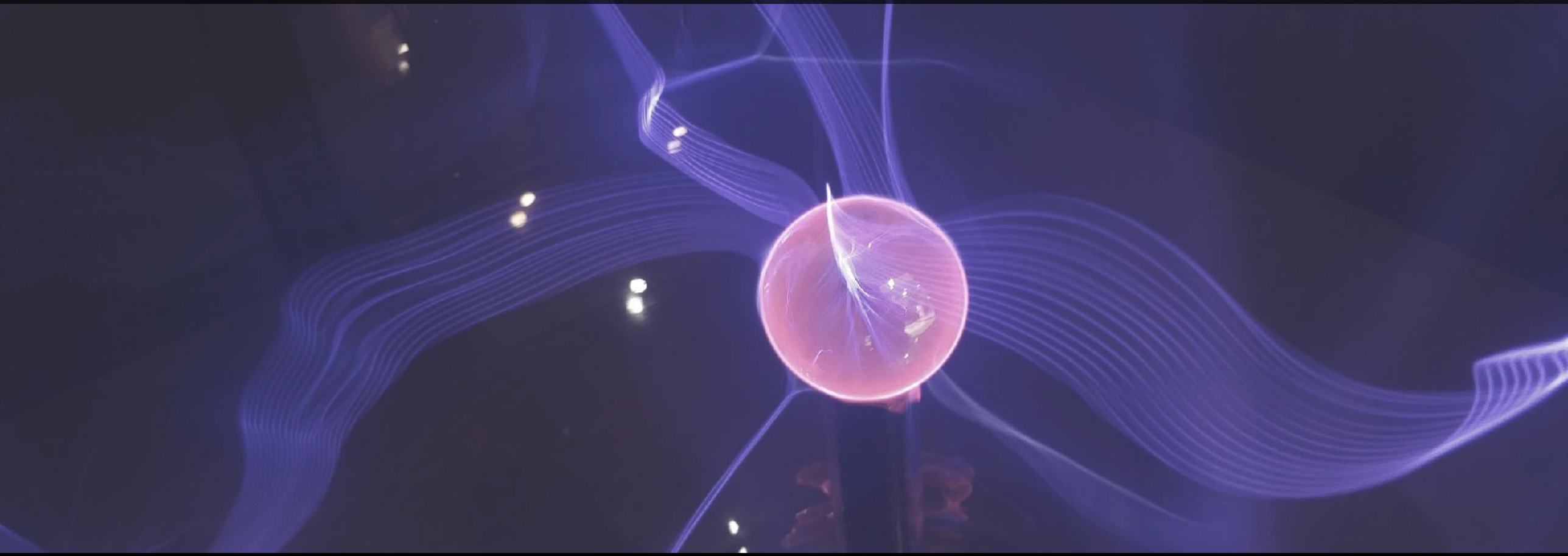
The Quantum Frontier of SE and its Adoption
Insights from Practitioners and a Systematic Mapping Study

sesa lab
SOFTWARE ENGINEERING SALERNO



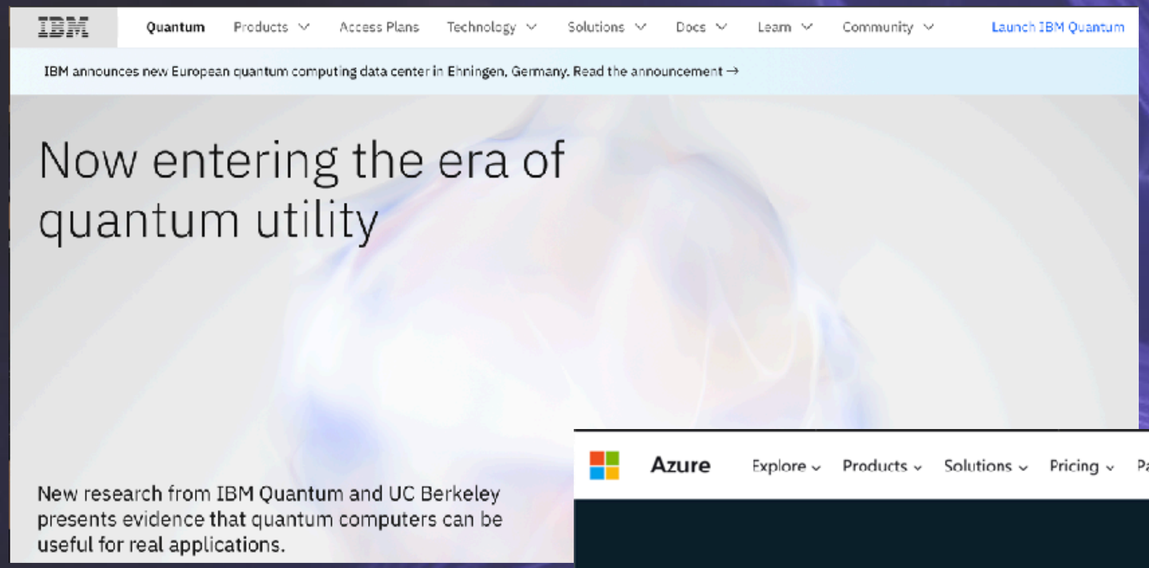
Dario Di Nucci

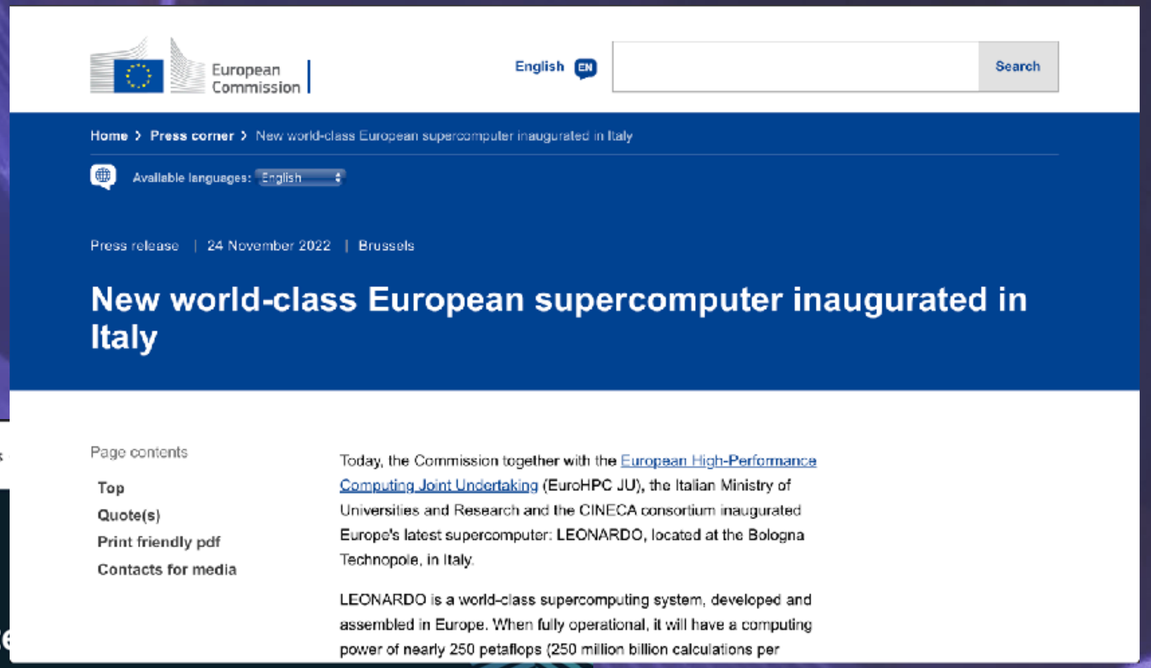
@dardin88

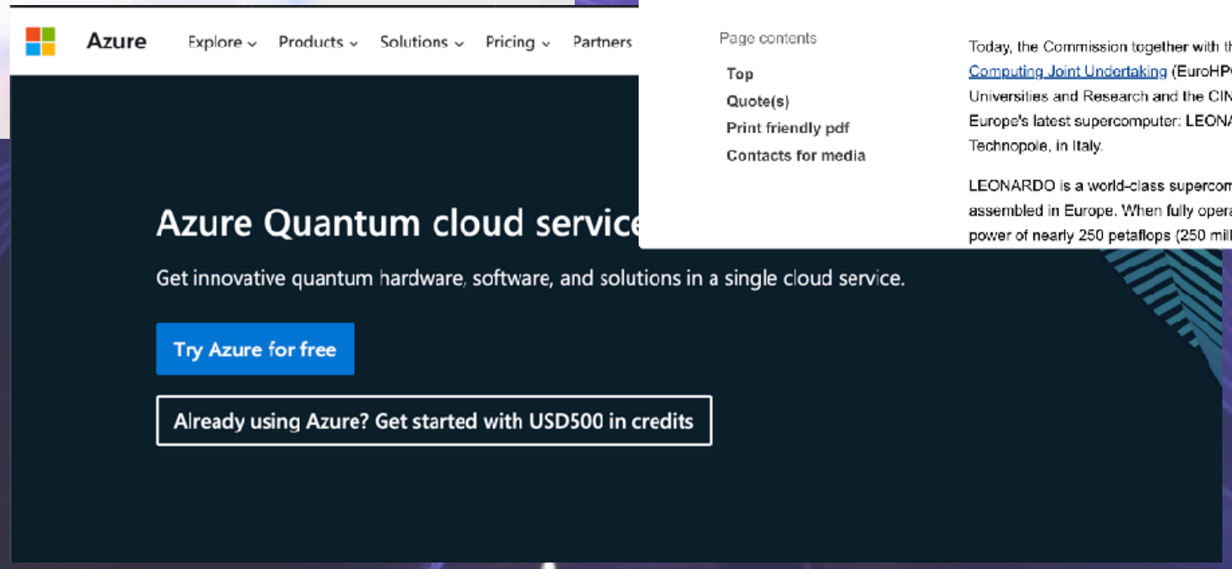
dardin88.github.io

ddinucci@unisa.it

Quantum software engineering: Insights from Practitioners



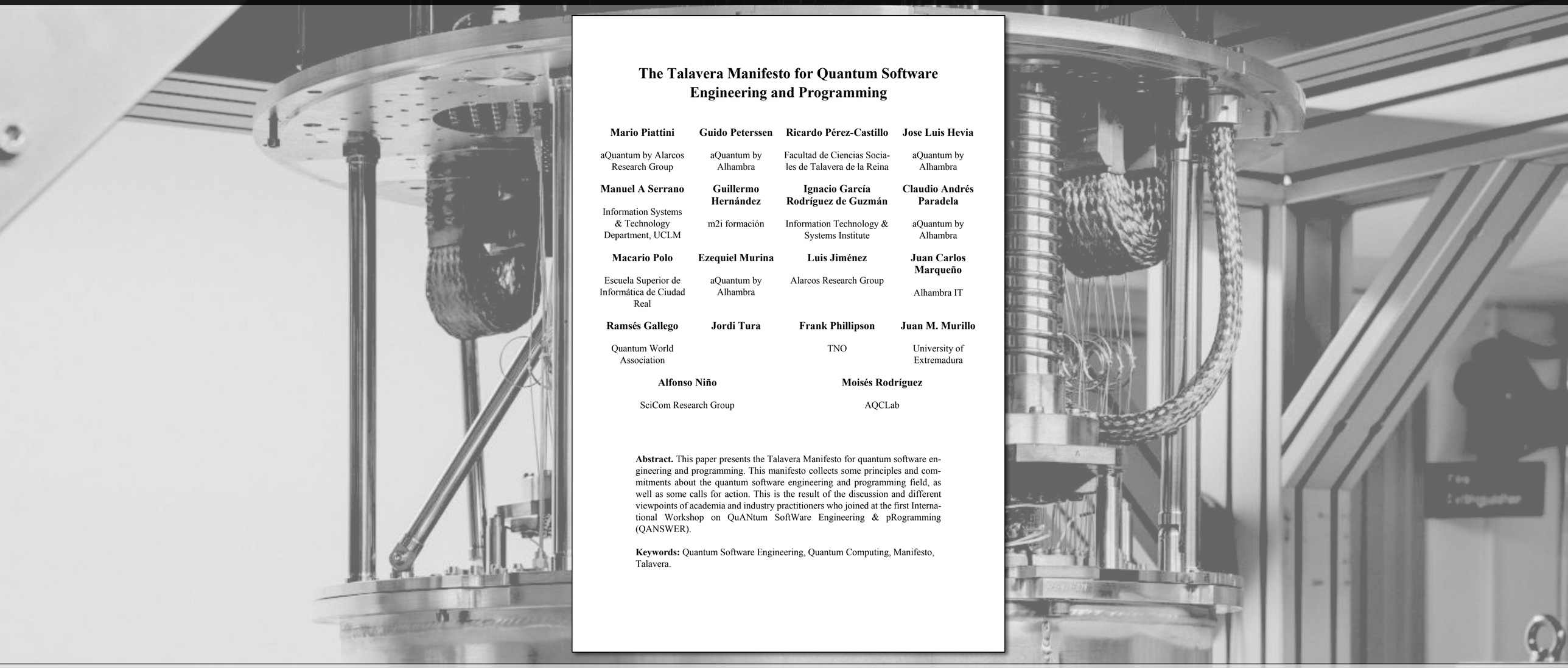




Quantum computing is now a reality!

sesa lab
SOFTWARE ENGINEERING SALERNO

The state of the art of Quantum Software Engineering



The Talavera Manifesto for Quantum Software Engineering and Programming

Mario Piattini
aQuantum by Alarcos Research Group

Guido Peterssen
aQuantum by Alhambra

Ricardo Pérez-Castillo
Facultad de Ciencias Sociales de Talavera de la Reina

Jose Luis Hevia
aQuantum by Alhambra

Manuel A Serrano
Information Systems & Technology Department, UCLM

Guillermo Hernández
m2i formación

Ignacio García Rodríguez de Guzmán
Information Technology & Systems Institute

Claudio Andrés Paradela
aQuantum by Alhambra

Macario Polo
Escuela Superior de Informática de Ciudad Real

Ezequiel Murina
aQuantum by Alhambra

Luis Jiménez
Alarcos Research Group

Juan Carlos Marqueño
Alhambra IT

Ramsés Gallego
Quantum World Association

Jordi Tura

Frank Phillipson
TNO

Juan M. Murillo
University of Extremadura

Alfonso Niño
SciCom Research Group

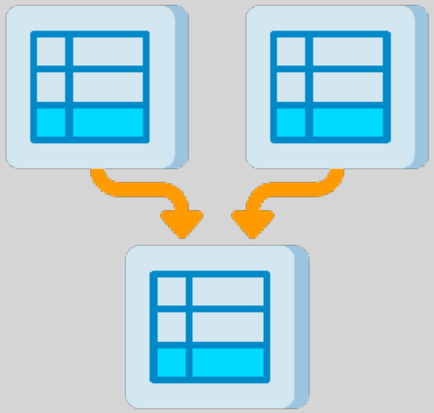
Moisés Rodríguez
AQCLab

Abstract. This paper presents the Talavera Manifesto for quantum software engineering and programming. This manifesto collects some principles and commitments about the quantum software engineering and programming field, as well as some calls for action. This is the result of the discussion and different viewpoints of academia and industry practitioners who joined at the first International Workshop on QuANtum SoftWare Engineering & pRogramming (QANSWER).

Keywords: Quantum Software Engineering, Quantum Computing, Manifesto, Talavera.

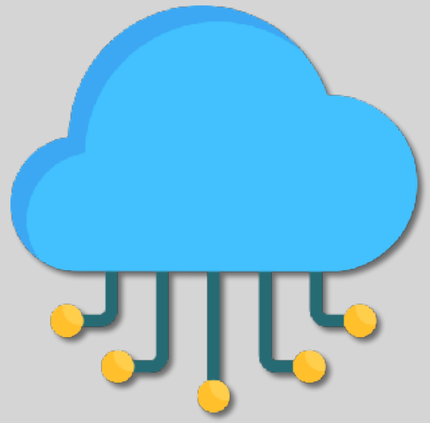
The Talavera Manifesto defines fundamental principles of Quantum Software Engineering.

Quantum software engineering: Insights from Practitioners



Modeling

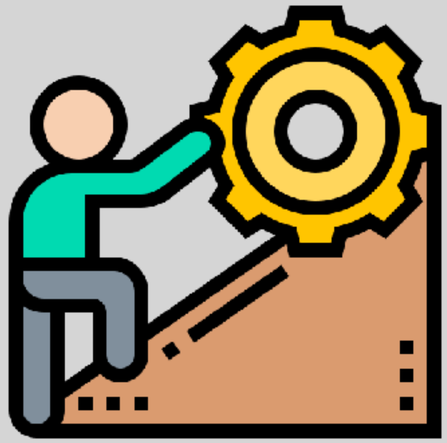
Testing



Cloud-Enabled Quantum Computing

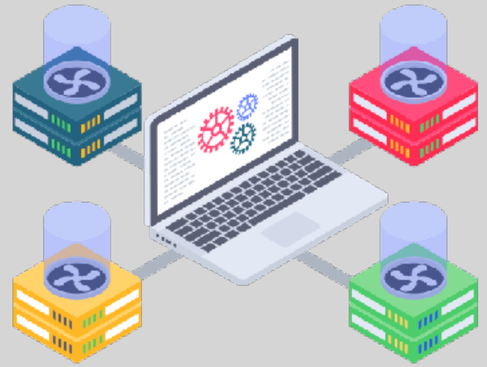
Several tools and techniques allows for modeling, testing, and cloud-enabled computing.

