Software Engineering For Quantum Computing

Thesis Proposal

Supervisor:
Prof. Antonio Brogi

Co-Supervisor:
Prof. Jose Garcia-Alonso

Candidate:
Giuseppe Bisicchia

Quantum Software Engineering

"The development of a ***full discipline*** *of* ***Quantum Software Engineering****, ready to exploit the* ***full potential of commercial quantum computer hardware****, once it arrives"*

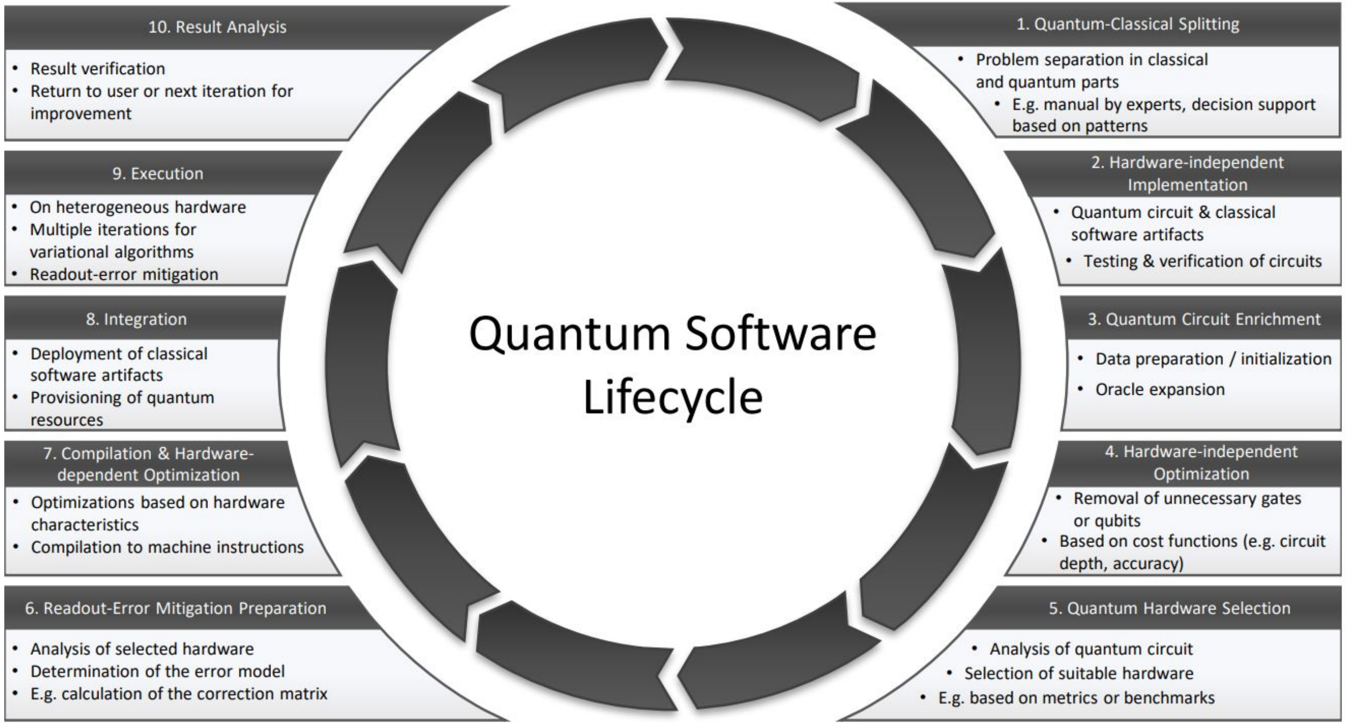
John Clark and Susan Stepney. 'Quantum software engineering'. 2002

Quantum Software Engineering

Jianjun Zhao. ‘Quantum software engineering: Landscapes and horizons’. In: arXiv preprint arXiv:2007.07047 (2020).

*“The use of **sound engineering principles** for the **development**, **operation**, and **maintenance** of **quantum software** and the associated document to obtain **economically** quantum software that is **reliable** and works **efficiently** on **quantum computers**”*

Quantum Software Lifecycles



Benjamin Weder et al. ‘The quantum software lifecycle’. 2020

The Talavera Manifesto for Quantum Software Engineering and Programming

Mario Piattini et al. ‘The talavera manifesto for quantum software engineering and programming.’ In: QANSWER. 2020, pp. 1–5.

QSE should be **agnostic** regarding **quantum programming languages** and **technologies**

embrace the **coexistence** of **classical** and **quantum computing**

support the **management** of **quantum software development projects**

consider the **evolution** of **quantum software**

aim at delivering quantum programs with **desirable zero defects**

promote **quantum software reuse**

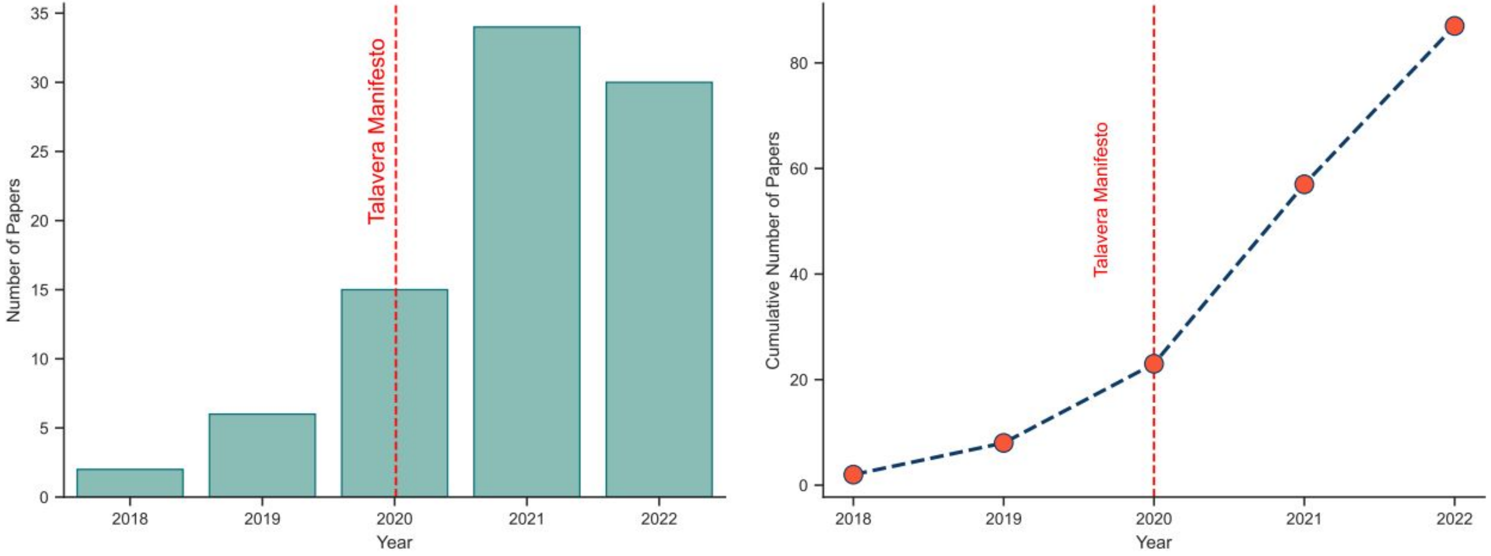
address **security** and **privacy by design**

