The Talavera Manifesto for Quantum Software Engineering and Programming

Mario Piattini

aQuantum by Alarcos Research Group

Guido Peterssen

aQuantum by Alhambra

Ricardo Pérez-Castillo

Facultad de Ciencias Sociales de Talavera de la Reina

Jose Luis Hevia

aQuantum by Alhambra

Manuel A Serrano

Information Systems & Technology Department, UCLM

Guillermo Hernández

m2i formación

Ignacio García Rodríguez de Guzmán

Information Technology & Systems Institute

Claudio Andrés Paradela

aQuantum by Alhambra

Macario Polo

Escuela Superior de Informática de Ciudad Real

Ezequiel Murina

aQuantum by Alhambra

Luis Jiménez

Alarcos Research Group

Juan Carlos Marqueño

Alhambra IT

Ramsés Gallego

Quantum World Association

Jordi Tura

Frank Phillipson

TNO

Juan M. Murillo

University of Extremadura

Alfonso Niño

SciCom Research Group

Moisés Rodríguez

AQCLab

Abstract. This paper presents the Talavera Manifesto for quantum software engineering and programming. This manifesto collects some principles and commitments about the quantum software engineering and programming field, as well as some calls for action. This is the result of the discussion and different viewpoints of academia and industry practitioners who joined at the first International Workshop on QuANtum SoftWare Engineering & pRogramming (QANSWER).

Keywords: Quantum Software Engineering, Quantum Computing, Manifesto, Talavera.

1 Introduction

At the beginning of the last century, the basis of a new physics theory, "quantum mechanics" was established by many exceptional scientists (such as Planck, Einstein, Bohr, Schrödinger, Born, Dirac, De Broglie, Heisenberg, Pauli, etc.); the theory describes the behaviour of nature at subatomic levels (photons, electrons, etc.), and led to the first quantum revolution that allowed researchers and engineers to apply the physical laws that had been set forth in many cutting-edge technologies such as the transistor, GPS, as well as solid-state lighting and lasers, among other advances.

In 1982, Nobel laureate Richard Feynman asked: *What kind of computer are we going to use to simulate physics?* This is how the "second quantum revolution" began, and the idea for quantum computing was thus born. Quantum computers 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) in the effort to provide faster computing speed. Apart from quantum computers, the second quantum revolution includes other key technologies like quantum internet, quantum clocks, quantum sensors, cryptography, etc.

We can already use quantum computers and we expect, in the near future, to take advantage of their huge computation capacity to solve problems which are considered very difficult for today's "classic" computers. This new computing paradigm has a great deal of direct applications, and many other potential ones. For example, applications include, but are not limited to:

- Privacy and cryptography: certification of randomness, authentication.
- Supply chain and logistics: optimization problems in procurement, production and distribution, vehicle routing optimization, etc.
- Chemistry: simulations of complex molecules, discovery of new materials, advanced molecular design, etc.
- Economics, and financial services: Portfolio risk optimization and fraud detection, actual randomness for financial models, simulations and scenario analysis, etc.
- Energy and agriculture: production of ammonia, better distribution of resources, asset degradation modelling, etc.
- Medicine and health: Protein folding and drug discovery, disease detection, non-invasive and high-precision surgeries, targeted drugs design, tailored medicine, improvement of the quality of life, prediction of therapeutic prescriptions, etc.
- Defence and national security programs.

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

Actually, this kind of breakthrough has already occurred several times during the history of software engineering. First, and most notably, with the diffusion of thirdgeneration languages such as COBOL in the 70s, which resulted in structured analysis, design and programming. Then came the advent of relational database management systems in the 80s, which led to Entity-Relationship (ER) analysis and design. Another example is object orientation, emerging in the 90s and leading to object-oriented analysis, design and patterns, which in turn were the seeds for the later model-driven architecture. Most recently, we can point to DevOps, born late on in the first decade of the 21st century, a methodology which has made continuous integration and continuous delivery a must today.

