



Quantum Computing

Jose Luis Hevia, Guido Peterssen, Christof Ebert, and Mario Piattini

From the Editor

Quantum computing has become a reality. Quantum computers are available to everybody via cloud service or simulation. Toolkits are available that invite practitioners to start their own quantum software projects and thus get used to this novel technology. In this article we evaluate technologies to help developers to start their own quantum software business. Practical guidance is provided from our own quantum technology projects. I look forward to hearing from you about this column and the technologies that matter most for your work.—*Christof Ebert*

HAVE YOU EVER tried to retrieve that forgotten key code for your suitcase? After one year without traveling, many of us found themselves having forgotten the combination and manually trying all permutations. The same situation, but more complex, would be to systematically try identifying that forgotten access code for an online app that you had not used for a while. Cyberattackers are doing exactly this, of course, at high speed and with increasing computing performance. The recommended security key length is thus getting longer by the year. Yet, the stepwise process to achieve this is tedious or consumes lots of computing power. Now imagine that all of these possible states could be tried in

a single step. This would be good for your own number lock, but frightening for our security infrastructure.

The promise of quantum computing is to vastly accelerate such complex algorithms.¹ Today, even supercomputers fail on high algorithmic complexity because many algorithms still work in sequences which build on results of a previous step. Of course, they use massively parallel hardware and algorithms, but networking imposes limits as does the memory needed to hold the myriad combinations of real-world problems.

Quantum computing has left the research domain and is ready to use in industry practice. It will rapidly advance industry applications in fields such as data science, pattern recognition, and cybersecurity.¹⁻⁴ Yet the actual software development for quantum computing is hampered

by a lack of appropriate methods and insufficiently scalable technology.²

Quantum Computing

Quantum computers use atomic-scale effects, such as electron spin, as underlying information.¹ Quantum computing uses what are called *quantum bits* or *qubits*: bits that are held in superposition and use quantum principles to complete calculations. A binary digit is always in one of two definite states, that is, either zero or one. Qubits are in a superposition of these classic binary states of zero and one.

Superposition is the ability of qubits to be in more than one physical state at a time, which allows us to parallelize combinations. Multiple qubits can also become entangled. If you measure the state of one qubit, the outcome of measuring the other

Digital Object Identifier 10.1109/MS.2021.3087755
Date of current version: 20 August 2021

qubit is correlated in some way with the first, even if the two qubits are far apart. Superposition and entanglement are used together for quantum

large number of logical states. Interference is used in that the incorrect answers of the specific problem destructively interfere and no longer

in which the quantum algorithm is simulated on classical hardware (CPUs), and real quantum computers, with quantum processing units (QPUs), in which qubits are built using a wide variety of technologies: ion-trap, superconducting, and photonic methods, among others.^{1,4,5}

Within quantum computers we find mainly two categories (Figure 1): quantum annealing computers, such as the D-Wave computers (suitable for running optimization problems since finding the largest or smallest value of an indicator can translate into minimizing the energy of a system), or gate-based computers, such as those from Google, IBM, Rigetti, IonQ, and Honeywell.⁵ All of this strongly affects the way in which applications are developed. Indeed, there are also two approaches: those based on building binary quadratic models for solving a problem or those based on the

Actual software development for quantum computing is hampered by a lack of appropriate methods and insufficiently scalable technology.

computation. Yet these effects also create practical challenges with real quantum computers: they require sophisticated lab environments; also, the information might decay when the state of the system is captured.

How do we deduce the result from these superimposed states? Many quantum algorithms first create superpositions of an exponentially

appear in the final output, leaving behind only the correct answer.