Quantum software engineering: Insights from Practitioners



Challenges

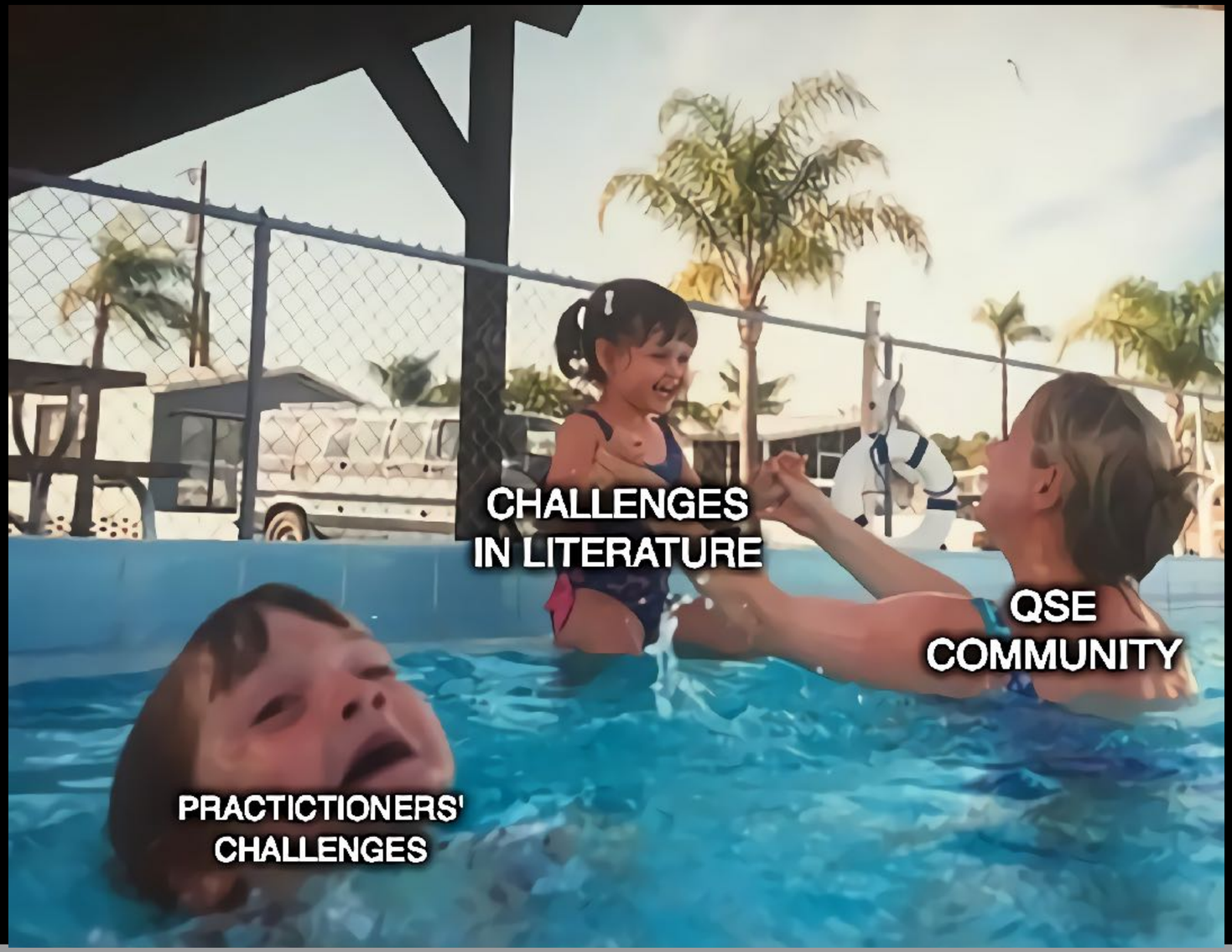
Testing



Architecture

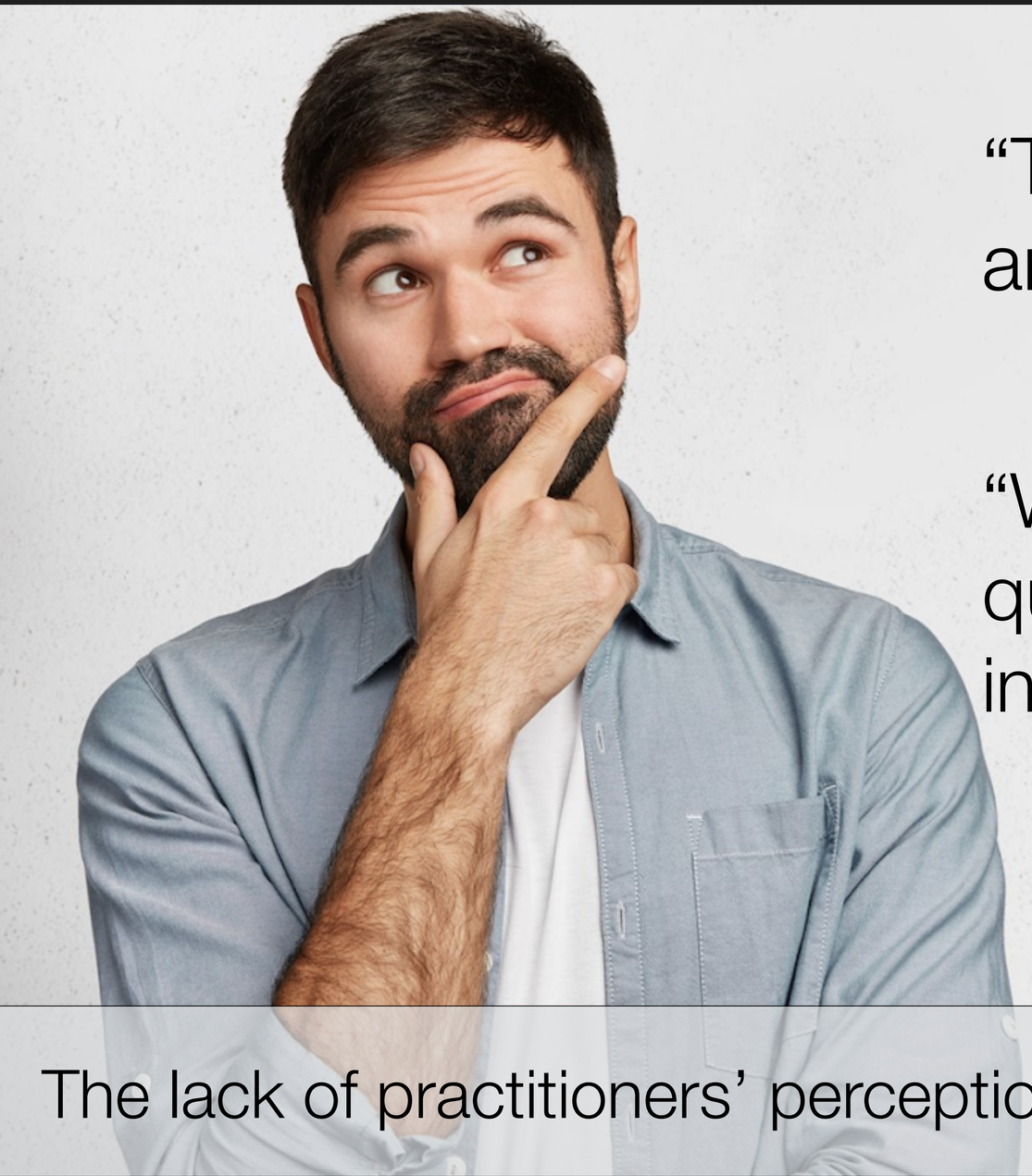
The main empirical studies carried out focused on challenges in using, debugging, testing, and architecture-related issues.

Quantum software engineering: Insights from Practitioners



Most of the challenges addressed in literature neglected the practitioners’ perception.

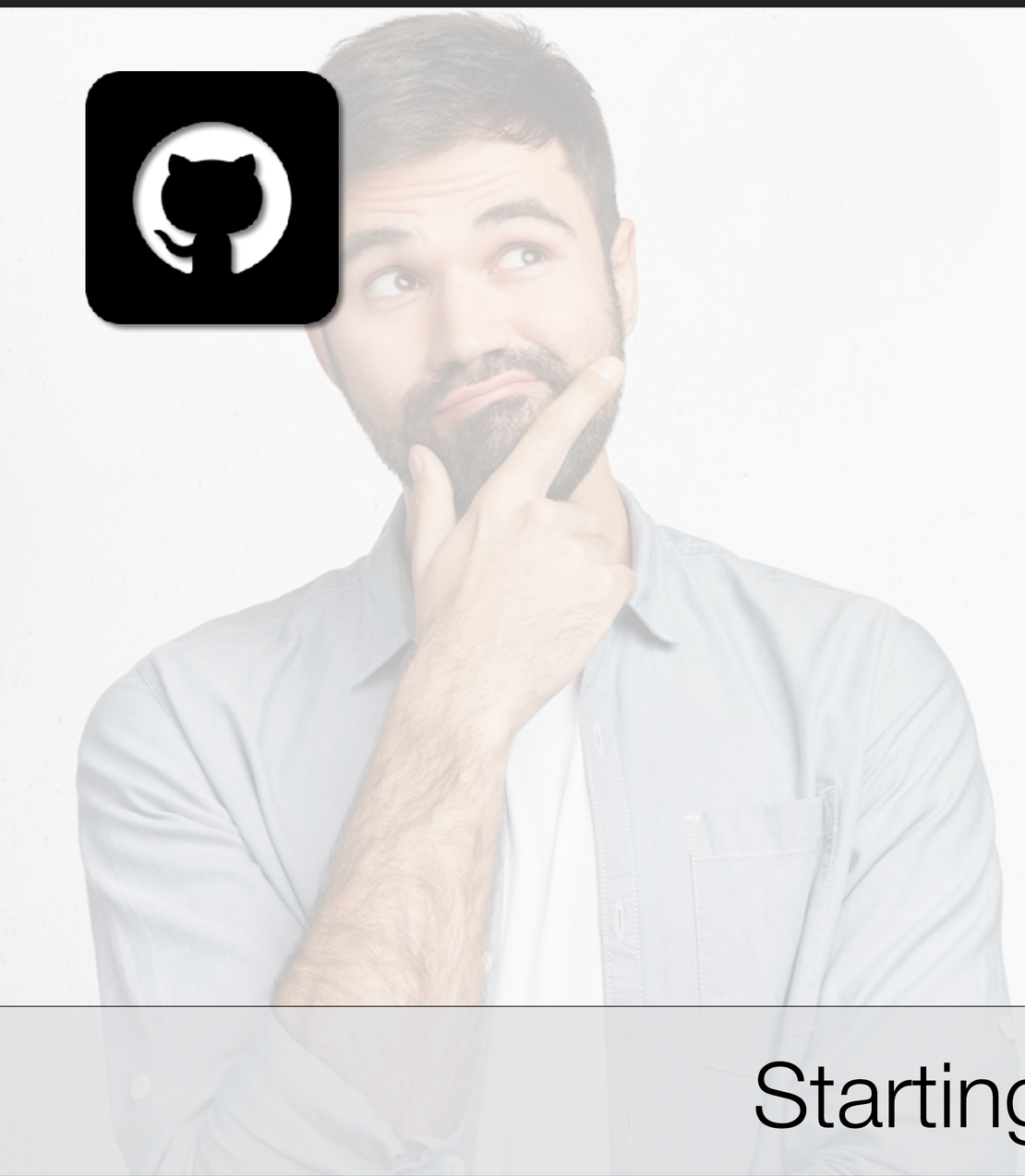
Quantum software engineering: Insights from Practitioners



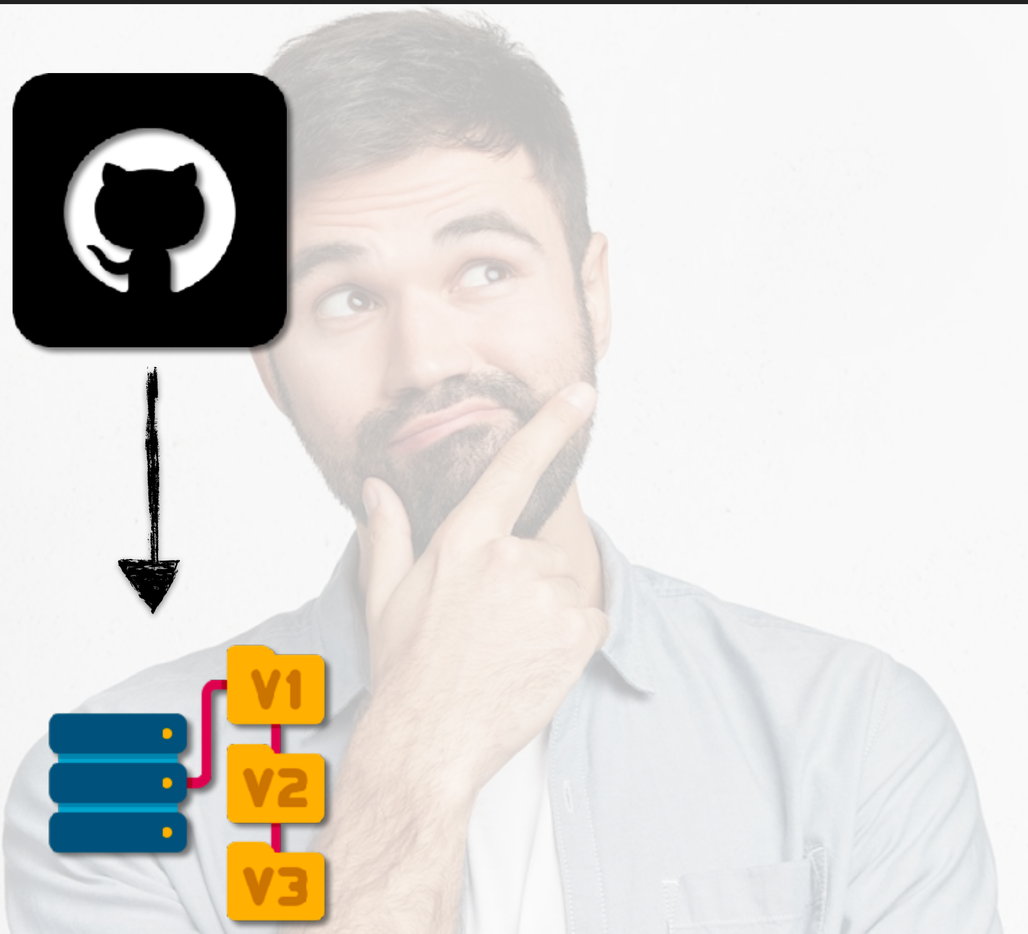
"To what extent and for what **purposes** are quantum frameworks used?"

"What are the main **challenges** that quantum developers experience when interacting with quantum frameworks?"