We believe Quantum Software Engineering (QSE) is a necessary contribution to the success of quantum computing. IEEE defines Software Engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, as well as to the study of these approaches; that is, the application of engineering to software. We feel that the time has come to take care of producing quantum software by applying knowledge and lessons learned from the software engineering field. This implies to apply or adapt the existing software engineering processes, methods, techniques, practices and principles for the development of quantum software (or it may imply creating new ones).

We recognize that there is a rapidly-increasing awareness of the need for quantum computing applications, and there is a great desire to produce quantum software in an industrial, controlled manner. However, this is ineffective unless we come to understand how software engineering can help.

2 Principles and commitments

We propose the following initial set of principles and commitments:

- **QSE is agnostic regarding quantum programming languages and technologies.** QSE serves to deliver quantum software with processes and methods that are understandable, controllable and repeatable by broad communities.

- **QSE embraces the coexistence of classical and quantum computing,** and advocates the use of reengineering techniques to integrate new quantum algorithms with the existing classical information systems. Reverse engineering techniques are also needed to parse and abstract quantum program information that is to be integrated into classical programs.

- **QSE supports the management of quantum software development projects**, delivering quantum software that fulfils the initial business goal and requirements, while at the same time ensuring that quality, time, and cost constraints are being properly observed. Methodologies for developing quantum programs must be created or adapted from the existing ones. Effort estimation methods for quantum software development need to be provided as well.

- **QSE considers the evolution of quantum software.** Quantum software should be maintained and evolved from inception to removal. Quantum software evolution must be handled throughout the whole quantum software lifecycle.

- **QSE aims at delivering quantum programs with desirable zero defects.** It is in charge of defining and applying testing and debugging techniques to quantum programs in such a way that most defects can be detected and solved before the program is released.

- **QSE assures the quality of quantum software**. Quality management for both process and product are essential if quantum software with expected quality levels is to be produced. Since we cannot improve what we cannot measure, new metrics for quantum programs and quantum processes have to be developed.

- **QSE promotes quantum software reuse.** QSE can help development teams to share, index, and find quantum software that can be reused. The study of design and architectural patterns for quantum programs must be addressed. It is also necessary to facilitate technical communication, and work on creating libraries of reference examples and application demonstrations.

- **QSE addresses security and privacy by design.** Quantum information systems must be secure and guarantee the privacy of data and of users. QSE offsets out to consider security and privacy from the initial phases of quantum software development, i.e., by design.

- **QSE covers the governance and management of software**. Managers should be aware of the particular processes, organizational structures, principles, policies and frameworks, information, culture, ethics and behaviour, people, skills and competences, as well as the services, infrastructure and applications that are associated with quantum software and that are (or should be) provided by organizations.

3 Call to Action

Each of the following stakeholders can do something right now to get started.

- **Software practitioners**: Try to identify the effects of your quantum projects on technical, economic and organizational contexts. Start asking questions about how to incorporate the principles and commitments into daily practice. Think about the social and individual dimensions. Talk about these issues with your colleagues.

- **Researchers**: Identify research questions in your field that can help us to better understand quantum software engineering. Discuss these with your peers, and think about how your experience in software engineering research could be transferred to the quantum software research field.

- **Educators**: Integrate quantum software engineering in curricula within the existing software engineering degrees and/or courses in this or other disciplines, and clearly specify which competences and skills are required for future quantum software engineers.

- **Government and funding bodies**: Analyse the commitments provided and consider quantum software engineering in research/industrial strategical plans. Provide adequate funding programs to support such plans, and contribute with dissemination of information about these plans.

- **Quantum technology vendors**: Be actively aware of the latest trends in quantum software engineering, and try to make every effort for the commitments mentioned to be fulfilled.

- **Professional associations**: Revise practice, attempting to incorporate principles; acknowledge explicitly the need to consider quantum software engineering as part of professional practice.

- **Customers**: Put your concerns on the table. Ask about these in the next project.

- **Users**: Demand that the software you use embraces all the principles mentioned.