Towards an Analyzability Model for Hybrid Software

Ana Díaz-Muñoz[1,2][a], José A. Cruz-Lemus[2][b], Moisés Rodríguez[2][c] and Teresa Baldasarre[3][d]

[1] *AQCLab Software Quality, Ciudad Real, Spain*

[2] *University of Castilla-La Mancha, Ciudad Real, Spain*

[3] *University of Bari Aldo Moro, Bari, Italy*

Keywords: Software Quality Model, Analyzability, Hybrid Systems, ISO/IEC 25010, Quantum Circuits, Quantum Metrics, Empirical Study.

Abstract: This paper presents an initial validation of a software quality model focused on analyzability, aligned with the ISO/IEC 25010 standard. The model targets hybrid systems that integrate classical and quantum components, combining established classical metrics with quantum-specific measures designed to capture the complexity of quantum circuits. In this first empirical study, we evaluate only the quantum dimension of the model through a quasi-experimental setup involving computer engineering students. The results show that the model's analyzability levels correlate with participants' comprehension performance, supporting its utility in distinguishing circuit complexity. These findings offer promising evidence for further model refinement and lay the groundwork for future evaluations involving real-world hybrid code bases.

1 INTRODUCTION

Quantum computing has emerged as a promising solution to problems intractable for classical computers, offering significantly enhanced processing capabilities (Bernhardt, 2019)(Piattini et al., 2021). However, current approaches do not entirely replace classical computing; instead, they aim to integrate both paradigms into hybrid systems that combine the strengths of classical and quantum processing.

As these systems grow in complexity, ensuring their quality becomes critical. Software quality is a key factor in their development and adoption (Rodríguez et al., 2015), as issues related to maintainability and usability may hinder industrial implementation (Rodríguez et al., 2016). The ISO/IEC 25010 standard (ISO/IEC, 2011) provides a structured approach to software quality, with analyzability (a key sub-characteristic of maintainability) being particularly relevant due to the interdisciplinary nature and dual architecture of hybrid systems.

Although established models exist for classical software (Piattini et al., 2020) (Verdugo et al., 2024), there are still significant gaps in assessing the quality of systems that integrate both classical and quantum components. Recent work has begun addressing this by proposing hybrid models that combine classical metrics—such as cyclomatic complexity—with quantum-specific ones, like quantum cyclomatic complexity and circuit depth (Díaz-Muñoz et al., 2024a)(Díaz-Muñoz et al., 2024b). These approaches aim to expand our understanding of hybrid software quality and promote its adoption in industrial contexts.

Nevertheless, the practical application of such models remains limited, and empirical validations are scarce. In particular, no consolidated methodology has yet emerged to operationalize integrating classical and quantum aspects within a unified and usable quality model.

This work represents a first step toward validating and refining the quantum part of a previously developed hybrid quality model (Díaz-Muñoz et al., 2024a)(Díaz-Muñoz et al., 2024b), which integrates classical and quantum metrics to assess analyzability and maintainability in hybrid systems. However, this initial empirical validation focuses exclusively on the quantum dimension of the model due to the lack of publicly available, integrated hybrid code bases. Through a quasi-experimental study with computer engineering students, we examine whether the ana-

[a] https://orcid.org/0000-0001-6515-8835

[b] https://orcid.org/0000-0002-0470-609X

[c] https://orcid.org/0000-0003-2155-7409

[d] https://orcid.org/0000-0001-8589-2850

Díaz-Muñoz, A., Cruz-Lemus, J. A., Rodríguez, M. and Baldasarre, T.
Towards an Analyzability Model for Hybrid Software.
DOI: 10.5220/0013533800004525
In *Proceedings of the 1st International Conference on Quantum Software (IQSOFT 2025)*, pages 97-104
ISBN: 978-989-758-761-0

lyzability levels indicated by the quantum metrics of the model align with participants' subjective understanding of quantum circuits. The insights gained from this study will contribute to refining the model and guiding future research toward a comprehensive framework for evaluating the quality of hybrid software systems.

