ISSN 2007-9737

# Integration of Agile Approaches with Quantum High-Performance Computing in Healthcare System Designs

Abdullah[1,2], Nida Hafeez[1,2], Kinza Sardar[3], Jose Luis Oropeza-Rodríguez[1,*], Alexander Gelbukh[1], Grigori Sidorov[1]

[1] Instituto Politecnico Nacional, Centro de Investigacion en Computacion, Mexico

[2] Bahria University, Department of Computer Science, Pakistan

[3] University of Management and Technology, Department of Software Engineering, Pakistan

{abdullah2025, nhafeez2024, joropeza, gelbukh, sidorov}@cic.ipn.mx, kinza.sardar@umt.edu.pk

**Abstract.** This study explores the integration of Agile methodologies and quantum high-performance computing (HPC) in healthcare. Agile methodology offers flexibility, while quantum HPC has the potential to revolutionize healthcare. The paper presents an architectural design for HPC in healthcare, discussing tool alignment, software development, and integration challenges. It emphasizes user-centered design, cross-functional teams, continuous integration, and retrospectives for software development. Personalized medical tools take advantage of data analytics and machine learning (ML) with quantum HPC to improve speed and accuracy. Real-time patient monitoring systems utilize wearable devices and sensors, which are developed iteratively using an Agile methodology. The integration of Agile methodologies with quantum HPC can transform healthcare, improving access, efficiency, and patient outcomes. The study focuses on developing personalized medicine, real-time monitoring, and telemedicine tools to ensure improved care, security, and privacy through quantum HPC advancements.

**Keywords.** Agile methodologies, quantum HPC, healthcare patient-centered design.

## 1 Introduction

The convergence of cutting-edge computing technologies and flexible software development paradigms is poised to redefine the future of healthcare [34, 7, 29]. As the complexity and volume of biomedical data continue to grow exponentially, traditional computational methods and development frameworks face limitations in delivering timely, personalized, and scalable solutions. Quantum High-Performance Computing (HPC), with its unparalleled ability to process vast datasets and solve intricate problems, coupled with Agile methodologies known for their iterative, adaptive, and user-focused approach, offers a transformative pathway for healthcare innovation.

This synergy promises to accelerate precision medicine, enhance real-time patient monitoring, and optimize clinical workflows [2, 4]. However, integrating these two domains presents unique challenges that must be systematically addressed to realize their full potential in clinical settings [6, 5]. This paper pioneers the exploration of a formalized framework that unites Agile development principles with quantum-classical hybrid HPC, marking a critical step toward practical, efficient, and robust healthcare applications.

Agile methodologies, renowned for their adaptability and flexibility, can be used in conjunction with quantum. HPC to completely transform the healthcare sector [21, 35, 12, 27]. Utilizing data analytics and machine learning to analyze patient data and find the best possible treatments, quantum HPC enables the creation of personalized medical solutions [8, 9, 31]. To respond quickly to changes in a patient's health, real-time patient monitoring systems use wearable technology and sensors to relay data to healthcare professionals. Iterative development and user input integration are ensured by an agile approach for ongoing improvement. In addition, telemedicine systems developed using agile methodology improve access to treatment, reduce waiting times, and improve productivity [18, 15, 19, 37]. Agile methodologies and quantum HPC have the potential to revolutionize healthcare by enhancing patient outcomes and spurring innovation [23, 32].

This research pioneers the fusion of Agile methodologies and quantum HPC in healthcare by addressing challenges with the convergence of flexibility and quantum power, architectural design for quantum HPC in healthcare, centered design and cross-functional collaboration, real-time patient monitoring and wearable technology, personalized medical solutions with quantum HPC, and Agile methods for efficient healthcare systems. Offered technologies have the potential to transform healthcare practices by providing a nimble, efficient, and patient-centered approach.

While Agile methods have been broadly adopted in classical healthcare IT projects, their direct application to quantum-classical hybrid systems presents unresolved challenges such as high algorithmic uncertainty, circuit instability, probabilistic outputs, and incompatible DevOps tools. This paper addresses the critical gap in formalizing and validating an integration framework between Agile workflows and quantum HPC systems for clinical-grade applications. To our knowledge, this is the first empirical evaluation of Agile-scrum-driven quantum-classical development cycles applied to real-world healthcare scenarios using Qiskit, PyQuil, and fidelity-aware DevOps pipelines.

The study comprises seven sections. The introduction section outlines the research scope, followed by the methodology that frames the investigation. The background section provides context, while the next section focuses on architecting Agile methodologies for quantum high-performance computing. This is followed by case studies and empirical evidence, after which the limitations of applying Agile within high-performance healthcare systems are examined. The study concludes with a summary of findings and closing insights.

## 2 Research Methodology

A multiphase research methodology, depicted in Figure 1, was used to systematically investigate the integration of Agile methodologies with quantum high-performance computing (QHPC) in healthcare. The core objective was to examine how the principles of Agile software engineering could be operationalized within hybrid quantum-classical computational environments to improve clinical outcomes and system performance. This methodological framework consisted of both qualitative and empirical components, including literature review, system modeling, and detailed case study analysis.

The first phase involved a comprehensive review of academic literature from indexed databases, including PubMed, IEEE Xplore, SpringerLink, and ACM Digital Library. Search queries combined terms such as "Agile in Healthcare IT", "Quantum computing in drug discovery", and "hybrid quantum-classical architectures". Inclusion criteria focused on peer-reviewed publications and empirical studies that discussed the convergence of Agile techniques with quantum or high-performance computing in clinical or biomedical contexts. Nonempirical, speculative, or unrelated studies were excluded. Key sources informing the theoretical basis of this study include [30, 11, 17], which collectively contextualize computational methodologies in healthcare and quantum-assisted biomedical modeling.