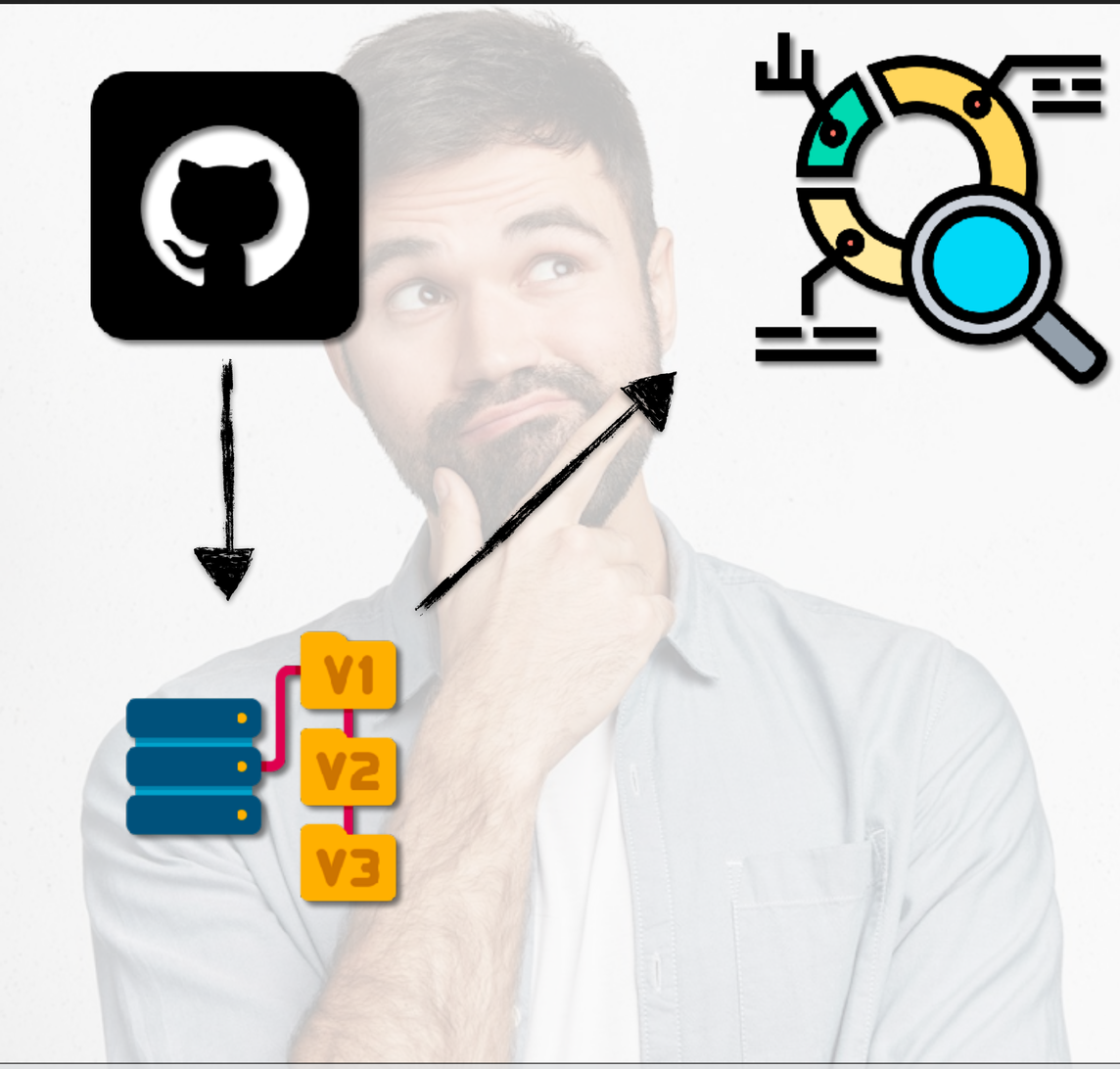
The lack of practitioners' perception motivated the first step of the research plan.



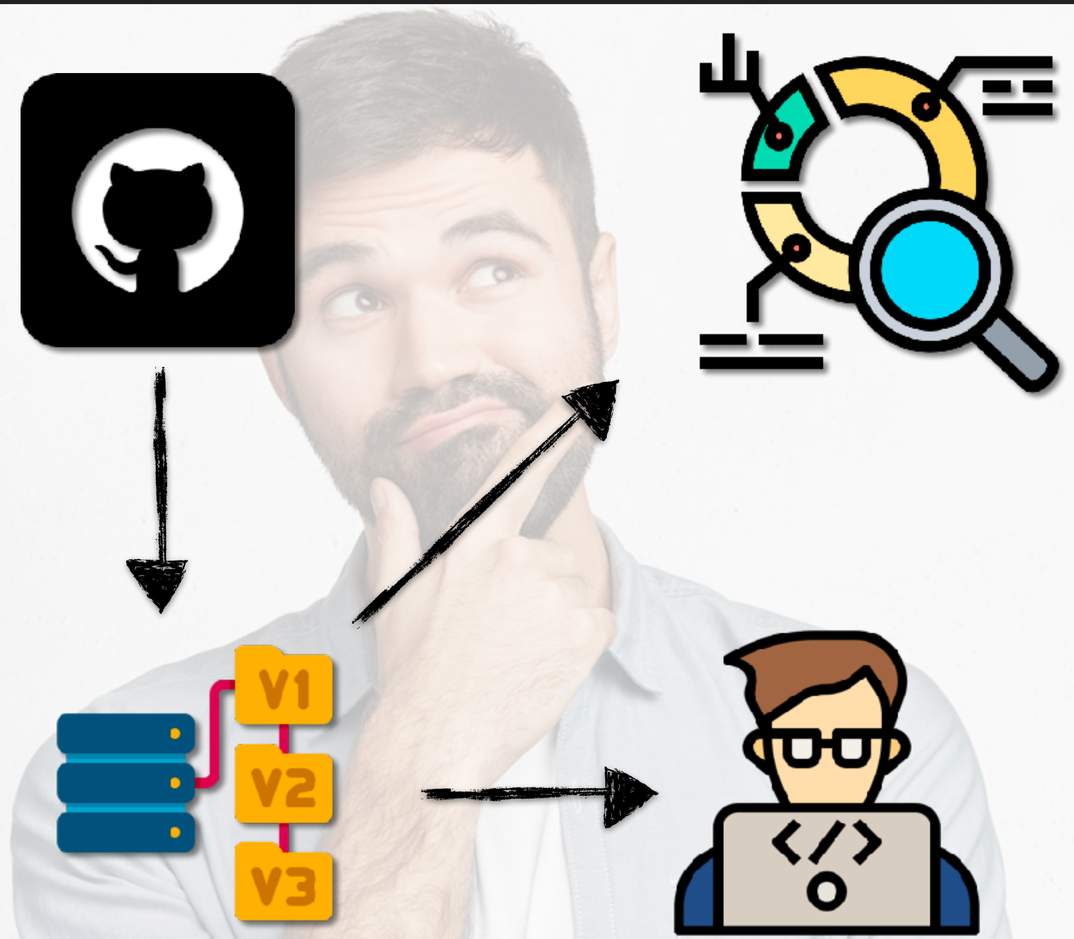
Starting from GitHub…



Repositories using Qiskit, Cirq, and Q# are selected.



Repositories are manually labeled to asses the purpose of their creation.



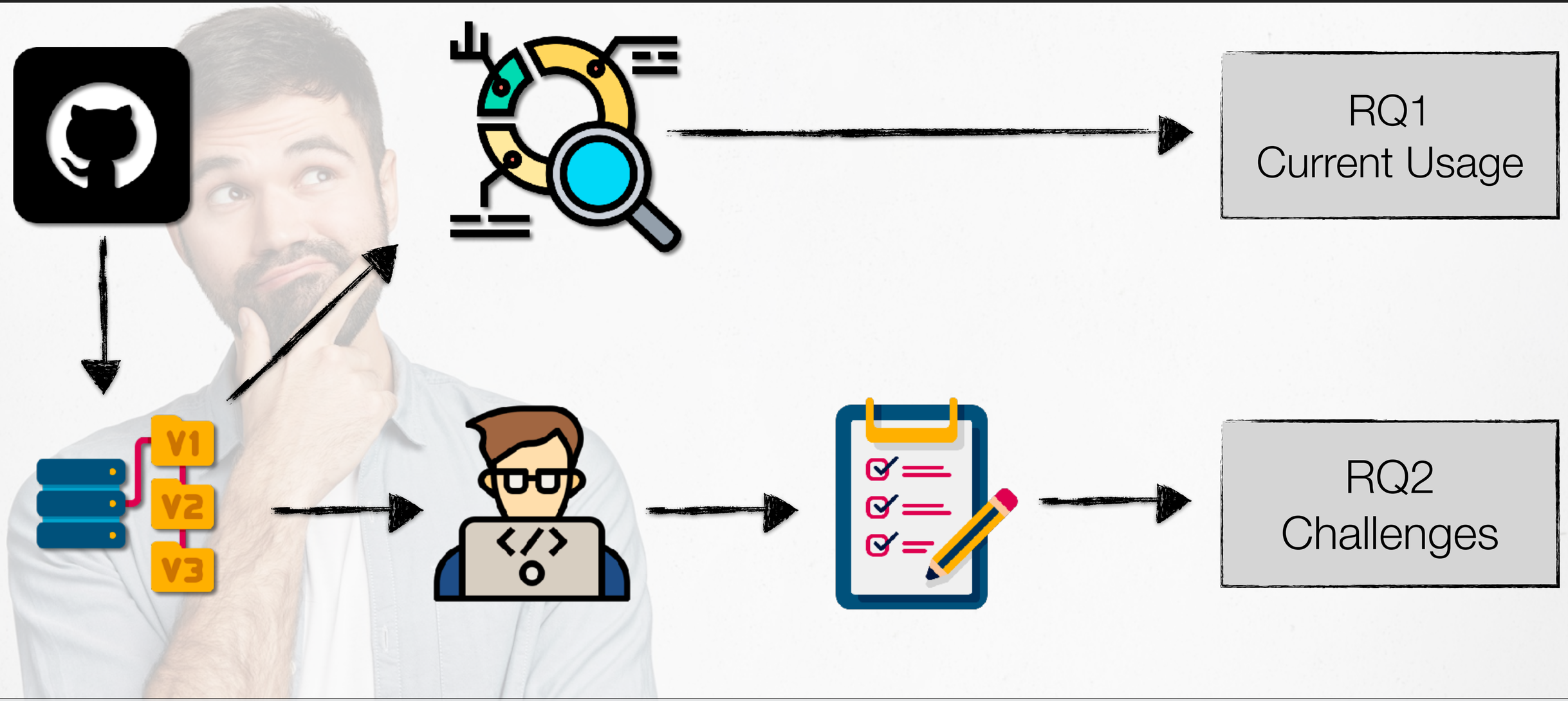
The list of contributors is extracted.

Practitioners are surveyed.

RQ1
Current Usage

The repository analysis and the second part of the survey answer RQ1.

Quantum software engineering: Insights from Practitioners



The third part of the survey answers RQ2.

Quantum software engineering: Insights from Practitioners

The Journal of Systems & Software 193 (2022) 111426

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Software engineering for quantum programming: How far are we?☆

Manuel De Stefano [a,*], Fabiano Pecorelli [b], Dario Di Nucci [a], Fabio Palomba [a], Andrea De Lucia [a]

[a] Sesa Lab - University of Salerno, Italy
[b] Tampere University, Finland

ARTICLE INFO

Article history:
Received 23 October 2021
Received in revised form 8 April 2022
Accepted 10 April 2022
Available online 20 April 2022

Keywords:
Quantum computing
Software engineering for quantum programming
Empirical software engineering

ABSTRACT

Quantum computing is no longer only a scientific interest but is rapidly becoming an industrially available technology that can potentially overcome the limits of classical computation. Over the last years, all major companies have profiled frameworks and programming languages that allow developers to create their quantum applications. This shift has led to the definition of a new discipline called *quantum software engineering*, which is demanded to define novel methods for engineering large-scale quantum applications. While the research community is successfully embracing this call, we notice a lack of systematic investigations into the state of the practice of quantum programming. Understanding the challenges that quantum developers face is vital to precisely define the aims of quantum software engineering. Hence, in this paper, we first mine all the GitHub repositories that make use of the most used quantum programming frameworks currently on the market and then conduct coding analysis sessions to produce a taxonomy of the purposes for which quantum technologies are used. In the second place, we conduct a survey study that involves the contributors of the considered repositories, which aims to elicit the developers' opinions on the current adoption and challenges of quantum programming. On the one hand, the results highlight that the current adoption of quantum programming is still limited. On the other hand, there are many challenges that the software engineering community should carefully consider; these do not strictly pertain to technical concerns but also socio-technical matters.

1. Introduction

The dream has come true (Knight, 2018): several physicists and computer scientists agree that the quantum technology is right around the corner (Knight, 2018; Hoare and Milner, 2005) and that the 21st century will be recalled as the "*quantum era*" (Piattini et al., 2021). Specific mechanic principles such as *superposition*, i.e., quantum objects may assume different states at the same time, and *entanglement*, i.e., quantum objects may be deeply connected without any direct physical interaction, promise to revolutionize program computation compared to classical computers (Nielsen, 2017). Quantum computers could eventually lead to resolving NP complete problems (Aaronson, 2005; Ghysa and Volovich, 2008)—often referred to as *quantum supremacy* (Arute et al., 2019), namely the point in time when a programmable quantum device would be able to solve problems that no classical computer can solve in any feasible amount of time.

For this reason, all major software companies, like IBM and Google, are currently investing hundreds of millions of dollars every year to produce novel hardware and software technologies that can support the execution of quantum programs.[1] For instance, IBM Quantum[2] has developed its programming framework, which allows developers to design, implement, and execute quantum applications on cloud-based quantum computers. Companies and researchers have also been developing several quantum programming languages (Omer, 2007; Q, 2021; Altenkirch and Grattage, 2005) and development toolkits (Aleksandrowicz et al., 2019; Broughton et al., 2020; Steiger et al., 2018) that provide developers with off-the-shelf instruments and APIs to create quantum programs.

While there have been already several promising applications of quantum programming to the resolution of various problems in the fields of machine learning (Diamonte et al., 2017), optimization (Gueneshi and Smelyanskiy, 2017), cryptography (Mailloux et al., 2016), and chemistry (Reiher et al., 2017), the development of large-scale quantum software seems to be still far from being

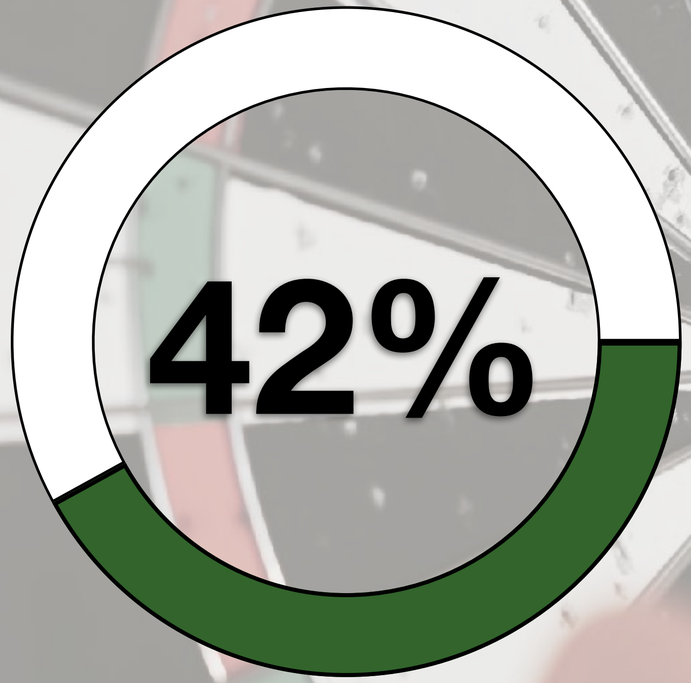
☆ Editor: Alexander Chatzigeorgiou.

* Corresponding author.
E-mail addresses: mdestefano@unisa.it (M. De Stefano), fabiano.pecorelli@tuni.fi (F. Pecorelli), ddinucci@unisa.it (D. Di Nucci), fpalomba@unisa.it (F. Palomba), adelucia@unisa.it (A. De Lucia).

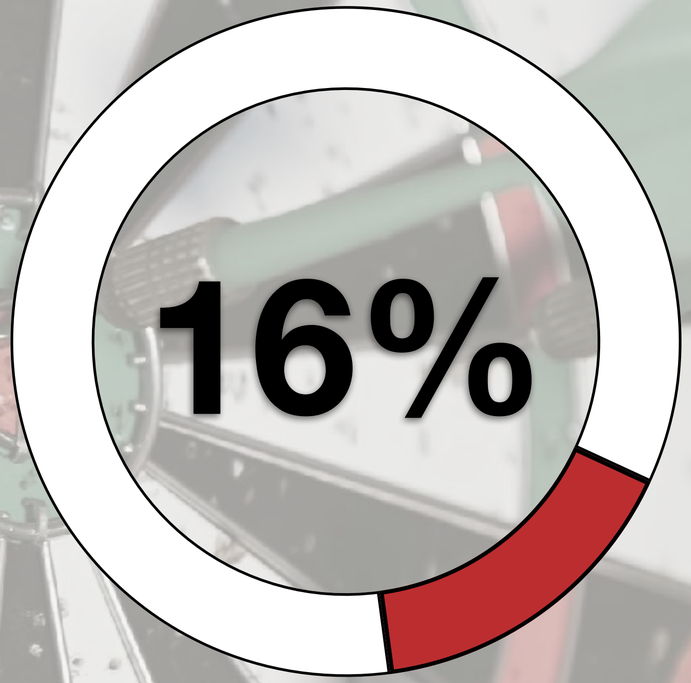
https://doi.org/10.1016/j.jss.2022.111426
0164-1212/

[1] Boston Consulting Group report: .
[2] IBM Quantum: https://www.ibm.com/quantum-computing.