2 RELATED WORK

The ISO/IEC 25010 quality model (ISO/IEC, 2011), part of the ISO/IEC 25000 series (SQuaRE) (ISO/IEC, 2014), establishes a structured framework for assessing the quality of software and systems. This standard defines eight fundamental quality characteristics encompassing various software evaluation dimensions: functionality, reliability, usability, efficiency, maintainability, portability, compatibility, and security. These characteristics are further divided into sub-characteristics, enabling a more detailed and specific assessment.

Among these, analyzability is a crucial sub-characteristic of maintainability. It evaluates how easily software can be understood, diagnosed, and modified, ensuring its long-term evolution without significant obstacles. In essence, analyzability determines how effectively the behavior and structure of software can be identified and comprehended, facilitating error detection, quality enhancements, and the integration of new functionalities.

In classical software, analyzability-related properties are essential for ensuring systems remain understandable, maintainable, and easy to modify. These properties focus on how software design and code organization impact developers' ability to interpret and alter systems efficiently. Table 1 presents the primary metrics used to evaluate these aspects (Rodríguez and Piattini, 2014).

These analyzability properties have proven effective in evaluating software quality in industrial environments. They are widely applied to identify problematic code areas and enhance maintainability and comprehension. The study in (Verdugo et al., 2024) provides an in-depth analysis of software evaluation and certification practices at the AQCLab laboratory [1] over the past 25 years. It highlights the practical use of classical quality metrics, such as analyzability, demonstrating their effectiveness in real-world industrial applications and emphasizing their continued relevance.

However, hybrid systems introduce new challenges due to integrating quantum components. The additional complexity of quantum computing —such as non-determinism, entanglement, and circuit depth— significantly impacts analyzability. These characteristics require new metrics to evaluate quantum circuits' complexity, understandability, and maintainability.

Several authors have proposed metrics for quantum software systems. For instance, Kumar (Kumar, 2023) formalized structural coverage criteria for quantum software testing. Other works have discussed metrics such as quantum circuit depth, gate complexity, and quantum cyclomatic complexity (Díaz-Muñoz et al., 2024b). However, these contributions focus exclusively on quantum systems and lack integration into broader software quality models.

Only a few studies have attempted to build unified models integrating classical and quantum perspectives. Our previous work (Díaz-Muñoz et al., 2024a) proposed a hybrid model that combines classical analyzability metrics with quantum-specific indicators. However, there is still a lack of systematic validation of the quantum part of such models, primarily through empirical studies in real environments.

This paper aims to fill that gap by empirically exploring the feasibility of applying the quantum properties of a hybrid quality model to evaluate the analyzability of software systems with quantum components. By doing so, we contribute to developing a structured and integrated framework for assessing quantum and hybrid software quality, which remains largely unexplored in current literature.

3 EXPERIMENTAL ANALYSIS

A controlled experiment was conducted to explore the feasibility of applying the proposed model in assessing the analyzability of quantum circuits. Although the model was originally conceived as hybrid —combining classical and quantum metrics— this first validation focuses exclusively on the quantum dimension. This focus is not only due to the lack of publicly available real-world hybrid code bases but also because the quantum part of the model had not yet been empirically validated on its own. Specifically, the study examined the relationship between the analyzability level of a set of quantum circuits and participants' performance in solving related tasks.

The experiment cannot be strictly controlled, as the sample was selected conveniently. Nonetheless, participants were randomly assigned to different treatments to minimize internal bias and enhance the validity of the results. Ethical considerations were also

[1] http://www.aqclab.es

Table 1: Classical metrics.

Metric	Description
Coding rules	Predefined coding guidelines that enhance code readability and comprehension. These include naming conventions, structural organization, and design principles. Adhering to these rules improves code clarity and reduces errors.
Code documentation	Essential for explicitly describing a program's functionality, objectives, and behavior. Inline comments, supplementary documents (e.g., `README` files), and design specifications enhance code comprehensibility, facilitating modifications by other developers.
Cyclomatic complexity	Introduced by McCabe, this metric quantifies the complexity of a program's control flow by counting the number of independent execution paths. Higher values indicate increased complexity, making code harder to analyze and modify. Reducing cyclomatic complexity improves software comprehension.
Package structuring	The organization of modules and packages directly influences analyzability. A well-structured system allows developers to identify relevant components without grasping the code base. Hierarchical organization and modularity enhance clarity and maintainability.
Class structuring	In object-oriented programming, clear and well-defined class responsibilities facilitate analyzability. The Single Responsibility Principle (SRP) and low-class coupling improve software organization and easier maintenance.
Method size	Excessively long methods complicate comprehension. Keeping methods concise and focused on a single functionality enhances readability. Best practices suggest limiting method length to 10–15 lines to simplify analysis and modifications.
Duplicate code	Code redundancy increases system complexity and complicates maintenance, as changes must be applied in multiple locations. Eliminating duplicate code through refactoring techniques significantly enhances analyzability and maintainability.

addressed: all participants took part voluntarily and signed informed consent forms by institutional guidelines.