cover the **governance** and **management** of software

*“The new QSE field needs to be considered as the application or adaptation of the **well-known methods**, **techniques**, and **practices** of **software engineering**. Some techniques can be **used just as they are** in classical computing. At the same time, however, **new methods** and **techniques** will be defined **specifically** for **quantum software production**”*

Manuel A Serrano, Ricardo Perez-Castillo, and Mario Piattini. Quantum Software Engineering. 2022

QSE: Current State

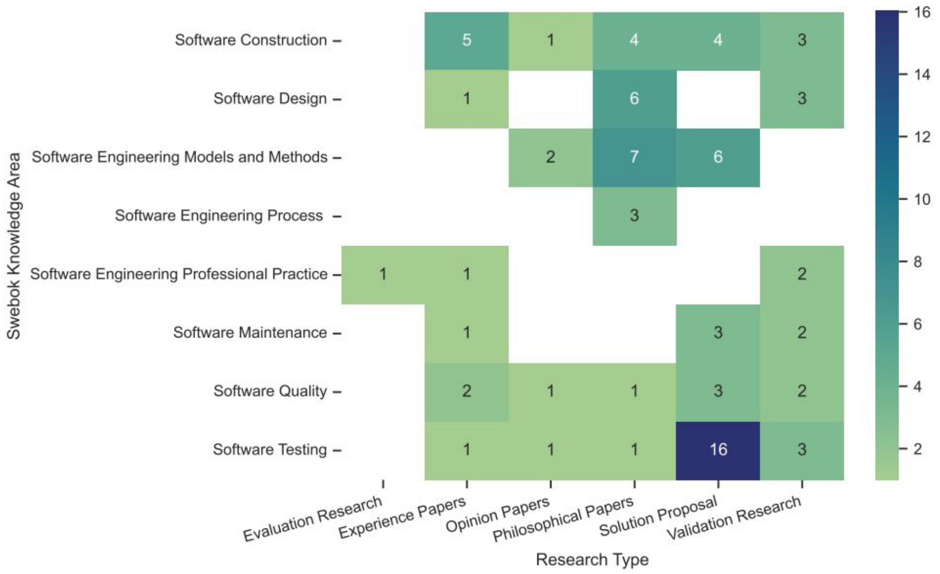


(a) Publication over years (b) Cumulative number of publications over the years

De Stefano, Manuel, et al. The Quantum Frontier of Software Engineering: A Systematic Mapping Study. 2023

QSE: Current State

4.1. RQ1: Main Topics and Studies in QSE



De Stefano, Manuel, et al. The Quantum Frontier of Software Engineering: A Systematic Mapping Study. 2023

QSE: Current State

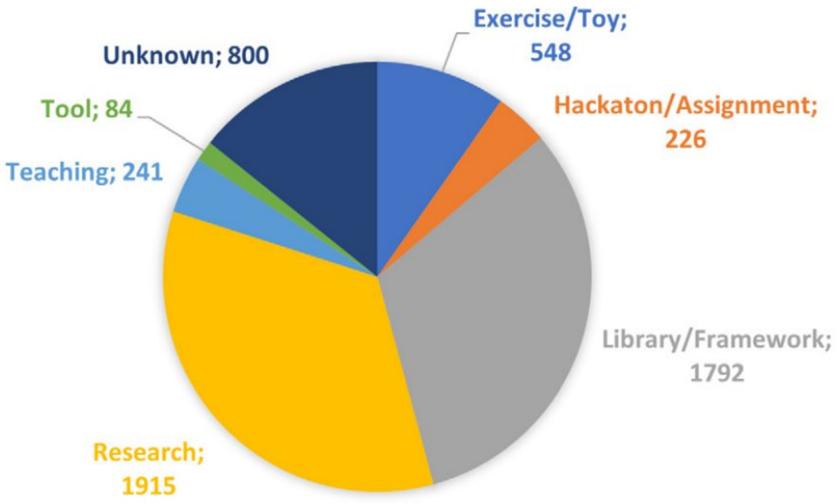
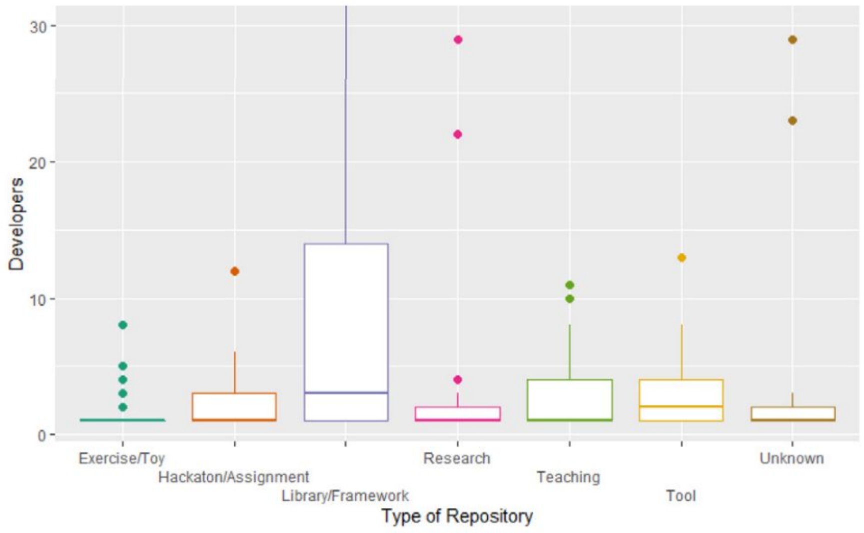


Fig. 3. Number of contributors per type of repository.



