



# An overview of quantum software engineering in Latin America

Álvaro M. Aparicio-Morales<sup>1</sup> · Enrique Moguel<sup>1</sup> · Luis Mariano Bibbo<sup>2</sup> · Alejandro Fernandez<sup>2,3</sup> · Jose Garcia-Alonso<sup>1</sup> · Juan M. Murillo<sup>1</sup>

Received: 4 March 2024 / Accepted: 31 October 2024  
© The Author(s) 2024

## Abstract

Quantum computing represents a revolutionary computational paradigm with the potential to address challenges beyond classical computers' capabilities. The development of robust quantum software is indispensable to unlock the full potential of quantum computing. Like classical software, quantum software is expected to be complex and extensive, needing the establishment of a specialized field known as Quantum Software Engineering. Recognizing the regional focus on Latin America within this special issue, we have boarded on an in-depth inquiry encompassing a systematic mapping study of existing literature and a comprehensive survey of experts in the field. This rigorous research effort aims to illuminate the current landscape of Quantum Software Engineering initiatives undertaken by universities, research institutes, and companies across Latin America. This exhaustive study aims to provide information on the progress, challenges, and opportunities in Quantum Software Engineering in the Latin American context. By promoting a more in-depth understanding of cutting-edge developments in this burgeoning field, our research aims to serve as a potential stimulus to initiate pioneering initiatives and encourage collaborative efforts among Latin American researchers.

**Keywords** Software engineering · Quantum Software Engineering · Quantum computing · Mapping study · Survey

## 1 Introduction

Quantum computing is postulated as a new computing paradigm capable of addressing several challenges that classic computers cannot [1], such as factorizing prime numbers [2], simulation of chemical reactions (quantum simulation), or resolving meteorological equations [3], among others. This is possible due to its quantum

---

Álvaro M. Aparicio-Morales, Enrique Moguel, Luis Mariano Bibbo, Alejandro Fernandez, Jose Garcia-Alonso, Juan M. Murillo have contributed equally to this work.

Extended author information available on the last page of the article

nature, superposition, and entanglement properties. These characteristics give quantum computers a huge computation capacity that fits with several applications, such as cryptography, medicine, and energy. Nevertheless, to develop those applications in a large-scale and industrial way, quantum software is needed [1].

As it happens in nowadays' software solutions, quantum software is expected to become large and complex to create high-quality solutions or applications. However, the necessary elements for developing it are still in a very early stage [4]. Given this context, Quantum Software Engineering (QSE) is presented as a necessary field to achieve that goal as proposed in the Talavera Manifesto [1] in which a synopsis outlining certain principles and commitments regarding Quantum Software Engineering and Programming is presented.

This new area, QSE, focuses on bringing the benefits of actual software engineering (SE) to quantum software development. In this way, efforts in researching, training, and organizing activities and/or events that promote the exchange of ideas and knowledge around QSE are necessary.

Leveraging the special issue focused on Latin America, it is important to know the development made by universities, research institutes, and companies to highlight these advancements in Quantum Software Engineering. This can be a starting point for pioneering initiatives and collaborations between Latin American countries to improve and develop this new area of Quantum Software Engineering. An initial example of such initiatives can be found at the RIPAISC network.<sup>1</sup>

In this regard, it is fundamental to establish cooperation between those stakeholders. Hence, it is necessary to identify who is working around QSE and their research domain inside this new field. Therefore, our research tries to provide a first approximation toward understanding the current landscape of Quantum Software Engineering research in Latin America. Additionally, we seek to offer an initial assessment of participants' perspectives on the prevalent issues and challenges in QSE, to promote research and development in this field.

For this purpose, we conducted a systematic mapping study of Quantum Software Engineering, identifying research works from Latin American universities or research centers to explore their research interests in Quantum Software Engineering. In addition, we surveyed experts in the field to capture information about affiliations, involvement with QSE, funding and resources, collaboration, participation in events and publications, opportunities, and challenges.

To detail this information, the rest of the paper is structured as follows. Section 2 briefly introduces Software Engineering, its procedures and practices, and their adaptation to Quantum Software development. Section 3 presents the methodology and results of the systematic literature mapping. Section 4 describes the survey we conducted to collect information on the current state of QSE in Latin America and the results derived from this survey. Finally, Sect. 5 details the conclusions of this work.

---

<sup>1</sup> <https://www.ripaisc.net> "Red Iberoamericana para el Avance de la Ingeniería de Software Cuántico".

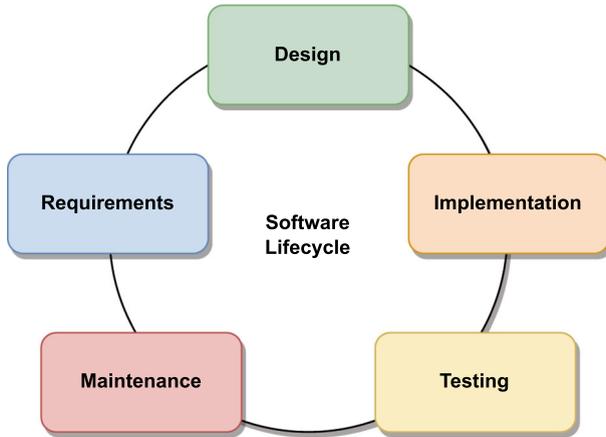


Fig. 1 Software development lifecycle (SDLC)

## 2 Background

Modern software applications are made possible by a set of established guidelines and techniques that allow to creation of large-scale and industrial software that works reliably, efficiently, and securely across diverse environments. This discipline is known as Software Engineering [5].

As well as it is being used in the development of nowadays software, it is assumed that the application of Software Engineering would be useful in quantum software development, as it is presented in the Talavera Manifesto [1].

In this section, a brief introduction to Software Engineering and its procedures and practices will be exposed and explained to understand the whole process of software development lifecycle (SDLC), including the current topics that are under discussion for their adaptation to quantum software development.

### 2.1 Software engineering

Considering the *standard* definition provided by IEEE, Software Engineering (SE) can be defined as follows:

*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).*

Each stage of software development involves applying specific techniques and methodologies. This entire process, from defining requirements to maintenance, is known as the Software Development Lifecycle (SDLC). The Software Engineering approach ensures the creation of robust, fault-tolerant, high-quality software [6].

SDLC stages diagram is shown in Fig. 1. Each stage represents an area of study and application in software engineering.

Throughout the software development process, transitions occur between higher-level stages (such as requirement engineering) and lower-level stages of abstraction, such as software implementation and testing. In the higher-level stages of abstraction, as indicated by some experts in the field of quantum computing and given the current state of this technology, it does not appear necessary to make many modifications. It would only be necessary to create elements that identify the quantum components of our system and adapt current techniques and methodologies to account for the coexistence and interaction of both computing models. Conversely, in the lower-level stages of abstraction, significant modifications, and new creations are anticipated to accommodate the peculiarities of quantum computing, such as handling qubits, quantum algorithms, and managing coherence, noise, and quantum entanglement. Additionally, it will be crucial to develop new tools and development environments that support quantum programming, as well as establish specific testing methodologies to verify and validate the correct functioning of quantum algorithms and their integration with classical systems. This includes simulating quantum environments, debugging quantum programs, and creating standards to ensure the interoperability and security of hybrid quantum-classical systems. However, the review of the software lifecycle stages is in an initial phase, and experts in the field are debating what needs to be created or how current methods applied in software engineering should be adapted to include quantum software [7–9]. Moreover, as progress is made in quantum hardware development and its technologies become standardized, it will be possible to identify new challenges and, perhaps, differences during the software development lifecycle.

Given the actual state of quantum computers, a way of using them is through their integration with classical computers, forming what is termed hybrid systems (classical-quantum) [10], in which quantum, and classical software will coexist and cooperate. In this context, the application of SE must take into account this particularity and it becomes necessary for any adaptation or even transformation of any stage of current SDLC.

Before defining the concept of Quantum Software Engineering, some basic ideas of differences between classical and quantum software will be presented for a better understanding.

To this end, we need to understand what is meant by Software. Software can be defined as the functional logic of a computer system. Until now, “logic” has primarily been employed to denote computational principles rooted in Boolean algebra (0-1). However, this new computing paradigm presents a new form of logic: Quantum Logic. Quantum Logic differs from the one formulated by *George Boole*. Still, the concept of computational logic should not be modified since both logic (quantum and classical) have the same purpose: to provide behavior to the computer system (either quantum or classical).

Taking this into account, the concept of software is not inherent to the computing model that the computer system possesses, therefore, if (in short) the definition of software engineering is the application of engineering processes to software development, this definition can be maintained in the field of quantum computing, bearing in mind that software can now be both, quantum and classical.