In what follows, we provide a detailed description of the study's components.

3.1 Analyzability Model

This section introduces the proposed model as a preliminary approach for exploring the analyzability of hybrid systems. Rather than presenting a definitive assessment framework, this study applies the quantum part of the model to investigate its feasibility, identify potential refinements, and contribute to its iterative development.

The classical properties included in the model align with those outlined in the previous section regarding software analyzability. Additionally, the model incorporates specific quantum metrics to assess quantum circuits, aiming to capture key complexity and understandability factors introduced by quantum computing. These adapted quantum metrics are shown in Table 2.

It is important to note that some quantum properties in the model are derived from classical software metrics, such as quantum cyclomatic complexity, which adapts the classical definition to the quantum context. However, other metrics are newly defined specifically for quantum systems, such as circuit width, circuit depth, and auxiliary qubit usage, as these concepts have no direct equivalent in classical software due to the fundamentally different nature of quantum computation.

3.2 Relevant Variables

The experimental design incorporates the variables and influencing factors outlined below.

The **independent variable** in the experiment is the analyzability level of the quantum circuits, categorized into three levels:

- **Low Analyzability:** A circuit with a highly complex structure characterized by deep circuit depth, a high density of operations, and excessive use of auxiliary qubits, making it difficult to understand. This level is rated 1/5 according to the model.
- **Medium Analyzability:** A circuit of moderate complexity, exhibiting a balanced structure with a reasonable number of operations and qubits, rated

Table 2: Quantum metrics.

Metric	Description
Circuit width	This metric quantifies the number of qubits utilized in a quantum circuit at any moment. It reflects the quantum resources required to execute an algorithm. A larger circuit width generally increases complexity, making the circuit more challenging to comprehend and debug.
Circuit depth	This refers to the number of sequential layers of quantum gates applied to qubits before obtaining a result. Greater depth typically implies higher complexity, making the circuit more difficult to analyze and optimize. Additionally, deeper circuits have longer execution times and are more susceptible to errors due to qubit interference.
Gate complexity	This metric evaluates the overall complexity of the quantum gates used in a circuit. It considers both the type and number of gates applied. While some gates, such as Hadamard or phase gates, are relatively simple, others, like swap or controlled gates, introduce greater complexity. Many complex gates can make a circuit more difficult to interpret and analyze.
Conditional instructions	These are operations where a quantum gate's execution depends on the state of a control qubit. While they enable decision-making based on quantum information, they also increase circuit complexity. Many conditional instructions can make the circuit more intricate and harder to understand, as execution paths may vary dynamically.
Quantum cyclomatic complexity	Adapted from the classical software metric, this evaluates the complexity of a quantum circuit based on the number of independent execution paths it contains (Kumar, 2023). Higher values indicate a more complex structure, making the circuit harder to analyze and debug.
Measurement operations	These operations extract results from qubits and are fundamental in quantum computing. This metric counts how frequently measurements occur within a circuit. Many measurements, particularly those dependent on complex conditions, may indicate increased difficulty understanding the circuit's behavior.
Initialization and reset operations	These refer to operations that either prepare qubits in a specific state before execution or reset them afterward. While essential for maintaining computational coherence, excessive or complex use of these operations can contribute to overall circuit complexity, complicating analysis.
Auxiliary qubits	Additional qubits used to facilitate complex quantum operations without interfering with the main computation. A higher number of auxiliary qubits can indicate increased circuit complexity, introducing more dependencies that must be managed and understood.

3/5 according to the model.

- **High Analyzability:** A well-structured, optimized circuit designed to minimize complexity and enhance clarity, making it easier to comprehend. This level is rated 5/5 according to the model.

