THE EMERGENCE OF A QUANTUM WORKFORCE

A GUIDELINE TO THE ORGANIZATIONAL QUANTUM WORKFORCE REQUIREMENTS

This article is part of Capgemini's Applications Unleashed 2022 Report which can be downloaded here.

HIGHLIGHTS

- As quantum computing is rapidly developing, companies should start thinking about building capable teams.
- Quantum computing is fundamentally different and, therefore, requires new roles across the whole organization.
- Depending on application specifics, and technological ambition, workforce needs will vary amongst companies.
- The stack model provides insight into the technical depth of quantum applications, whereas the software lifecycle helps understand the various steps in developing quantum applications.
- Using a combination of the software lifecycle and stack model, we outline a potential future quantum workforce.

Quantum computing promises to be the next big thing. The disruptive potential of quantum computing to accelerate drug discovery, accelerate the development of new materials, and to enable more complex financial models, puts the technology on top of the agenda of many companies. However, progress in developing quantum roadmaps and exploring how quantum computing will impact businesses seems to be limited by the availability of talent, not the amount of funding or access to the technology. How can companies build up the right teams?

The recent interest in quantum computing has spurred a boom of innovative startups, blue chip manufacturers are involved, and end-users are getting excited. Roadmaps are ambitiously targeting commercial value halfway through the decade. Companies are addressing every part of the stack, from hardware to programming languages to debugging tools. Quantum computing is around the corner, an outline of the future computational workflow is beginning to shape, and new tasks, roles and jobs are emerging.

But what profiles will be required to make the quantum dream a reality? Although current progress has been dominated by quantum information scientists, as the field matures, a fractionation of roles will occur. In 2023, progress is no longer exclusively made by academics and quantum information scientists. Increasingly, we need to start thinking about a role for our testers, infrastructure specialists, programmers, solution architects, cyber security officers, chief technology officers, and chief executive officers.

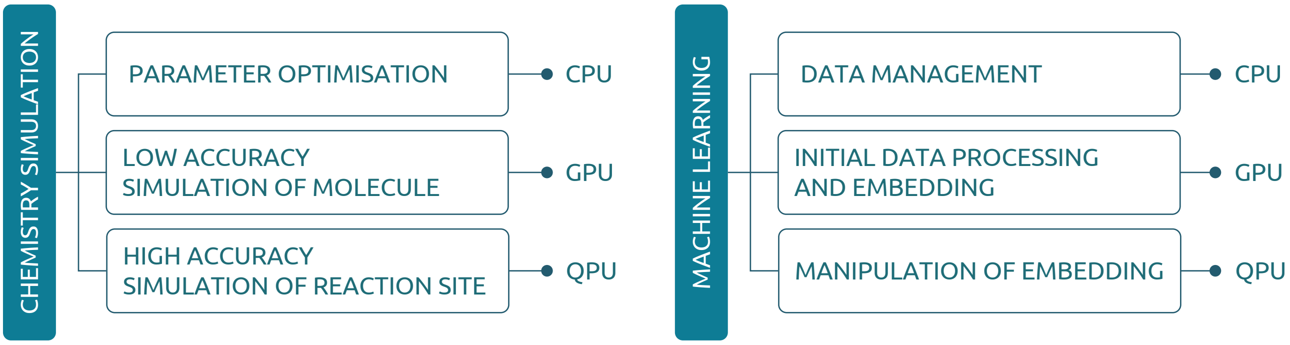
The quantum computing stack helps explain which quantum roles will emerge. As quantum hardware is fundamentally different from classical hardware, a complete quantum computing stack is only starting to emerge. As time progresses, complexity is added, and lower-level operations are abstracted away. However, low-level control will remain critical for high-performance applications which push the boundaries of what quantum computing can do.

- The hardware interface level of the stack interfaces the abstraction of digital programming with the physical implementation of the qubits. For example, superconducting qubits are encoded in the charge of a superconducting current (transmon qubits), where microwave pulses are used to implement quantum gates. Whilst each quantum computer comes with its own default settings, tools such as Qua from Quantum Machines provide a universal pulse-level language for quantum computers. Developers will be needs to manually optimize the control pulse for applications requiring very high performance.
- The gate level or circuit level of the stack creates quantum circuits and implements the quantum logic which powers quantum computation. Interfaces to this stack can include online tools such as IBM's Quantum Experience, where circuits can be defined through a graphical user interface. For specialist applications, developers may need to fine tune circuits to achieve optimal performance.
- The algorithmic or software level of the stack is defined by higher level programming languages where quantum circuits are abstracted away. For practical applications, millions of gates on thousands of qubits are required. Companies like Classiq are addressing this challenge by developing a declarative programming language where, instead of defining individual gates, the intent of an algorithm is defined, and quantum circuits are automatically generated. Validation and verification of code will be a challenge for developers as traditional

debugging techniques (such as breakpoints) cannot be implemented on quantum computers.