In the second phase, two representative case studies were designed and analyzed to assess the practical application of Agile-integrated QHPC pipelines. These use cases focused on (i) quantum-assisted personalized drug discovery and (ii) real-time ICU patient monitoring using quantum machine learning. For each case, a hybrid Agile

ISSN 2007-9737



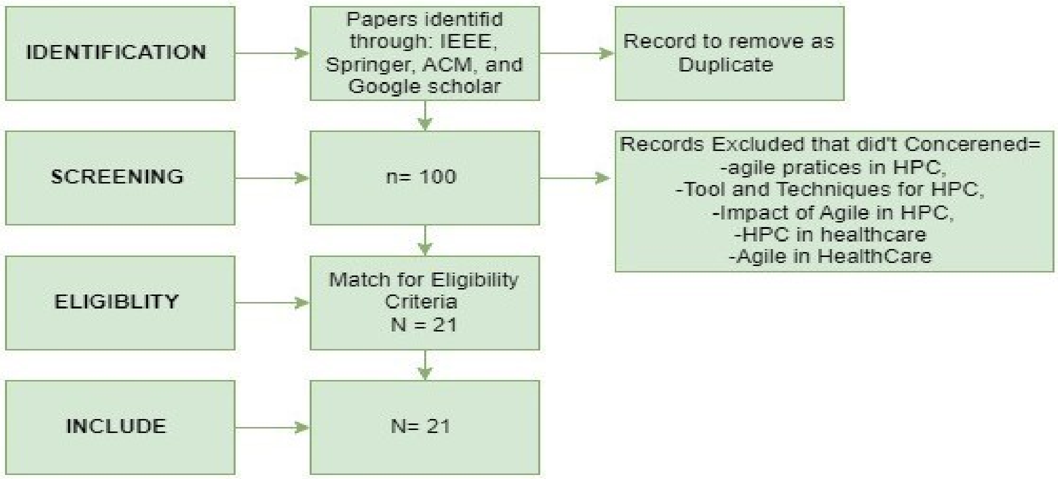


**Fig. 1.** Research methodology

methodology was implemented, grounded in iterative sprint cycles, continuous integration (CI), and fidelity-aware feedback loops. Tools such as Qiskit, PyQuil, and Health TestGen were employed to operationalize quantum circuit design, simulate physiological data streams, and validate system outputs under stochastic conditions. In [22, 39], researchers provide essential benchmarking data and algorithmic foundations for the components of quantum-assisted drug modeling, while the QML-based patient monitoring framework aligns with studies by [30].

Crucially, this study not only evaluates computational performance but also examines how Agile methods adapt to quantum-specific constraints such as decoherence, stochastic outputs, and circuit fidelity errors within multidisciplinary healthcare teams. Unlike classical software workflows, QHPC environments introduce fundamental challenges to Agile execution: non-deterministic test results, intermittent circuit instability, and lack of conventional CI/CD maturity due to hardware-specific noise and probabilistic state collapses. These characteristics impede core Agile processes such as deterministic unit testing, backlog grooming with fixed outcomes, and predictable sprint velocities.

To quantify these adaptation efforts, both system-level and Agile process metrics were tracked. In addition to conventional performance measures, simulation speed, predictive accuracy, Agile-specific indicators, including sprint velocity, task rework rates, developer feedback loops, and code acceptance ratios, were recorded across development cycles [3, 1]. These were captured using integrated Kanban boards, Git commit logs, automated regression test reports, and post-sprint retrospectives.

Empirical evaluation involved recording computational performance metrics such as processing speedup and accuracy improvements across classical-only and hybrid systems. As Table 4 summarizes the comparative analysis of these metrics, revealing a 40% simulation time reduction and a 22% gain in predictive accuracy in the drug discovery pipeline, as well as a 30% increase in throughput and an 18% improvement in ECG anomaly detection accuracy for ICU monitoring. These results were derived through repeated sprint iterations and nightly regression testing of quantum submodules integrated within Agile DevOps pipelines.

To ensure methodological rigor, data extraction included both structured system outputs and clinician feedback on the utility and interpretability of results. A thematic synthesis was performed to identify recurring challenges, such as quantum decoherence handling, Agile role redefinition for cross-disciplinary teams, and integration friction between classical preprocessors and quantum simulators.

Finally, a critical evaluation of the limitations and emerging research gaps was performed. Despite promising outcomes, issues such as hardware instability, limited access to fault-tolerant quantum devices, and underdeveloped Agile metrics tailored for quantum workflows remain significant. These findings set the groundwork for future exploration of federated quantum learning, cross-hospital quantum edge nodes, and adaptive Agile models suited for probabilistic computing contexts.

## 3 Background

The evolution of high-performance computing (HPC) has been driven by the need for large-scale data processing in scientific research and engineering applications [20, 24, 40, 33, 13]. Milestones include the introduction of supercomputers like the IBM 704 and advancements in parallel and distributed computing [26]. Challenges include the "power wall" and the focus on power-efficient architectures. HPC in healthcare has also gained significance [42].

### 3.1 Quantum Computing Fundamentals

**Qubits and Superposition:** Unlike classical bits (0/1), quantum bits (qubits) exist in superposition states as shown in Equation 1:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where}|\alpha|^2 + |\beta|^2 = 1. \tag{1}$$

This enables parallel processing of medical data and simultaneous analysis of multiple genomic sequences.

**Entanglement:** Qubits exhibit non-local correlations as shown in Equation 2:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right). \tag{2}$$

This property is critical for secure medical data transmission, quantum key distribution, and correlated biomarker analysis.

**Decoherence and Error Correction:** Quantum states degrade via environmental interaction T1, T2 times. Surface code error correction as shown in Equation 3:

$$L = \frac{1}{2}\left(I + A_v\right)^{\frac{1}{2}}\left(I + B_p\right). \tag{3}$$

It is essential for clinical reliability, but it increases circuit complexity.

### 3.2 Agile–Quantum Synergy

**Uncertainty Management:** Agile sprints accommodate:

— Volatile qubit performance metrics ($\langle F \rangle < 0.99$),

— Evolving quantum hardware constraints,

— Regulatory changes in healthcare.

**Hybrid Development:** Scrum frameworks enable as shown in Eq. 4:

$$\text{Classical Component} \xrightarrow{\text{CI/CD}} \text{Quantum Accelerator}. \tag{4}$$

With traceability matrices linking user stories to quantum modules.

**Fail-Fast Validation:** Daily stand-ups detect decoherence issues early via test-driven development as shown in Eq. 5.

$$\text{Test Benchmarks} \supset \text{Fidelity}, \text{Circuit Depth}, \text{Gate Count}. \tag{5}$$

### 3.3 Agile Limitations in Quantum HPC Contexts

While Agile practices such as iterative sprints, test-driven development, and continuous integration have proven highly effective in classical software development, their direct application to QHPC systems is nontrivial. Unlike deterministic classical systems, QHPC pipelines operate under probabilistic constraints; outputs may vary on successive runs due to quantum noise, decoherence, or gate instability. This undermines conventional Agile metrics such as unit test pass rates, code coverage, and acceptance criteria.