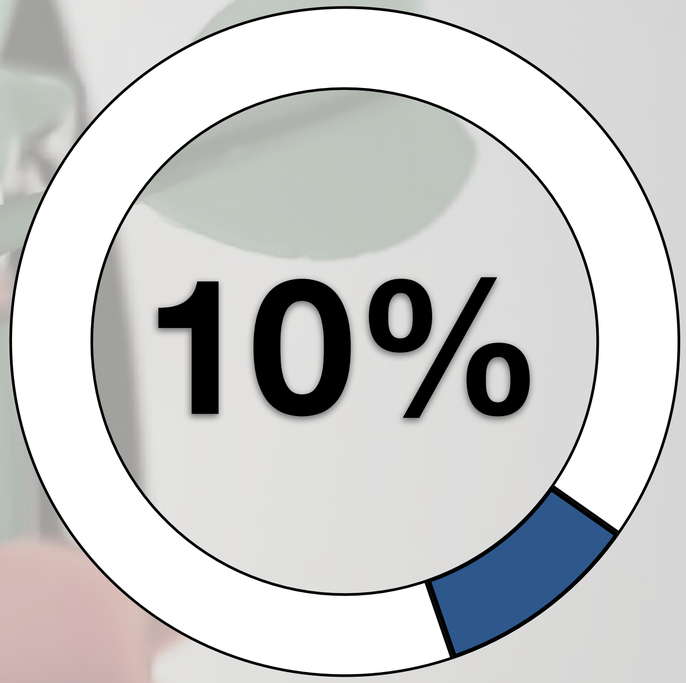
These results ended in our recent JSS paper



Toy projects



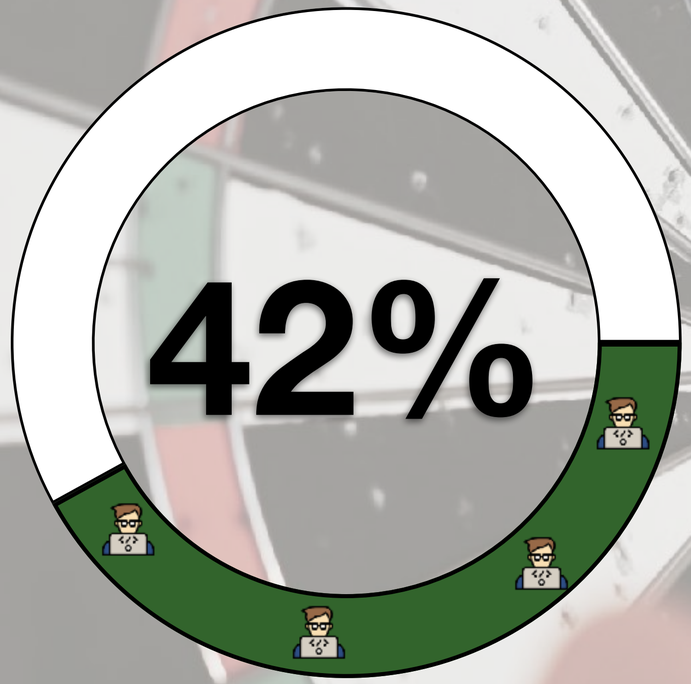
Libraries &
Frameworks



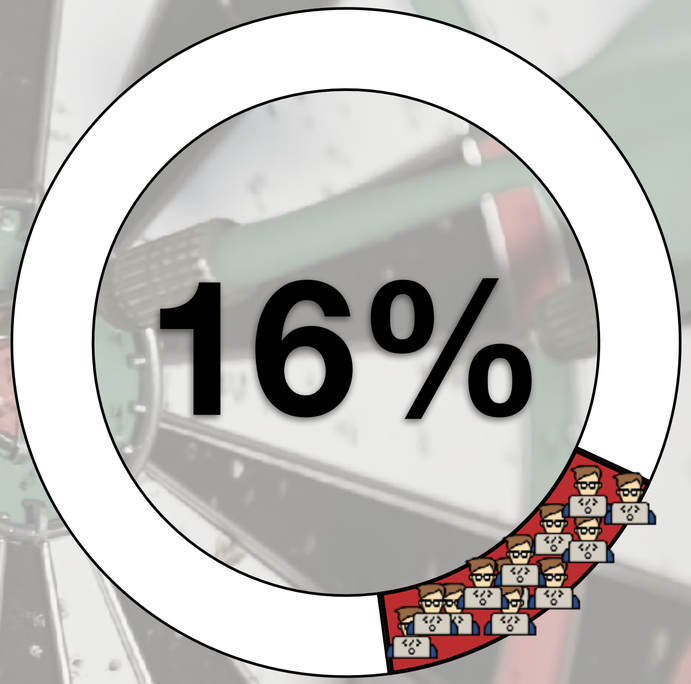
Research

Top three repository usage.

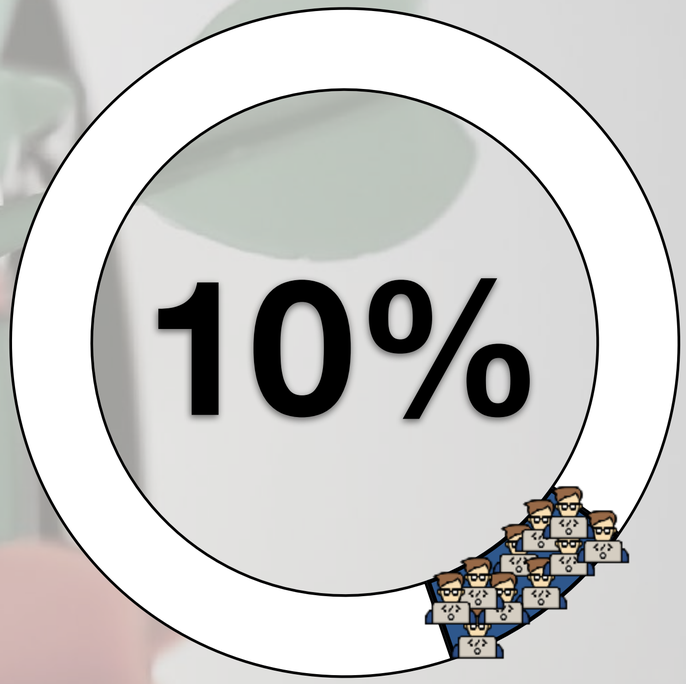
Quantum software engineering: Insights from Practitioners



Toy projects

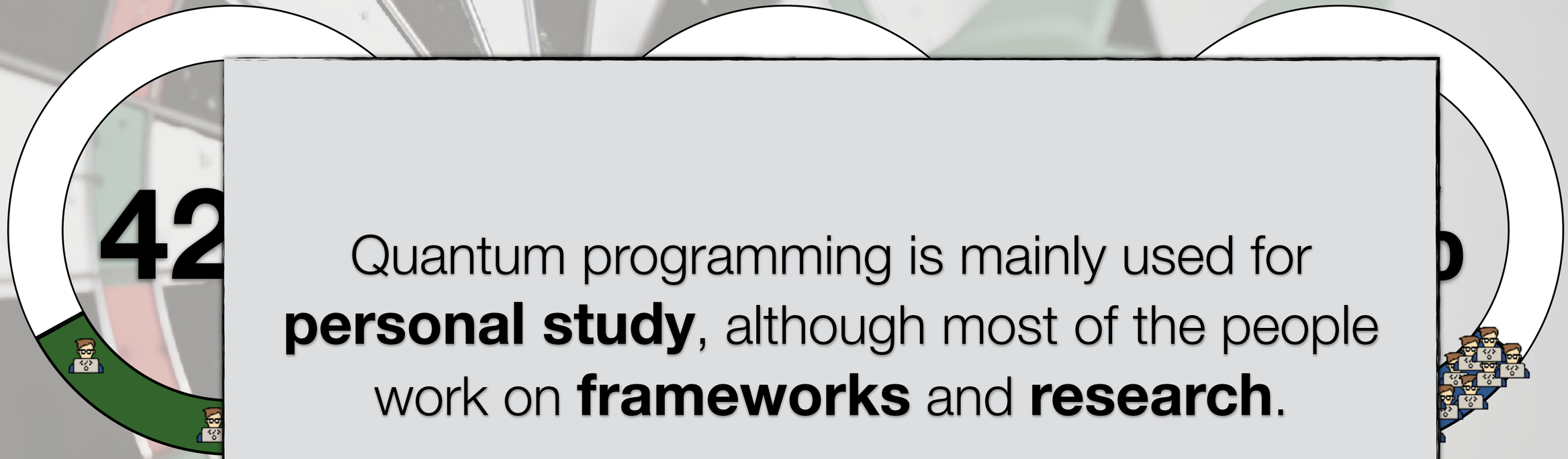


Libraries & Frameworks

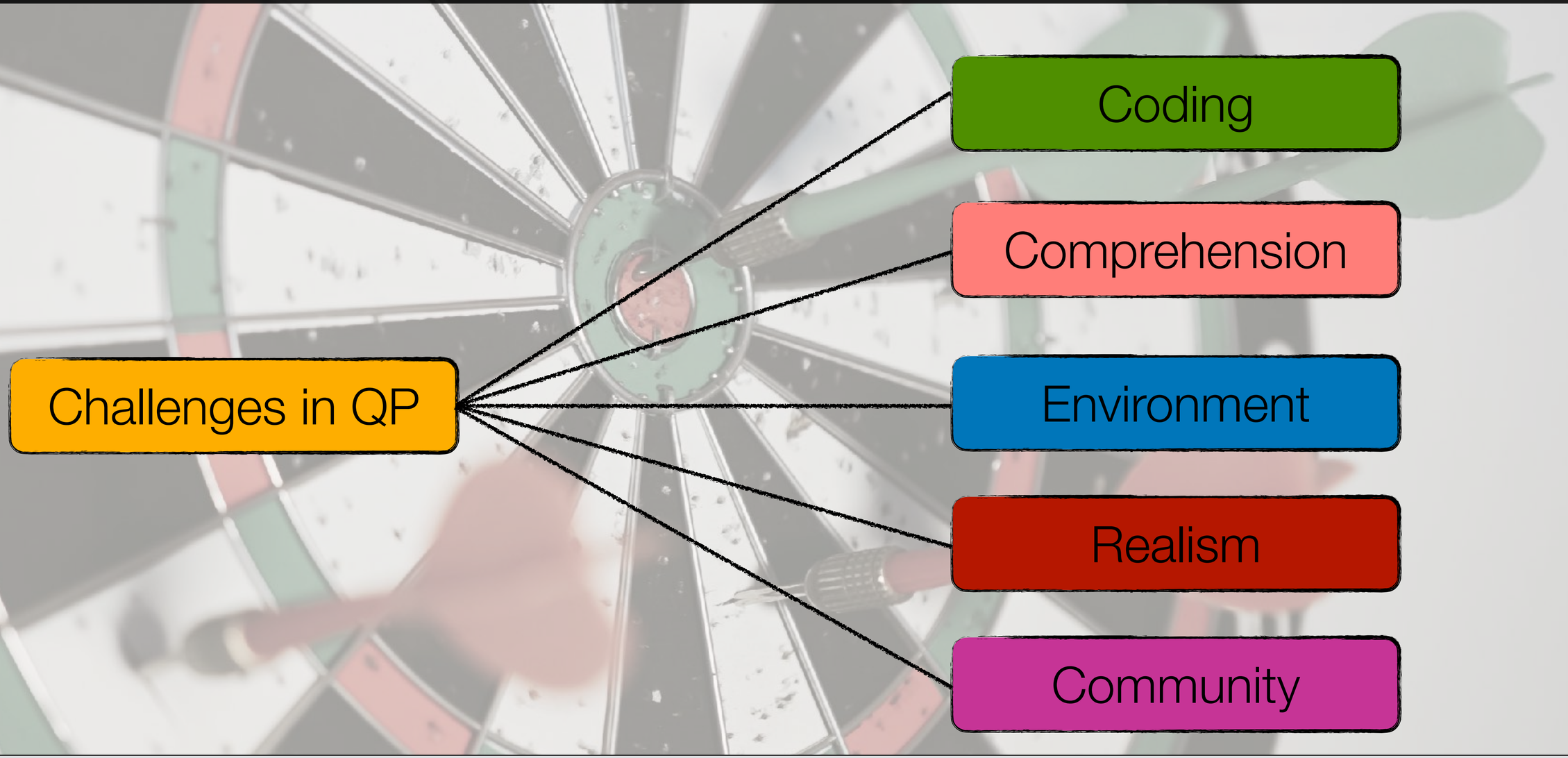


Research

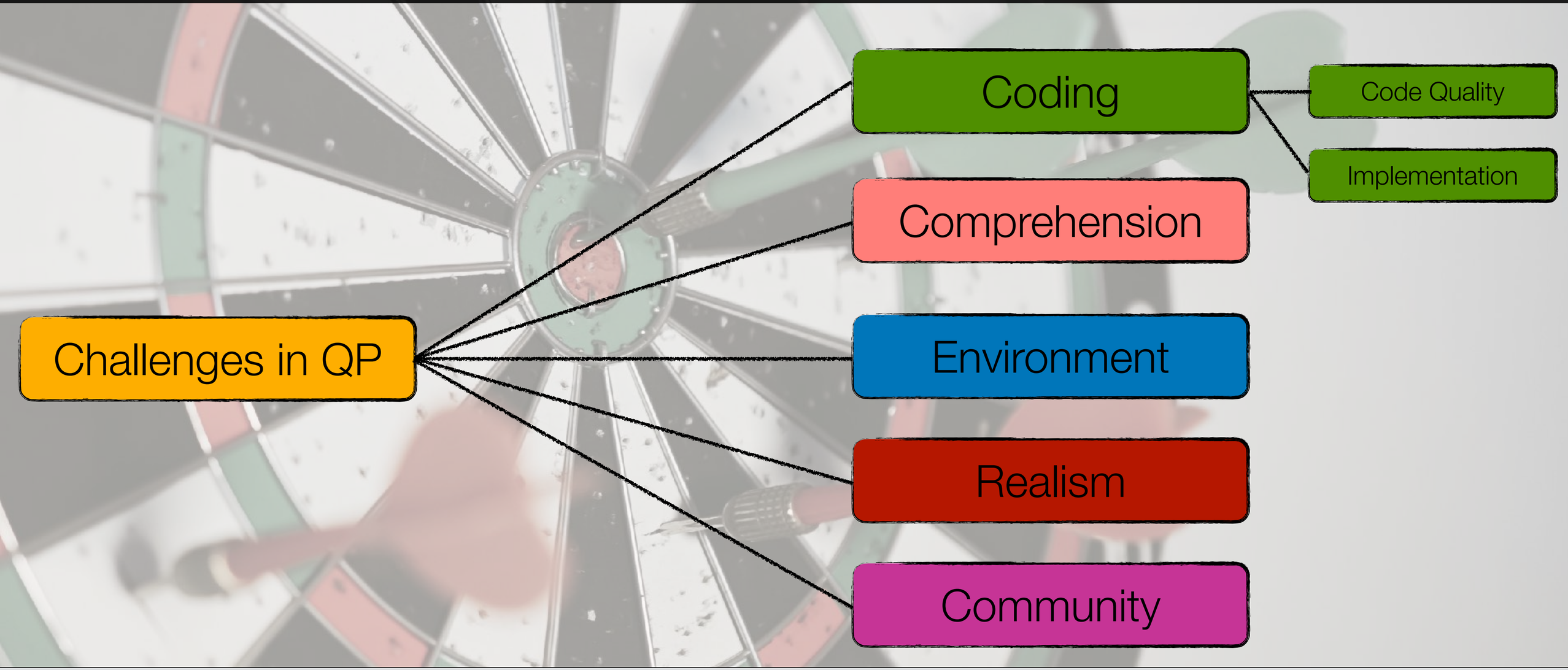
Contributors are distributed much differently.



Contributors are distributed much differently.



Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.

Challenges in C

Coding

Code Quality

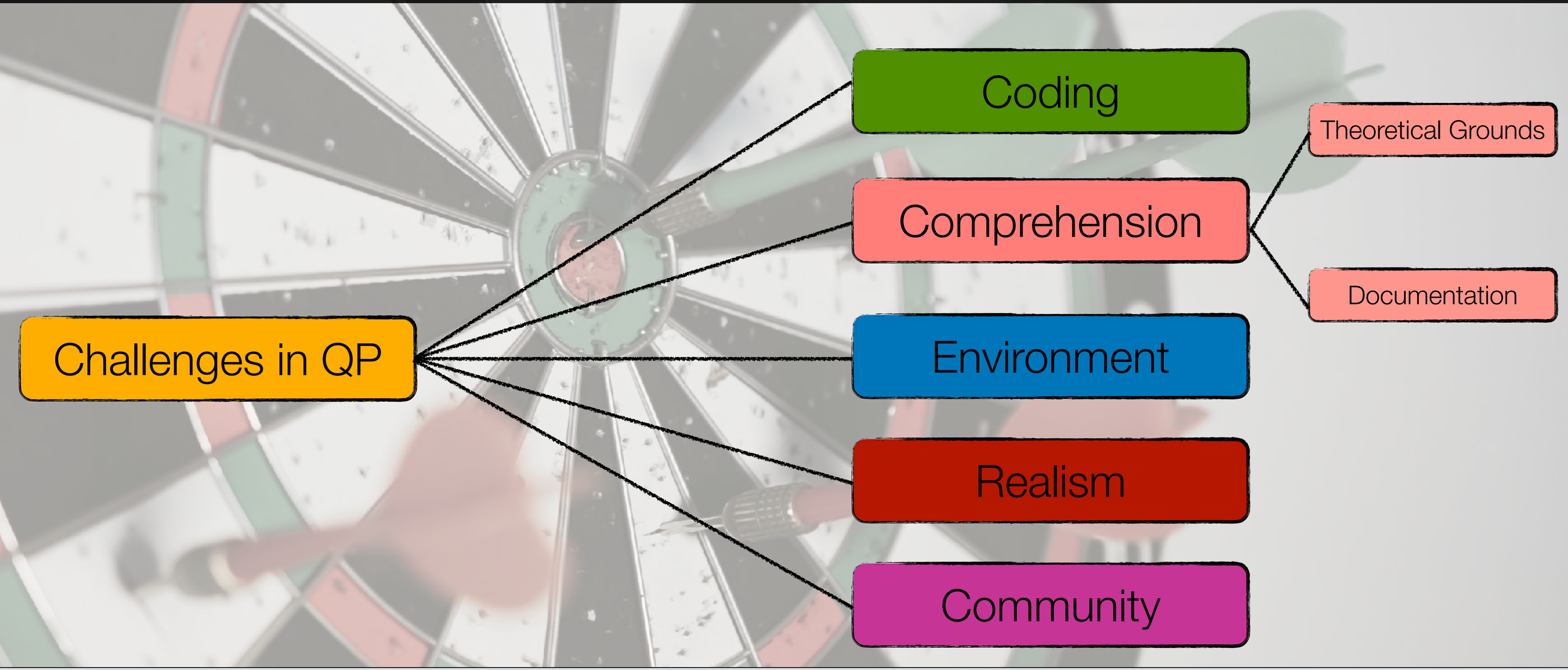
Implementation

“Being able to check that the circuit does what you want it to do”

“The little understanding of what should the result look like”

“Creating readable code ”

Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.