De Stefano, Manuel, et al. Software engineering for quantum programming: How far are we?. 2022

QSE: Tools & Languages

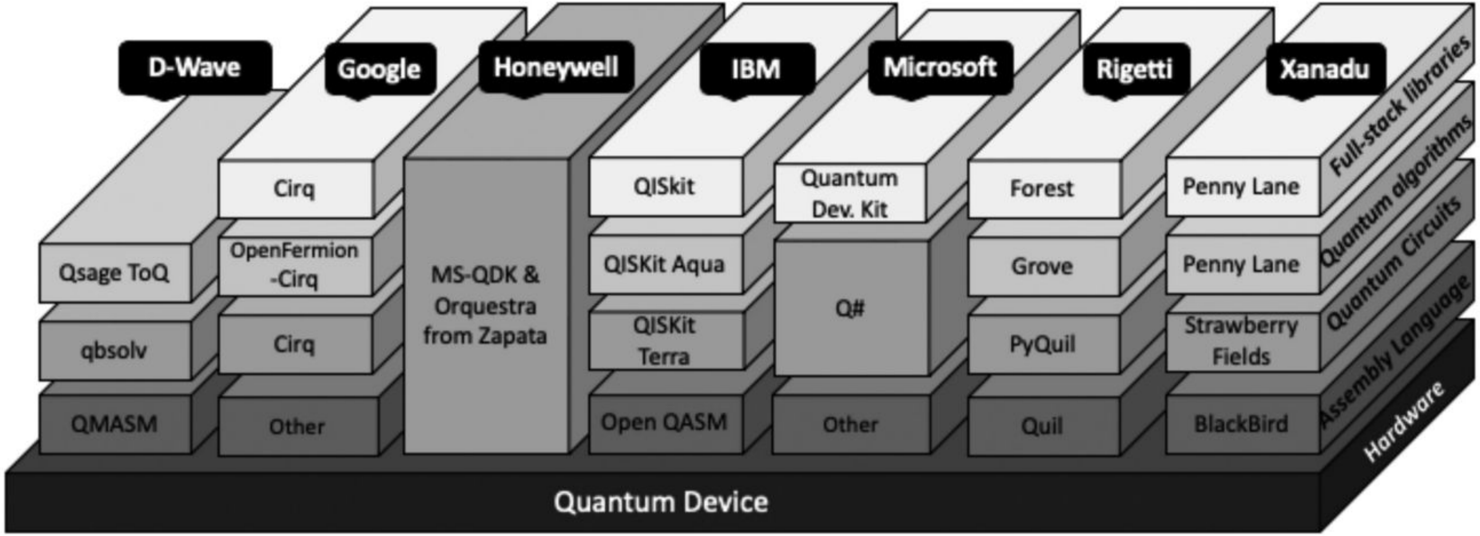


Fig. 10. Main quantum full stack platforms.

Serrano, Manuel A., et al. Quantum software components and platforms: Overview and quality assessment. 2022

QSE: Tools & Languages

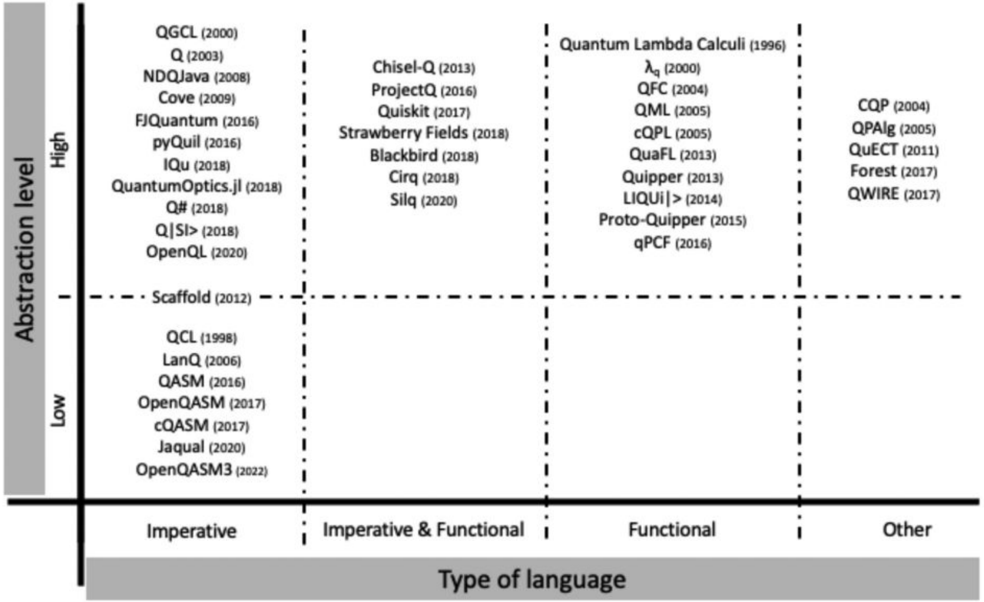


Fig. 7. Quantum programming languages landscape.

Serrano, Manuel A., et al. Quantum software components and platforms: Overview and quality assessment. 2022

QSE: Challenges

A TAXONOMY OF QUESTION CATEGORIES WHICH BASES ON AND EXTENDS [76]

Category	Description	Freq
API usage	Questions of this category are usually identified by "how to", i.e., how to use an API or how to implement a functionality.	85
Theoretical*	This category of questioners ask about theoretical explanations of quantum programs, algorithms, and concepts.	54
Errors	This category of questions search for explanations and solutions of errors and exceptions when developing or executing quantum programs.	49
Conceptual	Questions in this category are related to the limitation, background and the underlying concept of an API.	45
Discrepancy	Question of this category usually ask for explanations or solutions for unexpected results (e.g., "what is the problem", "why not work".	31
Learning	Questions in this category are searching for learning resources such as documentation, research papers, tutorials, or websites.	22
Review	This category describes questions like: "How/Why this is working?" or "Is there a better solution?". Generally, the questions in this category look for a better solution to a problem or for help reviewing the current solution.	17
Tooling*	This category describes questions like "I am looking for ...", "Is there a tool for ...". These questions search for tools to solve a specific problem or check the features of a tool.	16
API change	This category of questions concern about changes of an API and the associated compatibility issues and other implications.	2

*Categories newly identified in QSE-related questions.