From an algorithmic point of view, quantum computing can solve problems of higher complexity than classical computing—faster and also with cost and energy savings. The first quantum computers were built in the late 1990s. For practical usage, we distinguish quantum simulators,

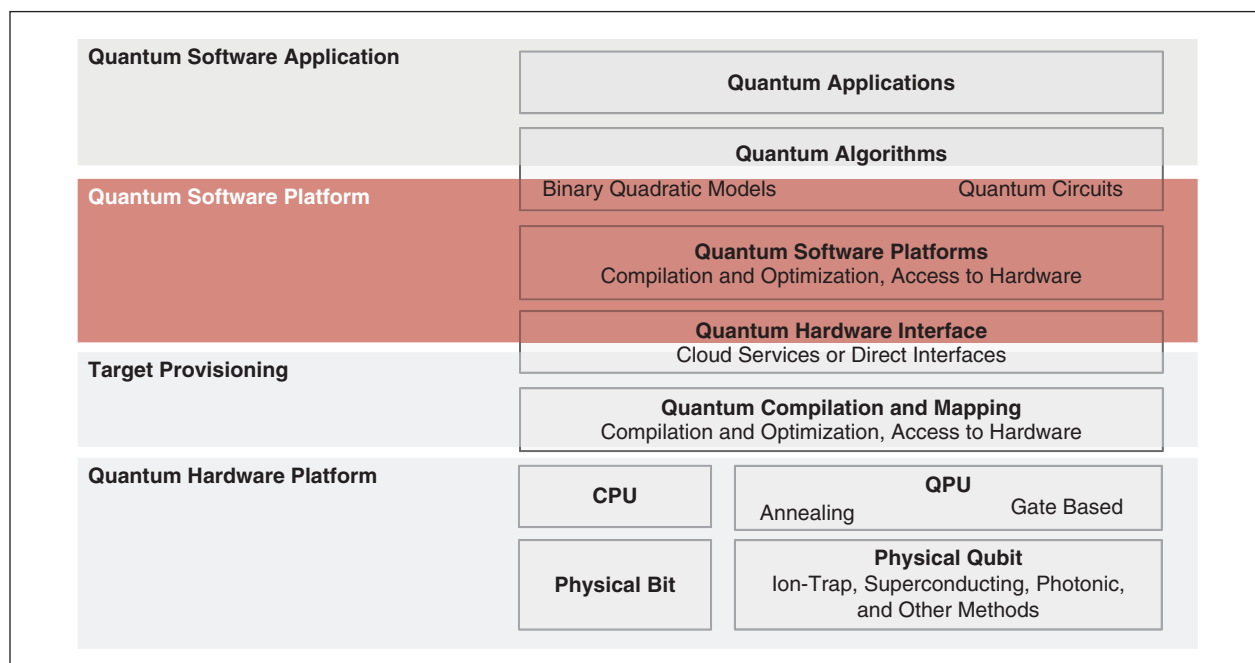


FIGURE 1. The quantum software and hardware stack.

construction of quantum circuits based on gates.

Today quantum hardware vendors such as IBM, Rigetti, and Google deliver some 100 qubits on a laboratory scale.^{4,5} This is impressive and demonstrates how fast the technology is evolving, but it is not yet sufficient to run actual software applications. Therefore, the quantum applications that we envisage today are separating the actual hardware stack from the software tier (Figure 1).

We expect that quantum computers will scale up at a pace like that of Moore's law. For the short term, a quantum network accessible by cloud services could show results from a software perspective. By connecting individual quantum devices, a quantum supercomputer could be created. A bigger step forward is a quantum network based on entangled qubits for fast information exchange. Cybersecurity is an obvious application domain of such a network to facilitate quantum key distribution with a cryptography protocol relying on interlinked quantum particles.

Quantum Computing Applications

Applications of quantum computing are manifold. Because of the extreme parallelism of quantum algorithms, some massive parallel challenges, such as data science and pattern recognition, can be accelerated by quantum computing.

Examples include, for example, identifying the optimal route of a delivery car or fleet of trucks to save on time and fuel costs. Or an investment company may need to balance its portfolio risk with numerous possible combinations of shares with different individual performances and related cluster risks. Pharmaceutical researchers need to simulate molecules to better

understand drug interactions, even if they do it using only known constraints and reported deficiencies. The latter is our case study in "QHealth."

On the dark side, massive parallel algorithms will also facilitate hacking any current cryptographic key with much less effort than is currently assumed. Shor's algorithm can factor large prime numbers down into two smaller ones.⁶ This is a very useful property for breaking encryption since the Rivest-Shamir-Adleman (RSA) system of encryption depends

algorithm can break elliptic curve cryptography even more easily than you might break RSA.⁶ As Grover's algorithm also accelerates mining, one further application is the evolution in bitcoin mining from GPUs, field-programmable gate arrays, and application-specified integrated circuits toward quantum computers.