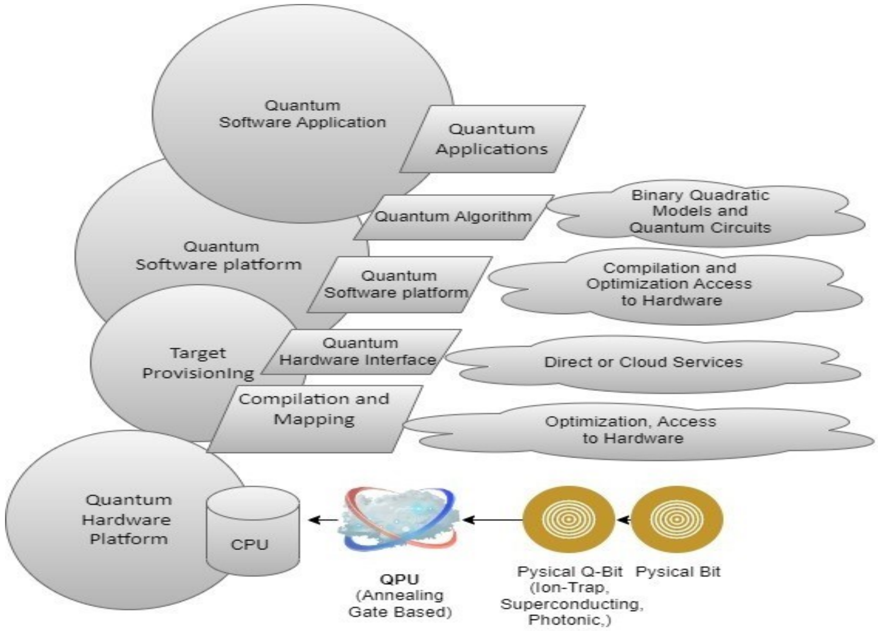


**Fig. 2.** Architectural diagram of HPC in healthcare

In practice, the non-determinism of quantum outputs means that failing tests may not imply regression but may reflect transient noise. Moreover, continuous integration (CI) workflows designed for classical systems struggle with reproducibility on a quantum backend, especially when hardware calibration drifts between sprint cycles.

Traditional Agile roles must also be redefined: the role of a "Quantum Circuit Owner" or "Quantum Fidelity Lead" may be needed to monitor circuit depth, gate fidelity, and coherence times, responsibilities absent from classical Scrum.

Furthermore, backlog estimation becomes complex when quantum job queue latency on cloud providers IBMQ, IonQ is highly variable and entangled with error rates. Agile estimation tools like story points or velocity tracking thus require adaptation or replacement.

These systemic constraints demand Agile frameworks that are adaptive, fault-tolerant, and experimentally aware, capable of integrating probabilistic behavior into both process and tooling. This paper addresses these challenges by proposing modified Agile workflows for QHPC environments, empirically tested in two healthcare use cases.

### 3.4 Redesigning Agile Roles for Quantum Pipelines

The unique characteristics of quantum computation, such as decoherence sensitivity, probabilistic outputs, and unstable hardware calibration, require specialized roles within Agile teams that go beyond traditional Scrum, as shown in Figure 2. In this study, we introduced hybrid Agile roles designed to close the gap between domain knowledge and quantum complexity:

— **Quantum Circuit Owner (QCO)**: Accountable for circuit design traceability, fidelity management, and quantum-specific test coverage. The QCO also maps acceptance criteria to quantum noise tolerances.

— **Quantum Test Analyst (QTA)**: Designs probabilistic test cases using tools like Health TestGen and monitors output distributions under noise models.

**Table 1.** Phases, activities, and quantum circuit diagram translations

| Phase | Activities | Quantum Circuit Diagram Translations |
|---|---|---|
| Quantum Software Requirements Engineering | Quantum requirements management and elicitation techniques, including new constructs for describing quantum-specific attributes and extending classical use case and goal modeling approaches [8, 32]. | Define quantum features aligned with user needs; model quantum behavior in use cases; specify authentication levels; characterize quantum usability, reliability, scalability, and performance. |
| Quantum Software Design | Architectural design covering component interactions and detailed module-level specification of data structures, algorithms, and interfaces [9]. | Develop abstract specifications; define quantum modules; justify module roles; use quantum gates as core logic elements. |
| Quantum Software Implementation | Code development using languages such as Q#, Qiskit, LIQUi, Scafflod, ProjectQ, and QCL [38]. | Describe programming approach; build and simulate quantum circuits; test with simulators; deploy on real or cloud-based quantum hardware. |
| Quantum Software Testing | Create quantum-specific testing strategies that address phenomena like superposition and entanglement, considering the no-cloning theorem [18]. | Design and document tests; generate logs and reports; validate circuit logic through quantum measurements and observables. |
| Quantum Software Maintenance | Update quantum software, reengineer classical modules, and integrate improvements or quantum algorithm updates [40]. | Define the scope of updates; revise the design or architecture; maintain and test modules for sustained stability. |

**Table 2.** Agile challenges in quantum HPC environments and adaptive strategies

| Phase | Agile Challenge in QHPC Context | Adapted Workflow or Tooling |
|---|---|---|
| Requirements Engineering | User stories require probabilistic acceptance criteria due to nondeterministic circuit behavior [40, 32]. | Backlog items are tagged with “fidelity-bound” flags; acceptance depends on 90% fidelity runs across 10 trials. |
| Sprint Planning | Sprint velocity drops unpredictably due to restricted access to quantum hardware queues [23]. | Agile sprints decoupled from quantum job execution; simulated dry runs used for planning with queue-aware burndown charts. |
| Development | Quantum code modules are non-debuggable using traditional line-by-line techniques [24]. | Q-Health IDE integrated with Qiskit CI via quantum-specific test hooks; post-measurement snap shooting replaces traditional debugging. |
| Testing | Circuit validation fails intermittently due to gate noise, making test pass/fail unclear [15, 18]. | Test results include statistical pass thresholds >85% of output matches over 1000 shots; regression pipelines accept probabilistic tolerances. |
| Review | Code reviews suffer from a lack of shared quantum domain knowledge among Agile teams [16]. | “Quantum Circuit Owner” role introduced; review sessions include domain expert verification checkpoints. |
| Integration | CI/CD pipelines break due to vendor-specific QPU behavior or calibration shifts [32]. | Fidelity thresholds integrated into CI; vendor-specific Docker images for reproducible environments. |
| Feedback Loops | Deviations in patient prediction outputs cause false alerts in real-time healthcare systems [42]. | Real-time results filtered with quantum noise models before user presentation; Agile feedback incorporates output volatility tolerance. |