Given this context, the explanation of the concept of Quantum Software Engineering is the same as for Classical Software Engineering by IEEE. Just keep in mind that

the concept of software currently associated with Classical Computing must also be associated with Quantum Computing.

The following sections provide a more detailed overview of Quantum Software Engineering, outlining the activities involved in each phase.

### 2.1.1 Software requirements

In Software Engineering, the stage of software requirements can be defined as the initial phase where the specific actions and operations (functional requirements) of our software solution and its circumstances (non-functional requirements), such as performance, security, usability, or other quality attributes, are collected by software engineers.

In this field, authors are not currently focusing much on identifying and describing functional requirements in Quantum Software as there are no differentiating elements compared to classical computing at this time. However, in the definition of non-functional requirements, certain elements must be considered given the current state of quantum computing, such as the availability of a computer, the algorithm's qubits, the number of shots, or the economic cost of execution.

### 2.1.2 Software design

As it is highlighted in Zhao's work [7], there are two main areas in the software design stage in classical software engineering. On one hand, we have the architectural aspect, which involves creating a high-level abstraction model of the software system's components and their relationships and interactions. On the other hand, we have the detailed specification of our software, which models its behavior and, along with the architectural model, must satisfy the requirements gathered in the previous phase. Up to this point, the concept of both areas remains the same for quantum software. The difference lies in the need to identify quantum components from classical ones when developing the architectural model.

As a result of this phase, an architectural model and software specifications are obtained. The architectural model is defined by a modeling language like UML (unified modeling language) which is considered a standard in software design [7]. Some works that design quantum software using UML are starting to appear such as the one of Pérez-Castillo et al. [11] proposing an extension of UML for two kinds of diagrams (class and sequence). A different perspective is the work of Ali et al. [12] which introduces an initial conceptual model of quantum software from the standpoint of model-based engineering.

On the other hand, software specifications are defined by using a specification language. In the quantum case, the specification language must consider the quantum characteristics due to its quantum nature. Because of that an extension will also be necessary to apply specification language to quantum software. In this line, we can highlight the article by Cartiere [13], which introduces a formal specification that can represent the basic notations of quantum computing and the work by Feng et al. [14] that proposes an extension of the computational logic tree to consider superposition and entanglement (Q-CTL). However, given the early stage of quantum software, the

use of quantum computation logic has not yet been widely adopted or integrated into existing tools and frameworks.

### 2.1.3 Software implementation

In Software Engineering, the implementation phase centers on creating the code according to the requirements, specifications, and model obtained in the previous steps. It involves translating the design blueprints into functional software components and validating their adherence to the established guidelines.

Current software has been possible due to the evolution of programming languages, frameworks, and tools that allow a transition from monolith applications to a modular and distributed way of computing software [15]. This kind of computing, known as Service-Oriented Computing, has made possible the creation of distributed, scalable, interoperable, and efficient software solutions that can be deployed on any computer.

Nowadays, quantum software programming unlike classical software programming, is mainly based on using quantum gates to implement existing quantum algorithm instead of calling to existing functions. This is a consequence of the lack of programming libraries and algorithms, although, in the last few years, new advanced algorithms have been developed due to the increasing number of qubits available in the new generation of quantum computers. Nevertheless, some problems may delay the advancement of quantum software development. Among them, the following can be highlighted:

- **Noise:** Errors and inaccuracies arising during quantum computing executions can propagate and affect the reliability of the computations performed by the quantum software. However, encountering software errors during the production stage could lead to severe issues, although quantum computing is in the Noisy Intermediate-Scale Quantum era, big efforts are being made by the scientific community to achieve quantum utility [16].
- **Accessibility:** Access to quantum computers today is primarily provided through cloud providers. However, long waiting times to collect and analyze results often arise due to the queuing system for accessing these computers.
- **Economic cost:** The novelty of quantum technology makes running algorithms in the Cloud expensive, limiting access for individuals, research groups, or companies with limited financial resources for quantum software execution.

Multiple providers offer access to different quantum computers, each with its own libraries and languages, creating a lack of standardization that hinders quantum software development.

Despite these challenges, significant efforts are underway to resolve these issues and simplify quantum software implementation for hybrid systems.

There are several lines of development for the implementation of quantum software, including the work by Rojo et al. [10], Moguel et al. [4], and Valencia et al. [17] focus on the development of a concept known as hybrid microservices. These involve encapsulating a quantum algorithm within a microservice framework based on classical technologies.

Another line of work consists of developing tools that make the process of programming a quantum algorithm easier. For instance, Romero-Álvarez et al. [18] demonstrate a method to define quantum services and automatically generate the corresponding source code by extending the Open API Specification. Another noteworthy contribution is the work conducted by Garcia-Alonso et al. [19], where the authors introduce an adaptation of the API Gateway pattern, considering the unique aspect that quantum services cannot be deployed in the same manner as traditional services. Instead, their Quantum API Gateway suggests the optimal quantum computer to execute a particular quantum service during runtime.

Therefore, it is crucial to have tools operating at higher abstraction levels to streamline the construction of intricate hybrid systems and widen quantum programming accessibility. Furthermore, addressing the essential skills, capabilities, and toolsets needed by future quantum software programmers is imperative.

### 2.1.4 Software testing

In the software testing phase, the software developed during the implementation phase is thoroughly examined to identify defects and errors and to ensure that it meets the specified requirements and quality standards. Software testing aims to validate and verify that the software functions correctly, performs as expected, and delivers the intended outcomes.

To test a classical program, each function is examined by comparing expected and actual results. Given the deterministic nature of classical programming, verifying results is straightforward. Tests require specifying inputs, execution conditions, and expected outcomes. If all tests pass, the program is ready for the next phase of the SDLC.

In quantum software, verifying if a function is working as expected is quite different because of the probabilistic nature of the results which implies using different methods and practices that consider uncertainties. However, just like in classical software, the tests should be conducted at the software/algorithm level. Additionally, during the verification process, evaluating statistical probabilities and distributions of outcomes becomes necessary.

Another aspect to consider is that in classical programming, there exists the possibility of adding breakpoints in the software. A breakpoint in computer programming is "*a point in a computer program at which execution can be suspended to permit manual or automated monitoring of program performance or results*" [5]. When the execution of a program is halted at a breakpoint, the testing software engineer can inspect the actual state of the computer program. Furthermore, this tool allows them to proceed with the execution of the software, either line by line, function by function, or from one breakpoint to another. Such functionality greatly aids testers in debugging by providing insights into why the computer program might malfunction. Given the quantum mechanics nature of this new computing paradigm, conducting a measurement significantly influences the observation [20]. This implies that placing breakpoints in the quantum computer program to know the state of the system and perform debugging tasks is impossible and it will be necessary to look for new alternative methods.

The previous aspects pose considerable challenges to modifying or creating new testing methodologies for the quantum software.

In quantum software testing, one of the relevant works is that of Miranskyy et al. [21]. They addressed several challenges linked to white-box and black-box testing, along with verifying and validating quantum programs. Another interesting work is the one proposed by de la Barrera et al. [22] in which strategies of classical software testing are discussed for being applied to quantum software testing.

### 2.1.5 Software maintenance

Maintenance is the final stage of the Software Development Lifecycle, involving bug fixes, updates, and improvements to ensure smooth operation.

Modern software relies on external libraries and packages, which also evolve over time. To maintain proper functionality, the software must stay compatible with these updated components.

Over time, the same thing will happen to quantum software, that will be built by new libraries and packages that will receive updates that may affect the quantum software's correct performance.

Nowadays, low-level updates to the software of quantum computers can interfere with the *correct* execution of our algorithm. Moreover, due to the variety of quantum computers implementing different hardware technologies, the result in the execution of our program may differ from one provider to another. Thus, in these early stages of quantum computing and quantum software development, performing proper maintenance of our quantum programs is necessary.

Another aspect is the continuous improvement of the processes performed by our computer program, either through new functionalities or improvements to existing functionalities, the latter is known as the re-engineering process.

Due to the coexistence of both types of software, certain functionalities currently performed in classical software may eventually be migrated to quantum software using the re-engineering process due to reductions in execution times. Also, new quantum functionality will be integrated into existing classical software. This is not a simple task, and it would be necessary to develop standards, models, and systematic transformations to preserve business rules and simplify migration across various classical and quantum environments [23].

There has been limited effort in applying reverse engineering techniques to quantum software within maintenance stages. However, Pérez-Castillo et al. [24] addresses this issue by proposing a software modernization approach. This approach aims to restructure classical systems alongside existing or new quantum algorithms, aiming to create target systems that combine classical and quantum information systems.