Novel quantum computation protocols are currently developed toward enhanced security. In such protocols, the client will encrypt its data so that the host or cloud computer cannot

Because of the extreme parallelism of quantum algorithms, some massive parallel challenges, such as data science and pattern recognition, can be accelerated by quantum computing.

on factoring large prime numbers. Already today, major cybersecurity algorithms are anticipating such quantum hacking and vastly enhancing the key length. Postquantum cryptography has started to be researched with encryption techniques that would operate and not be broken even with much larger quantum computers. Most of the encryption systems in modern cryptocurrencies are built on elliptic curve cryptography rather than RSA because elliptic curves are harder to crack than RSA—at least by classical computers. Current blockchain-based e-currencies thus use signatures that require the Elliptic Curve Digital Signature Algorithm (ECDSA). However, quantum computers seem to challenge ECDSA. With enough qubits, Grover's

learn anything about them yet can still perform the calculation. After the computation, the client will then decrypt the data again to get the real results of the calculation. Yet another application is a performance boost in network algorithms by using entangled qubits, which allows them to simultaneously calculate independent of their distance apart.^{1,7} The latter field of study is not yet mature, with distances only in the meter range and the entangling of only a few qubits, but the effects would be overwhelming if future networking no longer needed physical networks.

Given our analogy with Moore's law, large enough quantum computers will appear within a few years. Shor's algorithm works with

a quantum computer offering 10–100k qubits. Using Grover’s algorithm for database searching and hacking ECDSA will require some 100k qubits. All this assumes steady growth and the mastering of quantum-specific challenges, such as the noise and error rates caused by the inherent quantum effect that observing a superimposed quantum

- They provide abstractions between the underlying hardware and the actual software applications. This includes libraries to facilitate using the quantum computer either in simulation or as actual hardware.
- They offer development kits and computational platforms to ramp up end-user proficiency.

Building and even using a quantum computer involves a high investment because of the underlying quantum hardware stack.

state will influence its result, as described by the Schrödinger’s cat thought experiment. Given the longevity of embedded computing and the exponential growth rate, now is the time to prepare our software and IT systems for the impacts of quantum computing such as postquantum cryptography.

Quantum Software Development

To utilize quantum computing, quantum hardware vendors offer full stacks for the development of quantum software. As those are typically hardware specific, there are also third-party suppliers that provide platforms that claim to be hardware agnostic.

The quantum software platforms are portrayed in red in Figure 1. They offer the following functions:

- They provide users access to quantum computers to perform quantum computations via cloud services.

- They support software engineers in developing and testing their quantum algorithms.
- They enhance the reliability and performance of physical quantum computers. An inherent weakness of any quantum computing system is the errors in the transition from digital to quantum states. Random errors can occur due to the currently used hardware. Error-correcting software increases the stability and reliability of quantum computers.

Table 1 provides an overview of the currently available quantum software technologies. Toolkits from hardware suppliers are typically specific to their underlying hardware. Manufacturers provide both local simulators as well as cloud resources to access real machines.

Building and even using a quantum computer involves a high investment because of the underlying quantum hardware stack. Since actual quantum

computing hardware is much too expensive and complex, most available software platforms are based on cloud services. However, there are very few manufacturers capable of providing quantum services close to what is currently needed in terms of software business. Also, each manufacturer brings its own solutions, architectures, and specific hardware–software dependencies. To date there are no de facto standards for building an appropriate quantum software stack. In Figure 1 we have attempted to at least provide some abstraction levels between the different functional tiers.

Although there are many algorithms for quantum computers, it requires a good understanding of the underlying theory and technology to determine which algorithm can be used in a certain situation. Even if a suitable algorithm is transferred from traditional data science, its conversion into an executable program requires competence in the environment of the respective quantum computer, which data scientists and software engineers typically do not have.

