Persistent

Questions of Methodology of Requirements Engineering in Quantum Computing

April 2023

Christian Bucholdt

Persistent Systems

Christian Bucholdt
Head of Delivery Europe & Middle East – Industry Verticals (BFSI, HLS, HiTech, Manufacturing, Telco)

We are Persistent.

A trusted **Digital Engineering** and **Enterprise Modernization** partner with global presence.

$264.4M FY23 Q3 Revenue **+32.8%** YoY	**$1,057.4M** Annualized revenue run rate based on Q3FY23	**$978.8M** TTM Revenue **+39.6%** YoY	**₹2.4B** FY23 Q3 PAT **+34.9%** YoY
$440M Total Contract Value (TCV) Booking	**₹116.30** TTM EPS **+41.7%** YoY	**$3.6B** Market Cap**	**22,598** Employees **+33.0%** YoY

* 1 USD = INR 82.75 ** Market cap as on December 31, 2022

Persistent

Contents

1

Quantum Computing today*

$2.35 Bn

$15.3 Bn

$8.4 Bn

$3.7 Bn

US

China

EU

* Source: McKinsey Qunatum Computing Monitor 2023

Persistent

Quantum computing technologies

Quantum annealing: Measuring the energy state through quantum mechanics - for optimization only (D-Wave)
Quantum circuits: for general purpose computation problems (see below)
Challenges: Physical vs. Logical qubits; Error correction; 10^{-15} accuracy (99.9999999999999)

	Description	Players
Superconducting	Usage of materials with electric resistance Advantage: Most advanced research Problem: Requires ultra-low temperature	IBM, Google
Photonics	Usage of very precise light in an optical cavity Advantage: Can work at room temperature Problem: Scattering of light	China, Xanadu
Ion Trap	Manage ion's in magnetic fields and lasers for manipulation Advantage: Room temperature; coherence Problem: Scaling	IonQ, Quantinuum
Cold Atom	Cold atoms for a qubits array Advantage: Long coherence time and strong connectivity Problem: Slow performance	PASQUAL, ColdQuanta

Quantum Computing Use Cases

\ Optimization problem – Example: Revenue optimization for a rental car

Owners of a car rental company have determined that if they charge customers p dollars per day to rent a car, where $50 \leq p \leq 200$, the number of cars n they rent per day can be modelled by the linear function $n(p)=1000-5p$. If they charge $50 per day or less, they will rent all their cars. If they charge $200 per day or more, they will not rent any cars. Assuming the owners plan to charge customers between $50 per day and $200 per day to rent a car, how much should they charge to maximize their revenue?

\ Realworld challenge:

- There is a massive scheduling problem for security at airports. We have over 60,000 employees and we have to schedule them into 450 different airports. It's a huge challenge, because there are many, many factors that we need to consider; in fact, it's so challenging that we're typically creating schedules, six months in advance."
- It typically takes weeks to develop such schedules, and a wide variety to complicating factors (weather, personnel sickness, etc.) inevitably arise long after the schedules have been created.
- **Deloitte** has been developing an application to handle the problem, an optimization task, that uses the D-Wave computer.
- "Using the quantum annealer these [scheduling jobs] can be done in real-time very easily.

Quantum Computing Use Cases

- How to describe an optimization problem (search problem)
- *An optimization problem asks, what is the best solution?*
  - Find the optimal value for a formula (minimum or maximum)
  - Objective: Find the largest (or smallest) value of $f(x)$ when $a \le x \le b$.
  - Challenge: Sometimes a or b are infinite, but frequently the real world imposes some ***constraint*** on the values that x may have.

Persistent

Quantum Computing Use Cases

- Combinatorical problems (decision problem) – Example: Traveling salesman problem
  - finding a grouping, ordering, or assignment of a discrete, finite set of objects that satisfies given conditions

Given a set of N cities with travel distance between each pair, find the shortest path that visits each city exactly once. The full enumeration quickly becomes infeasible as N grows. Using a brute force search method, if a computer can check a solution in a microsecond, then it would take two microseconds to solve three cities, 3.6 seconds for 11 cities, and 3857 years for 20 cities.

- Realworld challenge: Chemical combinations for new drugs: side effects of new drugs with other drugs
  - One approach is the systematic high-throughput testing of pairwise drug combinations,
  - which, however, faces a combinatorial challenge: for 1000 U.S. Food and Drug Administration (FDA)-approved drugs, there are 499,500 possible pairwise combinations
  - that should be tested over approximately 3000 human diseases and multiple dosage combinations.

Quantum Computing Use Cases

- How to describe a combinatorical problems (decision problem)
- *A decision problem asks, is there a solution with a certain characteristic?*
  - Finding satisfying variable assignments of propositional formulae (SAT)
  - Given: Formula $F := (x1 \lor x2) \land (\neg x1 \lor \neg x2)$
  - Objective: Find an assignment of truth values to variables x1, x2 that renders F true, or decide that no such assignment exists.
  - Challenge: which algorithm to use to come to a possible and satisfying solution

Quantum Software Requirements Methodology 1/2

\ Quantum Lifecycle Model: Zhao, Quantum Software Engineering, 2021, https://arxiv.org/pdf/2007.07047.pdf

Understanding of what to resolve

- What is the problem to be solved by quantum software?
- What characteristics of the quantum software are used to solve the problem?
- How will the quantum software (and the solution) be implemented?
- What method will be used to detect errors made in the design and construction of the quantum software?
- How will the quantum software be supported in a long period, when the users request corrections, adaptations, and enhancements?