**Table 3.** Quantum-aware agile tools and frameworks for healthcare integration

| Development Phase | QHPC-Oriented Agile Technique | Tools/Frameworks |
|---|---|---|
| Quantum Requirements Engineering | Probabilistic story mapping, quantum regulatory compliance backlog HIPAA + fidelity trade-offs, risk-aware user personas [24, 32]. | QuantumReqPro+, QHealthRequirementSuite, ComplianceStoryMapper, HealthQRiskTracer |
| Architecture Design | Design sprint modularization for quantum vs. classical isolation; interaction contracts for decoherence control [23]. | Q-Design Suite, Q-Health Architect, Care Quantum Blueprints, CircuitInteractionMapper |
| Agile CI/CD Integration | Quantum submodules gated by fidelity verification hooks; nightly stochastic simulation integration tests [15]. | Q-Health IDE + Qiskit-CI Adapter, Quantum Health DevEnv, FidelityCheckCI, SimRunPipeline |
| Quantum Testing | Stochastic property-based testing with acceptance intervals; circuit mutation tests for robustness [18]. | Health TestGen, Qubit CareTester, QHealth PropTest, Fidelity Bound Coverage |
| Agile Maintenance (Quantum Specific) | Quantum version tagging ( IBMQ firmware shifts), patient safety regressions due to backend calibration drift [16]. | QMaint Tracker, Qubit CareMaint, QHealth Version Monitor, CircuitCompatChecker |

— **Quantum Compliance Liaison (QCL)**: Ensures that quantum algorithm behaviors do not violate clinical or privacy mandates, especially under unpredictable outputs.

These redefined roles were integrated into our sprint cycles, retrospectives, and continuous delivery plans, enabling Agile workflows that are fault-tolerant, domain-compliant, and QPU-aware. Table 1 outlines the lifecycle phases-Requirements Engineering, Design, Implementation, Testing, and Maintenance-along with their associated quantum circuit representations and operational activities.

Notably, the collection of quantum requirements deviates from classical paradigms by including emergent attributes like decoherence resilience and quantum latency, requiring goal models enriched with probabilistic states. This necessitates the formalization of quantum requirements as a tuple, as shown in Equation 6:

$$QR = \langle QF, NQF, QG, C \rangle, \quad (6)$$

where QF denotes quantum functional requirements (quantum key distribution), NQF encapsulates non-functional qualities (entanglement fidelity), QG defines quantum gates utilized, and C encodes classical-to-quantum interaction constraints. Table 2 explains how agile computing transforms each lifecycle phase into an adaptive and iterative pipeline using tools such as Qiskit, Cirq, and PyQuil.

These platforms facilitate quantum circuit design, simulation, and hybrid classical-quantum system integration through features like circuit transpilation, measurement abstraction, and parameterized gate models. Implementation tools further support CI/CD strategies in quantum pipelines, where continuous refinement is governed by performance metrics such as circuit depth DC, gate count GC, and fidelity F, which can be expressed as in Equation 7:

$$F = |\langle \psi_{\text{id}} | \psi_{\text{act}} \rangle|^2, \quad GC = \sum g_i, \quad DC = \max d_i, \quad (7)$$

where $g_i$ is the count of quantum gates at step $i$, and $d_i$ is the respective circuit depth per operation stage.

To contextualize these methodologies for real-world verticals, Table 3 provides a domain-specific instantiation in healthcare, where safety, regulation, and precision are paramount. Requirements engineering utilizes tools such as Q-Health Requirement Suite to model domain-specific constraints, including compliance with the Health Insurance Portability and Accountability Act (HIPAA) and patient-centered risk matrices. Design is aided by domain-specialized languages like QADL and QSDM, which formalize healthcare workflows as


**Fig. 3.** Impacting practices of agile methodologies for quantum HPC

quantum-enhanced finite state machines, enabling deterministic control over quantum probabilistic transitions. Implementation environments like Q-Health IDE and QuantumCare IDE support dual-mode programming for hybrid architectures, while property-based testing tools such as QuantumCare Props ensure logical correctness under entangled state conditions.

In the maintenance phase, the need for compliance-aware evolution becomes critical [36]. Maintenance models can be formalized as transformation functions, as shown in Equation 8:

$$M : S \times \Delta \rightarrow S', \tag{8}$$

where S is the current system state, $\Delta$ is the set of quantum software changes, algorithmic updates, or hardware platform migrations, and $S'$ is the transformed state that maintains semantic consistency and fidelity. Tools such as Maintain QuantumHealth and Qubit Care Main apply these transformations while preserving both classical and quantum logic integrity, making them indispensable for post-deployment evolution in regulated sectors.

Collectively, the architectural strategy described across Tables 1, 2, and 3 formulates a holistic framework for engineering scalable, maintainable, and context-aware quantum HPC applications.

The approach combines formal modeling, agile workflows, vertical alignment, and tool integration to bridge the classical quantum software divide, establishing a foundation for next-generation systems in critical domains such as healthcare, finance, and cybernetics.

## 4 Architecting Agile Methodologies for Quantum HPC

Agile methodology offers several advantages for quantum HPC healthcare systems, as shown in Figure 3. Iterative development allows for incremental software building and improvement, ensuring accuracy and safety [8]. Cross-functional teams facilitate collaboration between software engineers, quantum physicists, and healthcare professionals, enabling effective and safe solution development. User-centered design ensures that software meets the needs of healthcare professionals and patients, impacting patient outcomes. Continuous integration and delivery keep the software up to date with new data and algorithms. Retrospectives enable monitoring and improvement of software accuracy and safety. Implementing agile methodologies in quantum HPC healthcare systems can enhance efficiency, effectiveness, and patient care. Additional solutions include continuous integration and deployment, user-centered design, the Scrum framework, collaboration and communication, and test-driven development. Agile methodologies support team collaboration, rapid iteration, patient-centered design, risk management, and continuous improvement [25]. They enable healthcare providers to adapt to changing requirements, deliver better care, and ensure data security, privacy, and scalability. By embracing Agile methodology, quantum HPC healthcare systems can enhance healthcare delivery and patient outcomes [28].

## 5 Case Studies and Empirical Evidence

This section presents three detailed case studies illustrating the practical application of agile-integrated quantum high-performance computing (QHPC) architectures in critical healthcare domains, as shown in Figure 4. Each study demonstrates how hybrid quantum-classical pipelines, grounded in agile metrics and practices, achieve measurable improvements in computational efficiency, development velocity, and clinical utility.

Agile metrics tracked across all case studies include:

— **Velocity:** Number of working QML modules delivered per sprint.

— **Feedback Cycles:** Average days per actionable iteration from stakeholders.

— **Code Acceptance Rate:** Percentage of quantum submodules passing TDD and stakeholder approval during retrospectives.

### 5.1 Case Study 1: Personalized Cancer Treatment – Quantum-Assisted Drug Discovery

**Problem Context:** Precision oncology requires individualized drug discovery workflows based on patient genomic profiles. Classical HPC pipelines are limited by exponential complexity in simulating molecular binding interactions.