Li, Heng, et al. Understanding quantum software engineering challenges an empirical study on stack exchange forums and GitHub issues. 2021

QSE: Challenges – Pure SE

A TAXONOMY OF QUESTION CATEGORIES WHICH BASES ON AND EXTENDS [76]

Category	Description	Freq
API usage	Questions of this category are usually identified by "how to", i.e., how to use an API or how to implement a functionality.	85
Theoretical*	This category of questioners ask about theoretical explanations of quantum programs, algorithms, and concepts.	54
Errors	This category of questions search for explanations and solutions of errors and exceptions when developing or executing quantum programs.	49
Conceptual	Questions in this category are related to the limitation, background and the underlying concept of an API.	45
Discrepancy	Question of this category usually ask for explanations or solutions for unexpected results (e.g., "what is the problem", "why not work".	31
Learning	Questions in this category are searching for learning resources such as documentation, research papers, tutorials, or websites.	22
Review	This category describes questions like: "How/Why this is working?" or "Is there a better solution?". Generally, the questions in this category look for a better solution to a problem or for help reviewing the current solution.	17
Tooling*	This category describes questions like "I am looking for ...", "Is there a tool for ...". These questions search for tools to solve a specific problem or check the features of a tool.	16
API change	This category of questions concern about changes of an API and the associated compatibility issues and other implications.	2

*Categories newly identified in QSE-related questions.

Li, Heng, et al. Understanding quantum software engineering challenges an empirical study on stack exchange forums and GitHub issues. 2021

QSE: Challenges – Quantum Related

A TAXONOMY OF QUESTION CATEGORIES WHICH BASES ON AND EXTENDS [76]

Category	Description	Freq
API usage	Questions of this category are usually identified by "how to", i.e., how to use an API or how to implement a functionality.	85
Theoretical*	This category of questioners ask about theoretical explanations of quantum programs, algorithms, and concepts.	54
Errors	This category of questions search for explanations and solutions of errors and exceptions when developing or executing quantum programs.	49
Conceptual	Questions in this category are related to the limitation, background and the underlying concept of an API.	45
Discrepancy	Question of this category usually ask for explanations or solutions for unexpected results (e.g., "what is the problem", "why not work".	31
Learning	Questions in this category are searching for learning resources such as documentation, research papers, tutorials, or websites.	22
Review	This category describes questions like: "How/Why this is working?" or "Is there a better solution?". Generally, the questions in this category look for a better solution to a problem or for help reviewing the current solution.	17
Tooling*	This category describes questions like "I am looking for ...", "Is there a tool for ...". These questions search for tools to solve a specific problem or check the features of a tool.	16
API change	This category of questions concern about changes of an API and the associated compatibility issues and other implications.	2

*Categories newly identified in QSE-related questions.

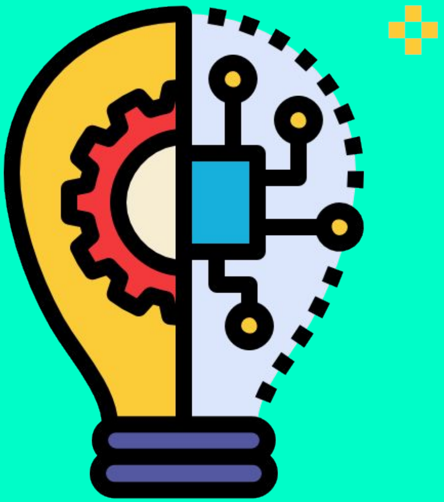
Li, Heng, et al. Understanding quantum software engineering challenges an empirical study on stack exchange forums and GitHub issues. 2021

QSE: Challenges

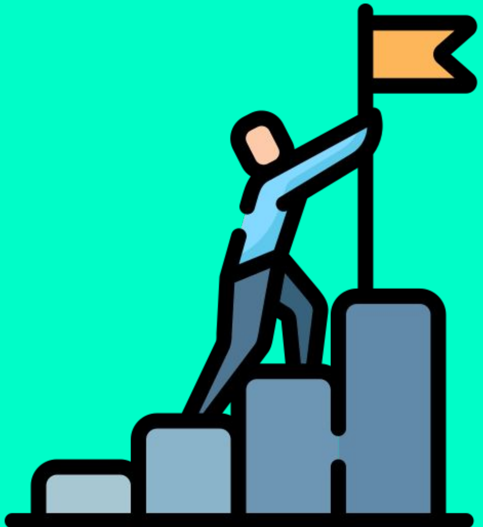
Fig. 8. Graphical representation of the obtained taxonomy of quantum computing challenges.

