ISSN 2186-7437

NII Shonan Meeting Report

No. 224

Quantum Software Engineering

Shaukat Ali
Paolo Arcaini
Andriy Miranskyy
Jianjun Zhao

July 28–31, 2025

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Quantum Software Engineering

Organizers:
Shaukat Ali (Simula Research Laboratory/
Oslo Metropolitan University, Oslo, Norway)
Paolo Arcaini (National Institute of Informatics, Japan)
Andriy Miranskyy (Toronto Metropolitan University, Canada)
Jianjun Zhao (Kyushu University, Japan)

July 28–31, 2025

Context and goal

Quantum Computing (QC) aims to solve many complex problems efficiently once sufficiently powerful quantum computers have been built. QC is becoming increasingly popular, with many companies (e.g., IQM, IBM, Google, Microsoft) now building their quantum computers using various hardware technologies. These companies are continuously making progress in increasing computational power while reducing noise, which is severely hindering the practical applications of QC. Such noise is inherent in today's quantum computers due to immature technology, which reduces the reliability of the computations performed by quantum computers.

Like classical computing, quantum software programs quantum computers. There have been efforts to make quantum programming languages such as Qiskit, Q#, Cirq, Scaffold, and Quipper, enabling quantum computers to be programmed for specific tasks. However, the current practice of programming remains at the quantum circuit level, which is too low-level and closer to hardware, and only people with specialized backgrounds can program quantum computers. Increasing efforts are now being made to bring novel abstractions to quantum circuits, supporting modularity, etc.

In the QC realm, Quantum software engineering (QSE) is an emerging area of research that focuses on developing scalable and cost-effective solutions to build quantum computing software that can be executed on real quantum computers. Given that it is a new area of research, there is still no agreed software development life cycle or development processes. However, researchers have begun to explore various possible phases of QSE by taking inspiration from classical computing.

Within the QSE, some areas have received more attention than others. For instance, quantum software testing has gained more attention than quantum software requirements engineering. However, most existing works within QSE are preliminary, requiring more novel research.

Details of the meeting

Currently, most quantum software is developed in a hybrid, classical-quantum fashion, i.e., both classical and quantum code is used. This means that certain parts of the code are executed on classical computers, whereas others are delegated to quantum computers for execution. This adds an additional view to consider while developing QSE solutions.

This Shonan meeting was focused on discussing various aspects (challenges, ideas, research roadmaps) of QSE, including, but not limited to:

- What are the possible phases of QSE (e.g., requirements engineering, modeling, testing, debugging, and repairing)? How do those phases differ from the classical phases?
- Is there any need for quantum software requirements engineering, or are classical methods and approaches sufficient?
- How can the classical-quantum split be handled across various phases of QSE?
- Given the inherent noise in the near-term quantum computers, shall QSE focus on specific solutions to deal with noise or focus exclusively on noise-free solutions?
- How can we develop best practices and solutions for developing quantum software? Are classical solutions, such as agile methods, still applicable, or are they valuable for quantum software?
- What are the strengths and weaknesses of benchmarks in the context of QSE? How can the community provide a systematic way of developing such benchmarks for QSE?

We invited researchers from academia and practitioners working in various aspects of QSE, such as quantum software testing and debugging, quantum software modeling, quantum software execution and optimization, and the application of classical AI algorithms for developing QSE solutions.

Introduction

This Shonan meeting brought together researchers and practitioners working on quantum software development, both for applications and across the quantum software stack, as well as on QSE methods throughout different phases of the QSE lifecycle. It aimed to discuss challenges in building, verifying, and validating quantum software, as well as other related aspects such as education.

In particular, the participants have been active in QSE for several years, enabling us to gather leading expertise in the field in one meeting room, with focused discussions on various challenges, practical aspects, and the future outlook of QSE.