Challenges in

Comprehension

"Many concepts are needed to get acquainted with quantum programming, particularly linear algebra"

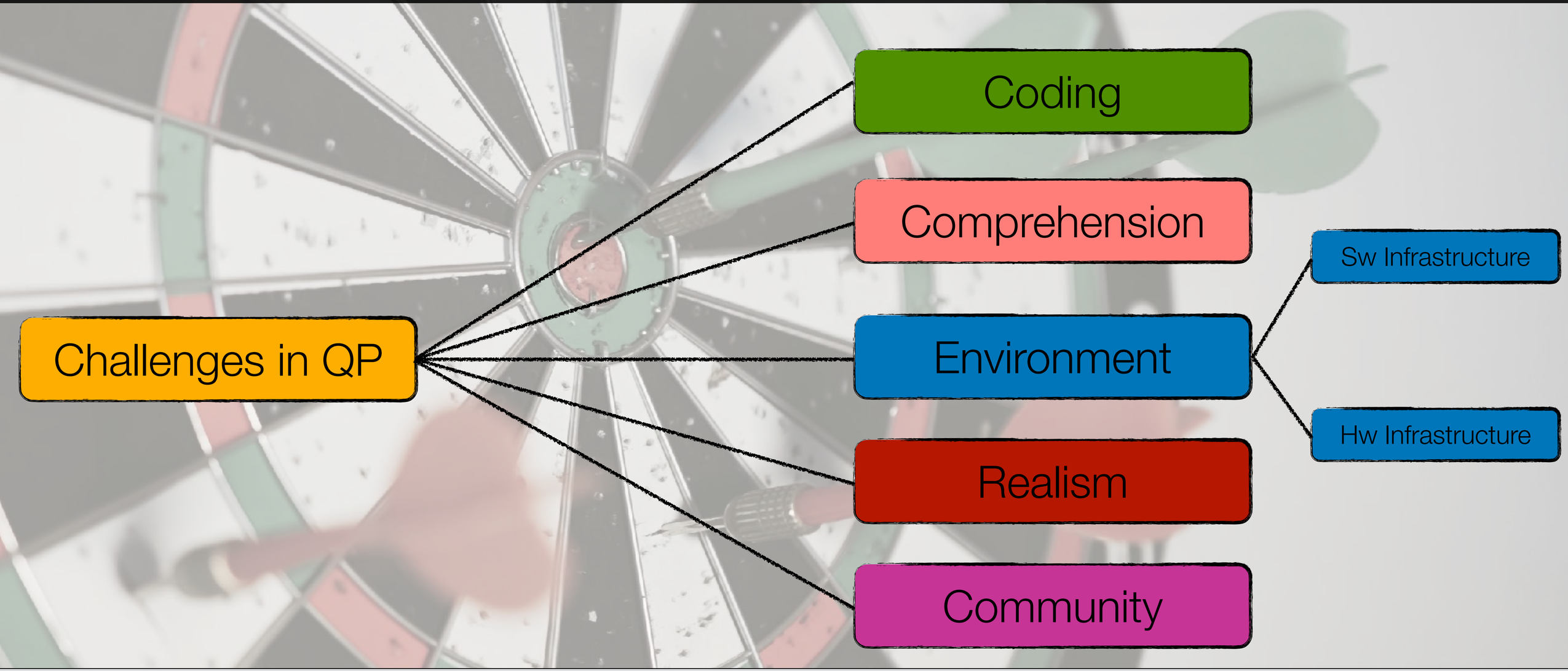
"Documentation out of date"

"Missing documentation"

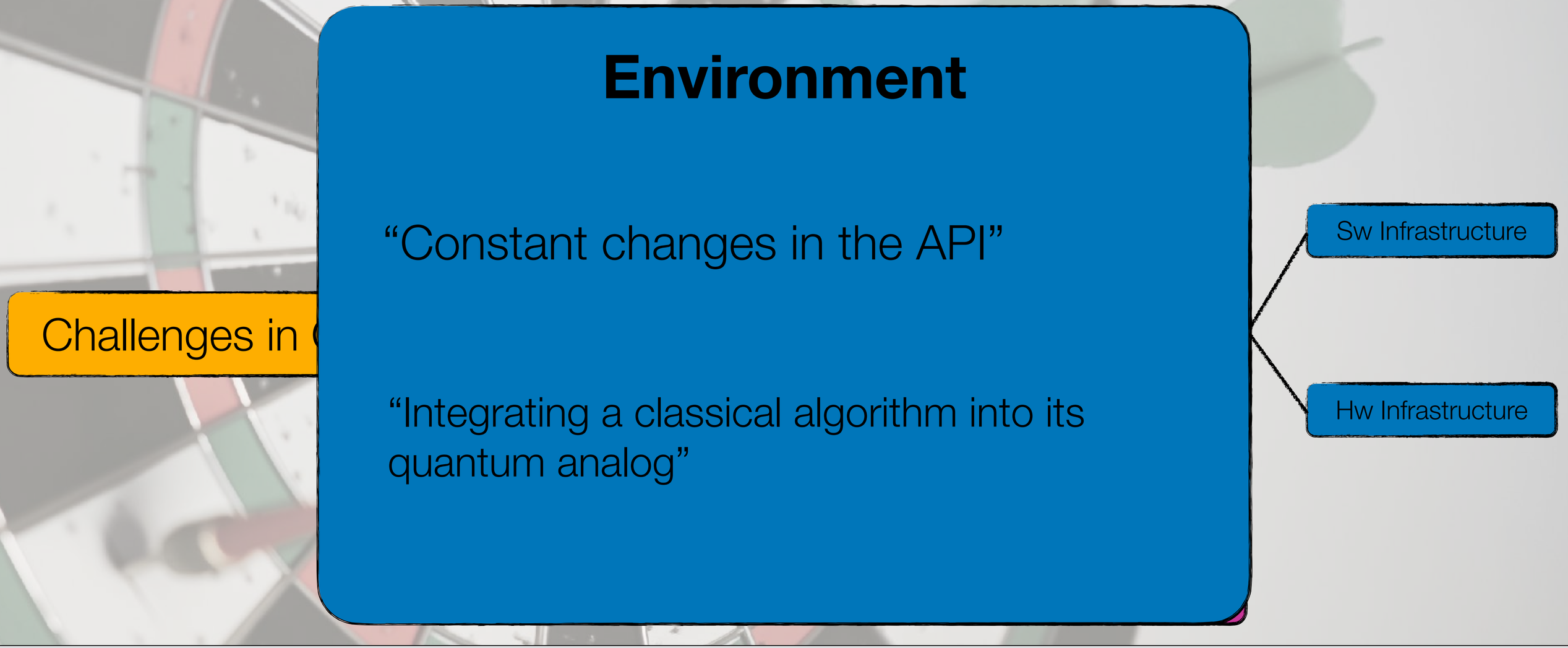
Theoretical Grounds

Documentation

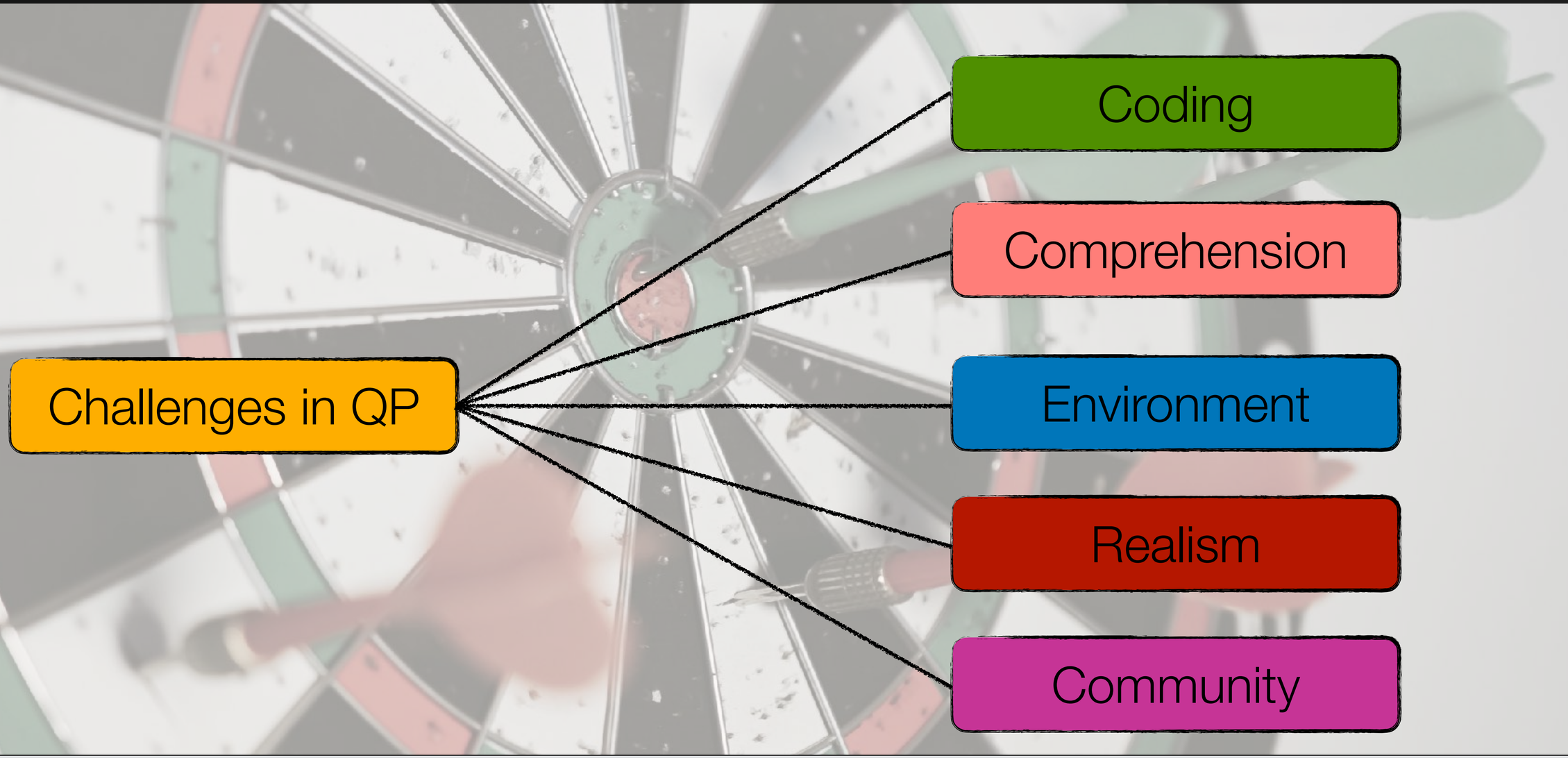
Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.

Challenges in

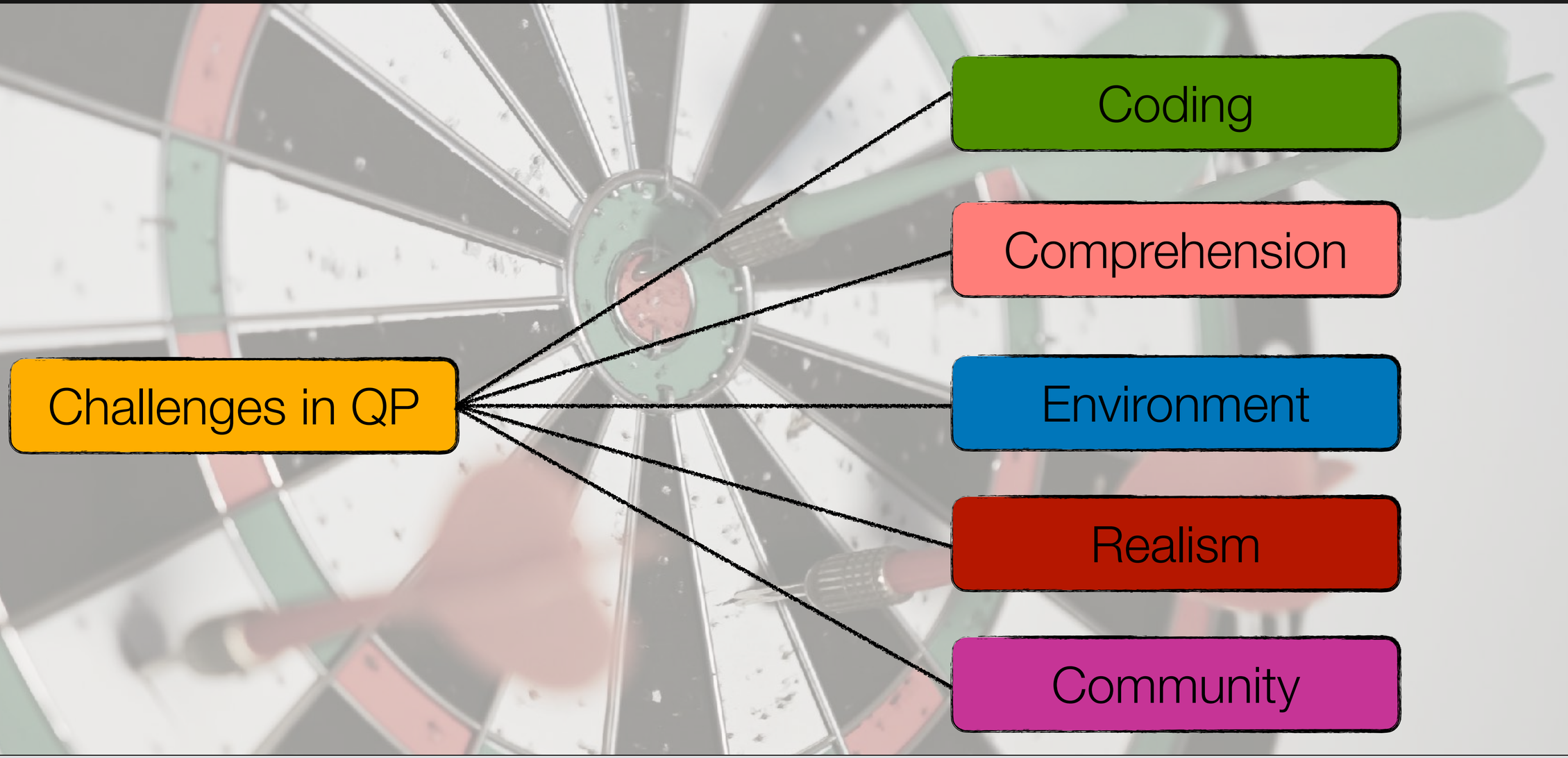
Realism

“Finding interesting use cases”

“Using QP on real products”

“Formulate a problem”

Taxonomy of challenges in QP perceived by developers.



Taxonomy of challenges in QP perceived by developers.

Challenges in Q

Community

“lack of professional connections; lack of peer guidance”

“learn to review the source code to understand how to use some features”

Taxonomy of challenges in QP perceived by developers.

The state of the art of Quantum Software Engineering

To understand the state of the art from an academic perspective,
we conducted a ***Systematic Mapping Study*** (under non-double-blind review).

RQ 1

What are the current research trends in Quantum Software Engineering?

RQ 2

What main results are reported, and which quantum computing tools/ frameworks are most being studied?

RQ 3

What is the historical evolution of QSE, including pre-discipline contributions and changes in the research community's interest over time?