The **dependent variable** is the **average score obtained for each circuit**. This metric quantifies participants' ability to read, interpret, and analyze the circuit, reflecting their capacity to understand its behavior and the final states of the qubits.

Additionally, several **external factors that may impact the results** were considered, including:

- **Prior Experience in Quantum Programming:** Participants with greater exposure to quantum programming are expected to demonstrate a higher ability to analyze complex circuits.
- **Background in Classical Programming:** Familiarity with traditional programming paradigms may influence participants' speed and accuracy in understanding quantum circuits.
- **Educational Background:** A higher level of education is assumed to correlate with improved comprehension skills.

Another potential confounding factor is the difference in abstraction between classical and quantum programming. Unlike classical code, quantum circuits rely on probabilistic logic and visual-spatial reasoning, which may affect participants' understanding regardless of their general programming experience.

3.3 Research Assumptions

The primary aim of this experiment is to examine the relationship between the analyzability level of quantum circuits, as assessed using the proposed model, and the performance outcomes of the participants. The following hypotheses were formulated:

Null Hypothesis (H_0): There is no statistically significant correlation between the analyzability level of the quantum circuits and the average score obtained by the participants. In other words, the analyzability level of the circuit does not influence the participants' performance in understanding and analyzing the circuit.

Alternative Hypothesis (H_a): This hypothesis contradicts the null hypothesis, asserting that there is a statistically significant relationship between the analyzability level and the participants' performance.

These hypotheses aim to evaluate whether the model's analyzability rating corresponds with measurable differences in participants' ability to understand and evaluate the circuits. Rejecting the null hypothesis would indicate the model effectively captures meaningful complexity differences among quantum circuits.

3.4 Subjects

The experiment involved 109 computer engineering students from Aldo Moro University in Bari, Italy. Tables 3 and 4 visually depict the distribution of participants across various factors, such as age, gender, and educational background. Most participants are between 18 and 21 years old, with a higher proportion of male participants. Regarding educational background, many participants pursue their undergraduate degree after completing secondary education, while a smaller percentage report holding advanced degrees, such as a master's.

Table 3: Number of participants classified by age group and gender.

Age group	Male	Female	Others
18-21 years old	75	8	1
22-25 years old	19	1	0
26-30 years old	4	0	0
31-40 years old	1	0	0

Data analysis reveals a relatively homogeneous sample of gender, age, and educational background. In terms of gender, males represent the majority of participants, comprising 79.57% of the sample, while females make up 18.28%, and a small percentage identify with a different gender. In terms of age,

Table 4: Number of participants classified by education level and gender.

Education level	Male	Female	Others
High school	90	8	1
Bachelor's degree	8	1	0
Master's degree	1	0	0

75.27% of participants are between 18 and 21 years old, with significantly smaller proportions in the 22-25 and 26-30 age groups, representing 18.28% and 6.45%, respectively. As for educational background, the majority of participants (90.32%) have completed secondary education (High School), with 10.75% holding a university degree and 2.15% possessing a master's degree.

3.5 Earlier Preparation

Before the experiment, participants underwent introductory training covering the basics of quantum computing and quantum circuit design over several sessions. This training aimed to equip participants with the necessary knowledge to understand the concepts involved in the experimental tasks. Topics included fundamental principles of quantum computing, such as qubit usage and operation, quantum gates, and the measurement of quantum states. Additionally, participants were introduced to practical examples of constructing and simulating quantum circuits using tools like Qiskit and Quirk, which would be employed during the experiment.

To further support the participants' preparation, guided exercises reflecting the structure and tasks they would encounter during the experiment were also conducted, albeit with reduced difficulty. This approach was intended to ensure participants became sufficiently familiar with the core concepts and tools, thereby minimizing potential biases arising from a lack of prior knowledge. Despite this preparation, it must be acknowledged that participants' actual expertise levels were not formally assessed before the experiment. Therefore, variability in prior experience may influence the results and is considered a potential threat to internal validity.

3.6 Data Collection Approach

A detailed procedure was established for data collection, ensuring the random assignment of exercises and the traceability of participant responses. For this purpose, each participant received a personalized `PDF` document via email. This document, written as an individualized letter, contained three links to randomly assigned exercises to mitigate biases related to fatigue and learning effects.