Overview of the meeting

The meeting began with an introduction by the organizers, explaining the overall scope, schedule, potential outcomes, and other practical matters. This was followed by a brief introduction of each participant—approximately three minutes each, with or without slides to present themselves, outline the topics they are working on, and share their expectations from the seminar. More detailed presentations from selected participants followed this, each lasting approximately 15 minutes. The titles and abstracts of these talks are presented in the "Overview of Talks" section. The result of these talks was the identification of key topics to be discussed, as well as the formation of groups for focused discussions for the rest of the seminar. These topics were identified for discussions:

(i) Abstraction

(ii) Testing

(iii) Verification

In the rest of the report, we first provide the meeting schedule, followed by the abstracts of the talks, and finally a summary of the discussions from the groups.

Meeting Schedule

Check-in Day: July 27, 2025 (Sun)

- Welcome Banquet

Day 1: July 28, 2025 (Mon)

- 9:00-9:30: Opening
- 9:30-10:30: Introductions (20 people)
- 10:30-11:00 Break
- 11:00-12:00: Introductions (5 people) and 3 presentations
- 12:00-13:30 Lunch
- 13:30-15:30 6 presentations
- 15:30-16:00 Break
- 16:00-17:30 Group formation
- 18:00- Dinner

Day 2: July 29, 2025 (Tue)

- 9:00-9:25 Introductions (3 people) and 1 presentation
- 9:25-10:30 Group Work
- 10:30-11:00 Break
- 11:00-12:00 Group Work
- 12:00-13:30 Lunch
- 13:30-14:00 2 presentations
- 14:00-15:30 Group Work
- 15:30-16:00 Break
- 16:00-17:30 Group Presentations + Discussion
- 18:00- Dinner

Day 3: July 30, 2025 (Wed)

- 9:00-10:30 Group Work
- 10:30-11:00 Break
- 11:00-11:40 Group Work
- 11:40-13:10 Lunch
- 13:10-13:30 Photo shoot

- 13:30-17:30 Free time
- 17:30:- Banquet

Day 4: July 31, 2025 (Thu)

- 09:00- 09:30 Group work
- 9:30-10:30 Group presentations and discussion on the next steps
- 10:30-11:00 Break
- 11:00-12:00 Wrap up

Overview of Talks

Automata-based quantum program verification

Yu-Fang Chen, Institute of Information Science, Academia Sinica Taiwan

I present an approach that enables fully automatic verification of quantum programs. The talk begins with a brief demo illustrating how to use AutoQ to verify simple quantum algorithms. We then introduce the semantics of quantum programs using a novel perspective: modeling quantum states as decision trees, and consequently, sets of quantum states as tree automata. Building on this perspective, we introduce a fully automatic verification framework in which both preconditions and postconditions are represented as tree automata.

Local Reasoning about Probabilistic Behaviour for Classical-Quantum Programs

Yuxin Deng, Shanghai University of Finance and Economics, China

Verifying the functional correctness of programs with both classical and quantum constructs is a challenging task. The presence of probabilistic behaviour entailed by quantum measurements and unbounded while loops complicate the verification task greatly. We propose a new quantum Hoare logic for local reasoning about probabilistic behaviour by introducing distribution formulas to specify probabilistic properties. We show that the proof rules in the logic are sound with respect to a denotational semantics. To demonstrate the effectiveness of the logic, we formally verify the correctness of non-trivial quantum algorithms, including the HHL and Shor's algorithms.

Formal Verification of Variational Quantum Algorithms

Alessandra Di Pierro, University of Verona, Italy

Variational quantum circuits (VQCs) are a central component of many quantum machine learning algorithms, offering a hybrid quantum-classical framework that, under certain aspects, can be considered similar to classical deep neural networks. A shared aspect is, for instance, their vulnerability to adversarial inputs—small perturbations that can lead to incorrect predictions. While formal verification techniques have been extensively developed for classical models, no comparable framework exists for certifying the robustness of VQCs. Here, we present the first in-depth theoretical and practical study of the formal verification problem for VQCs. Inspired by abstract interpretation methods used in deep learning, we analyze the applicability and limitations of interval-based reachability techniques in the quantum setting. We show that quantum-specific aspects, such as state normalization, introduce inter-variable dependencies that challenge existing approaches. We investigate these issues by introducing a novel semantic framework based on abstract interpretation, where the verification problem for VQCs can be formally defined, and its complexity analyzed.