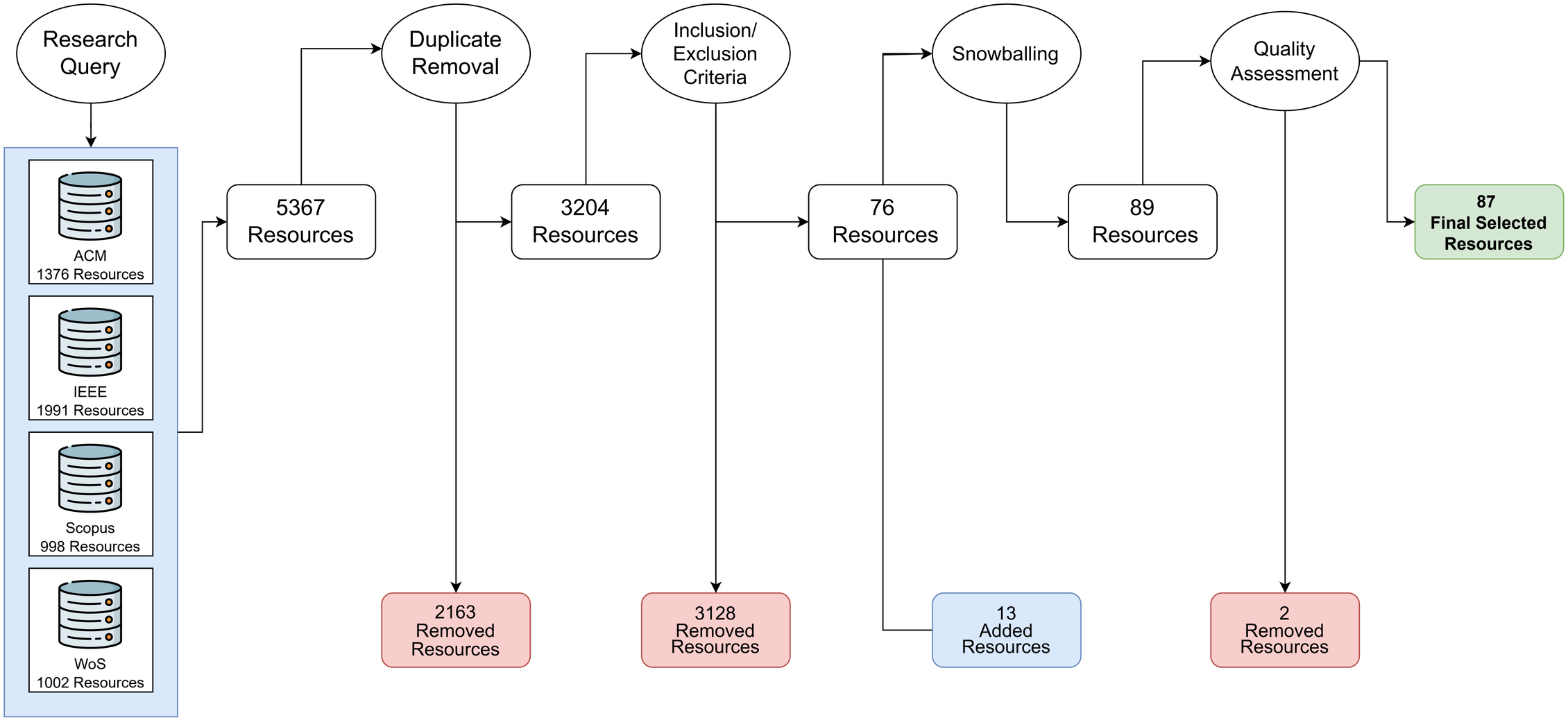
RQ 4

What is the current state of the authorship and collaboration network in QSE and how are the researchers distributed across various software engineering topics?

RQ 5

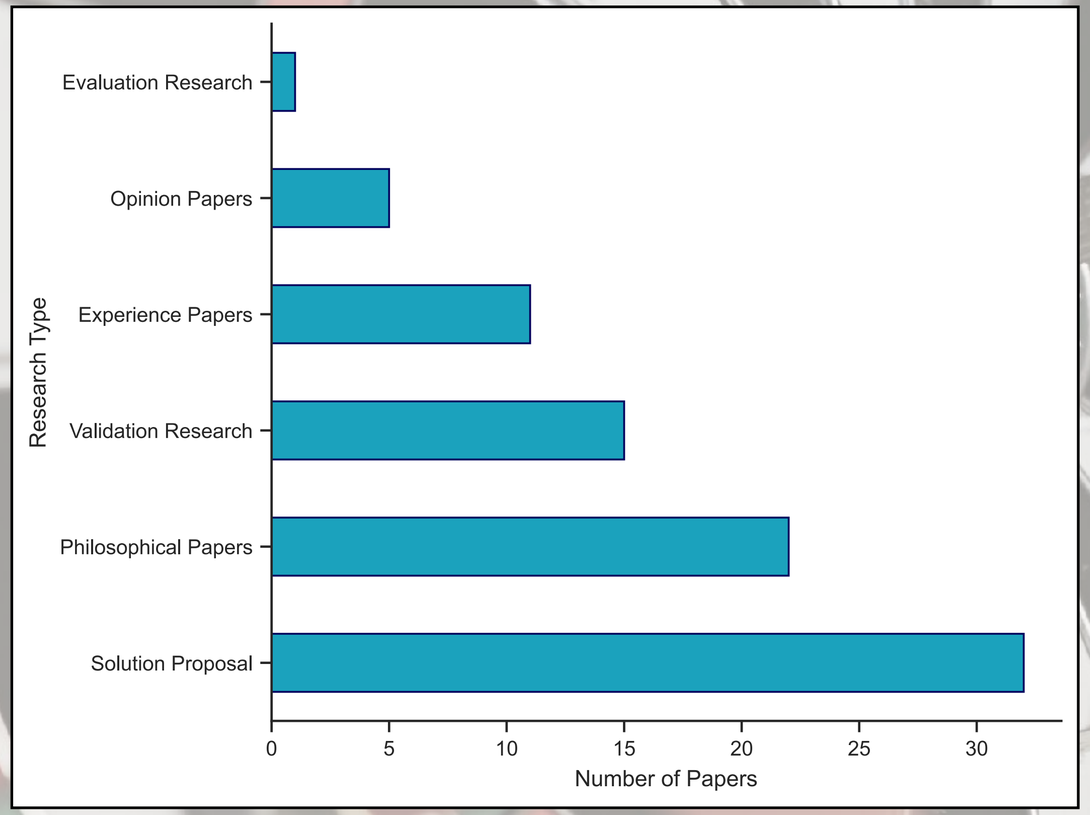
Which venues will most likely publish QSE articles outside thematic venues?

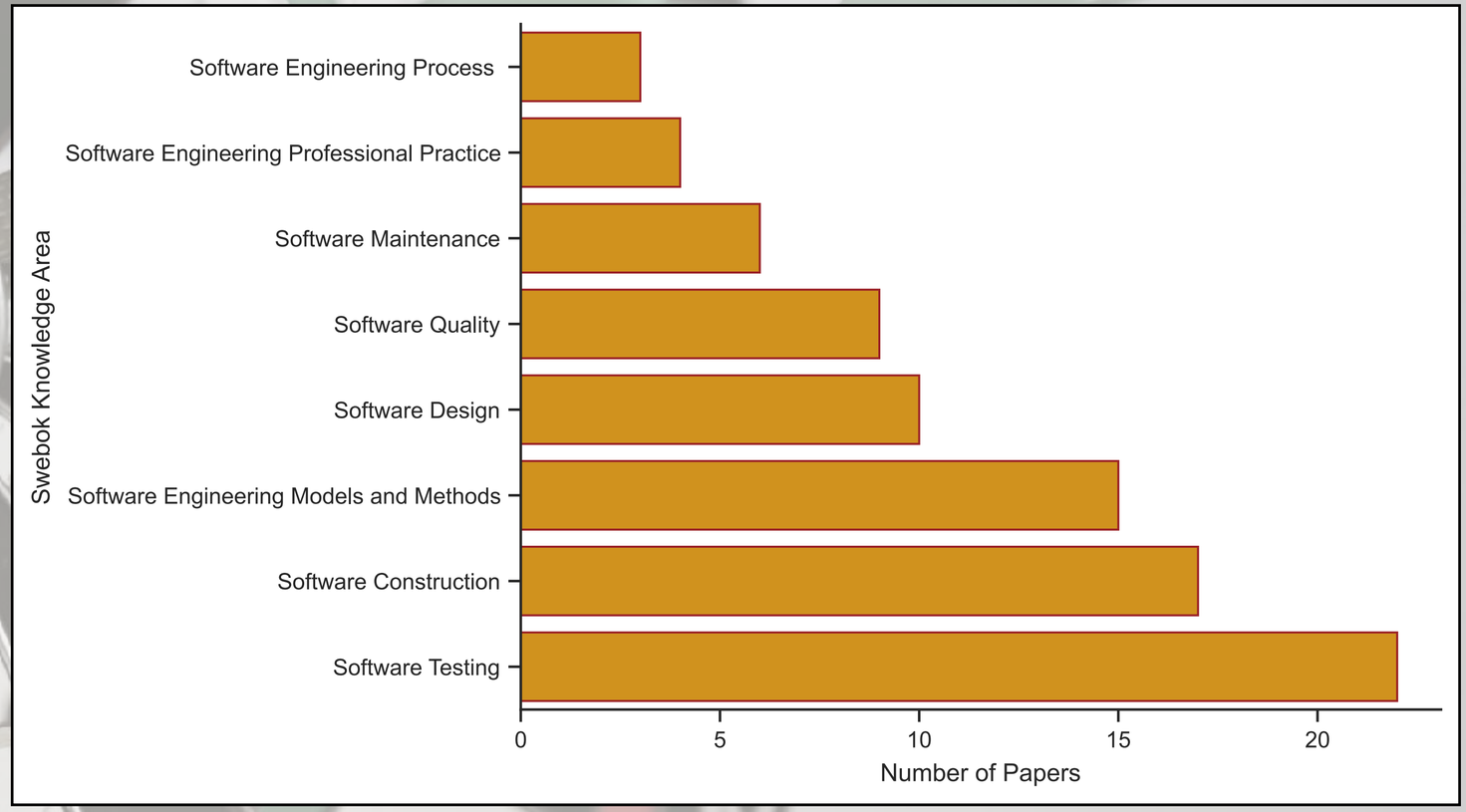
The state of the art of Quantum Software Engineering



Literature Review Process

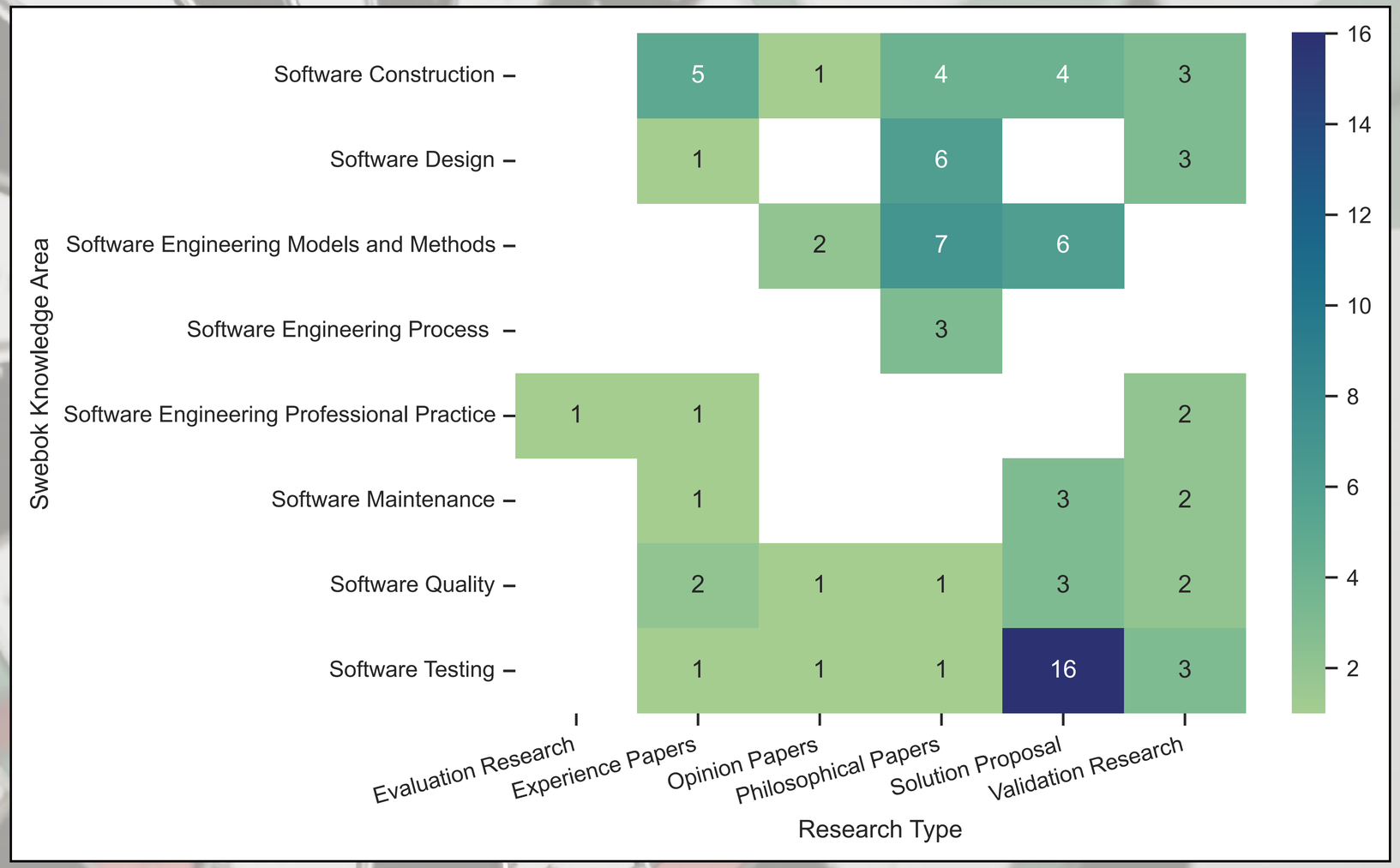
The state of the art of Quantum Software Engineering





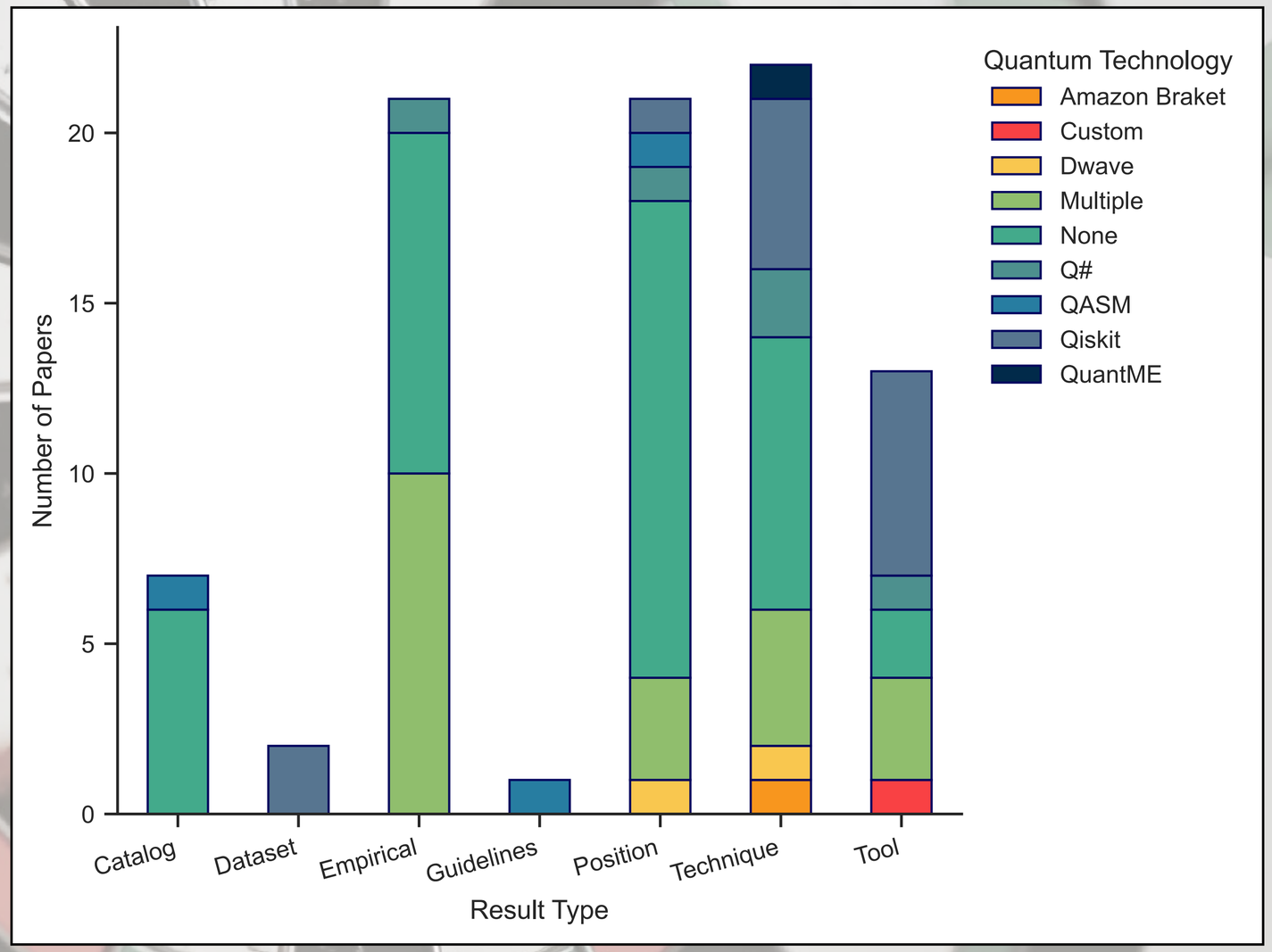
RQ1: Dominance of *Solution Proposal, Philosophical Papers* and *Software Testing,* neglecting *Empirical Studies* and *Software Engineering Process & Management*.