## 3 Systematic mapping study

The main objective of this study is to collect information of interest on the current State of the Art of Quantum Software Engineering in Latin America. For this purpose,

a systematic mapping study is carried out where, in addition to reaching the proposed objectives, it is sought to classify recent research works in Quantum Software Engineering by Latin American researchers.

A systematic mapping study, as opposed to the Systematic Literature Review, is a type of secondary study (study based on the analysis of previous research). Its objective is to determine the scope of research conducted on a specific research topic and to classify knowledge, as opposed to a systematic review that seeks to answer a specific research question [25, 26].

We will use this technique to search the defined scope, classify the most relevant research, and conduct a thematic analysis to offer a visual map of knowledge of the main research in Quantum Software Engineering in Latin America.

For this purpose, we follow the methodology proposed by Petersen et al. [26] in which they suggest a procedure consisting of 5 stages: 3.1 Define research questions; 3.2 Perform a literature search; 3.3 Select studies; 3.4 Classify papers; and 3.5 Extract and perform data aggregation.

Each defined step was carried out by the researchers signing this document. To improve readability and facilitate understanding of the results obtained, we decided to focus only on explaining the classification of the papers (stage 3.4) and the analysis (stage 3.5) in the following sections.

### 3.1 Research questions

Although the general objective of this study could be summarized as understanding what techniques, methodologies, and technologies exist in Quantum Software Engineering in Latin America, this objective is divided into three specific research questions to obtain more detailed knowledge and a global vision of the subject. The research questions to be answered in this study are the following:

*RQ1. What techniques and/or technologies exist in the field of Quantum Software Engineering in Latin America?*

*RQ2. Is there evidence of interest in Software Engineering applied to Quantum Computing in Latin American countries?*

*RQ3. What are the research areas and problems in which Quantum Software Engineering is being applied to in Latin America?*

### 3.2 Data sources and search strategy

The search strategy was developed taking into account the specific terminology related to Software Engineering for Quantum Computing. To perform the search, terms such as software, program, computation, circuit, or engineering and their main synonyms were considered. The search string was as follows:

(quantum)

AND

(software OR program\* OR comput\* OR circuit OR enginee\*)

The search was conducted in the Scopus<sup>2</sup> and IEEE<sup>3</sup> databases. These libraries have a wide coverage of publications in computer science (among others), index several publication catalogs (including other libraries such as ACM, Springer, Elsevier, etc.), and subsume other bibliographic information providers such as DBLP or WOS [27].

### 3.3 Study selection

Four simple criteria were defined for study selection:

- **Inclusion.** In performing a systematic mapping study of the literature, we took into account all papers dealing with any aspect related to the definition of Software Engineering and the explanation of each stage of the SDLC provided in the Background section or nearby, such as tools, techniques, methodologies, etc. Additionally, as inclusion criteria, we decided to perform a less restrictive search and took into account peer-reviewed journal and conference papers, books, book chapters, doctoral theses, technical reports, and even preprint papers.
- **Exclusion.** On the other hand, we decided to eliminate all those papers that use this new computing paradigm but do not deal with any aspect related to software engineering. We also took as exclusion criteria all those papers that were not written in English. In addition, for logical reasons, we discarded duplicate articles, taking into account for our study the most recently published manuscript.
- **Constraint.** As the mapping of the literature is focused on a special issue on quantum information in Latin America, we only considered papers in which at least one of the authors comes from a Latin American country (Argentina, Brazil, Chile, Colombia, Costa Rica, Ecuador, El Salvador, Guadeloupe, Honduras, Mexico, Paraguay, Peru, Puerto Rico, Uruguay or Venezuela).
- **Timing.** The concept of Quantum Software Engineering was coined by Stepney et al. in 2004 [28]. However, it was not until the concept of Quantum Supremacy was defined by John Preskill [29] that QSE took on special relevance. For this reason, we have considered restricting the search from 2012 onward. In addition, the interest of our proposal is to know the most active researchers in recent years, which is why we have restricted the search to the last 11 years.

### 3.4 Paper classification

Performing the search in the indicated scientific databases and applying the inclusion and exclusion criteria already detailed, the study's results revealed 1,316 works. The complete list of works analyzed in this paper can be found as supplementary material on the Zenodo platform.<sup>4</sup>

To facilitate the understanding of the analysis of the articles and due to the high number of proposals, the classification was carried out in three phases: 1) only the title, abstract, and keywords of each paper were analyzed according to the inclusion

<sup>2</sup> <https://www.scopus.com>.

<sup>3</sup> <https://ieeexplore.ieee.org/Xplore/home.jsp>.

<sup>4</sup> <https://doi.org/10.5281/zenodo.10718726> Supplementary material.

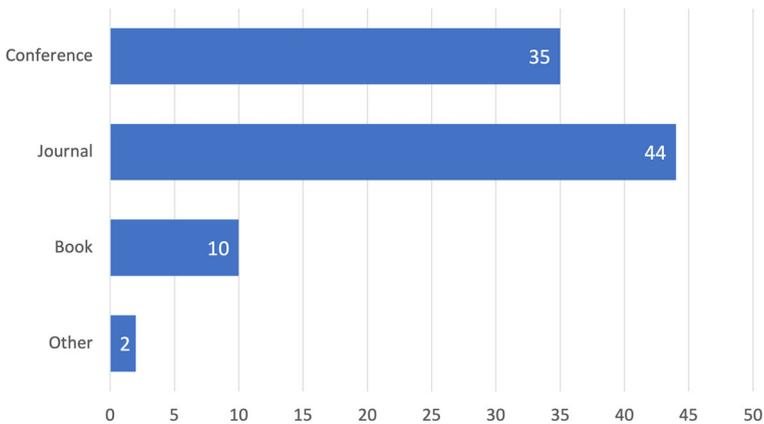


Fig. 2 Venue types

and exclusion criteria (detailed in the previous section); 2) the same elements of each paper were evaluated to classify them according to the Software Engineering topic; and 3) the full text of the papers were analyzed to refine the selected papers further to answer the defined research questions comprehensively.

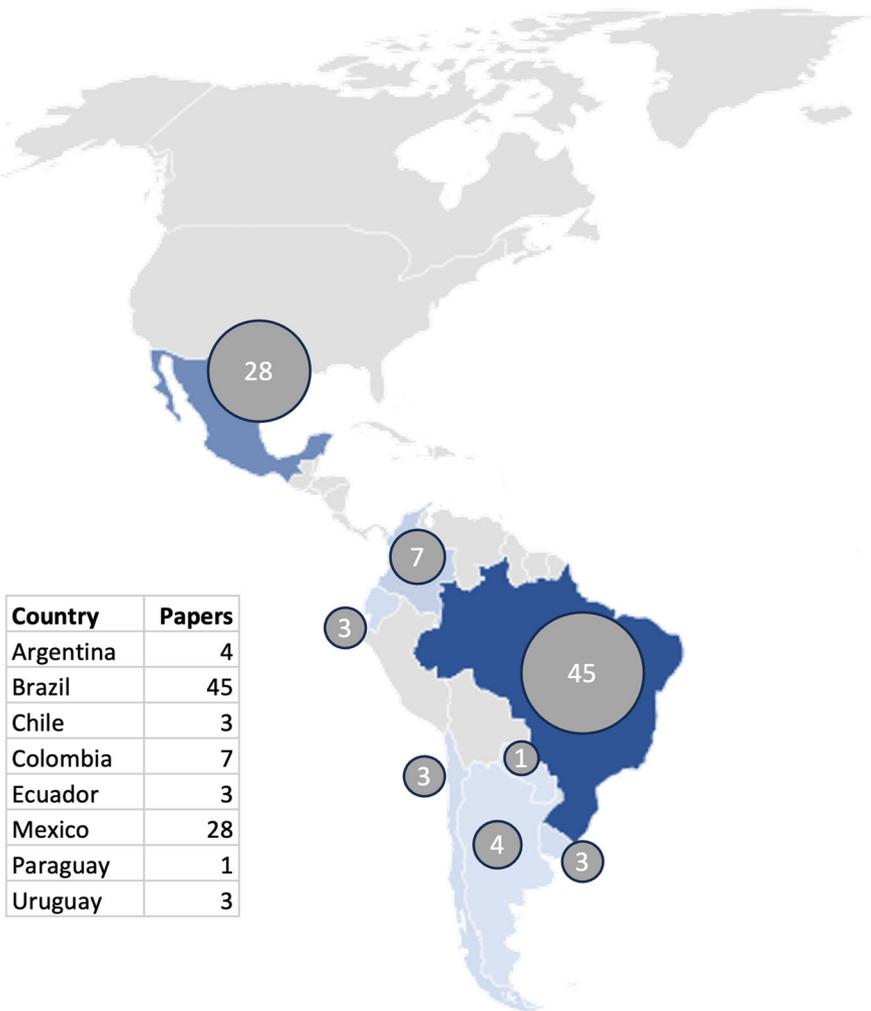
After the classification phase, the total number of papers classified as software engineering papers is 91 out of 1,316 (approximately 6.9%). This is a low number, but understandable because most of the papers from the search deal with this new computing paradigm applied to different contexts of physics, chemistry, cybersecurity, finance, etc., and do not refer to any aspect related to Software Engineering. Details of this selection of papers can be found as supplementary material on the Zenodo platform.<sup>5</sup>

These papers were published in different venues. Figure 2 presents a distribution of articles in terms of publishing venues. As it can be appreciated, 79 out of 91 Software Engineering papers were published in Journals and Conferences, standing out the publications of Journals with a total of 44 out of 91 papers, approximately 48.3%. Only 13.1% represent publications in books and other media.

