Quantum Computing: A New Software Engineering Golden Age

Mario Piattini
aQuantum by Alarcos Research Group
Pº de la Universidad 4, 13071 Ciudad Real, Spain
mario.piattini@uclm.es

Guido Peterssen
aQuantum by Alhambra
Calle Albasanz 16, 28037 Madrid, Spain
guido.peterssen@a-e.es

Ricardo Pérez-Castillo
aQuantum by Alarcos Research Group
Pº de la Universidad 4, 13071 Ciudad Real, Spain
ricardo.pdelcastillo@uclm.es

DOI: 10.1145/3402127.3402131
https://doi.acm.org/10.1145/3402127.3402131

ABSTRACT

Quantum computing, and to an even greater extent quantum technology, is changing the world. Quantum computing is not an evolution of classical computer science; it is actually a revolution that completely changes the computing paradigm. Quantum computers are based on the principles of quantum mechanics, such as superposition and entanglement, and they seek to boost computational power exponentially. Many problems that have until now been impossible to solve, in practical terms, might very well be able to be addressed by means of quantum computing. The fact is that at the present time quantum computing is influencing most business sectors and research fields, due to its various promising applications. To make such applications become reality, quantum algorithms must be specially coded for these extremely different computers. Although some well-known quantum algorithms already exist, the need for quantum software will increase dramatically in the next years. In that context, quantum software has to be produced in a more industrial and controlled way, i.e., aspects such as quality, delivery, project management, or evolution of quantum software must be addressed. We are sure that quantum computing will be the main driver for a new software engineering golden age during the present decade of the 2020s.

Categories and Subject Descriptors
D.2.0 [**Software Engineering**]: General; D.3.3 [**Programming Languages**]: Language Constructs and Features; C.5.0 [**Computer System Implementation**]: General.

General Terms
Algorithms, Management, Design, Languages.

Keywords
Quantum computing; Quantum software; Quantum software engineering.

1. THE QUANTUM COMPUTING MOMENTUM

During the first quarter of the twentieth century, the foundations of quantum mechanics were established by many exceptional scientists (such as Planck, Einstein, Bohr, Schrödinger, Born, Dirac, De Broglie, Heisenberg, Pauli, etc.). This new, astonishing physics theory describes the behaviour of matter at subatomic levels (photons, electrons, etc.). These foundations led to the first quantum revolution in the second half of the last century. The new physical laws were thus applied in the development of a host of cutting-edge technologies, such as the transistor, GPS, solid-state lighting and lasers, among other advances. Computer science did not lag behind, and it started to take advantage of quantum mechanics in the eighties, when Richard Feynman (Nobel laureate) claimed to be able to simulate physics with computers [1]. This was the seed for the concept of quantum computers, and the second quantum revolution began.

Quantum computers, in their effort to provide faster computing speed, attempt to use various “counterintuitive” principles, such as superposition (objects can be in different states at the same time) and entanglement (objects can be deeply connected without any direct physical interaction). Quantum computational power lies in the concept of qubits, the computational unit of quantum computers, which emerged as the evolution of digital bits. A qubit can be represented with different physics systems (e.g., the spin of an electron or the polarization of a single photon) where the status can be zero, one state and two states at the same time, by superposition. Thus, *n* qubits manage information corresponding to all the possible 2^n permutations; this is the key to the massive computational power.

The idea of building a functional quantum computer has not been abandoned throughout recent decades. Indeed, the quantum race has intensified dramatically in the last few years. Quantum computing has gained relevance in research and economic agendas of nations around the globe [2, 3], and private investment has been enlarged extensively [4, 5].

As a result, functional quantum computers, are part of breakthrough technology today [6]. We can already use quantum computers and in the near future we expect to take advantage of their huge computation capacity in the quest to solve problems which are considered very difficult for today’s classic computers. Quantum computing is actually, a transversal and interdisciplinary opportunity for digital transformation and social impact [7], with multiple direct applications in, among other fields, security and cryptography, medicine and pharmacology, chemistry and materials, economics and financial services, energy, and farming.

2. EMERGING QUANTUM SOFTWARE ENGINERING

All the above applications will not be accomplished with quantum computers alone; they need quantum software. With the rise of the first quantum computers, the initial programming languages and quantum algorithms came up with promising results [8]. Nevertheless, quantum software has not begun to be produced in a large-scale, industrial way yet. We expect this industrialization of quantum software production to come about in the next few years.

Having provided this review of the recent history of software engineering, we should point out that such a breakthrough has already happened several times before [9]. The evolution of Software

Engineering is "bottom-up", since it was developed after computer science foundations had already been laid. First, and most notably, the diffusion of third-generation languages such as COBOL in the 70s, resulted in the structured design techniques proposed by Myers, Yourdon and Constantine; subsequently, the E/R model was defined by Chen. More recently, Gane and Sarson, DeMarco and Weinberg came up with structured analysis. In the 1980s comprehensive methodologies (Merise, SSADM, Information Engineering, etc.) were published. This period is considered by Barry Booch to be the first "golden age" of software engineering [10]. Another clear example happened in the 90s; the hot topic became object orientation, which presented object-oriented analysis, design, and development patterns. This was, in turn, what provided the seeds for the later model-driven architecture at the beginning of the 21st century. In the latter part of the first decade of this century, DevOps was born as a development paradigm for boosting continuous integration and continuous delivery; this has become a must today, along with the consequent "continuous software engineering" [11].

We strongly believe that quantum computing will be the main driver for a new software engineering golden age during the 20s. Software Engineering has built up a broad knowledge base, and has learnt many lessons that should be applied to the production of quantum software. The new quantum software engineering field needs to be considered as the application or adaptation of the well-known methods, techniques, and practices of software engineering. At the same time, however, new methods and techniques will be defined specifically for quantum software production.