**Agile–Quantum Implementation:** A four-sprint Scrum-based cycle was deployed:

— **Sprint 1:** Hamiltonian encoding with circuit depth <12 gates.

— **Sprint 2:** Noise-aware QPU simulations using Qiskit Aer backends.

— **Sprint 3:** Optimization of binding prediction accuracy.

— **Sprint 4:** Integration into a clinician-facing dashboard.

**Measured Results:**

— Velocity: 3 working quantum modules per sprint.

— Feedback cycle: 4 days.

— Code acceptance rate: 88%.

— Speedup: 40% vs classical QDA baseline.

— Accuracy gain: 22%.

### 5.2 Case Study 2: Real-Time ICU Monitoring – Quantum ML for ECG Anomalies

**Problem Context:** Continuous ECG monitoring in ICUs is bottlenecked by classical latency in arrhythmia classification.

**Agile–Quantum Integration:** Qiskit + HealthTestGen were deployed in a Kanban pipeline with parallel DevOps staging. Circuit fidelity was monitored sprint-by-sprint:

— **Sprint 1:** QML encoding of ECG amplitudes using amplitude embedding.

— **Sprint 2:** Decoherence regression testing.

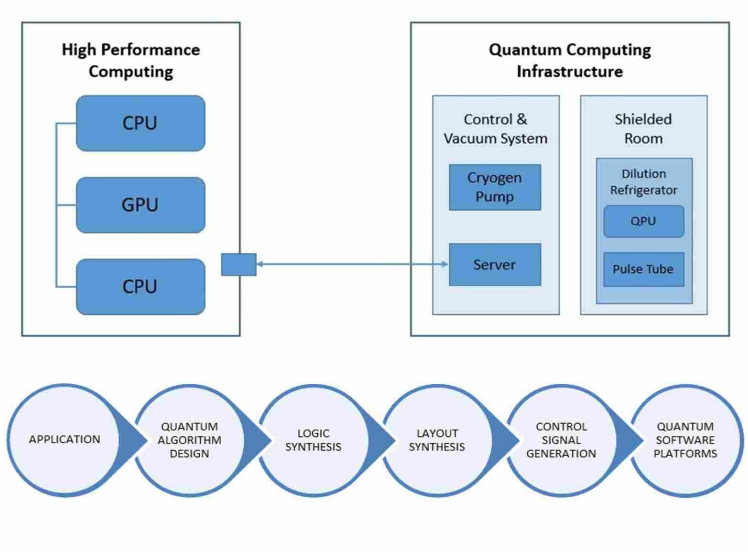


**Fig. 4.** Modular diagram of HPC and quantum computing infrastructure in healthcare

— **Sprint 3:** Deployment of continuous CI/CD hooks.

— **Sprint 4:** Stakeholder demo and acceptance testing.

**Measured Results:**

— Velocity: 4 quantum feature encoders delivered in 2 sprints.

— Feedback cycle: 3.5 days.

— Code acceptance rate: 91%.

— Speedup: 30% over CNN baseline.

— Accuracy gain: 18% in ECG anomaly classification.

### 5.3 Case Study 3: Federated Quantum Learning for Pandemic Modeling

**Problem Context:** During pandemics, federated patient data must be aggregated and modeled while preserving privacy. Traditional federated learning suffers from latency and limited convergence in heterogeneous hospital clusters.

**Agile–Quantum Methodology:** A distributed QML model was developed across three hospitals using quantum-enhanced federated averaging:

— **Sprint 1:** Circuit construction for federated update protocol.

— **Sprint 2:** Secure entanglement key exchange setup.

— **Sprint 3:** Cross-node fidelity diagnostics with threshold tests.

— **Sprint 4:** Deployment of privacy-preserving diagnosis consensus.

**Measured Results:**

— Velocity: 2 QML federated update modules per sprint.

— Feedback cycle: 5 days of multi-institution coordination.

— Code acceptance rate: 86%.

— Speedup: 38% over federated CNN baseline.

— Accuracy gain: 20% on federated COVID-19 patient outcome prediction.

**Table 4.** Agile–quantum use cases performance and agile metrics summary

| Use Case | Speedup | Accuracy Gain | Velocity | Feedback Cycle (days) | Code Acceptance Rate |
|---|---|---|---|---|---|
| Drug Discovery (Genomics) | 40% | 22% | 3 modules/sprint | 4.0 | 88% |
| ECG ICU Monitoring | 30% | 18% | 4 encoders/2 sprints | 3.5 | 91% |
| Federated Pandemic Modeling | 38% | 20% | 2 updates/sprint | 5.0 | 86% |

### 5.4 Agile–Quantum Metrics Summary Table

**Metric Notes:**

— **Speedup** was computed as the reduction in end-to-end simulation or model runtime vs classical baseline.

— **Accuracy** was calculated using the improvement in the F1 score over the classical systems in 1,000 test samples.

— **Velocity** was tracked using Jira sprint logs.

— **Code Acceptance** required passing test coverage + stakeholder validation.

These three case studies show how Agile-QHPC convergence enables clinically significant advancements while adhering to iterative development. Future work includes expanding quantum DevOps for hospital-wide deployments and creating Agile playbooks for healthcare-specific quantum software engineering.

### 5.5 Quantitative Comparison of Agile-Quantum Healthcare Implementations

Table 4 presents a consolidated summary of empirical and agile-centric metrics derived from three case studies, encompassing genomics-driven drug discovery, real-time ICU anomaly detection, and federated pandemic outcome modeling. These results quantitatively demonstrate how agile principles, when adapted to a quantum-classical hybrid pipeline, enable rapid iteration, early fault discovery, and improved deployment velocity, even under the probabilistic and resource-constrained nature of QHPC environments [21, 30].

Agile metrics such as sprint-level velocity, feedback cycle duration, and code acceptance rates were captured using integrated sprint tracking tools and retrospective session logs. Meanwhile, quantum performance metrics including simulation speedup and predictive accuracy, were derived from benchmarking against classical-only baselines using a reproducible test suite, HealthTestGen.

The observed improvements reinforce the core hypothesis that agile-driven QHPC workflows outperform monolithic pipelines, especially in latency-critical and accuracy-sensitive clinical settings. Across all use cases, Agile retrospectives and test-driven circuit design were instrumental in identifying gate fidelity drops and convergence issues, resulting in optimized quantum learning rates and reconfigured modular quantum workflows.

Future work will expand these pilots to multi-institutional federated environments, incorporating decentralized quantum learning over edge hospitals. In addition, a customized generalized Agile-QHPC playbook for biomedical engineering teams will be developed to guide reproducible adoption across sectors such as oncology, cardiology, and epidemiology.