On the other hand, in Fig. 3, a geographical distribution of the Latin American authors of the published papers is shown. A total of 8 Latin American countries are currently working on projects related to Software Engineering applied to Quantum Computing. Among the countries, two of them represent approximately 80.2% of the published papers. These countries are Brazil, with a total of 45 papers, and Mexico, with 28 papers.

It is worth mentioning that in Brazil, three authors can be highlighted because they are the most prolific ones. They are *Adenilton J. da Silva*, *Teresa B. Ludermir*, and *Marley Vellasco*, with 3 selected works each. *Da Silva* and *Ludermir* (2 selected works) usually work together because they have collaborations and their area of study in Software Engineering applied to Quantum Computing is the development of Neural Networks and development methodologies [30, 31].

<sup>5</sup> <https://doi.org/10.5281/zenodo.10718726> Supplementary material.



**Fig. 3** Bubble map by country

In addition, in Mexico, the most prolific ones are *César N. Plata*, *Raymundo Marcial Romero*, and *Antonio Hernández Servín*, with 3 selected works each. These three researchers work together to develop extensions for the quantum programming language QML (Qt modeling language) [32–34]. Also, we would like to highlight that in Argentina, from the 4 selected papers, 3 of them are from the researcher *Alejandro Diaz-Caro*, who works on formal aspects of quantum programming [35–37].

In addition, we would like to highlight that there are collaborations between researchers from different Latin American countries. Of the classified works, there are 3 papers in which collaborations exist between researchers from Argentina and Brazil [38], Brazil and Chile [39], and between Chile and Ecuador [40].

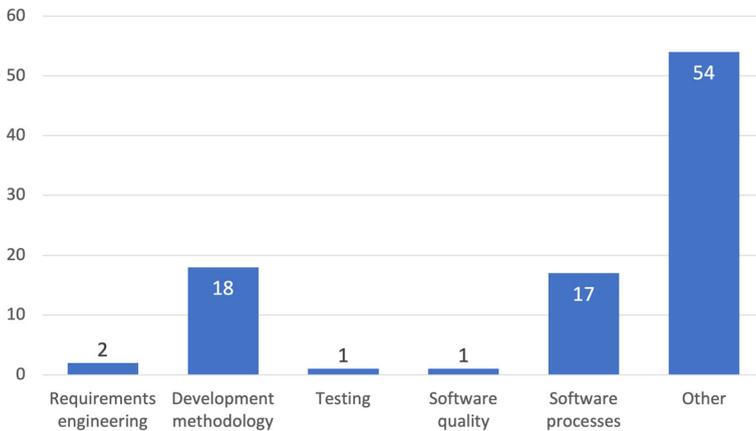


Fig. 4 Categories of software engineering

### 3.5 Data extraction and synthesis

For a systematic mapping study, Kitchenham et al. [25] and Petersen et al. [26] defined guidelines suggesting that other parts of the analyzed articles should only be evaluated in cases where they are not well structured or are imprecise. For this research, we decided to fully analyze each manuscript to answer all the questions raised better. Some important proposals could be left out by analyzing only titles, abstracts, and keywords.

Classifying the papers according to the different fields of Software Engineering, the result is shown in Fig. 4. It can be seen that most of the papers that exist so far are in the field of Development Methodologies (18 papers) and Software Processes (17 papers). Also, we have included a category of other works because most of them are works that make use of methodologies, techniques, or tools in conjunction with quantum computing, and not being software engineering its purpose.

As shown in Fig. 4, two papers address *Requirements Engineering*. In the first one, Saraiva et al. [41] analyzes the non-functional requirements of quantum programs based on experiences in quantum software development for real quantum hardware. Also, the authors analyze hardware-related constraints and derive a set of generic non-functional requirements for this type of program, identifying five non-functional requirements related to performance efficiency and reliability that should be considered when implementing a quantum program for a quantum device. The second of the papers, Diaz et al. [42], shows a study of different design techniques and implementations to simulate quantum computing with the support of classical computing, highlighting quantum resource consumption considerations.

Regarding *Development methodologies*, we have found 18 papers. Of all of them, we highlight the work of Feitosa et al. [43, 44] in which they propose a monadic semantics for FJQuantum, an object-oriented language based on Featherweight Java, created to reason and develop programs that handle quantum data and quantum operations. The authors also provide an interpreter for this semantics and examples of quantum

programs that run with it. On the other hand, we would also like to highlight the work of Gejea et al. [45] in which they present a high-performance simulation methodology of the Grover method that combines the advantages of the Grover technique with the parallel of cloud computing to improve multicore load balancing, memory utilization, and simulation efficiency.

On the other hand, in specific articles dedicated to **Testing**, we have only come across the work of Diaz et al. [42] who perform some tests on the tools that they implement. Given its nature and the topics it deals with, this work was the only one we have classified into 2 categories (Requirements Engineering and Testing).

Similarly, in the field of **Software quality**, we have only found the work of Oliveira et al. [46] in which they propose a framework to identify the sensitivity of quantum circuits to radiation-induced faults and the probability that a fault in a qubit propagates to the output. The researchers build on recent studies and experiments on real quantum machines, modeling transient faults in a qubit as a phase shift with a parameterized magnitude.

The Software Engineering category with the highest number of articles was **Software Processes** with 17 papers. Of these works, we would like to highlight the work of Dias et al. [47] in which they present a process based on QIEAs (quantum-inspired evolutionary algorithms) that make use of the evolution of machine code programs. Multilevel quantum systems inspire this process. Its operation is based on quantum individuals, representing a superposition of all programs (solutions) in the search space. On the other hand, we would also like to highlight the work of Silva et al. [48] in which they propose a software process for mapping the traveling salesman problem (TSP) based on pseudo-Boolean constraints to the D-Wave Systems. The authors formulate the problem as a set of constraints in propositional logic. Finally, we would also like to highlight the work of Ballinas et al. [49] in which they propose a process based on HQGA (Hybrid Quantum Genetic Algorithm) for the 0-1 Knapsack Problem (KP). The authors rely on the principles of quantum computation (superposition and entanglement of states) and use the Qiskit simulator.

In this study, we decided to go beyond the merely strict in terms of Software Engineering, and we wanted to evaluate other aspects related to methodologies, techniques, or tools in Quantum Computing that are applied in diverse areas. For them, we have decided to incorporate an additional category called **Others**. This category mainly includes works that make use of Neural Networks for different fields of application, for example, [50] or [30]; make use of Machine Learning techniques for different projects as [51], [52] or [53]; or algorithmic or cybersecurity issues as [54] or [55]; among other types of works.

Also, we would like to highlight some authors who go beyond the theoretical, applying quantum computing and quantum software engineering techniques to the world of industry. Some relevant cases are those of *Venegas-Andraca et al.* in the knowledge fields of image processing [56] and cybersecurity in smart cities and IoT devices [57]; On the other hand, *Portugal et al.* have made applications in molecular chemistry [58] or quantum walks [59]; Finally, we would like to highlight *Ludemir et al.* with its work related to quantum probabilistic associative memory architecture [60].