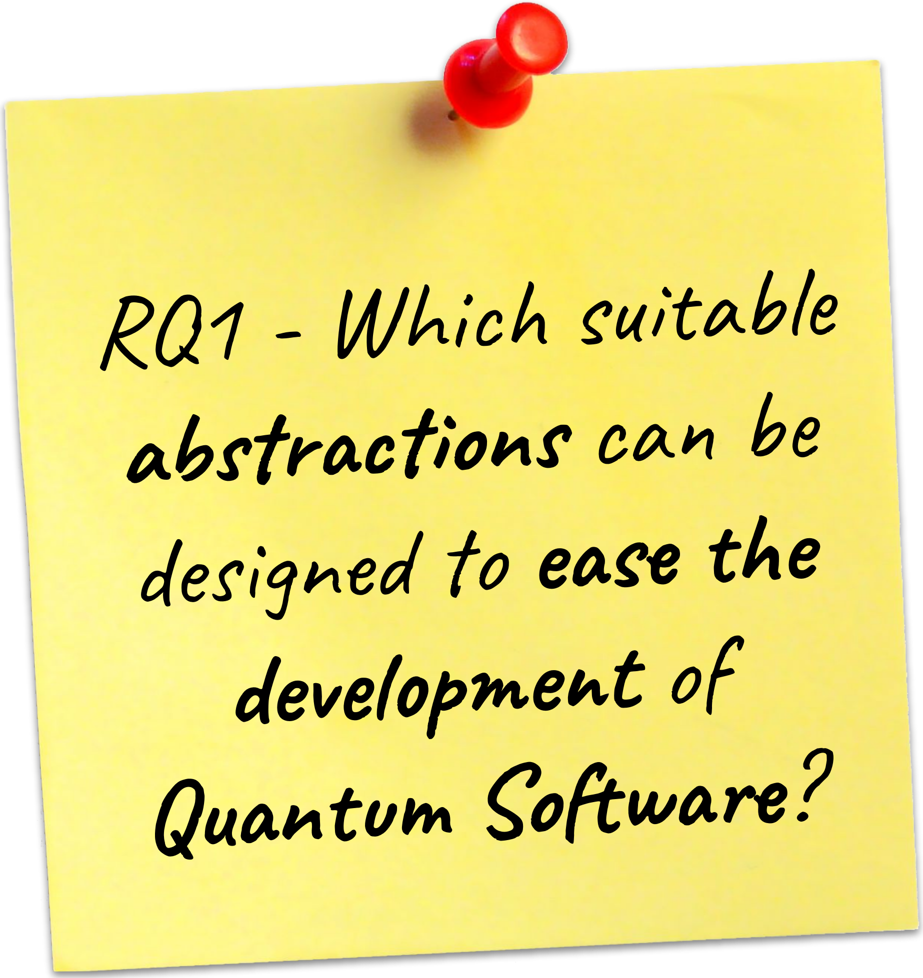
De Stefano, Manuel, et al. Software engineering for quantum programming: How far are we?. 2022

Thesis Objectives



To ***investigate, design*** *and* ***develop methodologies, techniques*** *and* ***prototypes*** *to* ***effectively*** *and* ***efficiently interact*** *and* ***use*** *current* ***NISQ devices,*** *and to* ***integrate Quantum Computations*** *in existing and next-generation* ***software pipelines*** *and* ***ecosystems***




RQ1 - Which suitable abstractions can be designed to ease the development of Quantum Software?

RQ2 - Which suitable methodologies and prototypes can be developed to support the Quantum Development Lifecycle?

(O1) Provide developers with **high-level abstractions** to **create**, and **operate quantum software (1)** on a quantum computer **without knowing the low-level details** of its **Quantum stack**, and **(2)** on **multiple quantum devices**

(02) Provide developers with **high-level abstractions** to **re-use** and **evolve quantum software** relieving them from knowing **how to write quantum circuits** from scratch

First Results

IBM Q

XANADU
Photonic Quantum Computing

D-Wave
The Quantum Computing Company™

Google AI
Quantum

rigetti

AQT

OQC

Amazon Braket

aws

Microsoft Azure

QCWARE
Forge

Quantum Inspire
By QuTech

ibm_cairo	27	64	2.4K	Online	16	Falcon r5.11	premium	OpenQASM 3
ibm_auckland Exploratory	27	64	2.4K	Online	1818	Falcon r5.11	premium	OpenQASM 3
ibm_hanoi	27	64	2.3K	Online - Queue paused	540	Falcon r5.11	premium	OpenQASM 3
ibm_peekskill Exploratory	27	-	-	Online	1	Falcon r8	premium	OpenQASM 3
ibmq_guadalupe	16	32	2.4K	Online - Queue paused	37	Falcon r4P	premium	
ibm_perth	7	32	2.9K	Online	126	Falcon r5.11H	open	OpenQASM 3
ibm_lagos	7	32	2.7K	Online - Reserved	64	Falcon r5.11H	open	OpenQASM 3

IonQ IonQ	Quantum Computing	Azure Quantum Credits
Microsoft Quantum Computing Microsoft	Quantum Computing	Learn & Develop
Quantinuum Quantinuum	Quantum Computing	Azure Quantum Credits
Rigetti Quantum Rigetti Computing	Quantum Computing	Azure Quantum Credits

Amazon Web Services	SV1	AVAILABLE NOW
Amazon Web Services	TN1	AVAILABLE NOW
Amazon Web Services	DM1	AVAILABLE NOW
IonQ	Harmony	02:44:47
IonQ	Aria 1	02:44:47
Oxford Quantum Circuits	Lucy	00:44:47
QuEra	Aquila	1 day 06:44:47
Rigetti	Aspen-M-3	05:44:47

The Quantum Daily's Guide to Qubit Implementations

THE QUANTUM DAILY
QUANTUM COMPUTING AND BEYOND

Classification	Description	Examples	Qubit lifetime (1)	Gate fidelity (2)	Gate operation time (3)	Connectivity	Scalability	Pros	Cons
Superconducting	Two level system of a superconducting circuit which forms a qubit (a transmon, first developed at Yale)	IBM, Google, Rigetti, Alibaba, Intel, Quantum Circuits	c.50-100μs	c.99.4%	c.10-50ns	Neighbours	Highly scalable (see OQC coaxmon tech)	- Fast gate times - Builds on existing semiconductor industry	- Typically low longevity - Must be kept very cold to work
Ion trap	Single charged ions trapped in magnetic fields. Energy level of its spin comprises the qubits	IonQ; Alpine Quantum Technologies; Honeywell	>1,000s	c.99.9%	c.3-50μs	All-to-all	TBC	- High gate fidelity - Very stable	- Slow operations
Photonics	Qubits made from single particles of light (photons) operating on silicon chips pathways	PsiQuantum, Xanadu	c. 150μs	c. 98.0%	c.1ns	Unknown	Highly scalable (see Psi Quantum)	- Highly scalable - Utilises existing SC industry infrastructure - No temperature requirements	- Nascent technology - Connectivity to be demonstrated
Neutral atoms	Qubits made from individual atoms (rather than ions which have a charge)	Atom Computing, PASQAL, QuEra	Similar to ion trap	c.95%	TBC	TBC	TBC	- Long qubit coherence times	- Must be kept cold - Nascent
Silicon	Artificial atoms made by adding an electron to a small piece of pure silicon and microwaves control the electrons state	Intel, Silicon Quantum Computing	c. 1-10s	c. 99%	c.1-10ns	Neighbours	Expect high scalability	- Stable - Utilises existing semiconductor industry infrastructure	- Must be kept cold - Nascent
Topological qubits	Qubits made from non-Abelian forms of matter	Microsoft (WIP)	Very high	Very high	Unknown	Unknown	Unknown	- Estimated long lifetime and high fidelities	- Existence to be confirmed

