Quantum Software Engineering: Future Trends in Software Engineering Body of Knowledge

1st Shaukat Ali
Simula Research Laboratory
and Oslo Metropolitan University
Oslo, Norway
shaukat@simula.no

2nd Paolo Arcaini
National Institute of Informatics
Tokyo, Japan
arcaini@nii.ac.jp

***Abstract*—Quantum Software Engineering (QSE) is an emerging research area focused on developing software that runs on quantum computers to solve complex problems. There has been growing interest among researchers in the classical software engineering community in devising methods for various phases of quantum software development, including requirements engineering, modeling and analysis, testing, debugging, and repair. As QSE evolves, it will become increasingly important within classical software engineering as well, mainly because it will integrate with classical software to create hybrid systems—combining both classical and quantum software. This presentation proposal aims to discuss QSE and advocate for its inclusion in the next Software Engineering Body of Knowledge (SWEBOK) edition.**

***Index Terms*—Software Engineering, Quantum Computing, Quantum-Classical Software**

I. Presentation Proposal

Quantum Computing (QC) is a computing paradigm that uses quantum-mechanical principles with the goal of potentially solving complex problems efficiently [1]. QC is expected to benefit various application areas, including medicine, finance, and climate science. Moreover, it shows promise for solving optimization problems, as early results on classical software test optimization using quantum optimization algorithms have already started to emerge [2], [3]. Irrespective of the application, quantum software must be developed to enable such applications to run on quantum computers. However, developing quantum software is challenging because QC operates fundamentally differently from classical computing, relying on features such as superposition and entanglement. This calls for devising new methods, tools, and guidelines for developing quantum software.

To devise the aforementioned methods, tools, and guidelines, Quantum Software Engineering (QSE) is emerging as a field focused on building dependable quantum software for quantum computers [4]. As interest in QSE grows within the software engineering community and quantum computers gain greater computational power, an increasing number of methods for QSE are emerging [5], [6]. This covers subareas such as requirements engineering, modeling and simulation, programming, testing, debugging, and repair. Moreover, researchers have recognized that the initial applications of quantum software will be hybrid, combining both classical and quantum components. Some aspects of an application will be implemented as classical software, while the rest will be quantum.

Given the growing interest in the field, it is time to consider including Quantum Software Engineering (QSE) as part of the Software Engineering Body of Knowledge (SWEBOK). Indeed, while some concepts of QSE are the same as in classical SE, others are specific to quantum computing and would require a separate treatment. In the last edition of the SWEBOK [7], SE for AI has become part of the book, recognizing the increasing importance of SE in developing AI-based applications. We believe that a similar case should be made for QSE, as developing quantum-based application will increasingly demand the expertise of the SE community.

This presentation will first provide a brief introduction to QSE, followed by an overview of key literature and major research challenges across various areas. Lastly, it will advocate for incorporating QSE into the next version of SWEBOK.

S. Ali is supported by Qu-Test (Project #299827) funded by the Research Council of Norway. P. Arcaini is supported by Engineerable AI Techniques for Practical Applications of High-Quality Machine Learning-based Systems Project (Grant Number JPMJMI20B8), JST-Mirai.

References

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

[2] X. Wang, A. Muqeet, T. Yue, S. Ali, and P. Arcaini, "Test case minimization with quantum annealers," *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 1, Dec. 2024. [Online]. Available: https://doi.org/10.1145/3680467

[3] X. Wang, S. Ali, T. Yue, and P. Arcaini, "Quantum approximate optimization algorithm for test case optimization," *IEEE Transactions on Software Engineering*, vol. 50, no. 12, pp. 3249–3264, 2024.

[4] S. Ali, T. Yue, and R. Abreu, "When software engineering meets quantum computing," *Commun. ACM*, vol. 65, no. 4, p. 84–88, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3512340

[5] J. M. Murillo, J. Garcia-Alonso, E. Moguel, J. Barzen, F. Leymann, S. Ali, T. Yue, P. Arcaini, R. Pérez-Castillo, I. García Rodríguez de Guzmán, M. Piattini, A. Ruiz-Cortés, A. Brogi, J. Zhao, A. Miranskyy, and M. Wimmer, "Quantum software engineering: Roadmap and challenges ahead," *ACM Trans. Softw. Eng. Methodol.*, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3712002

[6] N. C. L. Ramalho, H. Amario de Souza, and M. Lordello Chaim, "Testing and debugging quantum programs: The road to 2030," *ACM Trans. Softw. Eng. Methodol.*, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3715106

[7] H. Washizaki, Ed., *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0*. IEEE Computer Society, 2024. [Online]. Available: http://www.swebok.org