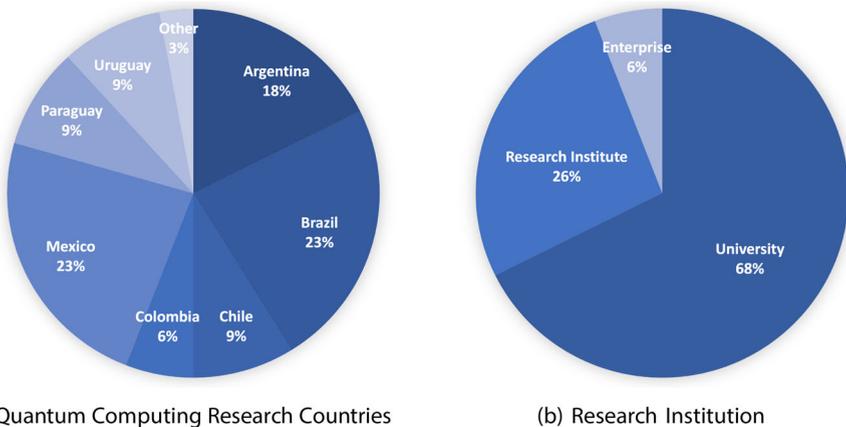


Fig. 5 Quantum research from Latin American countries and institutions

## 4 Surveying LATAM researchers' involvement in QSE

The main objective of this survey is to collect information about the present state of Quantum Software Engineering in Latin America. To gather this information, we created a survey using the Google Forms platform and sent it to a group currently working in research institutions in Latin America. The survey and the anonymized survey responses can be found as supplementary material on the Zenodo platform.<sup>6</sup>

This survey was sent to the 18 members of the RIPASC network<sup>7</sup> and to 113 researchers who were referred to us by the Network's researchers. This survey was also sent to the 39 Latin American researchers of the papers extracted from the Literature Mapping Study detailed in Sect. 3. A massive mailing of our survey was sent to 152 researchers. No question was defined as mandatory to make it easier for respondents to answer.

We obtained 34 responses from the Latin American scientific community. Below, we will detail the most important aspects extracted from the responses received.

### 4.1 Participation

Figure 5 provides an overview of the Latin American countries and types of institutions for the survey's participants. As shown in Fig. 5a, Brazil, Mexico, and Argentina have higher participation in the survey with 23%, 23%, and 18%, respectively. On the other hand, Fig. 5b shows that 68% of the survey participants come from university, 26% from research institutes, while only 6% come from the private sector.

On the other hand, we asked respondents whether they are currently involved in quantum computing in any form or topic. The results to that question can be seen in Fig. 6a, which shows that 94% of the respondents are working with aspects related to

<sup>6</sup> <https://doi.org/10.5281/zenodo.10718726> Supplementary material.

<sup>7</sup> <https://www.ripasc.net> "Red Iberoamericana para el Avance de la Ingeniería de Software Cuántico".

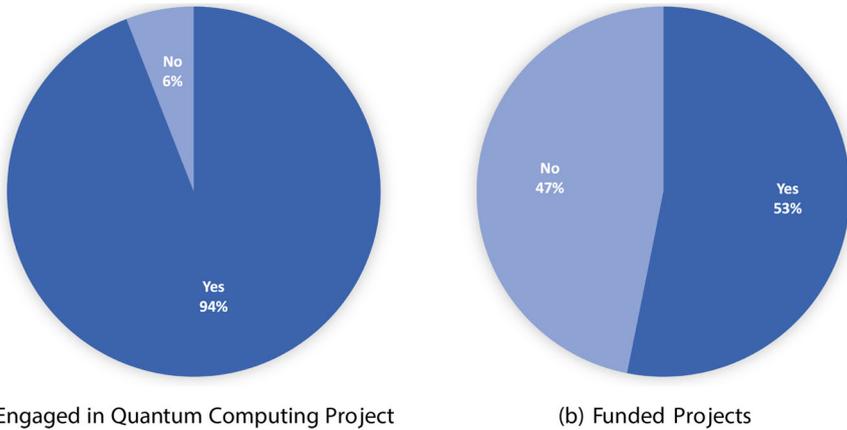


Fig. 6 Engaged in funded quantum computing projects

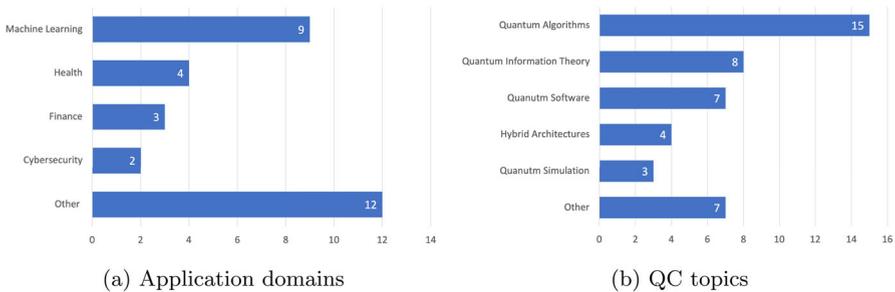


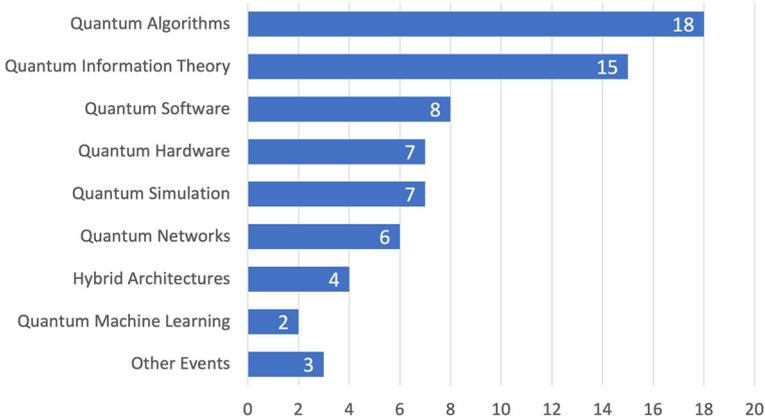
Fig. 7 Focus of respondents already involved in QC research

quantum computing. And of those researchers who are working in the field of quantum computing, 53% of respondents do so with public funding, as seen in Fig. 6b. It is worth noting that 35% of the researchers who indicated that they receive public funding for their research in quantum computing also receive private funding.

#### 4.2 Research focus

The researchers involved in QC projects reported that their main fields of application (see Fig. 7a) are “Machine learning” (30% of the participants), “Health”, “Finance” and “Cybersecurity” among others. As depicted in Fig. 7b, the most popular quantum computing topics among respondents are “Quantum Algorithms (New algorithms, Improve existing ones, etc.)” (34% of the participants), “Quantum Information Theory (Entanglement, error correction, etc.)”, and “Quantum Software (Design, Testing, Modeling, Quality, Development)”.

More than half (22 out of 36) of the participants already involved in QC research recognized working on a topic specific to QSE. When asked specifically about the QSE



**Fig. 8** Topics of QC events attended by participants

topic (from a list of predefined options), only three were mentioned: Quantum Software Design, Quantum Software Development, Quantum Software Modeling, and Method. None of the participants mentioned “Quantum Software Testing”, “Quantum Software Requirements”, “Quantum Software Maintenance”, “Quantum Software Quality”. These results can be analyzed from different points of view. On the one hand, there are researchers working on QSE that do not align themselves with any of the classical software engineering areas of interest. This can imply that QSE covers new areas that are not usually addressed in classical software. On the other hand, some of the classical areas of software engineering are not addressed by any researcher. This can be explained by two different reasons. First, there are areas of classical software engineering that do not need important adaptation for QSE and therefore pose no interest for researchers. And second, there are also areas of software engineering that are raising significant interest in QSE worldwide but that are not being addressed by Latin American researchers.

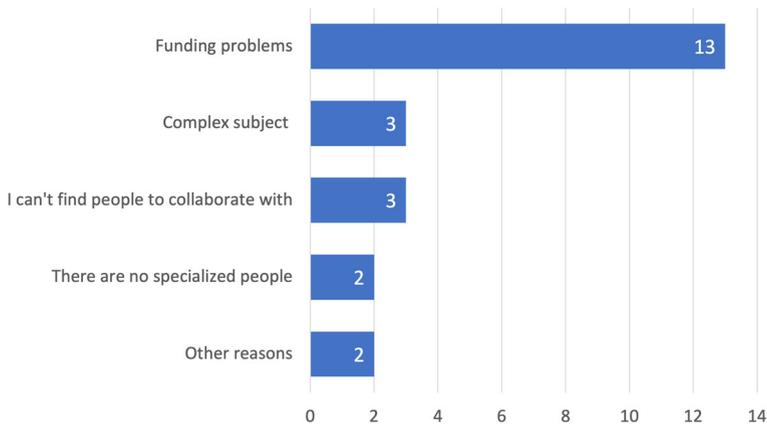
### 4.3 Collaboration and networking

Half of the participants report working in collaboration with researchers from other groups or institutions, mainly from their own country and from Europe, and to a lesser extent from Latin America and the USA.

Two-thirds of the participants report attending Quantum Computing events on varied topics (see Fig. 8).