Notes: (1) *Record coherence time for a single qubit position state; (2) Highest reported fidelity for two qubit gate operations; (3) Speed of gate operations*

Sources: Literature review, TQD expert interviews. Special reference to BCG reports, Science Mag and NAE report on quantum computing.

s = seconds

μs = microsecond (10^ -6 seconds)

ns = nanosecond (10^ -9 seconds)

D-Wave
The Quantum Computing Company™

IBM Q

XANADU
Photonic Quantum Computing

Google AI Quantum

rigetti

AQT

Microsoft Azure

aws
Amazon Braket

OQC

QCWARE Forge

Quantum Inspire
By QuTech

IBM Q

XANADU
Photonic Quantum Computing

D-Wave
The Quantum Computing Company™

Google AI Quantum

rigetti

AQT

Microsoft Azure

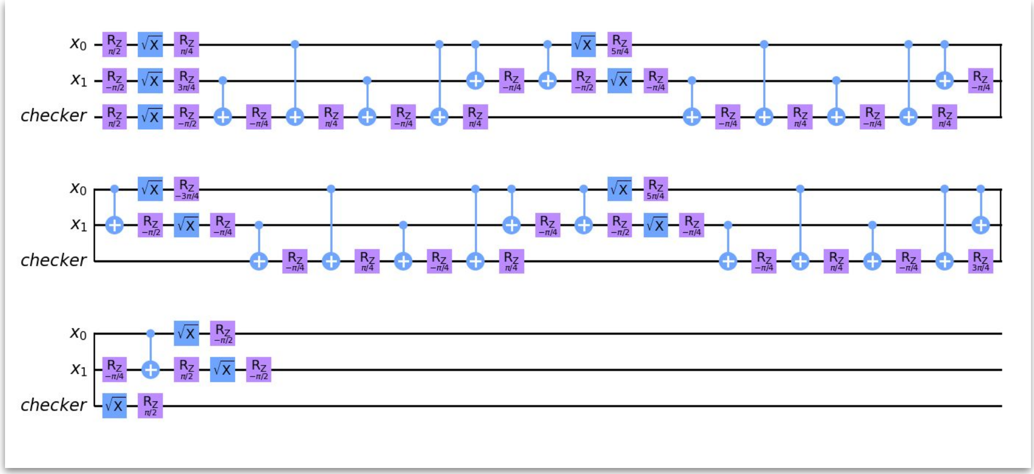
aws
Amazon Braket

OQC

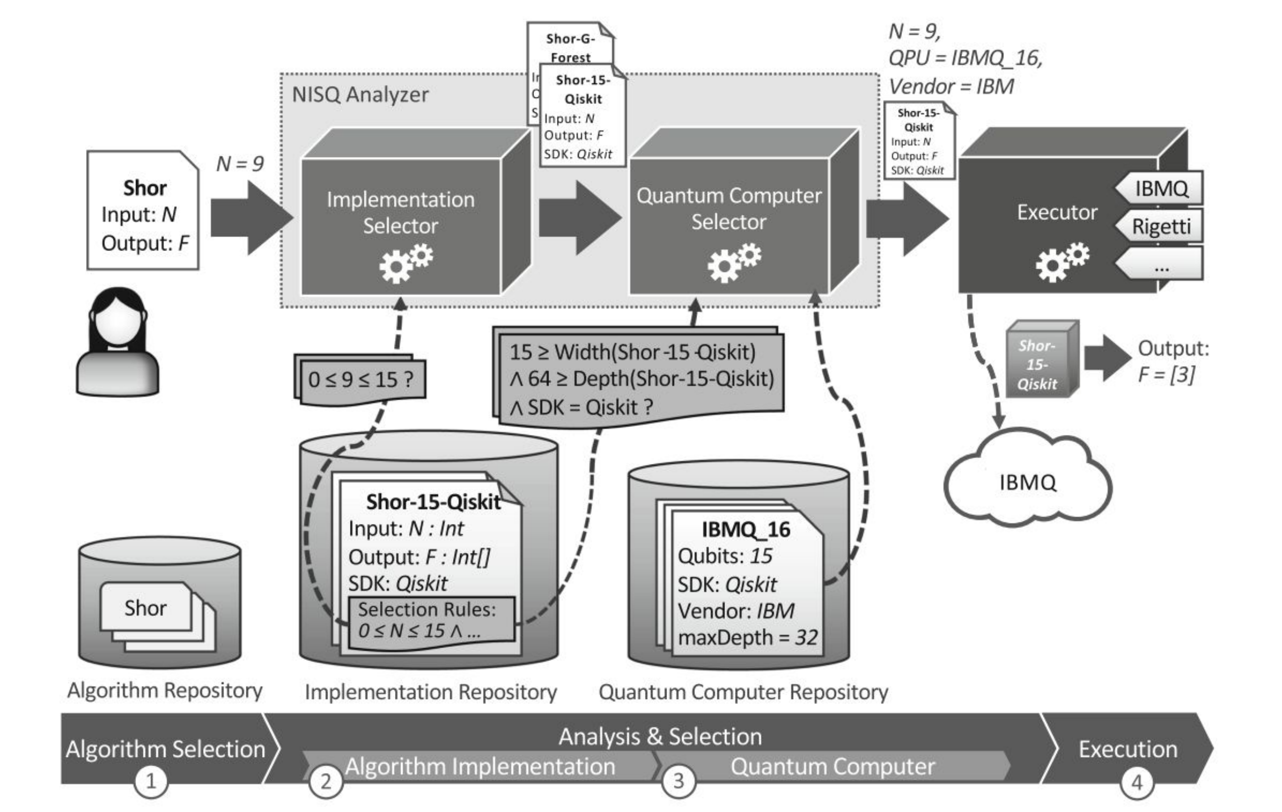
QCWARE Forge

Quantum Inspire
By QuTech

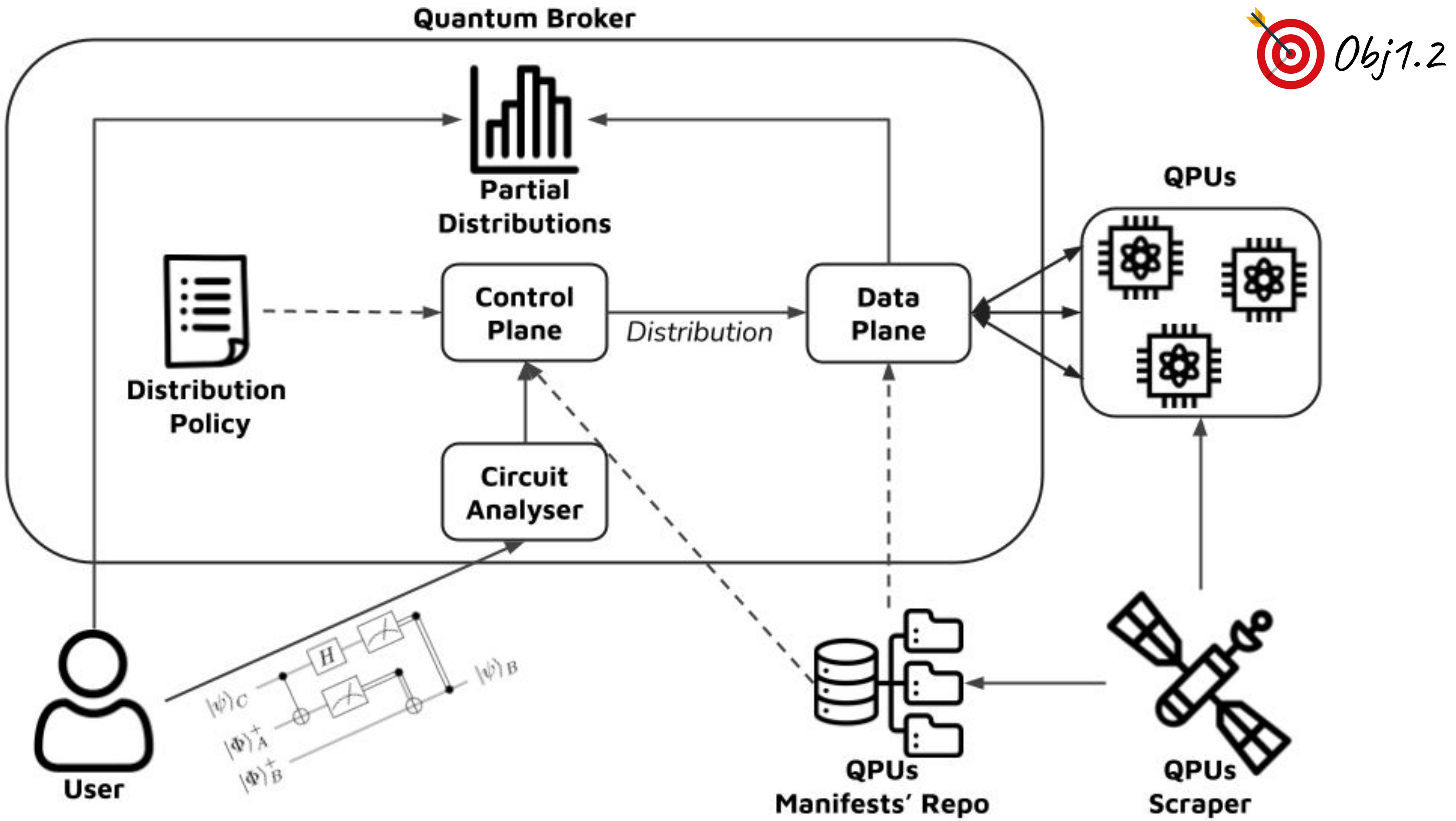
On which Quantum Computer should I run my circuit?



The NISQ Analyzer



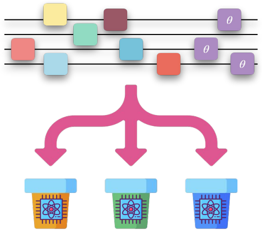