The state of the art of Quantum Software Engineering



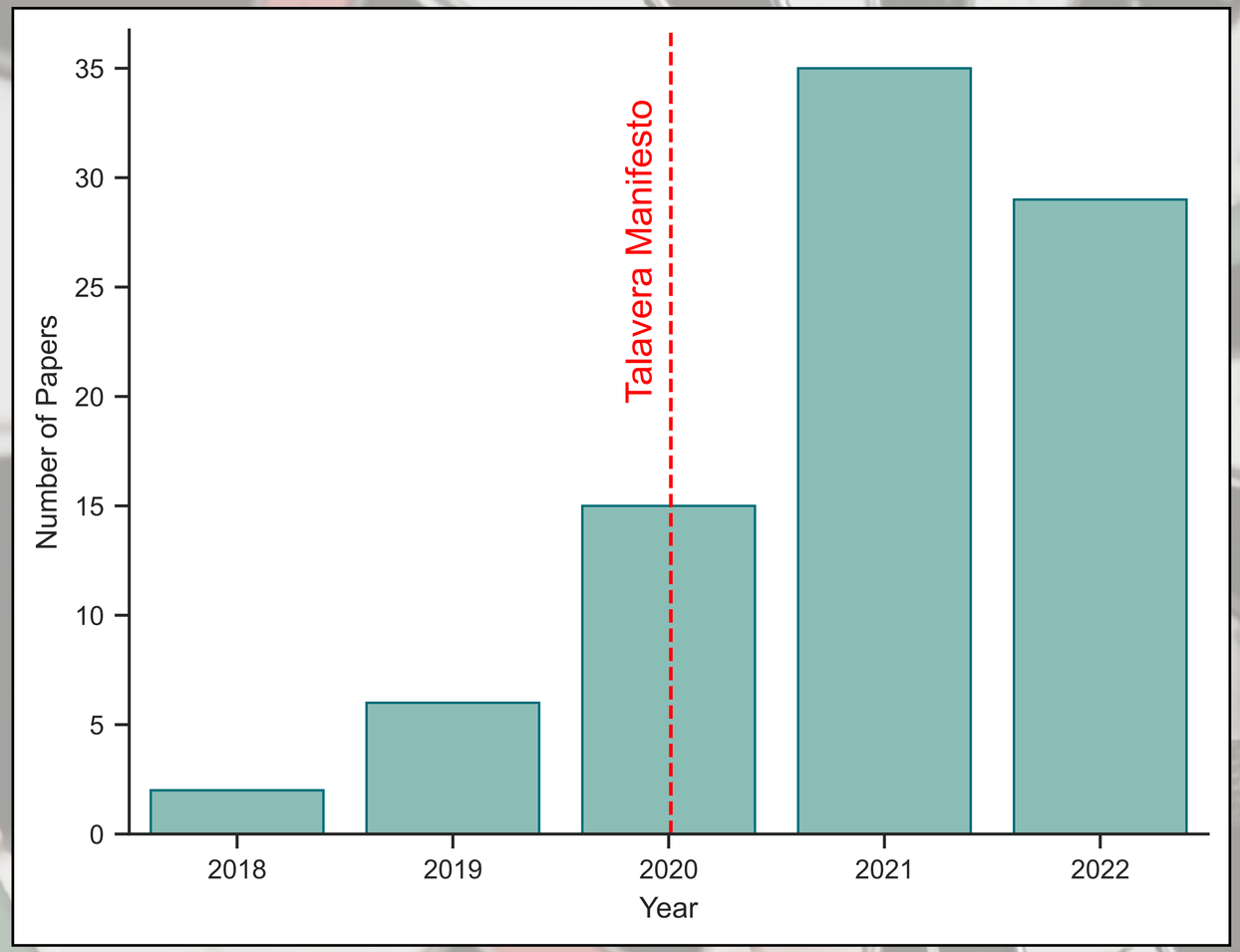
RQ1: Uneven concentration with abundance of *Solution Proposal* papers in *Software Testing* and relatively even distribution of *Validation Research* papers across most areas.

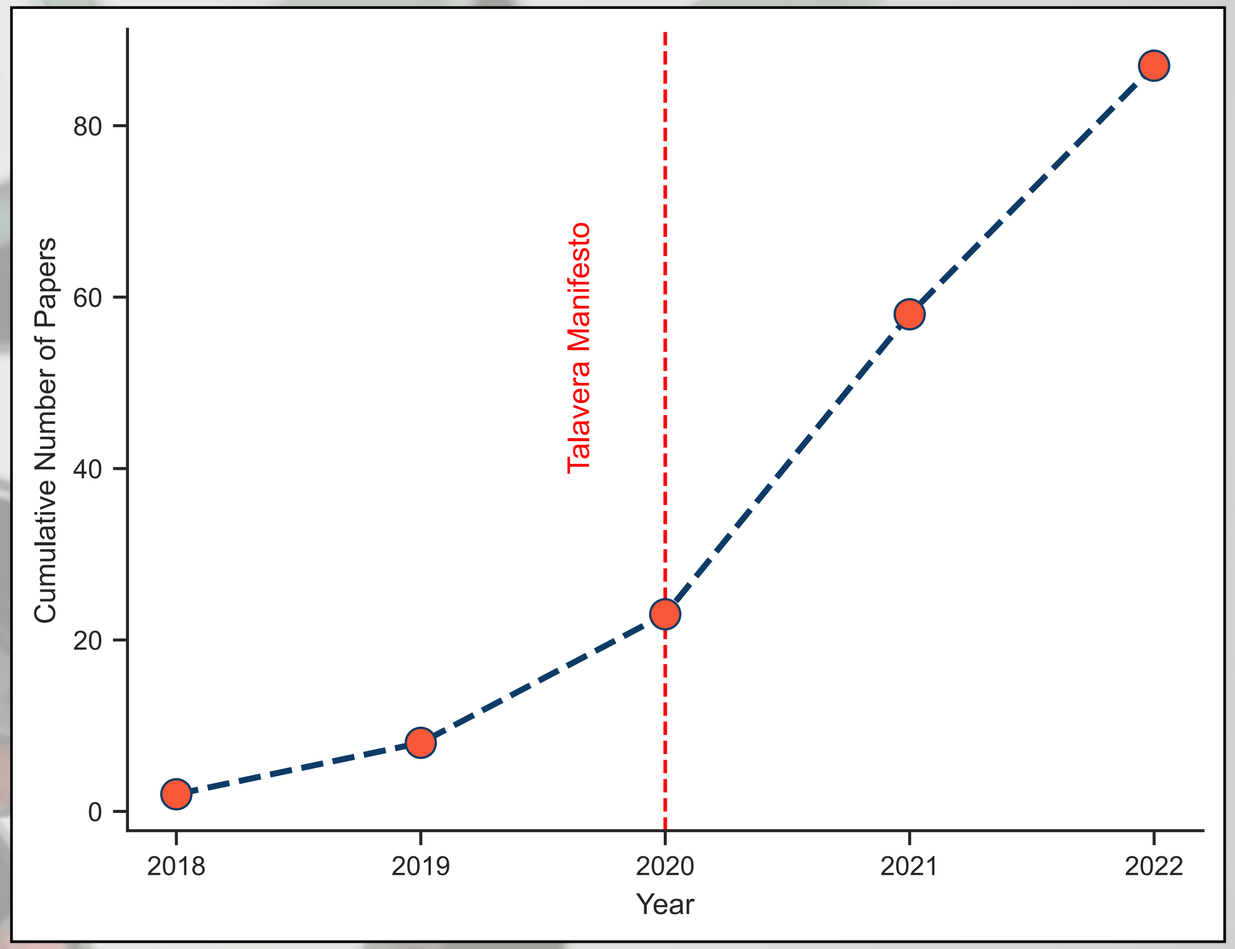
The state of the art of Quantum Software Engineering



RQ2: Most papers cover multiple technologies or none specific.
Qiskit is the most commonly studied technology.

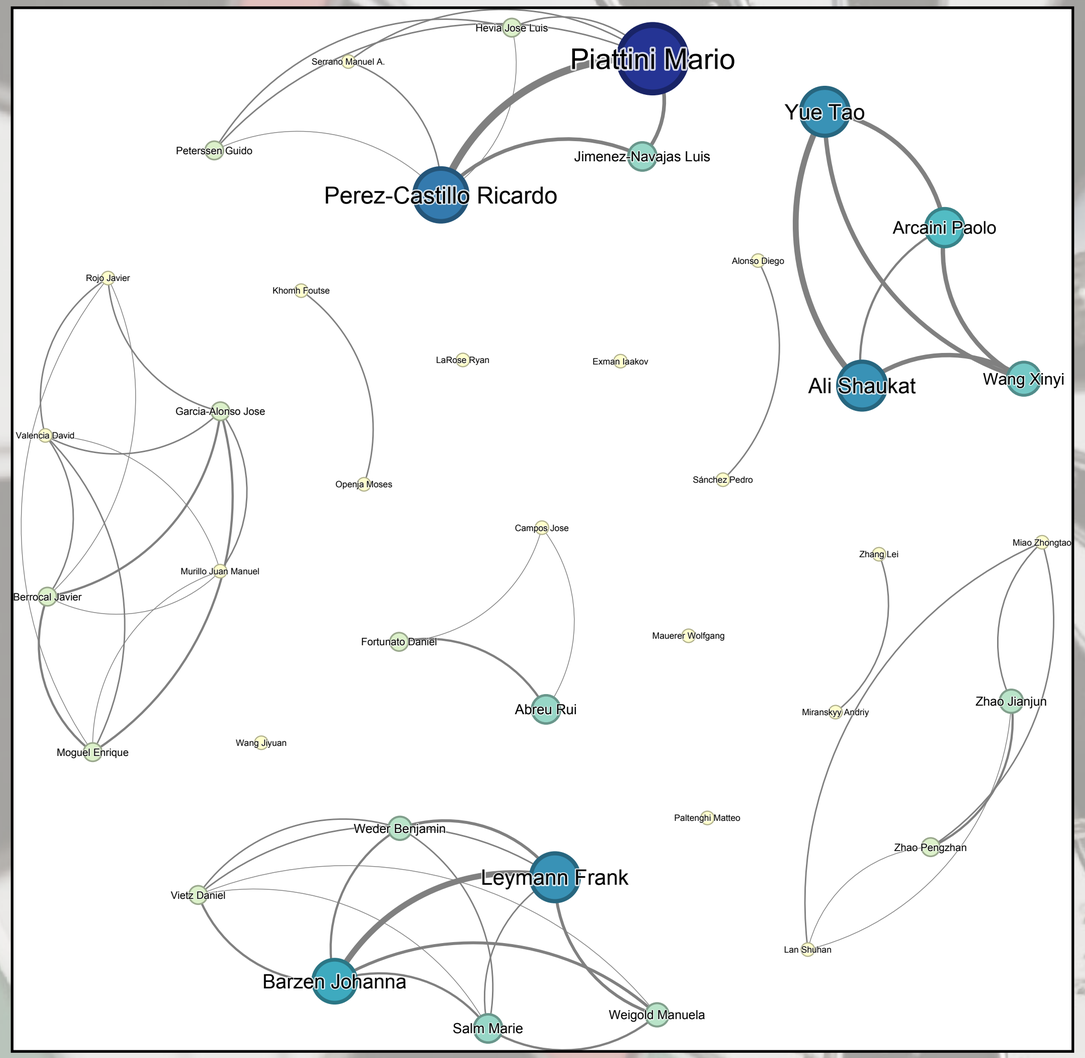
The state of the art of Quantum Software Engineering

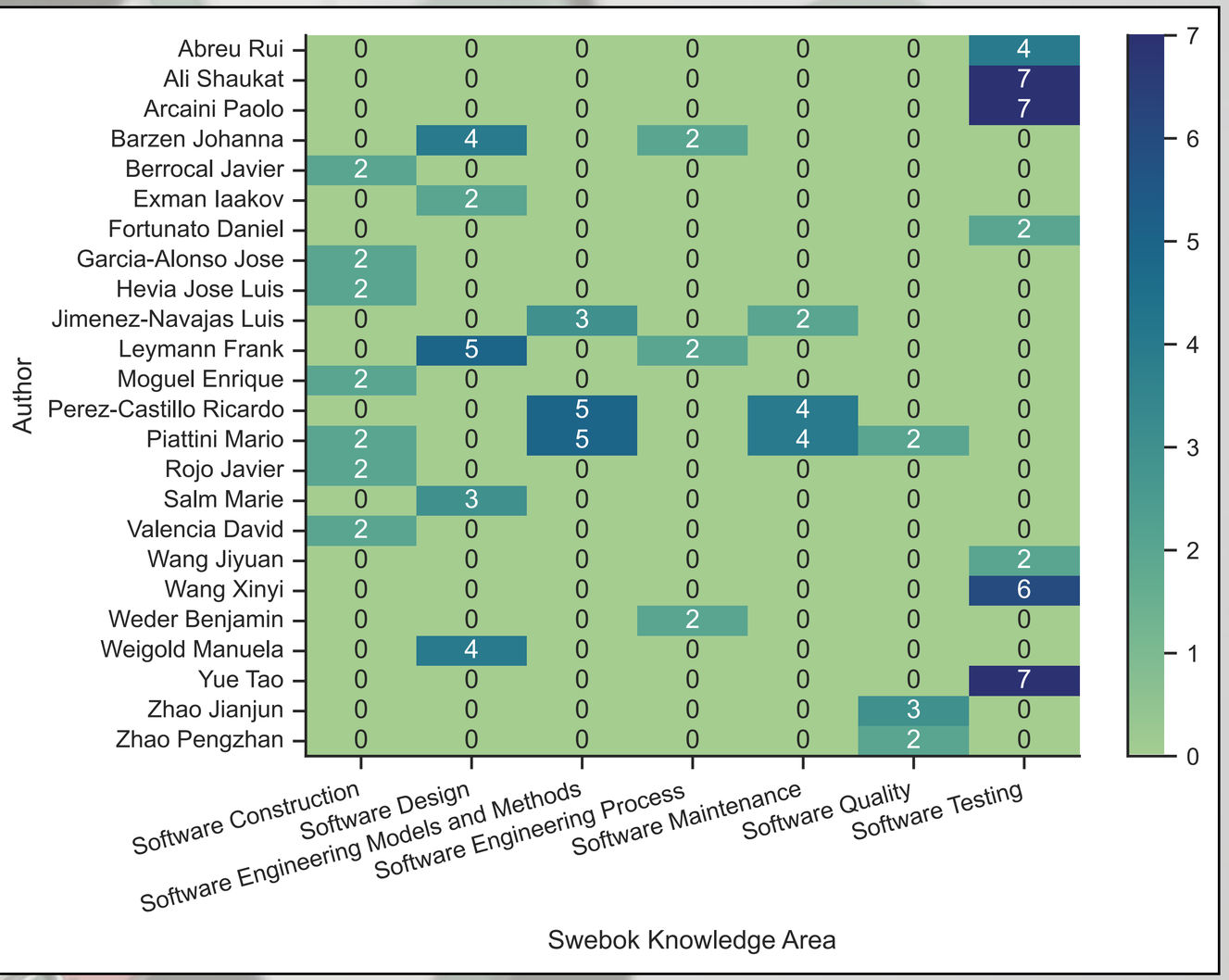




RQ3: Pre-establishment papers (8) and rapid growth of interest in the research community from 2020 to 2021.