Microsoft, IBM, and Google have their own respective environments, namely, Q#, Qiskit, and Cirq, which use the Python programming language. Microsoft’s Quantum Development Kit (QDK) delivers user-friendly code libraries, a debugger, and a resource estimator to assess how many qubits an algorithm will require. Each manufacturer provides its own access rules to the environments and its versions of approved languages. IBM offers access to a five-qubit machine free of charge. More powerful machines are available in its Quantum Network. Microsoft offers access to other companies’ quantum computers through its Azure Quantum platform.

Two distinct development technologies are visible: quantum gates

and quantum annealing. Most vendors offer an integrated development environment, but they are intended more as an environment for experiment and executing independent quantum algorithms/circuits than a business development environment. Most of the toolkits also include some quantum software optimization features, but usually modules are unconnected elements in a traditional or online file system, such as GitHub, or http-accessible files.

Several third-party tools can be connected to these toolkits, and high-level libraries are included. These libraries are sets of extensions to the programming language that encapsulate the manufacturer's specific components in a high-level unit: data normalizations, circuit classes, gates, calculation functions and utilities, error control, and many more. They are included because of the R&D and large investments of each manufacturer in this technology. In addition to these valuable resources, suppliers add through the Internet extensive repositories of information, code, algorithms, training materials, and a long list of other types of resources that make access to their quantum technologies much easier.

Third-party software platforms, although they bring quantum resources closer to the business world, do not yet provide the necessary core elements in the lifecycle and architecture of hybrid systems. While their capabilities and tools are good, it is necessary to invest time and effort in investigating how to fit them into a complete rigorous software lifecycle, to improve the productivity and ensure quality quantum software development. These development environments are hardware agnostic as they are intended to serve as development tools for various end-user environments. They are



QHEALTH

Quantum technology can be applied to multiple questions where data science meets algorithmic complexity. The aim of QHealth is to improve the quality of life of older adults. It correlates genetic and other variables related to a person's health history. The health history is analyzed as a function of the patient's drug consumption history, the reactions the older adult has experienced, and his/her physiological and genetic limitations.

The challenge in such an analysis involves the complexity of genetic pre-condition on one hand and also the number of drugs being used as part of the normal treatments of elderly persons. Even when looking only to the impacts of medication, there are multiple interactions and contraindications. For each active ingredient, in addition to variables such as genetic biomarkers, haplotypes, phenotypes, and so on, we must consider specific personal variables about the patient, such as sex, age, weight, blood pressure, recent drug history, and specific health impacts, among others. The underlying data analytics soon become intractable with classical computing.

QHealth builds a hybrid quantum system combining health-care applications and data analytics with quantum computing. Quantum technologies carry out optimizations and simulations whose realization in classical hardware is not possible in acceptable timescales. This hybrid system, in combination with classical health applications, will give its outputs to medical professionals involved in prescribing drugs to elderly adults. In a further extension, we also envisage application in the case of younger persons with difficult drug treatment and health conditions, trying to reduce the negative impacts of drugs due to their correlation and mutual side-effects when used in combination. Using the case histories and the socioeconomic and genetic variables of the persons being analyzed, we can then also make recommendations for suitable drug treatments and provide risk assessment before they are prescribed.

Using quantum technology for health care will vastly increase the possibilities to assess and optimize medical treatment applications, especially for persons who need multiple medications for coexisting illnesses. The proposed approach that we currently industrialize not only improves life and medical treatment but also has a financial impact because it will optimize the investments that health systems make in financing drugs and address the adverse effects that drugs often generate.

QHealth is founded by the Center for the Development of Industrial Technology (CDTI) of the Ministry of Science and Innovation of Spain and the European Regional Development Fund, in the 2020 CDTI Missions Program, with a total budget of several million euros. It involves a multidisciplinary team of researchers and technologists from the aQuantum, Gloin, and Madrija companies and the University Institute of Biosanitary Research of Extremadura in collaboration with the Pharmacogenetics and Personalized Medicine Unit, the University of Extremadura, and the University of Castilla-La Mancha.

Table 1. Quantum software development platforms.