Salm, Marie, et al. *"The NISQ analyzer: automating the selection of quantum computers for quantum algorithms"*, 2020.



G. Bisicchia *et al.* "*Distributing Quantum Computations, by Shots*", ICSOC 2023

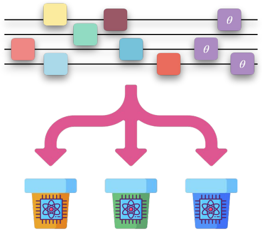
G. Bisicchia *et al.* "*Dispatching Shots Among Multiple Quantum Computers: an Architectural Proposal*", QCE 2023

RESULTS



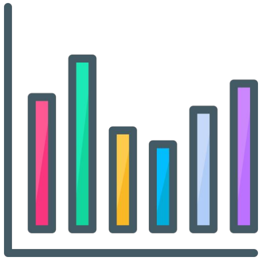
First proposal enabling the **distribution** of **shots** among **multiple Quantum Computers**

RESULTS

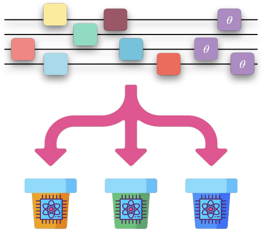


First proposal enabling the **distribution** of **shots** among **multiple Quantum Computers**

Users can access the **partial distributions** associated to a **circuit execution**

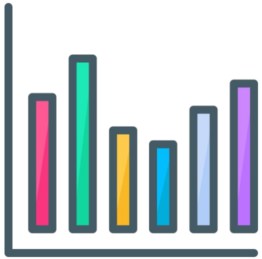


RESULTS



First proposal enabling the **distribution** of **shots** among **multiple Quantum Computers**

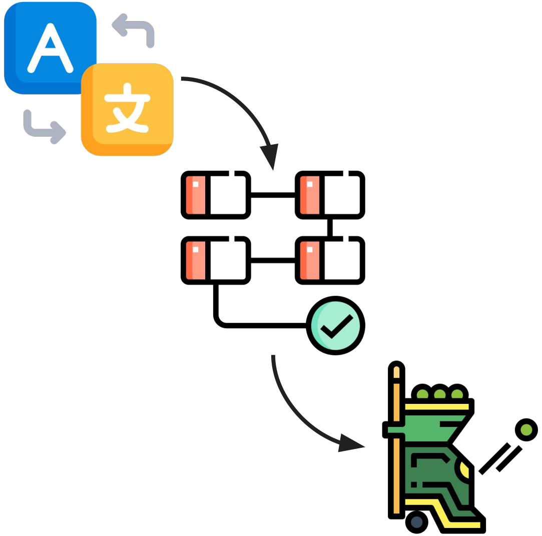
Users can access the **partial distributions** associated to a **circuit execution**