Verification challenges in quantum computing

Ross Duncan, University of Strathclyde/Quantinuum, UK

Verification usually means to guarantee that a computational system is correct with respect to some specification. Can this definition survive in the context of quantum computing? What are the distinctively quantum obstacles to verification? What parts of the system can be verified? What notions of correctness are helpful? What properties of realistic systems can be verified? What *should* we verify? I will survey the field and give some perspective on the challenges and opportunities for automated verification of quantum software. Spoiler: it's the compiler.

SoQ: When Services Meets Quantum, Jose Garcia-Alonso

Jose Garcia-Alonso, University of Extremadura, Spain

In this talk, we explored how quantum computing is evolving into a transformative technology capable of addressing problems beyond classical computational limits. As the field matures, integrating quantum capabilities into classical software ecosystems becomes essential. We introduced the concept of Service-Oriented Quantum, a paradigm inspired by Service-Oriented Computing, which promotes modular, scalable, and interoperable deployment of quantum services. The presentation covered how this approach supports hybrid quantum-classical workflows, simplifies access to quantum hardware, and lowers adoption barriers for industry. We also discussed the significant challenges that must be addressed to realize this vision, ranging from platform interoperability and orchestration of hybrid computations to pricing models and workforce development. Real-world case studies were shared to illustrate current progress, and we concluded by outlining a research agenda aimed at building a robust Quantum-as-a-Service ecosystem. The goal is to accelerate the transition from experimental quantum systems to practical, scalable solutions embedded within modern software infrastructures.

Free Quantum Computing

Chris Heunen, University of Edinburgh, UK

Quantum computing improves substantially on known classical algorithms for various important problems, but the nature of the relationship between quantum and classical computing is not yet fully understood. This relationship can be clarified by free models, that add to classical computing just enough physical principles to represent quantum computing and no more. I will discuss an axiomatisation of quantum computing that replaces the standard continuous postulates with a small number of discrete equations, as well as a free model that replaces the standard linear-algebraic model with a category-theoretical one. The axioms and model are based on reversible classical computing, and isolate quantum advantage in the ability to take certain well-behaved square roots. The free model may be interpreted as a programming language for quantum computers, that has the same expressivity and computational universality as the standard model, but additionally allows automated verification and reasoning.

Model-based / Search-based Circuit Synthesis

Stefan Klikovits, Johannes Kepler Universität Linz, Austria

This presentation introduces recent advances in Quantum Circuit Synthesis (QCS) conducted at the Institute of Business Informatics – Software Engineering (BISE), JKU Linz, as part of broader efforts in automated Quantum Software Engineering (QSE). QCS is framed as a search problem: given a desired quantum behavior, the goal is to construct a quantum circuit that satisfies functional and hardware-specific constraints. Leveraging principles from Search-Based Software Engineering (SBSE), the work applies Genetic Programming (GP) and hybrid optimization strategies to explore the large and error-prone design space. In particular, we demonstrated the successful use of hybrid search schemes that combine evolutionary algorithms with local parameter optimization (e.g., Nelder-Mead), resulting in improved circuit diversity, faster convergence, and better debugging and optimization capabilities across multiple benchmarks. These approaches have shown consistent performance gains over non-hybrid baselines.

Current research focuses on addressing three core challenges in QCS: the vastness of the search space, the high cost of quantum circuit simulation, and the difficulty of the fitness landscape. To mitigate these, we investigate strategies such as intelligent seeding of initial populations, caching of redundant simulations, and commutativity-aware circuit representations that exploit equivalences in gate orderings. Additional work explores the use of surrogate models to estimate circuit fitness more efficiently and the design of a general-purpose QCS testbed for systematic benchmarking. Altogether, the findings suggest that hybrid and search-informed strategies offer promising pathways toward scalable, automated quantum program synthesis.