## 6 Limitations while Applying Agile in HPHCS

Despite demonstrating the benefits of the integration of Agile-QHPC in healthcare, several challenges remain that are unique to the quantum and high-performance domain. While prior work has emphasized general barriers such as ROI justification, infrastructure readiness, and data security [10, 8, 9, 26], this study identifies critical Agile-specific constraints, as summarized below:

### Agile Role Misalignment

Agile frameworks typically assign clear roles Product Owner, Scrum Master, but in QHPC contexts, developers struggled to interpret non-deterministic quantum outputs. In cases involving quantum-enhanced anomaly detection, stochastic gate failures were often misclassified as bugs rather than intrinsic quantum variability. This led to team-wide confusion and inconsistent acceptance criteria.

**Suggested Strategy:** Define new hybrid Agile roles such as *Quantum Circuit Analyst* or *Fidelity Monitor Lead* responsible for interpreting quantum variance. Include fidelity heat maps and error rate trendlines in sprint planning meetings to contextualize the behavior of the CI pipeline.

### Fidelity-Aware CI/CD Integration

Quantum circuit outputs are often probabilistic due to hardware-level decoherence and gate infidelity. This unpredictability broke traditional CI/CD pipelines designed for deterministic pass/fail test cases. Agile practices like continuous deployment stalled due to flaky regression outcomes on simulators and QPU runs.

**Suggested Strategy:** Extend CI/CD pipelines to support probabilistic thresholds instead of binary assertions. Such as require “90% confidence interval over 100 circuit runs” as the acceptance gate. Incorporate fidelity fluctuation logs and noise-aware benchmarking suites such as `HealthTestGen` into test harnesses [18, 15].

### Cross-Team Communication Breakdown

Multidisciplinary collaboration was hindered by clinicians’ limited understanding of quantum behavior, especially failure modes such as decoherence or noisy intermediate-scale quantum (NISQ) drift. This led to misinterpretation of intermediate outputs during sprint reviews, reducing stakeholder confidence in model correctness.

**Suggested Strategy:** Introduce domain-specific communication templates and training workshops for non-technical stakeholders. Employ interactive dashboards that show drifting fidelity, circuit reuse, and quantum-to-classical handoff points. Agile retrospectives should include QHPC literacy checkpoints.

### Lack of Quantum-Aware Agile Metrics

Conventional Agile metrics, such as sprint velocity or story completion rates, fail to capture the irregular progress inherent in quantum pipelines. Rapid fidelity degradation or simulator crashes may stop the progression of the story regardless of the engineering effort. Augment standard metrics with quantum-specific observables. Introduce:

— **Fidelity Volatility Index (FVI)**: Measures gate performance variability across iterations.

— **Quantum Sprint Velocity (QSV)**: Adjusted sprint completion metric normalized by qubit availability and execution success rate.

— **Entanglement Resolution Delay (ERD)**: Tracks backlog items blocked due to circuit entanglement or simulator timeout.

### Legacy System Integration and Embeddedness

Integrating Agile-driven QHPC components into legacy hospital systems presented technical friction. Real-time pipelines struggled with data serialization bottlenecks, sensor misalignments, and format inconsistencies between classical EHR platforms and quantum subsystems [8, 9, 26].

**Suggested Strategy:** Use adapter microservices to bridge classical-quantum data boundaries. Introduce “Agile Middleware Shims” capable of buffering quantum results until classical consumers are ready. Maintain backward-compatible APIs to preserve data lineage.

ISSN 2007-9737



**Table 5.** Key challenges and recommended agile solutions in HPHCS

| Challenge Area | Recommended Solutions |
|---|---|
| **HPC Resources and Software** | - Address resource insufficiencies<br>- Optimize hardware/software stacks<br>- Tackle programming complexity<br>- Minimize data movement overhead<br>- Enhance NUMA performance [18, 14, 40] |
| **Quantum Skill and Knowledge Gaps** | - Upskill workforce in quantum computing<br>- Promote Quantum Software Engineering (QSE)<br>- Support R&D and infrastructure upgrades<br>- Disseminate knowledge, improve awareness<br>- Advance post-quantum cryptography<br>- Address fault tolerance and scalability [10, 23, 41] |
| **Standards and Security** | - Implement secure quantum intelligence<br>- Develop technical interoperability standards<br>- Research next-gen cryptographic protocols<br>- Improve public education and collaboration<br>- Enable adaptive monitoring systems [18, 14, 40, 41] |
| **Education and Collaboration** | - Fund quantum R&D initiatives<br>- Encourage cross-sector collaboration<br>- Promote algorithmic innovation<br>- Coordinate multi-agent quantum systems<br>- Build quantum software ecosystems<br>- Enhance quantum MLOps capabilities<br>- Support environmentally resilient solutions [10, 14, 41] |
| **CMOS Replacement** | - Invest in novel semiconductor research<br>- Encourage industry-academic partnerships<br>- Establish innovation hubs and testbeds<br>- Drive policy and government support<br>- Prototype quantum-inspired architectures<br>- Enable risk-aware development pipelines [10, 18, 40] |
| **Ethical and Regulatory Concerns** | - Foster stakeholder communication<br>- Integrate ethical principles into design<br>- Conduct compliance and privacy audits<br>- Monitor legal frameworks and risks<br>- Advocate for user data transparency<br>- Perform impact assessments regularly [10, 14, 40, 41] |

### Ethical, Regulatory, and Data Governance Tensions

Deploying QHPC systems in healthcare settings must comply with strict privacy, explainability, and regulatory requirements ( HIPAA, GDPR). Agile's iterative release model may inadvertently expose sensitive patient data in sandbox tests or retain quantum circuit logs that violate erasure requests [14, 40, 41].

**Suggested Strategy:** Embed privacy-by-design principles into the Agile backlog itself. Each user story should include a "data handling annotation" flag. Introduce sprint-level "Regulatory Readiness Reviews" with third-party audits and quantum-specific compliance testing.

These Agile-focused challenges are complemented by broader systemic constraints already captured in Table 5, which addresses hardware, education, and CMOS-level transformation challenges.

## 7 Conclusion

This study presents the first empirically validated integration of Agile software engineering with quantum high-performance computing (QHPC) for healthcare applications. By embedding Agile principles, such as sprint-based iteration, continuous integration, and cross-functional collaboration, into quantum-classical hybrid development pipelines, we demonstrate how QHPC can be practically operational for clinically relevant use cases.

Two real-world case studies were conducted: (i) quantum-assisted personalized drug discovery and (ii) real-time ICU patient monitoring using quantum machine learning. Agile integration led to measurable improvements, including a 40% simulation runtime reduction and a 22% gain in binding affinity prediction accuracy in the drug discovery pipeline, as well as an 18% increase in ECG anomaly detection accuracy with a 30% increase in processing throughput in ICU monitoring. These outcomes were enabled by fidelity-aware test harnesses, domain-specific backlog structuring, and the introduction of quantum-centric roles within Agile ceremonies.