- At the application level, quantum computing will always be heterogenous, working in parallel with classical resources. Tools which work on this level of the stack are only starting to be developed. For example, the European Commission procurement program of quantum computers aims to integrate quantum computing hardware with existing high-performance computing (HPC) infrastructure. Quantum developers at this level will need to draw on domain expertise to separate commercial problems into subtasks which can be allocated to CPUs, GPUs and quantum processors (QPUs) to achieve optimal performance. For example, within a chemistry application, a developer might optimize parameters on a CPU, run a low accuracy benchmark on a GPU, and perform a highly accurate simulation of a complex sub-problem on the QPU (Fig. 1).

Figure 1: Problem decomposition at the application level: on the left is an example of a heterogenic chemistry workflow using a combination of CPU, GPU, and QPU. On the right is an example of a heterogenic workflow for machine learning.



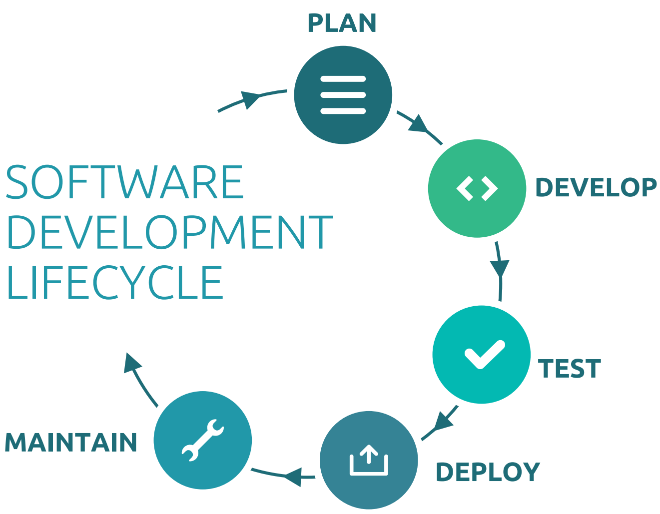
A SOFTWARE LIFECYCLE MODEL CAN HELP US PREDICT THE QUANTUM SOFTWARE DEVELOPMENT CYCLE FROM A PROCESS PERSPECTIVE

As quantum applications become commercially relevant, integration of quantum computing into business infrastructure becomes increasingly important. For secure, reliable, and scalable execution of quantum workloads, companies must start thinking beyond a PoC, and develop processes for testing, maintaining and deployment of code.

Current quantum computing hardware is not yet capable of delivering advantage over conventional computers for commercially valuable applications. To explore the potential of this technology, current quantum algorithm developers create PoCs and demonstrators to outline where quantum computing can solve relevant problems and what advantage may be expected. Integration into commercial workflows is hardly a relevant problem and application lifecycles often follow an experimental format where code is not rigorously optimized, tested, deployed or maintained.

However, when quantum algorithms show commercial value, companies must consider a spectrum of integration challenges when deploying quantum software to solve real-world problems. A software lifecycle model like those used in contemporary programming can help us predict the quantum software development cycle from a process perspective. To some degree, the depth of penetration of the lifecycle within the stack depends on the application and the maturity of the technology. Like high-value classical applications that still rely on dedicated hardware and low-level programming, high-value quantum applications might also require dedicated hardware or low-level programming. However, designing quantum software will follow a conventional development cycle of plan, develop, test, deploy and maintain (Fig.2)

Figure 2: Software Development Lifecycle



On the other hand, depending on application requirements, companies might choose to use out-of-the-box solutions or use specific-purpose, highly optimized quantum hardware and software

Plan

First, computational problems must be decomposed into a quantum and classical parts such that resources are allocated to the correct device. An example of such decomposition is given in figure three. However, the quantum/classical splitting could happen at different granularities. On the circuit layer, quantum/classical splitting offload part of the workload to intelligent, high-latency classical pre/post-processing. One level lower, on the hardware interface, quantum/classical spitting is used for error correction, noise mitigation, and memory access. To some degree, automated tools and compilers may help in such decomposition. Depending on the application requirements, custom programming might be required at all levels.

Develop

The next phase in the quantum software lifecycle is the development of quantum algorithms and circuits. Software packages are emerging that allow developers to easily experiment with algorithms. However, there is a complex design space to consider. On a high level, considerations such as cost/performance, hardware availability, and scalability should be considered. It is the gate level, where typically, new algorithms are devised. Existing algorithms can also be optimized using hardware-efficient mappings, noise-mitigation, and other techniques. For specific applications, particularly in the near term, it pays to optimize algorithms at the lowest level. Techniques such as optimal control, application-specific calibration, and adaptive compiling can help reduce the noise within applications and improve the performance of algorithms.