Methodology in its infancy

- Existing research very limited
- **Challenge:** Quantum computing problems hard to understand and require scientific/mathematical exact description and measurable results
- Requires classical requirements engineering and new aspects for hybrid approach
- Usage of modelling languages such as UML or more specific languages required for quantum computing problems

Continuous requirements analysis

3.5.1 ***Quantum Software Requirements Analysis.*** The quantum software life cycle model begins with the requirements analysis phase, where the stakeholders discuss the requirements of the software that needs to be developed to achieve a goal. The requirements analysis phase aims to capture the detail of each requirement and to make sure everyone understands the scope of the work and how each requirement is going to be fulfilled. The analysis creates a set of measurable quantum software requirements that specify, from the supplier's perspective, what characteristics, attributes, and functional and performance requirements the quantum software system is to possess, to satisfy stakeholder requirements. Later life cycle phases, including design, implementation, testing, maintenance for quantum software, assume that requirements analysis continues through the quantum software life cycle.

Scientific specification

5.2 Quantum Software Specification

Quantum computing relies on quantum mechanics, which is a subject more familiar to physicists rather than computer scientists and software engineers. Thus, we must be aware of the underlying theory before we reason about quantum computers. Even though delivering the principles of quantum mechanics is not an easy task, due to their counter-intuitive nature.

Recently, the method for reasoning about quantum computing is a mixture of linear algebra and Dirac representation, which is a subject more suitable for physicists than computer scientists (software engineers) [45]. Therefore, it is necessary to provide a more "intuitive" way to think and write quantum algorithms, thereby simplifying the design and implementation of quantum software. This can be achieved, for example, by introducing a specification language, which adopts the symbolism and reasoning principle of software engineering and will take on the role of Hilbert space algebra, allowing us to describe quantum structures and design quantum algorithms in a more natural way [115].

Persistent

Quantum Software Requirements Methodology 2/2

\ NFRs: Saraiva et. al., Non-Functional Requirements for Quantum Programs, 2021, https://ceur-ws.org/Vol-3008/paper4.pdf

- Quantum computing programs still very much tight to a specific concept and hardware (superconducting, photonics, Ion-Trap, Cold Atom) & Annealing – similar like embedded software
- Different Quantum programming concepts have impact on requirements description (intermediate state measure vs. state measure at the end of processing)

NFR1: number of qubits- the program should use a maximum of n qubits, where n is the number of qubits available in the target quantum device.

NFR2: Stability of compute process - the program should be designed considering the maximum circuit depth so that the target device can maintain a stable quantum state for the necessary period to execute the algorithm.

NFR3: Complexity of algorithms - the program should be designed considering the number of T gates so that it does not exceed the limit of the target device

NFR4: Qubit connectivity - the program should be implemented minimizing the number of gates between qubits that are not physically connected on the target device.

NFR5: Number of quantum gates - the program should be implemented minimizing the use of gates that are not available in the target quantum device.

NFR6: Readout error mitigation – Implement optimized readout error mitigation technique

Hybrid of Classical and Quantum Computing

A possible journey to quantum advantage

Classical Computing Stays	Classical computers will continue to be good for user interaction and workflow-oriented systems
Quantum takes care of the Hard Problems	Quantum computing will focus on the high computational needs such as Monte Carlo simulation or scheduling
Resource Management	This approach enables efficient management of the (expensive) quantum compute power.
Usability	With a hybrid approach the complexity of quantum computing can be «hidden» from non-experts
Extension and Adaptability	With a hybrid approach it is easier to extend or adapt a system with the further enhancements of quantum computing technology

Hybrid – Types of Classical & Quantum Computing

Batch Quantum Computing

- Client application creates a set of circuits for problem resolution
- All circuits are submitted together to the quantum processing unit

Saves time for initiation of quantum processing for each individual circuit

Interactive Quantum Computing

- Client application has multiple interaction with the quantum processing unit with different parameters
- Different prioritization is possible

Flexibility of quantum computing execution based on variable input

Integrated Quantum Computing

- Tight coupling of classical and quantum computing with coherent qubits
- Usage of common programming constructs

Mid-circuit measurement and reuse of qubits

Distributed Quantum Computing

- Classical computing works along with logical qubits
- Computation across different cloud resources

Large number of qubits and resolution of complex problems

Why bother?

Learning

Start Early

- Making first experiences with quantum computing leads to early learning
- New way of problem description is required and needs to be tested
- Quantum ready cryptography of todays IT

Value

Competitive Advantage

- With appropriate solution of QC the company can make a step ahead of competitors
- Quantum computing, although in its infancy, can bring new value in the future
- First quantum advantage possible today

Cost

Appropriate Resource Usage

- Cost conscious solutions
- Cost savings from early optimization adaptation (e.g. logistics or scheduling)

Persistent

See Beyond, Rise Above

Watch Brand Video