Quantum Intuition Over Quantum Formalism: Teaching Quantum Concepts to CS Students Without the Physics Prerequisites

Anila Mjeda, Munster Technological University, Ireland

Quantum computing presents a distinct pedagogical challenge when teaching computer science students, who typically lack the background in quantum mechanics and advanced mathematics required to engage with quantum formalism. This talk introduced an analogy-driven teaching approach designed to address this challenge by fostering quantum intuition, rather than full mathematical fluency, in learners. The goal is to integrate quantum computing into computer science curricula in a way that promotes understanding without requiring prior exposure to quantum physics. Grounded in cognitive load theory and analogical reasoning, the approach reframes key quantum concepts, such as superposition, entanglement, and Hamiltonian dynamics, using metaphors drawn from computer science. For example, the quantum Hamiltonian is recast as an operating system's task scheduler, enabling learners to intuitively grasp time evolution and energy optimization without delving into Hilbert spaces or differential equations. This work responds to a deeper contextual challenge:

quantum concepts often conflict with CS students' prior training in deterministic, logical, and discrete systems. These students arrive with strong computational schemas, powerful tools that can both aid and hinder their understanding of inherently probabilistic, non-intuitive phenomena. The proposed analogical method offers "cognitive on-ramps," epistemological scaffolds that foster intuitive pattern recognition, reduce abstraction anxiety, and facilitate knowledge transfer across disciplinary boundaries. Importantly, the talk also examined the limitations and pedagogical risks of analogies. While they are powerful teaching tools, analogies are not substitutes for formal understanding and must be dynamically recalibrated as students progress. Without such recalibration, there is a risk of reinforcing misconceptions or oversimplifications, particularly when analogies are created spontaneously without critical reflection, a practice observed in prior educational research. This approach seeks to bridge disciplinary divides, what we call Quantum Babel, the breakdown of understanding that occurs when learners confront conflicting notational and conceptual systems across fields, and to cultivate a generation of quantum-literate software engineers. Ultimately, this work contributes to the emerging field of quantum software engineering education, advocating for structured, reusable, and pedagogically sound analogy-based materials that enable learners to engage with quantum computing meaningfully and intuitively, lowering the entry barrier to the discipline.

Noise-aware property-based testing

Mohammadreza Mousavi, King's College London, UK

The Oracle Problem is a major issue in software testing: for complex systems, it is difficult to come up with oracles that can tell apart passing test-case executions from failing ones. This problem is exacerbated in quantum software. Property-based testing is an effective technique to address this problem by specifying high-level properties of the system under test and letting the test infrastructure generate tests and check the outputs against the specified properties. In this talk, we provide an overview of our line of work in adopting the idea of property-based testing in the domain of quantum software. We review our initial effort in developing QSharpCheck, a property-based testing framework for Q#, followed by QuCheck, a more substantial property-based testing framework for Qiskit, and finally the integration of QuCheck with a machine-learning-based error mitigation framework called Qoin.

Programming Quantum Computers

Juan Manuel Murillo, University of Extremadura / SCORE Excellence Unit, University of Seville, Spain

Computer programming requires an intellectual process on the part of the programmer that involves analysing a problem, devising a solution strategy that uses the computer's resources to implement that solution. Such a strategy must be expressible in the programming language chosen to program the computer.

Although this intellectual process is broadly similar whether one is programming classical or quantum computers, there are key differences between the two

that shape how strategies are composed. One of the most significant differences is the lack of suitable abstraction mechanisms in quantum programming languages. Motivated by this, the talk presents several proposals for building appropriate abstractions to support the development of quantum programs. These proposals range from considering n-qubit quantum registers that encode integers with n-bit precision—along with operations provided in the form of oracles—to the concept of a Locus, a place in a quantum program where functionality is intended to be constructed, with its implementation deferred to a later point in the program.
Some references:

- J. Sanchez-Rivero, D. Talavan, J. Garcia-Alonso, A. Ruiz-Cortes and J. M. Murillo. Automatic Generation of an Efficient Less-Than Oracle for Quantum Amplitude Amplification. In 2023 IEEE/ACM 4th International Workshop on Quantum Software Engineering (Q-SE), Melbourne, Australia, 2023, pp. 26-33, `https://ieeexplore.ieee.org/document/10189119`
- Sanchez-Rivero, J., Talaván, D., Garcia-Alonso, J., Ruiz-Cortés, A., Murillo, J.M. (2023). Operating with Quantum Integers: An Efficient 'Multiples of' Oracle. In: Aiello, M., Barzen, J., Dustdar, S., Leymann, F. (eds) Service-Oriented Computing. SummerSOC 2023. Communications in Computer and Information Science, vol 1847. Springer, Cham. `https://doi.org/10.1007/978-3-031-45728-9_7`
- J. M. Murillo. Composable Quantum Oracles for Shifting Quantum Circuits Abstraction Level. 2024 IEEE International Conference on Quantum Software (QSW), Shenzhen, China, 2024, pp. 9-11, `https://ieeexplore.ieee.org/document/10646506/`
- Sanchez-Rivero, J., Talaván, D., Garcia-Alonso, J., Ruiz-Cortés, A., Murillo, J.M. (2024). Some Initial Guidelines for Building Reusable Quantum Oracles. In: Monti, F., et al. Service-Oriented Computing – ICSOC 2023 Workshops. ICSOC 2023. Lecture Notes in Computer Science, vol 14518. Springer, Singapore. `https://doi.org/10.1007/978-981-97-0989-2_16`
- Juan Manuel Murillo, Jose Garcia-Alonso, Enrique Moguel, Johanna Barzen, Frank Leymann, Shaukat Ali, Tao Yue, Paolo Arcaini, Ricardo Pérez-Castillo, Ignacio García-Rodríguez de Guzmán, Mario Piattini, Antonio Ruiz-Cortés, Antonio Brogi, Jianjun Zhao, Andriy Miranskyy, and Manuel Wimmer. 2025. Quantum Software Engineering: Roadmap and Challenges Ahead. ACM Trans. Softw. Eng. Methodol. 34, 5, Article 154 (June 2025), 48 pages. `https://doi.org/10.1145/3712002`
- Javier Zayas Gallardo , Francisco Chicano, Carlos Canal, Juan Manuel Murillo. Locus: A Proposal for Quantum Software Composition. QP Workshop at Programming 2025 Conference (Proceedings under publication process).

Quantum recursive programming

Mingsheng Ying, University of Technology Sydney, Australia

In this talk, we introduce a new scheme of quantum recursive programming with a quantum if-statement. A simple programming language for supporting this kind of quantum recursion is defined, and its semantics is formally described. A series of examples is presented to show that some quantum algorithms can be elegantly written as quantum recursive programs. At the end, we will briefly discuss verification and compilation of quantum recursive programs (see `https://arxiv.org/abs/2404.05934`; `https://dl.acm.org/doi/10.1145/3729283`).

Distinguishing Quantum Software Bugs from Hardware Noise: A Statistical Approach

Lei Zhang, University of Maryland, USA

Quantum computing in the Noisy Intermediate-Scale Quantum (NISQ) era presents significant challenges in differentiating quantum software bugs from hardware noise. Traditional debugging techniques from classical software engineering cannot directly resolve this issue due to the inherently stochastic nature of quantum computation mixed with noise from NISQ computers. To address this gap, we propose a statistical approach leveraging probabilistic metrics to differentiate between quantum software bugs and hardware noise. We evaluate our methodology empirically using well-known quantum algorithms, including Grover's algorithm, Deutsch-Jozsa algorithm, and Simon's algorithm. Experimental results demonstrate the efficacy and practical applicability of our approach, providing quantum software developers with a reliable analytical tool to identify and classify unexpected behavior in quantum programs.

Focus group on "Abstraction"

To aid code maintenance, ease of training, interfacing quantum with other systems, algorithm discovery, and many more desirable software engineering aspects of quantum computing, more abstraction is needed in methods to specify and handle quantum computations. There are at least three problems in finding these abstractions:

- They should capture what sets quantum advantage or quantum utility apart from classical computing, but nobody knows how to isolate this.
- They should be elegant and composable, but it is unclear how to measure such properties. Additionally, even having such measures doesn't help in identifying good abstractions.
- They should extend current software engineering approaches, but these are not native to quantum computing, lack a large enough body of data in the quantum space, and are in danger of answering irrelevant questions.

A technical challenge that can help go toward solving these problems is the following:

- Given a big quantum circuit, how can we identify local subparts that are quantum and parts that are classical?

But the big underlying question is:

- How does one think quantumly?