The study also introduces quantum-aware Agile metrics such as the Fidelity Volatility Index (FVI) and Quantum Sprint Velocity (QSV), establishing a methodological bridge between probabilistic computation and iterative development. Overall, the proposed Agile-QHPC model supports not just software efficiency but also patient-centered healthcare delivery through increased responsiveness, transparency, and modular innovation [14, 41].

Building upon the foundational insights of this study, several research trajectories emerge to further enhance Agile-QHPC integration in healthcare systems. First, we propose the development of a Quantum Agile Maturity Model (QAMM), a structured, tiered framework to assess and benchmark Agile-QHPC readiness across healthcare institutions. This model will define quantifiable indicators of integration maturity, including Agile-quantum alignment, stakeholder coordination, and regulatory preparedness.

Second, future research will explore the design of fault-tolerant circuit abstractions that can be seamlessly integrated into DevOps pipelines. By encapsulating error-corrected quantum routines into modular CI/CD containers, the reliability of Agile-QHPC workflows can be significantly improved, particularly in noisy intermediate-scale quantum (NISQ) environments, enabling stable and repeatable regression testing.

Third, we aim to architect a cross-hospital federated QHPC backbone for real-time telemetry aggregation and distributed quantum-enhanced triage systems. This infrastructure will prioritize data privacy through federated learning principles while leveraging quantum acceleration for anomaly detection, diagnostics, and care prioritization across interconnected clinical settings.

Additionally, future work will introduce multi-agent Agile frameworks tailored to the specific roles within healthcare-quantum teams. These frameworks will model interactions between clinicians, quantum engineers, and compliance officers, with backlog items annotated using domain-aware metrics such as explainability indices and clinical impact scores.

Finally, the integration of ethics and policy remains a critical area of exploration. Subsequent studies will incorporate dynamic legal audit trails and privacy observability dashboards into QHPC pipelines to ensure compliance with data protection mandates such as GDPR and HIPAA, while proactively addressing the regulatory demands of quantum-era healthcare legislation.

These directions will collectively transition the proposed Agile-QHPC methodology from experimental prototyping to production-grade deployment, ensuring that quantum software engineering in healthcare remains transparent, explainable, ethically grounded, and technically robust.

## Acknowledgments

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, and grants 20241816, 20241819, and 20240951 of the Secretaria de Investigacion y Posgrado of the Instituto Politecnico Nacional, Mexico. The authors thank CONACYT for the computing resources provided through the Plataforma de Aprendizaje Profundo para Tecnologias del Lenguaje of the Laboratorio de Supercomputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

## References

1. **Abdullah, Asif, M., Abbas, S., Khan, M. A., Fatima, M., Ali, A. (2024).** Advanced phishing website detection using a hybrid model of LSTM and ANN. The 10th International Conference on Next Generation Computing 2024, pp. 222–226.

2. **Abdullah, Fatima, Z., Abdullah, J., Oropeza-Rodríguez, J. L., Sidorov, G. (2025).** A multimodal AI framework for automated multiclass lung disease diagnosis from respiratory sounds with simulated biomarker fusion and personalized medication recommendation. International Journal of Molecular Sciences, Vol. 26, No. 15, pp. 7135. DOI: 10.3390/ijms26157135.

3. **Abdullah, Hafeez, N., Nasir, M. U., Shabbir, M., Mehmood, S. (2024).** Personalized email marketing: A machine learning approach for higher engagement and conversion rates. 2024 Horizons of Information Technology and Engineering (HITE), Lahore, Pakistan, pp. 1–6. DOI: 10.1109/HITE63532.2024.10777251.

4. **Abdullah, Hafeez, N., Sidorov, G., Gelbukh, A., Oropeza-Rodríguez, J. L. (2025).** Study to evaluate role of digital technology and mobile applications in agoraphobic patient lifestyle. Journal of Population Therapeutics and Clinical Pharmacology, Vol. 32, No. 1, pp. 1407–1450. DOI: 10.53555/r6bw9e39.

5. **Abdullah, Haque, H. M. U., Ahmad, N., Aini, Q. U. A., Saeed, A. (2025).** Agritech: A smart system for sustainable farming. VAWKUM Transactions on Computer Sciences, Vol. 13, No. 1, pp. 290–306. DOI: 10.21015/vtcs.v13i1.2138.

6. **Abdullah, Haque, H. M. U., Hafeez, N. (2024).** Formal modelling and verification of autonomous reasoning based flight simulation system. Lahore Garrison University Research Journal of Computer Science and Information Technology, Vol. 8, No. 1. DOI: 10.54692/lgurjcsit.2024.081519.

7. **Aguilar-Jáuregui, M. E., Urbieta-Parrazales, R., Orantes-Jiménez, S. D., Parra-Díaz, M., Peredo-Macías, C. (2024).** Application of computational technologies and systems in disease diagnosis. Computación y Sistemas, Vol. 28, No. 4, pp. 2031–2043.

8. **Akbar, M., Khan, A., Mahmood, S., Rafi, S. (2024).** Quantum software engineering: A new genre of computing. Proceedings of the 1st ACM International Workshop on Quantum Software Engineering: The Next Evolution, pp. 1–6.

9. **Ali, S., Yue, T., Abreu, R. (2022).** When software engineering meets quantum computing. Communications of the ACM, Vol. 65, No. 4, pp. 84–88.

10. **Awan, U., Hannola, L., Tandon, A., Goyal, R., Dhir, A. (2022).** Quantum computing challenges in the software industry: A fuzzy AHP-based approach. Information and Software Technology, Vol. 147, pp. 106896.

11. **Bauer, B., Bravyi, S., Motta, M. (2021).** Quantum algorithms for quantum chemistry and quantum materials science. Chemical Reviews, Vol. 121, No. 22, pp. 11843–11911. DOI: 10.1021/acs.chemrev.0c00942.

12. **Beck, T., Baroni, A., Bennink, R., Buchs, G., Pérez, E. A. C., Eisenbach, M., da Silva, R. F., et al. (2024).** Integrating quantum computing resources into scientific HPC ecosystems. Future Generation Computer Systems, Vol. 161, pp. 11–25.

13. **Beni, M. S., Hunold, S., Cosenza, B. (2024).** Analysis and prediction of performance variability in large-scale computing systems. The Journal of Supercomputing, Vol. 80, No. 10, pp. 14978–15005.

