quantum-prepared software engineering for hybrid systems with quantum and classical computing elements. The structure of the framework has layered support in terms of maintainability, scalability, orchestration, and verification to cater to the complexities of the application for systematic and reliable development. Besides the conceptual model, the methodology offers practical programming templates and evaluation benchmarks for real cases.

The study underscores the pressing requirement for a disciplined engineering approach that could mend the existing gap between experimental quantum prototypes and large-scale commercial applications. Further research needs to endeavor to formalize quantum-oriented design patterns, automated modules and extensive test plans, and integrate quantum processes into the DevOps workflow. These advances will be critical to ensure that hybrid architectures stand the test of time: they should be scalable and easily operational for industrial use once quantum computing is in its beginning stages of operational maturity.

Acknowledgments

The authors have no specific acknowledgments to make for this research.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hasanain Hazim Azeez	✓	✓			✓	✓	✓		✓	✓		✓	✓	✓

C: **C**onceptualization
M: **M**ethodology
So: **So**ftware
Va: **Va**lidation
Fo: **Fo**rmal analysis
I: **I**nvestigation
R: **R**esources
D: **D**ata Curation
O: Writing- **O**riginal Draft
E: Writing- Review& **E**diting
Vi: **Vi**sualization
Su: **Su**pervision
P: **P**roject administration
Fu: **Fu**nding acquisition

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Informed Consent

All participants were informed about the purpose of the study, and their voluntary consent was obtained prior to data collection.

Ethical Approval

The study was conducted in compliance with the ethical principles outlined in the Declaration of Helsinki and approved by the relevant institutional authorities.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. Schuld and N. Killoran, 'Quantum machine learning in feature Hilbert spaces', Phys. Rev. Lett, vol. 122, no. 4, 2019.doi.org/10.1103/PhysRevLett.122.040504

[2] A. Abbas, 'The power of quantum neural networks', Nat. Comput. Sci, vol. 1, pp. 403-409, 2021.doi.org/10.1038/s43588-021-00084-1

[3] T. Häner, 'Software engineering of quantum programs: Challenges and opportunities', ACM Trans. Comput. Theory, vol. 12, no. 4, 2021.

[4] S. Pistoia, 'Quantum computing for finance: State-of-the-art and future prospects', IEEE Trans. Quantum Eng, vol. 3, pp. 1–21, 2022.

[5] IBM Quantum, "Qiskit: An open-source SDK for quantum computing," 2023. Available: https://qiskit.org

[6] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, pp. 79, 2018. doi.org/10.22331/q-2018-08-06-79

[7] D. Venturelli and A. Kondratyev, 'Reverse quantum annealing approach to portfolio optimization problems', Quantum Mach. Intell, vol. 1, pp. 17-30, 2019. doi.org/10.1007/s42484-019-00001-w

[8] S. Guerreschi and J. Park, 'Factoring: A quantum software engineering perspective', npj Quantum Inf, vol. 7, 2021.

[9] I. Google, 'Quantum supremacy using a programmable superconducting processor', Nature, vol. 574, pp. 505–510, 2019.

[10] R. Van Meter, 'Quantum networking and software engineering', IEEE Internet Comput, vol. 24, no. 5, pp. 15–24, 2020.

[11] A. Ruqayyah, 'Media Convergence and the Change in Media Content Production and Distribution in Nigeria', Culture and Communication(JMCC), vol. 2, no. 04, pp. 40-48, 2022. doi.org/10.55529/jmcc24.40.48

[12] A. Kishore, 'Populism, Protest and the Social Norms', Tourism and Hospitality (JSRTH), vol. 2, pp. 16-23, 2022. doi.org/10.55529/jsrth.23.16.23

[13] ere too. A. Yousuf Rather, 'Gandhian Philosophy is More Relevant Today than Yesterday', Tourism and Hospitality (JSRTH), vol. 2, pp. 19-24, 2022. doi.org/10.55529/jsrth.24.19.24

[14] D. P. J. Bachhav, 'The Implication of Information and Communication Technologies (ICT) in Enhancing Communication Skills of MBA Students (with Reference to Four MBA Colleges in Pune)', Culture and Communication (JMCC), vol. 2, no. 06, pp. 10-20, 2022. doi.org/10.55529/jmcc26.10.20

[15] D. R. Bhat, M. H. Kaur, and M. P. Sarkar, 'Role of FM Radio in the Dissemination of Important Messages in the Time of Covid - A Case Study on Delhi /NCR', Culture and Communication (JMCC), vol. 2, no. 06, pp. 21-32, 2022. doi.org/10.55529/jmcc26.21.32

[16] A. Chowdhury, 'History Beyond the Frame: Exploring the Art in Spiegelman's Maus', Culture and Communication (JMCC), vol. 2, no. 05, pp. 6-10, 2022. doi.org/10.55529/jmcc25.6.10

[17] D. Chima Paul ph and M. Joseph, Eds, 'Towards Digital-Driven Tourism for National Development in Nigeria: Lessons from Dubai Experience', Tourism and Hospitality (JSRTH), vol. 1, pp. 1-13, 2021. doi.org/10.55529/jsrth.11.1.13

[18] T. Dr, 'Role of Volunteers Developing Social Responsibility During an Outbreak of Epidemic Diseases', Tourism and Hospitality (JSRTH), vol. 1, pp. 35-40, 2021. doi.org/10.55529/jsrth.11.35.40

[19] 'Impact Of Covid-19 Induced Job Insecurity on Psychological Wellbeing of Tourism and Hospitality Sector Employees in India', Tourism and Hospitality (JSRTH), vol. 1, pp. 41-50, 2021. doi.org/10.55529/jsrth.11.41.50

[20] R. Dr, 'A Conceptual Review on Ego-Strength and Organisational Performance', Tourism and Hospitality (JSRTH), vol. 2, pp. 14-17, 2022. doi.org/10.55529/jsrth.21.14.17

[21] R. Dr, 'An Empirical Study on Employee's Welfare Facilities In Hospitals In Madurai', Tourism and Hospitality (JSRTH), vol. 2, pp. 1-5, 2022. doi.org/10.55529/jsrth.22.1.5

[22] D. Echeta, 'Impact of Event Tourism on Lagos Residents Image', Tourism and Hospitality (JSRTH), vol. 2, pp. 24-34, 2022. doi.org/10.55529/jsrth.23.32.41

[23] I. Leo, 'Caste Discrimination and Social Change in India', Tourism and Hospitality (JSRTH), vol. 1, pp. 1-6, 2021. doi.org/10.55529/21.1.6

[24] D. K. Mishra, 'Media Literacy: Building Critical Media Awareness', Culture and Communication (JMCC), vol. 2, no. 05, pp. 11-17, 2022. doi.org/10.55529/jmcc25.11.17

[25] D. P. Ma, 'A Review on Media Tradition and Realities in Natural Disasters', Culture and Communication (JMCC), vol. 2, no. 06, pp. 1-4, 2022. doi.org/10.55529/jmcc.26.1.4

How to Cite: Hasanain Hazim Azeez. (2025). Quantum-ready software engineering: principles, frameworks, and hybrid development pipelines. International Journal of Information Technology and Computer Engineering (IJITC), 5(2), 1-7. https://doi.org/10.55529/ijitc.52.1.7

BIOGRAPHY OF AUTHOR

	**Hasanain Hazim Azeez**, is a researcher and academic in the Department of Computer Science and IT at Wasit University, Kut, Iraq. His work focuses on computer science applications, information technology, and emerging technologies. He actively contributes to teaching, research, and academic development, aiming to enhance innovation and knowledge dissemination. Email: hbashagha@uowasit.edu.iq