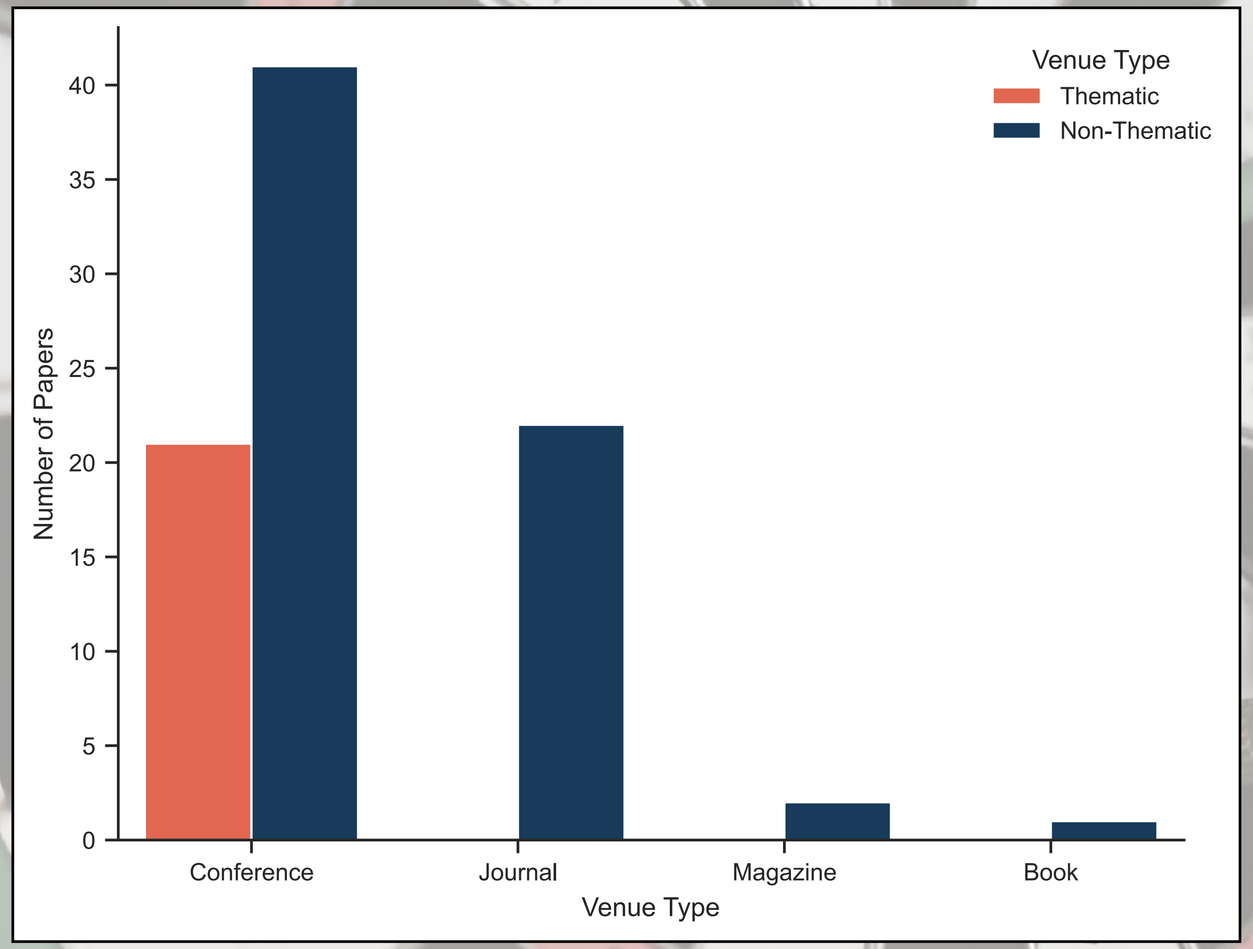
The state of the art of Quantum Software Engineering

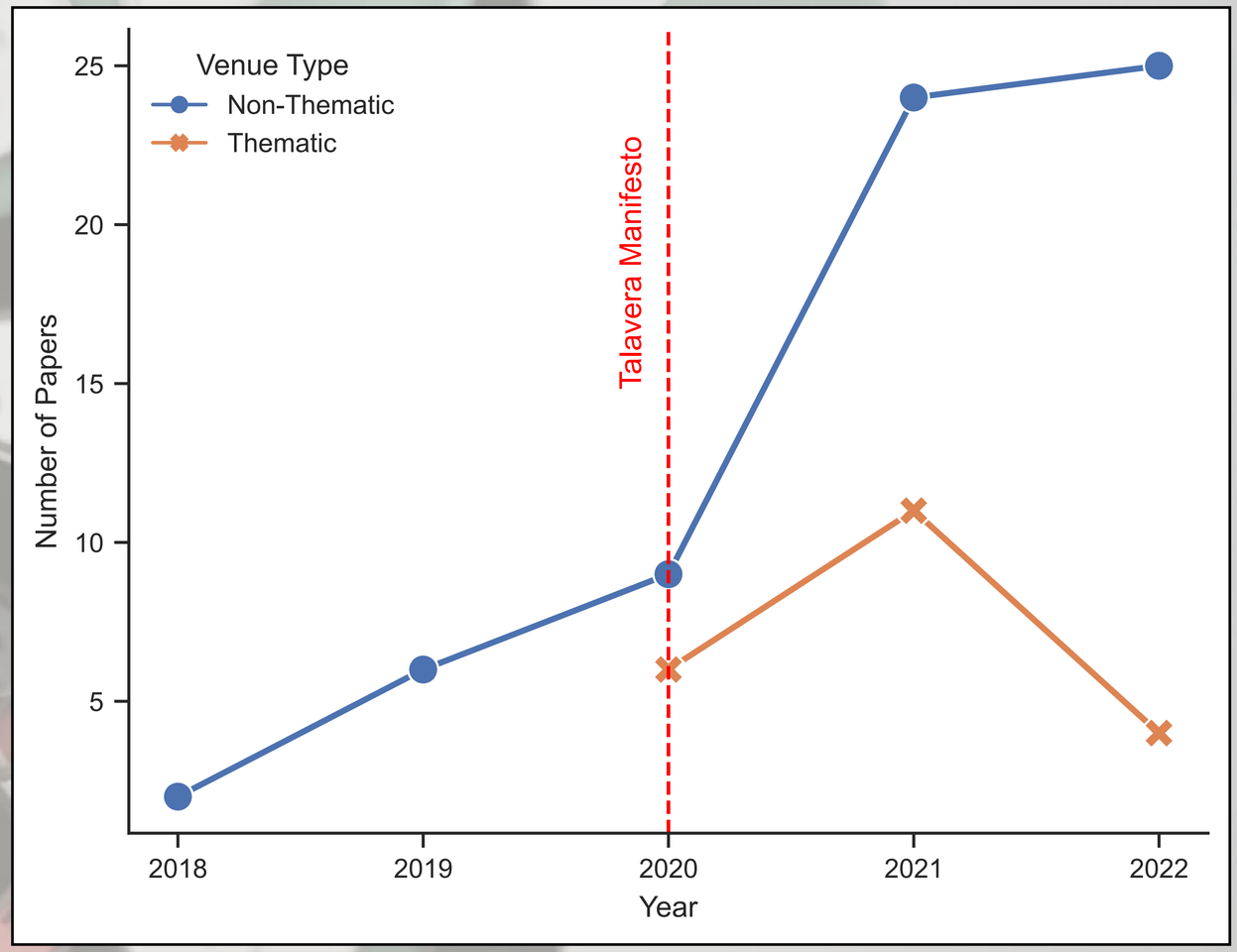




RQ4: Mario Piattini, Ricardo Perez-Castillo, Shaukat Ali, Frank Leymann, and Tao Yue, with Piattini as the most productive author (13 Publications). Most researchers concentrate on specific SE topics, with Piattini as the only author publishing in more than two areas.

The state of the art of Quantum Software Engineering

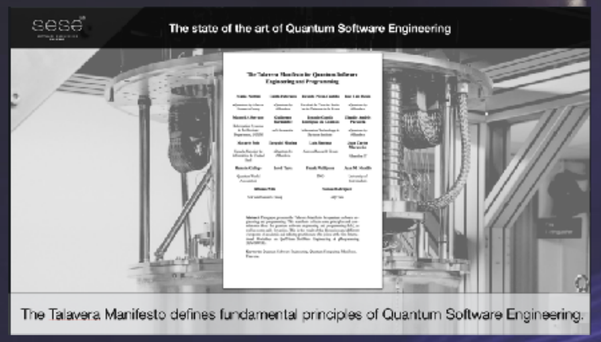




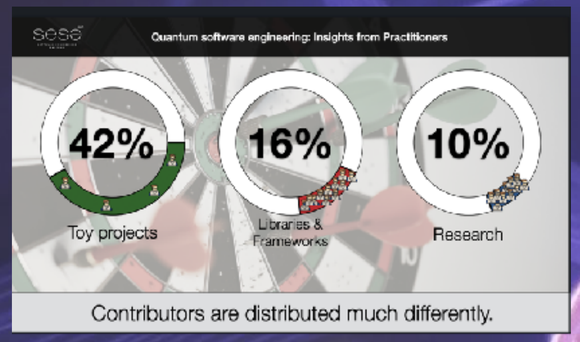
RQ5: Significant presence of Thematic and Non-Thematic Venues.
Conferences, particularly ICSE and ASE (including their workshops), as the most preferred preferred choices.

The Quantum Frontier of SE and its Adoption
Insights from Practitioners and a Systematic Mapping Study

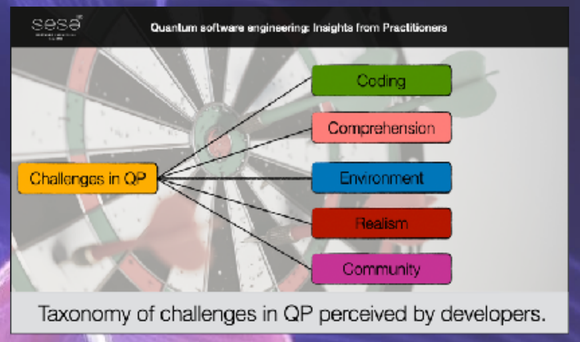
sesa lab
SOFTWARE ENGINEERING SALERNO



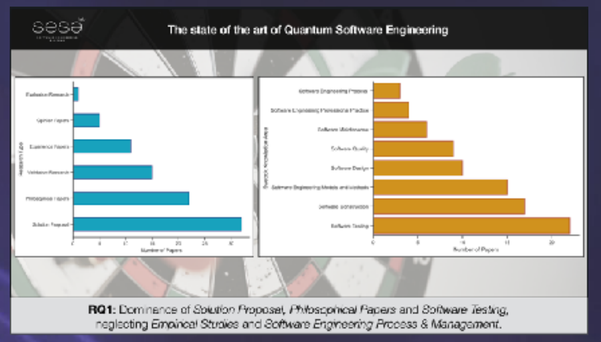
The Talavera Manifesto defines fundamental principles of Quantum Software Engineering.



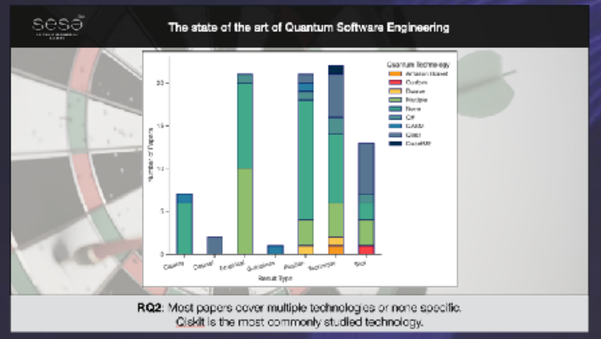
Contributors are distributed much differently.



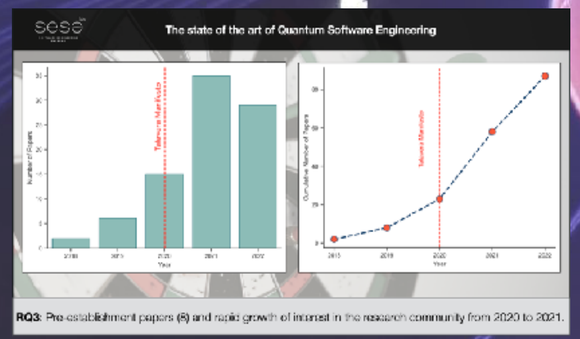
Taxonomy of challenges in QP perceived by developers.



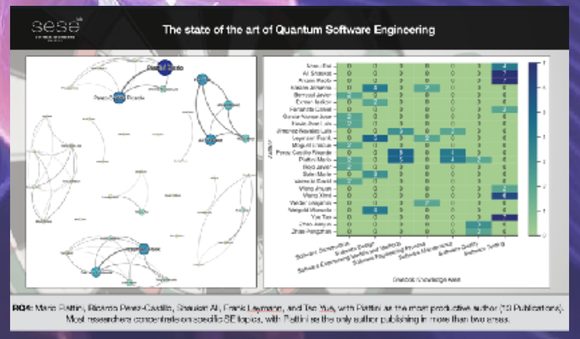
RQ1: Dominance of *Solution Proposal*, *Philosophical Papers* and *Software Testing*, neglecting *Empirical Studies* and *Software Engineering Process & Management*.



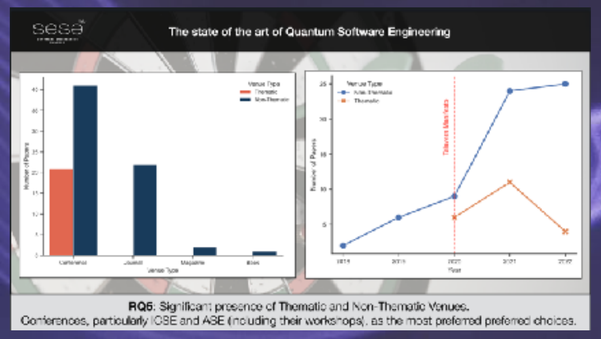
RQ2: Most papers cover multiple technologies or none specific. Ciskit is the most commonly studied technology.



RQ3: Pre-establishment papers (6) and rapid growth of interest in the research community from 2020 to 2021.



RQ4: Mario Piattini, Ricardo Perez-Castillo, Shaukat Ali, Frank Leymann, and Tao Yue, with Piattini as the most productive author (13 Publications). Most researchers concentrate on specific SE topics, with Piattini as the only author publishing in more than two areas.



RQ5: Significant presence of Thematic and Non-Thematic Venues. Conferences, particularly ICSE and ASE (including their workshops), as the most preferred preferred choices.

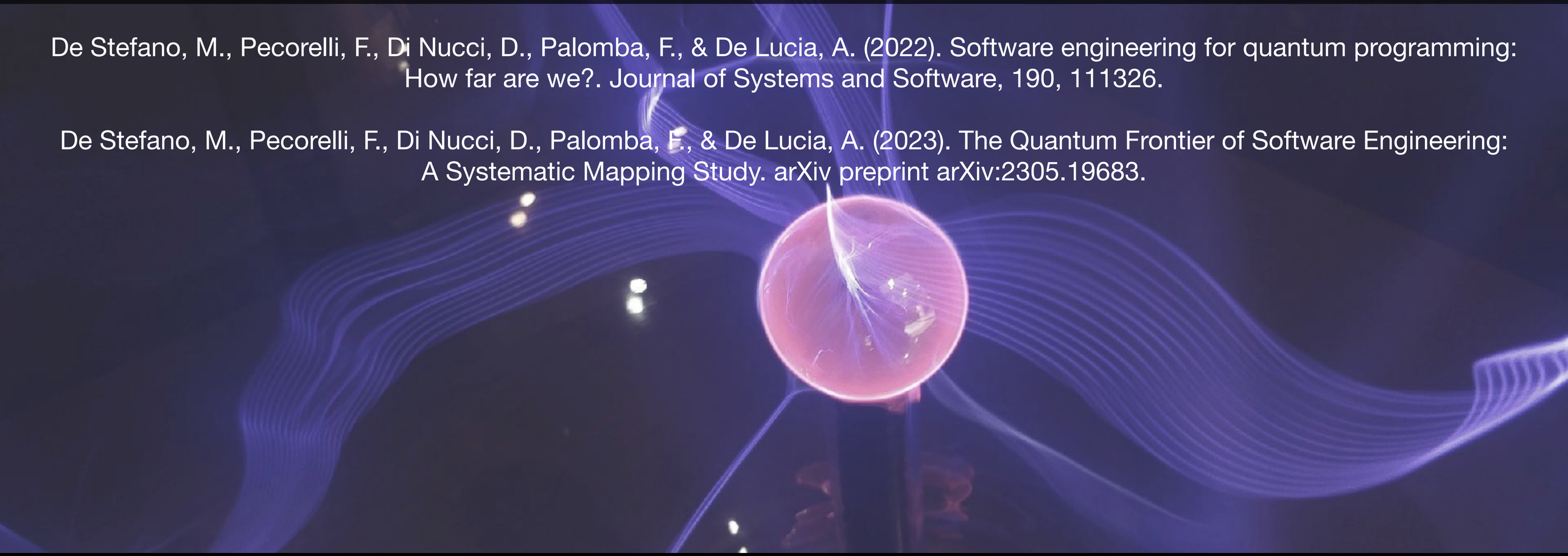
Dario Di Nucci

THANK YOU!

@dardin88
dardin88.github.io
ddinucci@unisa.it

The Quantum Frontier of SE and its Adoption
Insights from Practitioners and a Systematic Mapping Study

sesa lab
SOFTWARE ENGINEERING SALERNO



References

De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., & De Lucia, A. (2022). Software engineering for quantum programming: How far are we?. Journal of Systems and Software, 190, 111326.

De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., & De Lucia, A. (2023). The Quantum Frontier of Software Engineering: A Systematic Mapping Study. arXiv preprint arXiv:2305.19683.

Dario Di Nucci

@dardin88

dardin88.github.io

ddinucci@unisa.it