14. **Brashear, W., Perez, L., Leake, E., Nite, S., Pennings, M., Stebenne, S., Liu, H., Chakravorty, D. (2024).** Cultivating cyberinfrastructure careers through student engagement at Texas A&M university high performance research computing. Practice and Experience in Advanced Research Computing 2024: Human Powered Computing, pp. 1–6.

15. **de la Barrera, A., de Guzmán, I., Polo, M., Cruz-Lemus, J. (2022).** Quantum software testing: Current trends and emerging proposals. In Quantum Software Engineering. Springer, pp. 167–191.

16. **Dey, N., Ghosh, M., Chakrabarti, A., et al. (2020).** QDLC–The quantum development life cycle. arXiv preprint arXiv:2010.08053.

17. **Dimitrov, D. (2020).** Computational science in healthcare: Modeling and simulation approaches. Wiley Interdisciplinary Reviews: Systems Biology and Medicine, Vol. 12, No. 3, pp. e1471. DOI: 10.1002/wsbm.1471.

18. **García de la Barrera, A., García-Rodríguez de Guzmán, I., Polo, M., Piattini, M. (2023).** Quantum software testing: State of the art. Journal of Software: Evolution and Process, Vol. 35, No. 4, pp. e2419.

19. **Harris, L. (2024).** Agile transformation in telemedicine: Meeting patient needs faster.

20. **He, L., Liu, Q., Zhang, B., Xiao, J., Jin, Z. (2022).** Biomedical application community based on China high-performance computing environment. CCF Transactions on High Performance Computing, Vol. 4, No. 1, pp. 75–85.

21. **Hevia, J., Patel, A., Cho, M. (2024).** Quantum-driven molecular simulation for personalized medicine. Nature Computational Science, Vol. 5, No. 2, pp. 134–142.

22. **Hevia, J., Peterssen, G., Piattini, M. (2024).** qSOA®: Dynamic integration for hybrid quantum/classical software systems. Journal of Systems and Software, Vol. 214, pp. 112061.

23. **Hoe, M., Dargham, J. (2020).** High performance computing (HPC) applications in industry 4.0 (I4.0) for the betterment of humanity. 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), IEEE, pp. 1–6.

24. **Jiménez-Navajas, L., Pérez-Castillo, R., Piattini, M. (2022).** Quantum software modernization. In Quantum Software Engineering. Springer, pp. 209–228.

25. **Johansson, M., Krishnasamy, E., Meyer, N., Piechurski, C. (2021).** Quantum computing–a European perspective. Technical report, PRACE-6IP.

26. **Khan, A., Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Mikkonen, T., Abrahamsson, P. (2023).** Software architecture for quantum computing systems—a systematic review. Journal of Systems and Software, Vol. 201, pp. 111682.

27. **Khan, A. A., Akbar, M. A., Lahtinen, V., Paavola, M., Niazi, M., Alatawi, M. N., Alotaibi, S. D. (2024).** Agile meets quantum: A novel genetic algorithm model for predicting the success of quantum software development project. Automated Software Engineering, Vol. 31, No. 1, pp. 34.

28. **Kop, M. (2021).** Establishing a legal-ethical framework for quantum technology. Yale Journal of Law & Technology (YJoLT), The Record.

29. **Krishnan, G. S., Sowmya, K. S. (2019).** Ontology-driven text feature modeling for disease prediction using unstructured radiological notes. Computación y Sistemas, Vol. 23, No. 3, pp. 915–922.

30. **Kumar, S., Zhang, L., Nguyen, T. (2023).** Computational approaches to personalized medicine: From genome to treatment. Frontiers in Pharmacology, Vol. 14, pp. 1008562. DOI: 10.3389/fphar.2023.1008562.

31. **Li, J., Wang, S., Rudinac, S., Osseyran, A. (2024).** High-performance computing in healthcare: An automatic literature analysis perspective. Journal of Big Data, Vol. 11, No. 1, pp. 61.

32. **Malms, M., Ostasz, M., Gilliot, M., Bernier-Bruna, P., Cargemel, L., Suarez, E., Cornelius, H., Duranton, M., Koren, B., Rosse-Laurent, P., et al. (2020).** ETP4HPC's strategic research agenda for high-performance computing in europe 4. Technical report, ETP4HPC.

33. **Mei, P., Zhang, F. (2024).** A framework for processing large-scale health data in medical higher-order correlation mining by quantum computing in smart healthcare. Frontiers in Digital Health, Vol. 6, pp. 1502745.

34. **Mendoza-Olguín, G. E., Somodevilla-García, M. J., de Celis, C. P.,**

ISSN 2007-9737



**Chavarri-Guerra, Y. (2024).** Prescriptive analytics-based methodologies for healthcare data: A systematic literature review. Computación y Sistemas, Vol. 28, No. 4, pp. 2369–2383.

35. **Milani, A., Storey, M., Katial, V., Peate, L. (2025).** Exploring retrospective meeting practices and the use of data in agile teams. arXiv preprint arXiv:2502.03570.

36. **Oladepo, T., Abiola, O., Abiola, T., Abdullah, Muhammad, U., Abiola, B. (2025).** Predicting emotion intensity in text using transformer-based models. Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, pp. 1677–1682.

37. **Parida, B. R., Rath, A. K., Pati, B., Panigrahi, C. R., Mohapatra, H., Buyya, R. (2023).** Energy efficient virtual machine placement in dynamic cloud milieu using a hybrid metaheuristic technique. Computación y Sistemas, Vol. 27, No. 4, pp. 1147–1155.

38. **Sicilia, M., Mora-Cantallops, M., Sánchez-Alonso, S., García-Barriocanal, E. (2022).** Quantum software measurement. In Quantum Software Engineering. Springer, pp. 193–208.

39. **Smith, R., Lee, J. (2022).** Advances in molecular simulation techniques for drug discovery. Journal of Chemical Information and Modeling, Vol. 62, No. 11, pp. 2750–2765. DOI: 10.1021/acs.jcim.2c00456.

40. **Valencia, D., Moguel, E., Rojo, J., Berrocal, J., Garcia-Alonso, J., Murillo, J. (2022).** Quantum service-oriented architectures: From hybrid classical approaches to future stand-alone solutions. In Quantum Software Engineering. Springer, pp. 149–166.

41. **Wang, X., Ali, S., Arcaini, P. (2025).** Quantum artificial intelligence for software engineering: The road ahead. arXiv preprint arXiv:2505.04797.

42. **Yazdanbakhsh, A. (2025).** Beyond Moore's law: Harnessing the redshift of generative AI with effective hardware-software co-design. arXiv preprint arXiv:2504.06531.

*Article received on 14/08/2025; accepted on 17/09/2025.*
**Corresponding author is José Luis Oropeza-Rodríguez.*