### 4.4 Relevance awareness and challenges

To learn about the participants’ perception of the relevance of Quantum Software Engineering, we asked them to indicate if they considered it was relevant now or if it will become relevant in the mid-future, in the long term, or only after the consolidation



**Fig. 9** Challenges to get involved in QC research

of Quantum Computing. Responses are almost evenly distributed between the four options, with only a small tendency toward the mid and long-term alternatives.

When asked about the challenges they face in getting involved in QSE projects (see Fig. 9), participants identified the lack of funding, the complexity of the subject, and the lack of collaboration partners and experts as the main issues.

#### 4.5 Discussion

Once we have analyzed the data obtained from the survey that we have carried out, we would like to point out a possible threat to the validity of the survey itself, which is that the data may be unconsciously biased due to the selection of the respondents. In this sense, we have tried to get the survey to as many researchers in the field of quantum computing as possible, and we believe that the resulting data are close to the reality in which we find ourselves.

To continue, and after performing a knowledge analysis, we can draw several significant conclusions from the point of view of Quantum Software Engineering research in Latin America:

- *Geographical concentration of research.* There is a direct correlation between the number of survey participants and the authors' affiliation from the systematic review conducted in the previous section. This suggests that the survey reasonably reflects the general opinion of researchers in Latin America. Regarding the concentration of research, two countries (Brazil and Mexico) have the largest number of researchers in this area of knowledge.
- *Crucial role of Universities and Research Institutes.* The overwhelming majority of research participants are affiliated with universities and research institutes. This highlights these institutions' fundamental role in promoting and conducting research activities in the region, especially in cutting-edge research such as in the case.

- *Financial challenges.* It is concerning that only just over half of the participants have funding for their research. This situation may limit the scope and quality of research projects, as well as the ability of researchers to carry out their work effectively.
- *Need for international collaboration.* Although a significant proportion of participants report collaborations with researchers from other countries, collaboration between research groups within Latin America is considerably lower. Fostering and strengthening regional collaboration could be key to improving the quality and impact of research in the region and addressing common challenges.

The implications for research communities in Latin America derived from the presented information are diverse and significant. On the one hand, the low percentage of researchers with funding suggests a significant dependency on external funding sources. This may subject researchers to external research agendas and limit their ability to address local issues or develop innovative research lines. On the other hand, the lack of collaboration between research groups within Latin America can restrict the exchange of knowledge and resources among countries in the region. This may result in duplicated efforts, lack of synergies, and reduced capacity to address common challenges effectively. It is crucial to strengthen research infrastructure and institutional support throughout the region to address these implications. This includes increasing funding for research, promoting regional collaboration, and creating conducive environments for developing scientific careers in Latin America. In summary, these implications highlight the importance of addressing the structural challenges facing the research community in Latin America to promote a more equitable, collaborative, and sustainable research environment in the region. This will require a concerted effort from governments, academic institutions, and the scientific community.

Considering the identified implications and current needs in the research community in Latin America, some future directions that could be explored include:

- *Promotion of regional collaboration.* Promoting and facilitating collaboration among research groups within Latin America and researchers from other regions of the world. This can be achieved through exchange programs, research networks, and collaborative projects addressing common challenges.
- *Diversification of funding sources.* Seeking and developing new funding sources for research in Latin America, including public, private, and international funds. This may involve strengthening national funding agencies, promoting public-private partnerships, and participating in international scientific cooperation programs.
- *Strengthening research infrastructure.* Investing in improving and expanding research infrastructure in Latin America, including laboratories, computing centers, libraries, and other resources necessary for conducting high-quality research. This can enhance researchers' capacity to conduct cutting-edge and internationally competitive research.
- *Support for scientific talent development.* Promoting the development of scientific talent in Latin America through training programs, scholarships, and professional development opportunities. This may include creating high-quality

graduate programs, supporting academic mobility, and promoting the inclusion of underrepresented groups in science.

In summary, these future directions can contribute to strengthening research capacity in Latin America and promoting sustainable scientific and technological development in the region. These initiatives must be developed collaboratively and participatively, involving diverse stakeholders from the scientific, academic, governmental, and business communities.

## 5 Conclusions

Quantum computing is advancing rapidly and presents a transformative potential in solving problems that are intractable for classical systems. Our systematic mapping study and survey of Quantum Software Engineering (QSE) in Latin America revealed key insights into the current state and future directions of this emerging field. While the quantum software landscape is still developing, it is clear that engineering principles are crucial to ensuring scalable, reliable, and maintainable quantum applications.

Our findings show that the majority of research in Latin America is concentrated in Brazil and Mexico, reflecting regional strengths in areas like quantum algorithms and quantum-inspired methodologies. However, the overall number of studies specifically addressing Software Engineering aspects, such as testing, maintenance, and quality assurance in quantum systems, remains limited. This points to the need for further research in these critical areas, particularly as quantum computing hardware matures and becomes more accessible.

Key challenges identified by our study include limited funding and collaboration across Latin American countries. While there is a strong focus on algorithmic development, there is a notable gap in practical applications of QSE principles in real-world scenarios, particularly in fields like healthcare, finance, and cybersecurity. Addressing these gaps will require enhanced cooperation, both regionally and internationally, and the development of more robust funding mechanisms to support long-term research efforts.

Continuing our study, we surveyed to extract information about the vision and opinion of experts in this field. We sent this survey to the 18 researchers of the RIPAISC network (*Red Iberoamericana para el Avance de la Ingeniería de Software Cuántico*), to 113 researchers who were referred to us by the network researchers themselves, and 39 Latin American researchers from the papers extracted from the systematic mapping study. From this survey, we extracted revealing results such as the fact that most of the researchers in this field come from universities, that more than half of the researchers in this field obtain public funding for their research, or that most of them work in the field of quantum algorithm development; among other results already described.

In the mid-term, we expect that as quantum technologies transition from laboratory to industrial applications, the demand for comprehensive software engineering frameworks tailored to the unique characteristics of quantum systems will grow. This includes the development of tools and environments that facilitate the integration of

classical and quantum computing, a task that requires new methods for debugging, testing, and deploying hybrid quantum-classical systems.

In short, the future of Quantum Software Engineering in Latin America hinges on strengthening regional collaboration, fostering interdisciplinary research, and investing in education tailored to quantum computing. By enhancing cooperation across countries, sharing knowledge, and pooling resources, the region can overcome fragmentation in research efforts. Emphasizing interdisciplinary approaches, combining insights from physics, engineering, and computer science, will drive innovation in quantum algorithms and software tools. Additionally, promoting open science and democratizing access to quantum computing resources will accelerate progress and allow broader participation. Ultimately, the development of robust software engineering frameworks for hybrid quantum-classical systems will be essential for the successful industrial application of quantum technologies.

**Acknowledgements** This work was partially funded by the European Union “Next GenerationEU /PRTR” by the Ministry of Science, Innovation and Universities (projects PID2021-1240454OB-C31, TED2021-130913B-I00, and PDC2022-133465-I00). It was also supported by QSERV: Quantum Service Engineering: Development Quality, Testing and Security of Quantum Microservices project funded by the Spanish Ministry of Science and Innovation and ERDF; by the Regional Ministry of Economy, Science and Digital Agenda of the Regional Government of Extremadura (GR21133); and by the European Union under Agreement 101083667 of the Project “TECH4E -Tech4efficiencyEDIH” regarding the Call: DIGITAL-2021-EDIH-01 supported by the European Commission through the Digital Europe Program. This study was also supported by the QSALUD project (EXP 00135977 / MIG-20201059) in line with the actions of the Center for the Development of Industrial Technology (CDTI). This study was also supported by the “Ingeniería de Software Cuántico” project, funded by the Faculty of Informatics of the National University of La Plata.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J.L., Serrano, M.A., Hernández, G., Guzmán, I.G.R., Paradela, C.A., Polo, M., Murina, E., et al: The talavera manifesto for quantum software engineering and programming. In: QANSWER, pp. 1–5 (2020). [https://www.aquantum.es/wp-content/uploads/2020/03/Talavera\\_Manifesto.pdf](https://www.aquantum.es/wp-content/uploads/2020/03/Talavera_Manifesto.pdf)
2. Salm, M., Barzen, J., Breitenbücher, U., Leymann, F., Weder, B., Wild, K.: The nisq analyzer: automating the selection of quantum computers for quantum algorithms. In: Service-oriented computing: 14th symposium and summer school on service-oriented computing, summerSOC 2020, Crete, Greece, September 13–19, 2020 14, pp. 66–85 (2020). [https://doi.org/10.1007/978-3-030-64846-6\\_5](https://doi.org/10.1007/978-3-030-64846-6_5). Springer
3. Gaitan, F.: Finding flows of a navier-stokes fluid through quantum computing. npj Quantum Inf. 6(1), 1–6 (2020). <https://doi.org/10.1038/s41534-020-00291-0>
4. Moguel, E., Rojo, J., Valencia, D., Berrocal, J., Garcia-Alonso, J., Murillo, J.M.: Quantum service-oriented computing: current landscape and challenges. Softw. Qual. J. 30(4), 983–1002 (2022). <https://doi.org/10.1007/s11219-022-09589-y>