Each exercise included two key components:

- **Qiskit Code:** Each exercise provided the code in the Qiskit quantum programming language, which participants were to use as a reference for analyzing the circuit's behavior.
- **Quirk Simulator:** As part of the task, participants were required to design the corresponding circuits for the Qiskit code they received in the Quirk simulator[2], an interactive tool that enables real-time visualization of qubit evolution.

The results of the exercises were gathered using `Google Forms`, designed to assess participants' ability to understand and analyze quantum circuits. The goal of the exercises was for participants to analyze the provided circuit and determine the final states of the qubits after its execution. These results would be entered into the forms as responses to specific questions.

The exercises were designed to assess participants' ability to determine the final quantum state of each qubit after circuit execution. Each task consisted of eight multiple-choice and short-answer questions, targeting core aspects of comprehension such as gate behavior, measurement interpretation, and identification of quantum interference. Each correct answer contributed one point to the total circuit score, with a maximum of eight points per circuit. After the data collection phase, a manual review was conducted to discard incomplete or invalid responses. The cleaned dataset was exported in `CSV` format and processed in `Excel` and Python for statistical analysis.

4 ANALYSIS AND REFLECTION ON RESULTS

This section provides an in-depth discussion of the experiment's results, aiming to place the findings in context and analyze their significance concerning the analyzability model for quantum circuits. The results are interpreted to assess the validity of the proposed hypotheses and to reflect on the implications and limitations of this initial validation. Additionally, the potential influence of these findings on future research and practical applications in the quantum computing field is discussed.

[2]https://algassert.com/quirk

4.1 Analysis of Data Distribution

A descriptive analysis was first performed on the scores obtained by participants, considering various demographic variables. This analysis helped identify patterns and preliminary differences relevant to interpreting the results of the statistical tests.

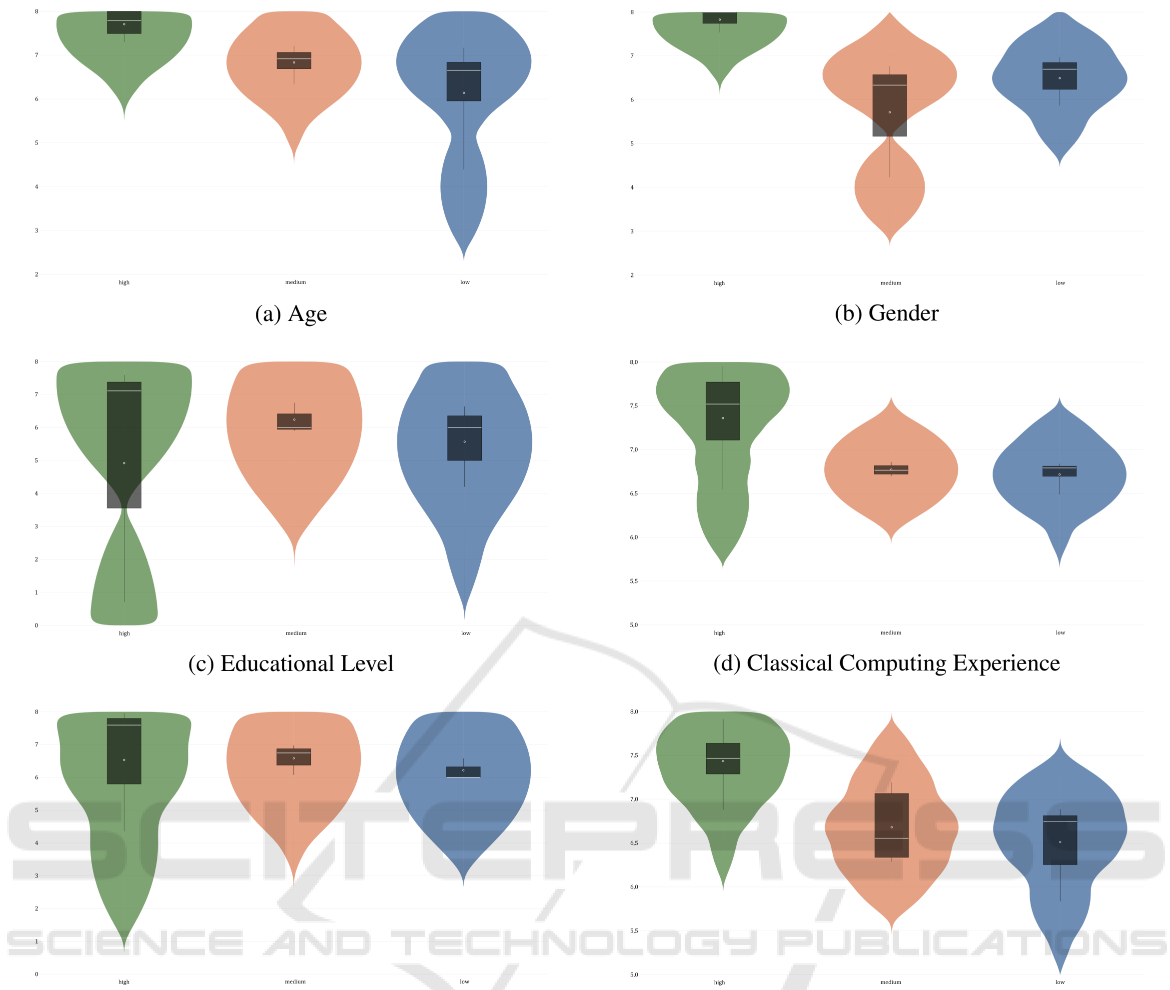
Using violin plots (Figure 1), the distribution of scores was visualized across different analyzability levels and demographic subgroups. The X-axis represents the analyzability level of the circuit —"high" (green), "medium" (orange), and "low" (dark blue)— and the Y-axis ranges from 0 to 8, representing the number of correct responses out of eight questions per circuit.