Product Functionality	D-Wave Leap–Ocean	Fujitsu Quantum-Inspired Services	Google Cirq	IBM Quantum Experience and Qiskit	Microsoft Azure QDK and Q#	
URL	https://www.dwavesys.com/take-leap	https://www.fujitsu.com/es/services/business-services/digital-annealer	https://quantumai.google/cirq	https://quantum-computing.ibm.com	https://azure.microsoft.com	
Hardware agnosticity	NO Only D-WAVE	NO Only Fujitsu	NO Only Google	NO Only IBM	YES	
Programming language	Python Platform-specific language	Python	Python, platform-specific language	Python QASM Platform-specific language	Q# Python	
Integrated development environment	LEAP for executing quantum algorithms	None. It depends on Jupyter and Python	None. It depends on Jupyter and Python	QEXPERIENCE for executing quantum algorithms	Visual Studio Code	
Optimization	YES	NO	YES	YES	NO	
Modularity	YES if Python is used	YES if Python is used	YES if Python is used	YES	YES	
Out of the box functions	NO	NO	NO	NO	YES	
Service integration	API for executing solvers as a service	API for executing solvers as a service	API for executing circuits as a service	API for executing circuits as a service	API for executing circuits as a service	
Third-party software	Jupyter Strangeworks QuantumPath	Jupyter Strangeworks QuantumPath	Jupyter Strangeworks Zapata Orquestra QuantumPath	Jupyter Strangeworks Zapata Orquestra QuantumPath	Jupyter Strangeworks Zapata Orquestra QuantumPath	
High-level libraries	YES	YES	YES	YES	YES	

URL: uniform resource locator; QDK: Quantum Development Kit; OOTB: out of the box; API: application programming interface.

evolving as these kinds of toolkits try to create, in most cases, an integrated development environment. Only a few of them provide optimization facilities and out-of-the-box (OOTB) functionality.

In our evaluation (Table 1) we reflect these different attributes and functionalities. OOTB functionality reflects whether the toolkit is stand-alone or whether it needs to install third-party software to be able to produce software professionally.

Regarding service integration, all of the platforms provide an application programming interface (API)—in the case of quantum gate-based computers for executing quantum circuits as a service and in the case of quantum annealing ones for executing solvers as a service.

Challenges in Using Quantum Software

Designing software for quantum computers requires additional skills

compared to creating software for traditional computers. To benefit from the fast pace of quantum hardware evolution, it is urgent that we mature the technology and methods for quantum software. It is not enough to stress the importance of quantum software;¹ we must go a step further and raise the awareness of quantum software engineering (QSE).^{5,7} Distinguishing different layers of complex systems by simulation and networked smaller elements

	Rigetti Forest	Xanadu– Strawberry Fields and Penny Lane	Orquestra	Quantum Inspire	QuantumPath	Quantum Programming Studio	Strangeworks QC
	https://www.rigetti.com/quantum-computing/	https://strawberryfields.ai/	https://www.zapatacomputing.com/orquestra/	https://www.quantum-inspire.com/	https://www.quantumpath.es/	https://quantum-circuit.com/	https://strangeworks.com/
	NO Only RIGETTI	Partially IBM	YES, but few integrated providers	YES, mainly connected with IBM Quantum Experience.	YES	NO. The circuit is only directly exportable to Rigetti’s hardware	YES
	Python QUIL Platform-specific language	Python	Python	Python cQASM	Python Q#	Multiple languages	Python
	FOREST for executing quantum algorithms	None It depends on Jupyter and Python	NO. It depends on Jupyter and Python	Quantum Experience	YES	YES	YES
	YES	YES	NO	YES	YES	NO	NO
	YES	YES if Python is used	YES, if Python is used	YES, if Python is used	YES	NO	YES
	NO	NO	NO	NO	YES	NO	YES
	API for executing circuits as a service	API for executing circuits as a service	NO	API for executing circuits as a service	API for executing circuits as a service	PARTIAL	PARTIAL
	Jupyter Strangeworks Zapata Orquestra QuantumPath	Jupyter Strangeworks	Jupyter	Jupyter QuantumPath	Jupyter Visual Studio Java	NO	NO
	YES	YES	YES	YES	YES	YES	YES

will allow us to target innovation in parallel for the underlying hardware and software.³

Quantum software with industry-scale performance, robustness, and reliability will mean a next level in software technology. We strongly believe that quantum computing could also bring a new “golden age” to software engineering.⁸ But it is necessary to address all the challenges and opportunities faced in QSE⁷ and adapt or create the necessary models,

standards, or methods to help us in the creation of new quantum systems and the migration of current ones.⁹ One step in such advances is having the right development toolkits and knowing their characteristics.