5. IEEE: Ieee standard glossary of software engineering terminology. IEEE Std 610.12-1990, 1–84 (1990). <https://doi.org/10.1109/IEEESTD.1990.101064>
6. Basiri, A., Behnam, N., Rooij, R.D., Hochstein, L., Kosewski, L., Reynolds, J., Rosenthal, C.: Chaos engineering. *IEEE Softw.* **33**, 35–41 (2016). <https://doi.org/10.1109/MS.2016.60>
7. Zhao, J.: Quantum software engineering: Landscapes and horizons. arXiv (2020). <https://doi.org/10.48550/arXiv.2007.07047>
8. Juan M.: Garcia-Alonso, E.E.A.M. Jose; Moguel: Challenges of quantum software engineering for the next decade: the road ahead. In: 2030 Software engineering on ACM international conference on the foundations of software engineering (FSE 2024) (2024). <https://doi.org/10.48550/arXiv.2404.06825>
9. Juárez-Ramírez, R., Navarro, C., Jiménez, S., Ramírez, A., Tapia-Ibarra, V., Guerra-García, C., Perez-Gonzalez, H.G., Fernández-y-Fernández, C.: A taxonomic view of the fundamental concepts of quantum computing-a software engineering perspective. *Program. Comput. Softw.* **49**(8), 682–704 (2023). <https://doi.org/10.1134/S0361768823080108>
10. Rojo, J., Valencia, D., Berrocal, J., Moguel, E., García-Alonso, J., Murillo, J.M.: Trials and tribulations of developing hybrid quantum-classical microservices systems. In: Perez-Castillo, R., Serrano, M.A., Piattini, M. (eds.) *International Workshop on Software Engineering & Technology 2021(Q-SET2021)*, vol. 3008, pp. 38–53 (2021). <https://ceur-ws.org/Vol-3008/paper2.pdf>
11. Pérez-Delgado, C.A., Perez-Gonzalez, H.G.: Towards a quantum software modeling language. In: *Proceedings of the IEEE/ACM 42nd international conference on software engineering workshops. ICSEW'20*, pp. 442–444. Association for computing machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3387940.3392183>
12. Ali, S., Yue, T.: Modeling quantum programs: challenges, initial results, and research directions. In: *Proceedings of the 1st ACM SIGSOFT international workshop on architectures and paradigms for engineering quantum software. APEQS 2020*, pp. 14–21. Association for computing machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3412451.3428499>
13. Cartiere, C.R.: Formal methods for quantum software engineering. *Quantum Softw. Eng.* (2022). [https://doi.org/10.1007/978-3-031-05324-5\\_5](https://doi.org/10.1007/978-3-031-05324-5_5)
14. Feng, Y., Yu, N., Ying, M.: Model checking quantum markov chains. *J. Comput. Syst. Sci.* **79**(7), 1181–1198 (2013). <https://doi.org/10.1016/j.jcss.2013.04.002>
15. Aparicio-Morales, Á.M., Herrera, J.L., Moguel, E., Berrocal, J., García-Alonso, J., Murillo, J.M.: Minimizing deployment cost of hybrid applications. In: *2023 IEEE international conference on quantum computing and engineering (QCE)*, vol. 02, pp. 191–194 (2023). <https://doi.org/10.1109/QCE57702.2023.10209>
16. Kim, Y., Eddins, A., Anand, S., Wei, K.X., Van Den Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K., et al.: Evidence for the utility of quantum computing before fault tolerance. *Nature* **618**(7965), 500–505 (2023). <https://doi.org/10.1038/s41586-023-06096-3>
17. Valencia, D., Moguel, E., Rojo, J., Berrocal, J., García-Alonso, J., Murillo, J.M.: Quantum service-oriented architectures: from hybrid classical approaches to future stand-alone solutions. *Quantum Softw. Eng.* (2022). [https://doi.org/10.1007/978-3-031-05324-5\\_8](https://doi.org/10.1007/978-3-031-05324-5_8)
18. Romero-Álvarez, J., Alvarado-Valiente, J., Moguel, E., García-Alonso, J., Murillo, J.M.: Using open api for the development of hybrid classical-quantum services. In: *International conference on service-oriented computing*, pp. 364–368 (2022). [https://doi.org/10.1007/978-3-031-26507-5\\_34](https://doi.org/10.1007/978-3-031-26507-5_34)
19. García-Alonso, J., Rojo, J., Valencia, D., Moguel, E., Berrocal, J., Murillo, J.M.: Quantum software as a service through a quantum api gateway. *IEEE Internet Comput.* **26**(1), 34–41 (2022). <https://doi.org/10.1109/MIC.2021.3132688>
20. Hidary, J.D., Hidary, J.D.: *Quantum Computing: An Applied Approach*. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-83274-2>
21. Miranskyy, A., Zhang, L.: On testing quantum programs. In: *2019 IEEE/ACM 41st international conference on software engineering: new ideas and emerging results (ICSE-NIER)*, pp. 57–60 (2019). <https://doi.org/10.1109/ICSE-NIER.2019.00023>. IEEE
22. Barrera, A.G., Guzmán, I.G.-R., Polo, M., Cruz-Lemus, J.A.: In: Serrano, M.A., Pérez-Castillo, R., Piattini, M. (eds.) *Quantum Software Testing: Current Trends and Emerging Proposals*, pp. 167–191. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-05324-5\\_9](https://doi.org/10.1007/978-3-031-05324-5_9)
23. Piattini, M., Serrano, M., Perez-Castillo, R., Petersen, G., Hevia, J.L.: Toward a quantum software engineering. *IT Professional* **23**(1), 62–66 (2021). <https://doi.org/10.1109/MITP.2020.3019522>
24. Pérez-Castillo, R.: Reengineering of information systems toward classical-quantum systems. In: *QANSWER*, pp. 64–70 (2020). <https://ceur-ws.org/Vol-2561/paper7.pdf>