The current state of science has no one unified answer. Towards finding one, we plan to survey the quantum community and aggregate all fragmented pieces of answers.

Focus group on "Testing"

The working group met to explore the emerging challenges of quantum software testing [7, 6, 2] as quantum programs advance towards scales of 100–200 qubits and beyond, where classical simulation can no longer serve as a viable validation method. Discussions covered a range of topics, including dynamic testing methodologies, property definition, hybrid classical-quantum systems, interoperability, and reproducibility, framed as research directions with the potential to shape the next phase of quantum software engineering.

A central theme was the construction of test oracles for large-scale quantum programs. Without the ability to rely on exhaustive simulation or deterministic validation, new strategies are required. Possible directions include tolerance-based or "fuzzy" comparisons of quantum states and operations to accommodate noise and probabilistic output distributions; frameworks for approximate oracles that produce computationally feasible estimates of properties; the use of quantum algorithms, such as search techniques, to synthesise property-specific test circuits; and metamorphic testing [1, 4]. Approaches for noise-resilient oracle evaluation, where thresholds adapt dynamically to the performance of the target hardware, were also identified as an important research path.

System-to-component property translation was also identified as a critical research challenge. Scalable verification of complex quantum systems depends on translating high-level, system-wide properties into component-level specifications that can be tested in isolation. While assume–guarantee reasoning [10] offers a conceptual framework for such decomposition, its adaptation to quantum software raises distinctive challenges: mapping system contracts into quantum circuit-level constraints, including pre- and post-conditions in probabilistic domains; developing bottom-up approaches where component properties demonstrably satisfy system requirements under composition; and defining abstraction layers that preserve behavioural guarantees across heterogeneous subsystems in hybrid quantum–classical architectures.

The topic of property-based testing [5, 9] was also discussed in this context. This includes formalising high-level system properties, generating representative test inputs, and defining meaningful adequacy measures in probabilistic domains. Work in this area will need to address both the probabilistic nature of quantum outputs and the difficulty of full-state verification at scale.

Collaboration with domain experts was recognised as essential to ensuring the precision and relevance of quantum testing properties. Many application areas, such as chemistry, physics, and finance, depend on domain-specific constraints and properties that require specialist knowledge to define and interpret. Opportunities include embedding domain-level invariants into formal property definitions, designing interfaces that allow domain experts to specify properties without programming expertise, and adapting collaboration models from classical software engineering research. AI-assisted approaches to requirements engineering were noted as potentially valuable for supporting property formulation.

In testing tools and automation frameworks, the need for toolchains that can automate the translation from high-level property definitions to executable test artefacts was emphasised. Desirable capabilities include code generation for test circuits, translation from property statements to low-level assertions and circuit constructs, and test explanation tools that connect observed failures to probable root causes. Quantum algorithm-based test generation could accelerate the exploration of large input spaces and improve coverage.

The group also discussed coverage and adequacy metrics [3, 8] that better reflect the probabilistic and high-dimensional nature of quantum programs. Suggested directions include moving from binary coverage to diversity metrics and entropy-based measures [3], modelling adequacy over continuous coverage spaces, extending mutation testing to incorporate realistic fault models, and addressing the heavy-tailed nature of measurement costs, where some faults require disproportionately many shots to detect.

Addressing scale limitations and hardware constraints will be central to enabling testing for large-scale systems. Strategies may include execution budgeting and scheduling to maximise scarce hardware access, abstractions that move beyond qubit-by-qubit reasoning, and optimisation of data preparation pipelines for large classical inputs. Lessons from other engineering domains, where test execution is treated as a constrained physical resource, could be adapted for quantum contexts. Analogies were drawn to staged validation pipelines in the automotive industry, where software-in-the-loop and hardware-in-the-loop approaches complement each other.

References

[1] R. Abreu, J. a. P. Fernandes, L. Llana, and G. Tavares. Metamorphic testing of oracle quantum programs. In *Proceedings of the 3rd International Workshop on Quantum Software Engineering*, Q-SE '22, pages 16–23, New York, NY, USA, 2023. Association for Computing Machinery.

[2] S. Ali, J. Barzen, A. Delgado, H. A. Müller, and J. M. Murillo. Quantum Software Engineering (Dagstuhl Seminar 24512). *Dagstuhl Reports*, 14(12):63–84, 2025.