Quantum software platforms and toolkits are difficult for practical industry usage. They do not bring much context support to the quantum algorithm generation, assuming that the quantum software engineer will know how to incorporate each product to its

corresponding platform. In the meantime, collections of quantum software algorithms are available, such as the quite exhaustive quantum algorithm zoo.⁶ So, to be able to work with the different quantum hardware, it is necessary to be knowledgeable about the requirements and libraries of each one of them.

Where Do We Go From Here?

Software and system technology innovation will further evolve at a fast



JOSE LUIS HEVIA is the quantum chief technology officer and the software architect and software solutions IT manager at Alhambra IT, Madrid, 28037, Spain. Further information about him can be found at <https://www.aquantum.es>. Contact him at jluis.hevia@alhambrait.com.



GUIDO PETERSSEN is the quantum chief operating officer and the director of R&D and software solutions at Alhambra IT, Madrid, 28037, Spain. Further information about him can be found at <https://www.aquantum.es>. Contact him at guido.peterssen@alhambrait.com.



CHRISTOF EBERT is the managing director of Vector Consulting Services, Stuttgart, 70499, Germany. He serves on the editorial board of *IEEE Software* and is a Senior Member of IEEE. Further information about him can be found at <https://twitter.com/christofebert>. Contact him at christof.ebert@vector.com.



MARIO PIATTINI is the quantum chief research officer and leader of the Alarcos Research Group, University of Castilla-La Mancha, Ciudad Real, 13001, Spain. Further information about him can be found at <https://www.uclm.es>. Contact him at mario.piattini@uclm.es.

pace in fields such as nanotechnology, biotechnology, genomics, and quantum computing.³ Today we can already use quantum computers and profit from their huge computation capacity to solve problems considered very difficult or unaffordable for “classic” computing. Quantum computing speeds up the process of solving algorithms that require massive parallel computations and so allows us to better simulate nature. All of this brings very new, disruptive, and potentially useful innovations.

Our focus here is on quantum software platforms to get started in

industry-scale software engineering. Quantum hardware suppliers have provided software technologies for their respective computers and quantum effect simulators. Results look promising as there are several platforms available which allow a smooth learning curve.


The state of quantum technology is improving at an accelerating rate. To produce useful and trusted quantum software, applications still must solve relevant issues, such as the resolution of qubits and the control of their errors. The results of quantum machines create new types of errors, and

we must learn to interpret the results. However, each vendor provides the results in different ways, which again leads to the need to rely on a particular vendor or to build a homogenized channel to consolidate the results.

The importance of professional software engineering for quantum computing has been neglected so far.⁷⁻⁹ New software engineering methods must be conceived based on experiences from software engineering for data science and machine learning.^{3,9} They must be enhanced to manage specific quantum challenges, such as uncertainties, noise, and interpretation. Along those lines, development tools are not yet suitable from a business point of view. The resources resulting from the use of vendor software development kits are individual elements that are not yet incorporated into enterprise development resources.

Software technology and development methodologies need to advance to make these assets part of a complete quantum software project lifecycle. The increasing awareness of quantum computing applications demands the production of quality quantum software. Without proper software technology platforms and suitable software engineering methods, quantum software remains a mere research topic. Especially in trusted environments, such as medicine, and others where defects will have severe consequences, quantum software must prove the same high-quality standards that we demand from any other software.

Physics Nobel laureate and quantum pioneer Niels Bohr once remarked: “Those who are not shocked when they first come across quantum theory cannot

possibly have understood it.” His observation still applies today, especially in using quantum effects to actually produce software. There is still a way to go to deliver quality quantum applications. Yet now is the time to start. To scale from research to industry, quantum software must adopt sound software engineering methods for the development of quantum software—and enhance them as we once did when scaling agile development. Good enough may be sufficient for today, but it is certainly not for tomorrow. 