Test

An increasingly researched area, and critical part of a quantum software lifecycle, is the testing and verification of quantum applications. Current systems are limited in qubit number and can therefore be simulated on classical hardware. However, for anything above fifty qubits, we cannot expect a full classical representation. We will therefore need a variety of approaches on all levels of the stack. At the hardware level, benchmarks such as qubit number, gate fidelity, quantum volume and QLOPS aim to address different hardware characteristics, all with their limitations and merits. At the algorithm level, performance is typically explored using a complexity-centric approach i.e., based on the scaling of the algorithms. However, how much that holds up for larger applications remains unclear. Other techniques focus on quantifying the 'quantumness' of data and algorithms, suggesting that quantum algorithms can be more 'expressive' and more efficiently represent certain problem classes than classical solvers. On an application level, quantum results could be benchmarked against GPU-based benchmarks. However, as quantum applications start to deliver results to problems classically intractable, this might become increasingly difficult. Traditional testing tools, such as unit tests or breakpoints, will be largely unavailable for quantum computers, providing an additional challenge.

Deploy

Quantum applications should be packaged, deployed, and executed on quantum and classical clouds. From a workflow perspective, workloads should be managed, controlled, and allocated. Dependencies should be resolved. Applications should be packaged and compiled to dedicated hardware. On a low-level, runtimes must be virtualized. Additionally, complicated by the fact that classical and quantum resources might be physically located in different locations. Central within each layer of the stack is to create secure runtimes that efficiently use available resources, given the variety of low-latency constraints.

Maintain

Finally, quantum software will need to be maintained to ensure that solutions are available when needed and that quantum-enabled systems are robust against software updates, configuration changes, system downtime for maintenance and other operational challenges.

As we move forward, specializations within quantum software roles will continue to emerge depending on the maturity and application requirements. Advances in hardware, such as error correction, optimal control, or quantum memory, and in software, such as orchestration platforms and algorithm libraries, will open the door for computer scientists and domain experts to develop quantum applications within larger computational workflows. Increasingly, these workflows will be integrated into high-performance compute infrastructure and connected to classical software lifecycles. On the other hand, depending on application requirements, companies might choose to use out-of-the-box solutions or use specific-purpose, highly optimized quantum hardware and software. Just like Java developers might use JVM without thinking about the underlying hardware, while energy grid providers might use dedicated hardware and software for quality assurance.

Either way, new roles for testers, software developers, quantum data scientists, quantum data engineers, and others will emerge in the quantum field. To understand the roles required, we can use the stack model to understand the required technical depth. The software lifecycle, on the other hand, helps understand the various steps in developing quantum applications. It helps us to predict who will be involved at each stage of quantum software development, and it helps us to understand what is required for running quantum applications for real-world applications. In table one, we have summarized potential roles based on this taxonomy.

An exciting future is beginning to take shape, and recent progress provides plentiful opportunities. However, no one knows what the future will bring. The most disruptive application of quantum computing will be one we haven't thought of yet. With such rapid technological advances, it becomes ever more essential to adopt an open innovation culture.

Table 1: Sample of quantum technology roles

	PLAN	DEVELOP	TEST	DEPLOY
Hardware interface level	Engineer at hardware vendor	Error correction/ optimal control specialist	Hardware characterisation engineer	Hardware characterisation engineer
Gate level	Quantum algorithm optimisation specialist	Quantum algorithm optimisation specialist	Statistical noise analyst	Configuration manager
Software level	Quantum performance analyst?	Quantum software developer	Benchmark specialist	Quantum DevOPS
Application level	Quantum innovation lead	Quantum innovation lead	Quality assurance tester	Quantum cloud devops

ABOUT THE AUTHORS

Edmund Owen

Head of Research, Capgemini Quantum Lab

Edmund Owen is a quantum physicist at Cambridge Consultants with a background in modelling quantum systems and quantum-enabled product development. He works closely with businesses and technologists to identify how quantum technologies can be used to meet the needs of the most ambitious clients.

edmund.owen@cambridgeconsultants.com

https://www.linkedin.com/in/edmund-owen-274381164/

Julian van Velzen

CTIO & Head of Capgemini's Quantum Lab

Julian is the Head of the Capgemini quantum lab; a global network of quantum experts, partners, and facilities, focused on 3 key areas: Sensing, Communication and Computing. From this Lab, Capgemini is exploring with its clients how to research and build demos to help solve business and societal problems that up until now are seemingly intractable. Additionally, Julian has a special interest in sustainable development, he is part of the group CTIO community, he is the Dutch representative of the European quantum consortium (QuIC), and he is a member of the Forbes Technology Council.

julian.van.velzen@capgemini.com

https://www.linkedin.com/in/julian-van-velzen-60877283/

About Capgemini

Capgemini is a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of 325,000 team members in nearly 50 countries. With its strong 55 year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2021 global revenues of €18 billion.

Get the Future You Want | www.capgemini.nl

IN/2A-005.23d

For more information:

Capgemini Nederland B.V. P.O. Box 2575, 3500 GN Utrecht Tel. + 31 30 203 05 00 www.capgemini.nl