The **distribution decision process** is **decoupled** from the **control panel**: users can **customise** their own **policies**

Future Work

An Interoperable Pipeline for Quantum Computing



Develop a *Quantum Pipeline* to **abstract** the **complexity** of dealing with **numerous quantum languages**, **compilers**, **SDKs** and **providers**

Developers could exploit a *custom* **sequence of multiple compilers**, **optimisers** and **SDKs** (as *plugins*)

The final optimised quantum program could be **submitted** to one or **multiple quantum providers** (as *plugins*)

Obj1.1

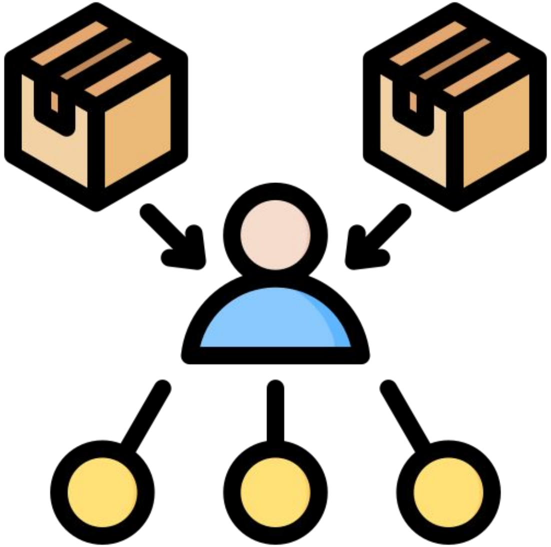
Re-Engineering the Quantum Broker

Develop **various distribution policies** and **heuristics**

Integrate the *Quantum Pipeline* with the *Quantum Broker*

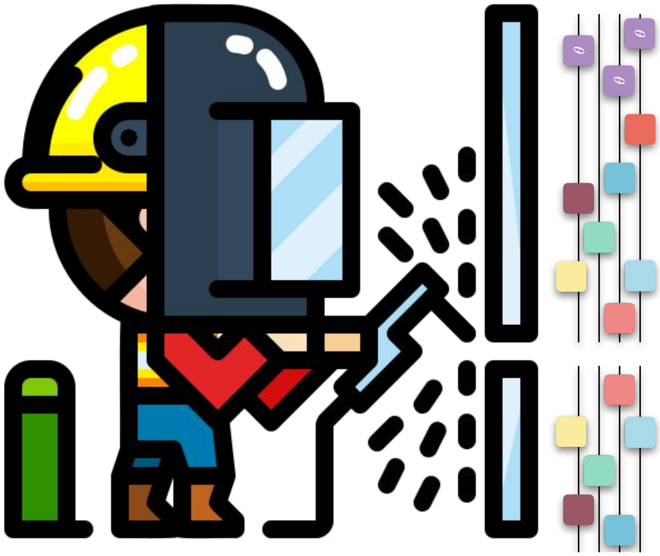
Define an **experimental setup** and **metrics**

Assess the *Quantum Broker* through **lifelike experiments**



Obj1.2

Welding Quantum Circuits



Investigate how to **improve re-usability, modularity and interoperability** in quantum development through **composing** existing **quantum circuits** as **fundamental building blocks**

Offer the solution *as-a-Service* also employing a **repository of composable quantum circuits** and the *Quantum Pipeline*

Develop **verification strategies** to prove when a composition is ***sound*** (what does it **mean** to be **sound** in **quantum programming**?)

Obj2

Develop a *Proof-of-Concept* of an **ecosystem** of **quantum prototypes** to guide the developers in their **quantum development lifecycles** and enable them to **seamless interactions** with *Quantum Computers*

Concluding

What have I done this year?

Passed all **6** required **exams**

Attended all **3** required **seminar series**

(**Organised** one of them!)

Attended the *Bertinoro International Spring School*

(Including the *Hybrid Quantum Computing course*!)

Teaching Assistant for the *Cloud Computing* course

What have I done this year?

Presented in **2(+1) conferences:**

- *CLOSER*
- *QCE/Q-SET*
- +*ICSOC* (this **November**!)

Published 1 journal article

2 journal and **1 conference** papers **under peer review**

1 month research stay at the *University of Extremadura*

[C1] G. Bisicchia, J. Garcia-Alonso, J. M. Murillo, and A. Brogi. "*Distributing Quantum Computations, by Shots*". Proceedings of the 21st International Conference on Service-Oriented Computing, ICSOC. 2023, *In Press*.

[C2] G. Bisicchia, J. Garcia-Alonso, J. M. Murillo, and A. Brogi. "*Dispatching Shots Among Multiple Quantum Computers: an Architectural Proposal*". Proceedings of the 4th IEEE International Conference on Quantum Computing and Engineering, QCE. 2023, *In Press*.

[J1] G. Bisicchia, S. Forti, and A. Brogi. "*Sustainable goal-oriented smart environments: a declarative programming approach*". Journal of Logic and Computation 33.4. 2023, pp. 864–899.

[C3] G. Bisicchia, S. Forti, A. Colla, and A. Brogi. "*Customisable Fault and Performance Monitoring Across Multiple Clouds*". Proceedings of the 13th International Conference on Cloud Computing and Services Science, CLOSER. 2023, pp. 212–219.

[J2] S. Forti, G. Bisicchia, and A. Brogi. "*Declarative continuous reasoning in the cloud-IoT continuum*". Journal of Logic and Computation 32.2. 2022, pp. 206–232.

[C4] G. Bisicchia, S. Forti, and A. Brogi. "*Declarative Goal Mediation in Smart Environments*". Proceedings of the 7th IEEE International Conference on Smart Computing, SMARTCOMP. 2021, pp. 389–391.

[C5] G. Bisicchia, S. Forti, and A. Brogi. "*A Declarative Goal-oriented Framework for Smart Environments with LPaaS*". Proceedings of the 36th Italian Conference on Computational Logic, CILC. Vol. 3002. CEUR Workshop Proceedings. 2021, pp. 143–157.