References

1. J. D. Hidary, *Quantum Computing: An Applied Approach*. New York: Springer, 2019.
2. Quantum Software Manifesto. <https://www.qusoft.org/quantum-software-manifesto> (accessed June 20, 2021).
3. C. Ebert and B. Tavernier, “Technology trends: Strategies for the new normal,” *IEEE Softw.*, vol. 38, no. 2, pp. 7–14, Mar. 2021. doi: 10.1109/MS.2020.3043407.
4. European Quantum Flagship, “Strategic research agenda.” <https://digital-strategy.ec.europa.eu/en/library/quantum-flagship-major-boost-european-quantum-research> (accessed June 20, 2021).
5. “Quantum devices and simulators.” IBM. <https://www.research.ibm.com/ibmq-technology/devices/#ibmq-20-tokyo> (accessed June 20, 2021).
6. “Algebraic and number theoretic algorithms.” Quantum Algorithm Zoo. <https://quantumalgorithmzoo.org/> (accessed June 5, 2021).
7. J. Zhao, “Quantum software engineering: Landscapes and horizons,” July 14, 2020, arXiv:2007.07047v1 [cs.SE].
8. M. Piattini, G. Peterssen, and R. Pérez-Castillo, “Quantum computing: A new software engineering golden age,” *ACM SIGSOFT Softw. Eng. Newslett.*, vol. 45, no. 3, pp. 12–14, June 2020. doi: 10.1145/3402127.3402131.
9. M. Piattini, M. Serrano, R. Pérez-Castillo, G. Peterssen, and J. L. Hevia, “Towards a quantum software engineering,” *IT Prof.*, vol. 23, no. 1, pp. 62–66, Jan.-Feb. 2021. doi: 10.1109/MITP.2020.3019522.

ICCQ

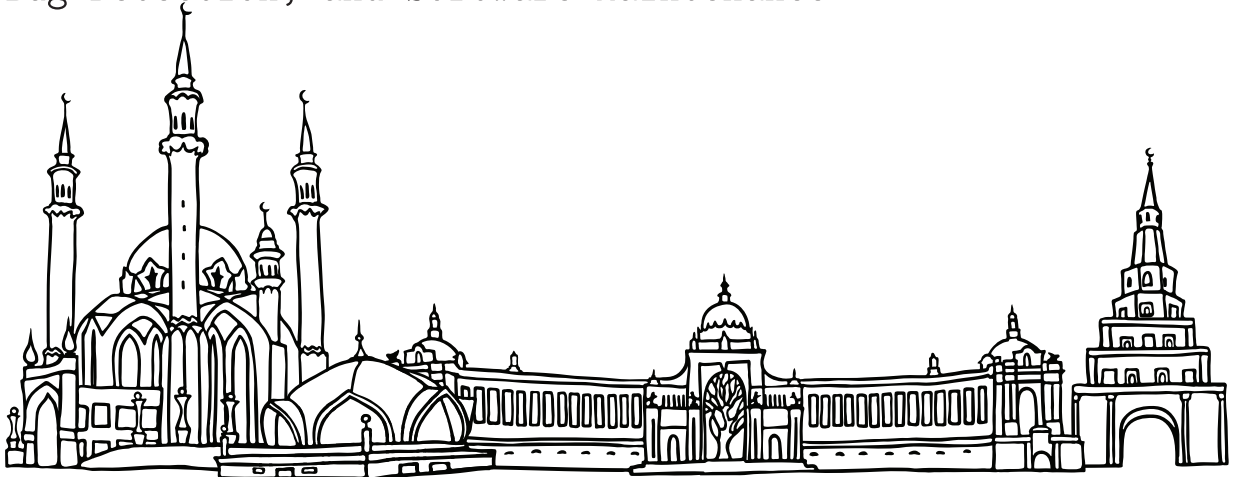
The Second International Conference on Code Quality (23 Apr, online)

Static/Dynamic Analysis, Program Verification, Bug Detection, and Software Maintenance

www.iccq.ru

CfP closes on 18 Dec

In cooperation with
ACM SIGPLAN and SIGSOFT
IEEE Computer Society



Digital Object Identifier 10.1109/MS.2021.3099661