3. QUANTUM SOFTWARE ENGINEERING PRINCIPLES

In the "Talavera Manifesto" several principles and commitments for Quantum Software Engineering have been gathered [12]. This manifesto was developed as a result of the discussion and presentation of different viewpoints from academia representatives and industry practitioners who joined in the 1st International Workshop on Quantum Software Engineering & Programming (QANSWER'20). The 9 principles referring to Quantum Software Engineering (QSE) are:

1. **QSE is agnostic regarding quantum programming languages and technologies**.
2. **QSE embraces the coexistence of classical and quantum computing**.
3. **QSE supports the management of quantum software development projects**
4. **QSE considers the evolution of quantum software**.
5. **QSE aims to deliver quantum programs with desirable zero defects**.
6. **QSE assures the quality of quantum software**.
7. **QSE promotes quantum software reuse**.
8. **QSE addresses security and privacy by design**.
9. **QSE covers the governance and management of software**.

4. MAIN IMPLICATIONS

The Talavera Manifesto [12] has already been signed by 100 researchers and practitioners from 20 different countries in the first month subsequent to its publication. This fact reflects the importance that both software engineering and quantum computing communities perceive with regard to the quantum software engineering challenge. This challenge has important current and future implications.

Software practitioners should identify how quantum software development projects influence technical, economic, and organizational aspects. They might start by considering how to adopt the principles mentioned in their daily practices. These practitioners could launch discussions with their peers about the issues set out above, giving due thought to the individual and social dimensions.

Academia should highlight valuable research questions in each specific area of the software engineering field (but also in the quantum computing field) that could help us to understand quantum software engineering better. Researchers from different knowledge areas should investigate how the previous experience in software engineering research could be transferred to the new quantum software engineering field.

Academia, together with industry, must be concerned about the increasing need for a quantum workforce [13, 14]. Educational bodies should attempt to address the lack of skilled professionals in the context of quantum computing and quantum software engineering. To that end, they will have to integrate quantum software engineering in the curricula of the existing software engineering degrees, and/or courses in this and other disciplines. New syllabi must clearly specify which competences and skills are required for future quantum software engineers.

Government and funding bodies should take quantum software engineering principles into account and consider quantum software engineering in strategic research/industrial plans. These plans should be properly supported with a sufficient level of founding schemas and ambitious dissemination programs. Governments should also encourage education bodies to include quantum computing and quantum software engineering in training programs.

Quantum technology vendors also have an active part to play in ensuring the success of quantum software engineering. They must be actively aware of the latest trends in quantum software engineering (from both academia and industry). Vendors should make efforts that align with the principles mentioned above, providing industry with enough tools and trained people to combat the lack of a skilled workforce.

Similarly, professional associations need to revise methods, techniques and practices, and incorporate the quantum software engineering principles to an appropriate degree. These associations can also help in the overall endeavour by spreading information regarding the challenges and benefits of the quantum software engineering field.

If you are interested in supporting the ideas of the manifesto, please endorse it at: https://www.aquantum.es/endorse-talavera-manifesto/

REFERENCES

[1] Feynman, R.P., *Simulating physics with computers.* International Journal of Theoretical Physics, 1982. **21**(6): p. 467-488.

[2] European Quantum Flagship. *Strategic Research Agenda on Quantum technologies*. 2020 https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=65402.

[3] NSTC. *National Strategic Overview for Quantum Information Science.* Subcommittee on Quantum Information Science under the Committee on Science of the National Science & Technology Council 2018 [cited 2020 13/04/2020]; Available from: https://www.whitehouse.gov/wp-content/uploads/2018/09/National-Strategic-Overview-for-Quantum-Information-Science.pdf.

[4] Langione, M., et al., *Where Will Quantum Computers Create Value—and When?* Boston Consulting Group.

https://www.bcg.com/en-es/publications/2019/quantum-computers-create-value-when.aspx, 2019: p. 19.

[5] Mohseni, M., et al., *Commercialize quantum technologies in five years.* Nature News, 2017. **543**(7644): p. 171.

[6] Arute, F., et al., *Quantum supremacy using a programmable superconducting processor.* Nature, 2019. **574**(7779): p. 505-510.

[7] López, M.A. and S.M.D. Silva. *Tecnologías Cuánticas. Una oportunidad transversal e interdisciplinar para la transformación digital y el impacto social*. 2019.

[8] Chong, F.T., D. Franklin, and M. Martonosi, *Programming languages and compiler design for realistic quantum hardware.* Nature, 2017. **549**(7671): p. 180.

[9] Boehm, B., *A view of 20th and 21st century software engineering*, in *Proceedings of the 28th international conference on Software engineering*. 2006, Association for Computing Machinery: Shanghai, China. p. 12–29.

[10] Booch, G., *The History of Software Engineering.* IEEE Software, 2018. **35**(5): p. 108-114.

[11] Fitzgerald, B. and K.-J. Stol, *Continuous software engineering: A roadmap and agenda.* Journal of Systems and Software, 2017. **123**: p. 176-189.

[12] Piattini, M., et al., *The Talavera Manifesto for Quantum Software Engineering and Programming*, in *QANSWER 2020. QuANtum SoftWare Engineering & pRogramming*, M. Piattini, et al., Editors. 2020, CEUR-WS: Talavera de la Reina. p. 1-5.

[13] Peterssen, G., *Quantum technology impact: the necessary workforce for developing quantum software*, in *QANSWER'20 - QuANtum SoftWare Engineering & pRogramming*, M. Piattini, et al., Editors. 2020, CEUR-WS: Talavera de la Reina. p. 6-22.

[14] Hilton, J., *Building The Quantum Workforce Of The Future*, in *Forbes Technology Council*. 2019, Forbes.