Next, an interpretation of the various results obtained is provided:

- **Score Dispersion.** The width of the violin indicates score variability. Educational level, for example, shows a larger dispersion, possibly due to differences in academic preparation or familiarity with abstract reasoning. In contrast, gender and order of exercises show narrower distributions, suggesting more consistent performance among those groups.
- **Score Distribution.** Most participants, especially in younger and less-experienced subgroups, scored higher when analyzing highly analyzable circuits. This suggests that improved circuit structure benefits a wide range of users, reinforcing the utility of the analyzability rating in distinguishing complexity levels.
- **Comparison of Average Scores.** The mean scores (white circles) confirm a clear positive trend between analyzability level and participant performance.
- **Comparison Between Analyzability Levels.** For nearly all demographic groups, scores increased with circuit analyzability. This reinforces the hypothesis that more analyzable circuits facilitate understanding. However, the relative flatness observed in some subgroups (e.g., educational level) suggests that factors such as cognitive style, abstraction ability, or prior training may also play an important role.

4.2 Evaluation of Assumptions

To statistically evaluate the relationship between circuit analyzability and participant performance, the Kruskal-Wallis test (Kruskal and Wallis, 1952) was applied. This non-parametric method is appropriate for comparing three or more independent groups



(a) Age

(b) Gender

(c) Educational Level

(d) Classical Computing Experience

(e) Quantum Computing Experience

(f) Order of Exercises

Figure 1: Descriptive Analysis of the Data.

when normal distribution cannot be assumed, as in this study.

The analysis, conducted with Python's `scipy.stats` library, yielded a test statistic of **58.3928** and a p-value **<0.001**. This result is well below the 0.05 threshold, so the null hypothesis (H_0) can be rejected. Therefore, it can be concluded that the differences between groups are statistically significant.

This supports the interpretation that the analyzability levels assigned to the circuits using the model correlate with measurable differences in comprehension performance. However, it is essential to emphasize that this validation applies only to the quantum portion of the model and within the study sample and environment constraints. Further testing must assess whether similar results would be obtained with hybrid (classical–quantum) code in real-world scenarios.

4.3 Limitations and Impact on Validity

Despite promising results, this study has several limitations. The sample was limited to undergraduate students from a single institution, reducing diversity in academic background and technical experience. While participants received training, their quantum and classical programming proficiency was not formally assessed. Moreover, although the model targets hybrid systems, this evaluation focused only on quantum circuits due to the lack of publicly available hybrid code bases. Additional confounding factors—such as differences in reasoning style, cognitive load, or familiarity with graphical tools like Quirk—may also have influenced performance.