25. Kitchenham, B.A., Budgen, D., Brereton, O.P.: Using mapping studies as the basis for further research - a participant-observer case study. *Inf. Softw. Technol.* **53**, 638–651 (2011). <https://doi.org/10.1016/J.INFSOF.2010.12.011>
26. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering, pp. 71–80 (2008). <https://dl.acm.org/doi/10.5555/2227115.2227123>
27. Rosenfeld, A.: Is dblp a good computer science journals database? *Computer* **56**(3), 101–108 (2023). <https://doi.org/10.1109/MC.2022.3181977>
28. Stepney, S., Braunstein, S.L., Clark, J.A., Tyrrell, A., Adamatzky, A., Smith, R.E., Addis, T., Johnson, C., Timmis, J., Welch, P., Milner, R., Partridge, D.: Journeys in non-classical computation the challenge (2004). <https://www.cs.york.ac.uk/nature/gc7/journeys.pdf>
29. Preskill, J.: Quantum computing and the entanglement frontier. arxiv (2012). <https://arxiv.org/abs/1203.5813v3>
30. Neto, F.M.D.P., Ludermir, T.B., Oliveira, W.R.D., Silva, A.J.D.: Implementing any nonlinear quantum neuron. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(9), 3741–3746 (2020). <https://doi.org/10.1109/TNNLS.2019.2938899>
31. Araujo, I.F., Park, D.K., Ludermir, T.B., Oliveira, W.R., Petruccione, F., Silva, A.J.: Configurable sublinear circuits for quantum state preparation. *Quantum Inf. Process.* (2023). <https://doi.org/10.1007/s11128-023-03869-7>
32. Plata-Cesar, N., Marcial-Romero, J.R., Hernandez-Servin, J.A.: Reversibility for quantum programming language QML. *IEEE Latin Am. Trans.* **18**, 1692–1698 (2020). <https://doi.org/10.1109/TLA.2020.9387639>
33. Plata-Cesar, N., Marcial-Romero, J.R., Hernández-Servín, J.A.: Measurements in quantum programming language QML. *CEUR-WS 2585*, 159–168 (2019). <https://ceur-ws.org/Vol-2585/paper15.pdf>
34. César, N.P., Romero, J.R.M., Servín, J.A.H.: A history and reversibility for quantum programming language QML. *CEUR-WS 2264*, 25–39 (2018). <https://ceur-ws.org/Vol-2264/paper3.pdf>
35. Díaz-Caro, A., Martínez, G.: Confluence in probabilistic rewriting. *Electron. Notes in Theor. Comput. Sci.* **338**, 115–131 (2018). <https://doi.org/10.1016/j.entcs.2018.10.008>
36. Díaz-Caro, A., Dowek, G.: Typing quantum superpositions and measurement. *LNCS* **281–29**, 10687 (2017). [https://doi.org/10.1007/978-3-319-71069-3\\_22](https://doi.org/10.1007/978-3-319-71069-3_22)
37. Assaf, A., Díaz-Caro, A., Perdrix, S., Tasson, C., Valiron, B.: Call-by-value, call-by-name and the vectorial behaviour of the algebraic  $\lambda$ -calculus. *Log. Methods in Comput. Sci.* (2014). [https://doi.org/10.2168/LMCS-10\(4:8\)2014](https://doi.org/10.2168/LMCS-10(4:8)2014)
38. Sousa, R.B.D., Pereira, E.J.S., Cipolletti, M.P., Ferreira, T.A.E.: A proposal of quantum data representation to improve the discrimination power. *Nat. Comput.* **19**, 577–591 (2020). <https://doi.org/10.1007/S11047-019-09734-W>
39. Dinani, H.T., Tancara, D., Fanchini, F.F., Norambuena, A., Coto, R.: Estimating the degree of non-markovianity using variational quantum circuits. *Quantum Machine Intell.* **5**, 29 (2023). <https://doi.org/10.1007/S42484-023-00120-5>
40. Cárdenas-López, F.A., Sanz, M., Retamal, J.C., Solano, E.: Enhanced quantum synchronization via quantum machine learning. *Adv. Quantum Technol.* **2**, 7–8 (2019). <https://doi.org/10.1002/qute.201800076>
41. Saraiva, L., Haeusler, E.H., Costa, V., Kalinowski, M.: Non-functional requirements for quantum programs. *CEUR-WS 3008*, 89–100 (2021). <https://ceur-ws.org/Vol-3008/paper4.pdf>
42. Díaz, T.G.J., Barrios, H.C.J., Steffene, L.A., Couturier, J.F.: Nearly quantum computing by simulation **1660 CCIS**, 205–219 (2022). [https://doi.org/10.1007/978-3-031-23821-5\\_15](https://doi.org/10.1007/978-3-031-23821-5_15)
43. Silva Feitosa, S., Vizzotto, J.K., Piveta, E.K., Bois, A.R.D.: A monadic semantics for quantum computing in an object oriented language. *Sci. Comput. Program.* **173**, 37–55 (2019). <https://doi.org/10.1016/j.scico.2018.03.003>
44. Feitosa, S.S., Vizzotto, J.K., Piveta, E.K., Bois, A.R.D.: Fjquantum - a quantum object oriented language. *Electron. Notes in Theor. Comput. Sci.* **324**, 67–77 (2016). <https://doi.org/10.1016/j.entcs.2016.09.007>
45. Gejea, A.M., Mayakannan, S., Palacios, R.M., Hamad, A.A., Sundaram, B., Alghamdi, W.: A novel approach to grover’s quantum algorithm simulation: cloud-based parallel computing enhancements, 1740–1745 (2023). <https://doi.org/10.1109/ICOCSE58147.2023.10276383>

46. Oliveira, D., Giusto, E., Dri, E., Casciola, N., Baheri, B., Guan, Q., Montrucchio, B., Rech, P.: Qufi: a quantum fault injector to measure the reliability of qubits and quantum circuits, 137–149 (2022). <https://doi.org/10.1109/DSN53405.2022.00025>
47. Dias, D.M., Pacheco, M.A.C.: Quantum-inspired linear genetic programming as a knowledge management system. *Comput. J.* **56**, 1043–1062 (2013). <https://doi.org/10.1093/comjnl/bxs108>
48. Silva, C., Aguiar, A., Lima, P.M.V., Dutra, I.: Mapping a logical representation of tsp to quantum annealing. *Quantum Inf. Process.* (2021). <https://doi.org/10.1007/s11128-021-03321-8>
49. Ballinas, E., Montiel, O. (2022) Hybrid quantum genetic algorithm for the 0-1 knapsack problem in the ibm qiskit simulator. *Computación y Sistemas.* **26**, 725–742. 10.13053/CyS-26-2-4253
50. Leal, D., Lima, T.D., Silva, A.J.D.: Training ensembles of quantum binary neural networks **2021**, (2021). <https://doi.org/10.1109/IJCNN52387.2021.9534253>
51. Payares, E., Martínez-Santos, J.C.: Advancements in quantum machine learning for intrusion detection: a comprehensive overview, pp. 167–176. IGI Global, Hershey, PA, USA (2023). <https://doi.org/10.4018/978-1-6684-8422-7.ch009>
52. Dinani, H.T., Tancara, D., Fanchini, F.F., Norambuena, A., Coto, R.: Estimating the degree of non-markovianity using variational quantum circuits. *Quantum Machine Intell.* **5**, 29 (2023). <https://doi.org/10.1007/s42484-023-00120-5>
53. Brokowski, T.J., Chowdhury, F.T., Smith, L.D., Alvarez, P., Sandeep, S., Aiello, C.: Spin chemistry simulation via hybrid-quantum machine learning, 867–868 (2022). <https://doi.org/10.1109/QCE53715.2022.00147>
54. Yadav, S.P., Singh, R., Yadav, V., Al-Turjman, F., Kumar, S.A.: Quantum-safe cryptography algorithms and approaches: impacts of quantum computing on cybersecur, pp. 1–326. De Gruyter, Berlin, Germany (2023). <https://doi.org/10.1515/9783110798159>
55. Singamaneni, K.K., Nauman, A., Juneja, S., Dhiman, G., Viriyasitavat, W., Hamid, Y., Anajemba, J.H.: An efficient hybrid qhpc-abe model to improve cloud data integrity and confidentiality. *Electronics* **11**(21), 3510 (2022). <https://doi.org/10.3390/electronics11213510>
56. Yan, F., Venegas-Andraca, S.E., Hirota, K.: Toward implementing efficient image processing algorithms on quantum computers. *Soft. Comput.* **27**, 13115–13127 (2023). <https://doi.org/10.1007/s00500-021-06669-2>
57. El-Latif, A.A.A., Abd-El-Atty, B., Mehmood, I., Muhammad, K., Venegas-Andraca, S.E., Peng, J.: Quantum-inspired blockchain-based cybersecurity: Securing smart edge utilities in iot-based smart cities. *Inf. Process. Manage.* **58**, 102549 (2021). <https://doi.org/10.1016/j.ipm.2021.102549>
58. Lavor, C., Marquezino, F., Oliveira, A., Portugal, R.: A quantum approach to the discretizable molecular distance geometry problem. *Quantum Inf. Process.* **21**, 239 (2022). <https://doi.org/10.1007/s11128-022-03583-w>
59. Acasiete, F., Agostini, F.P., Moqadam, J.K., Portugal, R.: Implementation of quantum walks on ibm quantum computers. *Quantum Inf. Process.* **19**, 426 (2020). <https://doi.org/10.1007/s11128-020-02938-5>
60. Paula Neto, F.M., Silva, A.J., Oliveira, W.R., Ludermer, T.B.: Quantum probabilistic associative memory architecture. *Neurocomputing* **351**, 101–110 (2019). <https://doi.org/10.1016/j.neucom.2019.03.078>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

Álvaro M. Aparicio-Morales<sup>1</sup> · Enrique Moguel<sup>1</sup> · Luis Mariano Bibbo<sup>2</sup> · Alejandro Fernandez<sup>2,3</sup> · Jose Garcia-Alonso<sup>1</sup> · Juan M. Murillo<sup>1</sup>

✉ Álvaro M. Aparicio-Morales  
amapamor@unex.es

Enrique Moguel  
enrique@unex.es

Luis Mariano Bibbo  
lmbibbo@lifia.info.unlp.edu.ar

Alejandro Fernandez  
casco@lifia.info.unlp.edu.ar

Jose Garcia-Alonso  
jgaralo@unex.es

Juan M. Murillo  
juanmamu@unex.es

- 1 Departamento de Ingeniería de Sistemas Informáticos y Telemáticos, Universidad de Extremadura, Avenida de la Universidad, s/n, Cáceres 10004, Extremadura, Spain
- 2 LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Calles 50 y 120, La Plata 1900, Province of Buenos Aires, Argentina
- 3 Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Calle 526 e/ 10 y 11, La Plata 1900, Province of Buenos Aires, Argentina