[3] J. Campos, R. Abreu, G. Fraser, and M. d'Amorim. Entropy-based test generation for improved fault localization. In *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering*, ASE '13, pages 257–267. IEEE Press, 2013.

[4] N. Costa, J. a. P. Fernandes, and R. Abreu. Asserting the correctness of shor implementations using metamorphic testing. In *Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering*, QP4SE 2022, page 32–36, New York, NY, USA, 2022. Association for Computing Machinery.

[5] S. Honarvar, M. R. Mousavi, and R. Nagarajan. Property-based testing of quantum programs in Q#. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20, pages 430–435, New York, NY, USA, 2020. Association for Computing Machinery.

[6] N. C. Leite Ramalho, H. Amario de Souza, and M. Lordello Chaim. Testing and debugging quantum programs: The road to 2030. *ACM Trans. Softw. Eng. Methodol.*, 34(5), May 2025.

[7] 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.*, 34(5), May 2025.

[8] A. Perez, R. Abreu, and A. van Deursen. A test-suite diagnosability metric for spectrum-based fault localization approaches. In *Proceedings of the 39th International Conference on Software Engineering*, ICSE '17, pages 654–664. IEEE Press, 2017.

[9] G. Pontolillo, M. R. Mousavi, and M. Grzesiuk. QuCheck: A property-based testing framework for quantum programs in Qiskit. *CoRR*, abs/2503.22641, 2025.

[10] M. Yamaguchi, N. Yoshioka, and F. Ishikawa. Practical design by contract framework for quantum applications. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, FSE Companion '25, page 1699–1709, New York, NY, USA, 2025. Association for Computing Machinery.

Focus group on “Verification”

Ensuring the correctness of quantum programs, particularly those integrating classical and quantum components, is a critical challenge in quantum software development. Traditional formal verification techniques, such as model checking, Hoare logic, and abstract interpretation, have been adapted to the quantum setting, yielding a growing body of research on quantum formal methods. However, despite these advances, their practical application remains limited due to the inherent complexity of quantum systems. To bridge this gap, we propose to focus on developing verification techniques tailored to key components of the quantum software stack, including intermediate representations, error correction codes, compilers, and compilation validation.

Hoare logic

A major obstacle in quantum program verification is the scalability of existing methods. To address this, we propose to use a new quantum Hoare logic to make it more tractable for real-world applications. A better design of quantum assertion languages is also needed. In our discussions, we explored the idea of designing an assertion language capable of expressing hybrid properties, meaning properties that involve both classical and quantum aspects simultaneously. We considered an intuitive syntax with high-level primitives and a set of useful atomic propositions to support reasoning about quantum programs and algorithms. Another topic was the compilation of such an assertion language into representations used by verification tools, such as AutoQ or CoqQ.

Bisimulation checking

Additionally, we explore bisimulation checking methods to verify equivalence between quantum programs, ensuring correctness across different levels of abstraction, such as source code and compiled output. Besides verification, we also consider the potential use of this relation in quantum program/circuit optimization (by merging equivalent nodes in control flow graphs).

Coabstract interpretation

We discussed the need for tractable foil theories of quantum mechanics to avoid the need to simulate the evolution of quantum memory. One candidate’s ”coabstract interpretation” was the relational models pioneered by Robert Spekkens, etc. The idea is to replace the standard Hilbert space model with finite sets and relations. However, initial efforts did not produce a functorial interpretation of FDHilb in Rel. We will keep looking!

Session types for DQC

Another important direction of scalable quantum computing is distributed quantum computation (DQC). In such computation, entangled states (or quantum communication) between computational nodes are crucial resources to realize remote quantum operations. In our idea, (multiparty) session types may be a solution for modeling and analyzing the behaviours of DQC. Further direction is to enable hardware-aware analysis for distributed quantum programs with session types. Designing a new ISA (or giving a good abstraction) to describe distributed quantum programs is also essential.

This research line is driven by the need for robust verification tools that can keep pace with the rapid advancement of quantum computing. We seek to strengthen the foundations of quantum software correctness, ultimately supporting the development of dependable quantum algorithms and applications in areas such as cryptography, optimization, and simulation.