Future work should involve a more diverse, international sample with varying levels of quantum experience, ideally classified through pre-tests. The model must also be validated on hybrid systems integrating classical and quantum components. Comparative studies across tools and environments will further support the generalization and scalability of the model. These steps are key to refining its applicability in real-world development and quality assurance settings.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a preliminary validation of a quality model for assessing the analyzability of classical–quantum software. The model integrates classical and quantum metrics within a unified framework. Although hybrid in design, this first evaluation focused solely on quantum circuits due to the lack of accessible, mature hybrid code bases.

An empirical study with 109 participants showed statistically significant differences in comprehension performance across analyzability levels assigned by the model. These results support the model's utility in distinguishing circuit complexity and reinforce the need for structured metrics in hybrid software quality evaluation.

However, this study is an initial step. Its scope was limited to a homogeneous participant sample and quantum-only code. Future work will expand the study to include diverse profiles, assess prior expertise, and test the model on real hybrid systems. Further iterations will refine the metrics and assess their scalability across tools and contexts, moving toward a generalizable hybrid software quality assessment framework.

ACKNOWLEDGEMENTS

This research was supported by the projects QSERV: Quantum Service Engineering: Development Quality, Testing & Security of Quantum Microservices (PID2021-124054OB-C32), funded by the Spanish Ministry of Science and Innovation and ERDF; Q2SM: Quality Quantum Software Model (13/22/IN/032) project financed by the Junta de Comunidades de Castilla-La Mancha and FEDER funds; and AETHER-UCLM: A holistic approach to smart data for context-guided data analysis (PID2020-112540RB-C42) funded by MCIN/AEI/10.13039/501100011033/. It also received financial support for the execution of applied research projects within the UNION - UCLM Own Research Plan framework, co-financed at 85% by the European Regional Development Fund (FEDER) (2022-GRIN-34110).

REFERENCES

Bernhardt, C. (2019). *Quantum Computing for Everyone*. MIT Press.

Díaz-Muñoz, A., Rodríguez, M., and Piattini, M. (2024a). Implementing an environment for hybrid software evaluation. *Science of Computer Programming*, 236:103109.

Díaz-Muñoz, A., Rodríguez, M., and Piattini, M. (2024b). Towards a set of metrics for hybrid (quantum/classical) systems maintainability. *JUCS - Journal of Universal Computer Science*, 30(1):25–48.

ISO/IEC (2011). Iso/iec 25010:2011 systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models.

ISO/IEC (2014). Iso/iec 25000:2014: Systems and software engineering — systems and software quality requirements and evaluation (square) — guide to square.

Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47:583–621.

Kumar, A. (2023). Formalization of structural test cases coverage criteria for quantum software testing. *International Journal of Theoretical Physics*, 62:1–16.

Piattini, M., Hevia, J. L. O., Pérez-Castillo, R., Serrano, M., and Nodarse, G. P. (2021). Toward a quantum software engineering. *IT Professional*, 23:62 – 66.

Piattini, M., Nodarse, G. P., Pérez-Castillo, R., Oliver, J. L. H., Serrano, M., González, G. H., Guzmán, I., Paradela, C. A., Polo, M., Murina, E., Navajas, L. J., Marqueño, J., Gallego, R., Tura, J., Phillipson, F., Murillo, J., Niño, A., and Rodríguez, M. (2020). The talavera manifesto for quantum software engineering and programming.

Rodríguez, M., Oviedo, J. R., and Piattini, M. (2016). Evaluation of software product functional suitability: A case study. *Software Quality Professional Magazine*, 18.

Rodríguez, M. and Piattini, M. (2014). Software product quality evaluation using iso/iec 25000. *ERCIM News*, 99:41–42.

Rodríguez, M., Piattini, M., and Fernández, C. M. (2015). A hard look at software quality: Pilot program uses iso/iec 25000 family to evaluate, improve and certify software products. *IEEE Software*, 48(09):30–36.

Verdugo, J., Oviedo, J. R., Rodríguez, M., and Piattini, M. (2024). Connecting research and practice for software product quality evaluation and certification: A software laboratory's 25-year journey. *IEEE Software*, 41:33–40.