List of Participants

- Rui Abreu, University of Porto, Portugal
- Shaukat Ali, Simula Research Laboratory and Oslo Metropolitan University, Norway
- Paolo Arcaini, National Institute of Informatics, Japan
- Jose Campos, University of Porto, Portugal
- Yu-Fang Chen, Institute of Information Science, Academia Sinica, Taiwan
- Yuxin Deng, Shanghai University of Finance and Economics, China
- Alessandra Di Pierro, University of Verona, Italy
- Ross Duncan, University of Strathclyde/Quantinuum, UK
- Michael Felderer, German Aerospace Center (DLR) / University of Cologne, Germany
- Jose Garcia-Alonso, University of Extremadura, Spain
- Claude Gravel, Toronto Metropolitan University, Canada
- Chris Heunen, University of Edinburgh, UK
- Fuyuki Ishikawa, National Institute of Informatics, Japan
- Stefan Klikovits, Johannes Kepler Universität Linz, Austria
- Lei Ma, The University of Tokyo, Japan
- Wolfgang Mauerer, Technical University of Applied Science, Regensburg / Siemens AG, Germany
- Andriy Miranskyy, Toronto Metropolitan University, Canada
- Anila Mjeda, Munster Technological University, Ireland
- Mohammad Reza Mousavi, King's College London, UK
- Hausi A. Müller, University of Victoria, Canada
- Juan Manuel Murillo, University of Extremadura / SCORE Excellence Unit, University of Seville, Spain
- Shinobu Saito, NTT, Japan
- Davide Taibi, University of Oulu, Finland
- Ryo Wakizaka, Kyoto University, Japan
- Masaomi Yamaguchi, Fujitsu Limited, Japan
- Mingsheng Ying, University of Technology Sydney, Australia
- Nobuko Yoshida, University of Oxford, UK
- Lei Zhang, University of Maryland, USA
- Jianjun Zhao, Kyushu University, Japan

Group photos

Figure 1: Group photos on July 30, 2025

Yosegaki

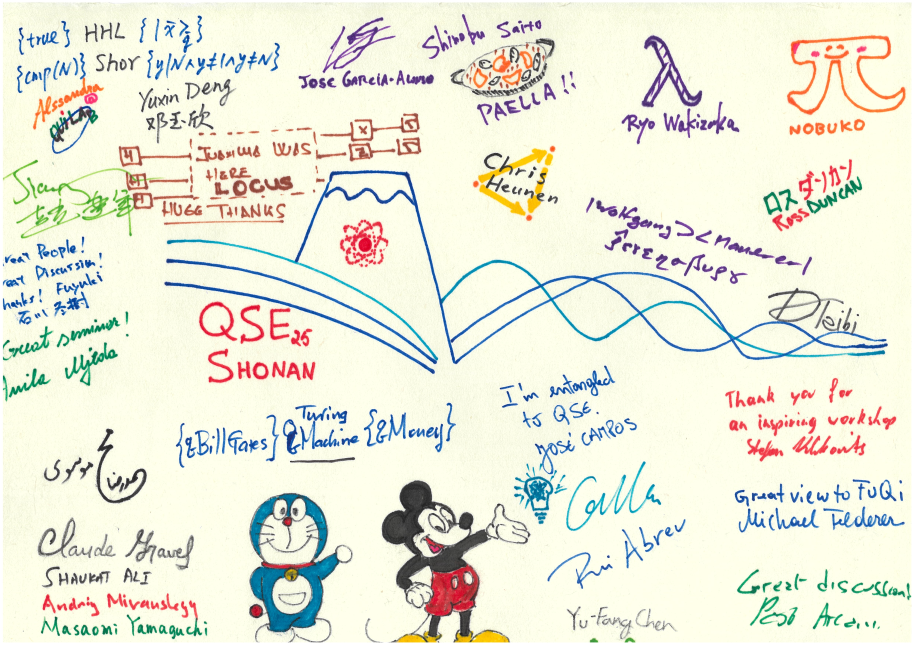

Figure 2: Yosegaki