Quantum Computing and Software Engineering Superposition:

$\frac{1}{\sqrt{2}}(|QC\rangle + |SE\rangle)$

Prof. DSc. Filipe Fernandes

filipe.fernandes@ifsudestemg.edu.br

L I P E S

Laboratório de Inovação, Pesquisa e Engenharia de Software

Outline I

LIPES

Outline II

1. Who Am I?

Who Am I?

FILIPE FERNANDES

- Professor at IF Sudeste MG [since 2016]
- Head of LIPES
- Doctoral degree COPPE/UFRJ [2023]
- Master's degree COPPE/UFRJ [2017]
- Software Engineering (SE) Areas:
  - Software Visualization
  - SE Education
  - Metaverse Engineering
  - Immersive Learning
  - Quantum SE (*on going*)
  - Software-powered Innovation (*on going*)

LIPES

Professor and Researcher

- IF Sudeste MG - Campus Manhuaçu
- Software Engineering and Human-Computer Interaction disciplines
- Current projects:
  - Development of a Metaverse Platform as an Innovative Technological Strategy for Basic Education (FAPEMIG)
  - Identification of solutions, problems and challenges in methods, techniques and tools that support the development of applications for the Metaverse (FAPEMIG)
  - Transforming Manhuaçu into a Smart City (IF Sudeste MG)
- NITTEC coordinator

LIPES

LIPES

Laboratório de Inovação, Pesquisa e Engenharia de Software

@lipes.ifsemg

Master's dissertation

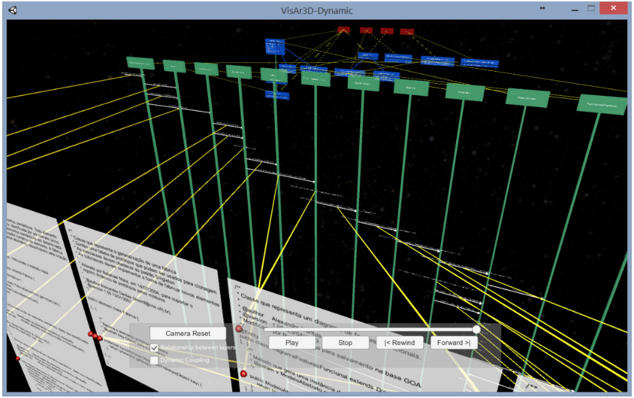


Figure: Perspectives visualization of VisAr3D-Dynamic (FERNANDES, 2017)

Thesis

Metaverse-based Software Engineering Education (MetaSEE)

- *How to support SEE through Immersive Learning?*

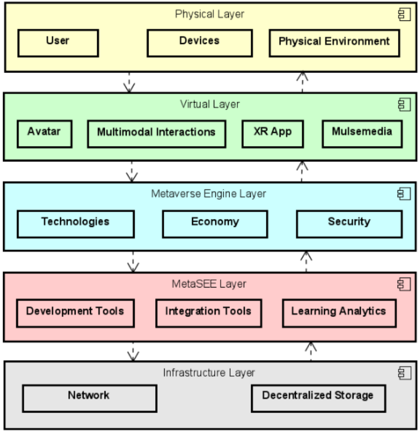


Figure: (FERNANDES, 2023)

Thesis

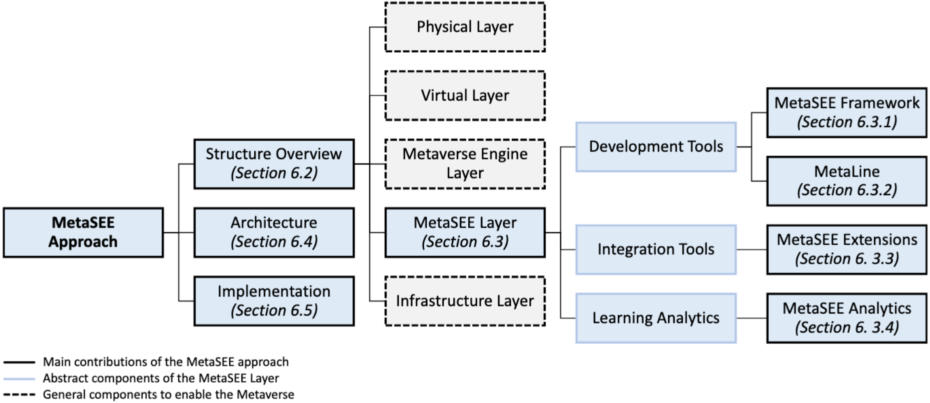


Figure: MetaSEE approach contributions (FERNANDES, 2023)

2. Introduction

Introduction

LIPES

Computing Types

Classical Computing

- Governed by the laws of classical mechanics
- Deterministic theory
- Uses digital computers to perform computation

Quantum Computing (QC)

- Governed by the laws of quantum mechanics
- Probabilistic theory
- Uses quantum computers to perform computation
  - Quantum circuits and
  - Adiabatic computing

Information Unit

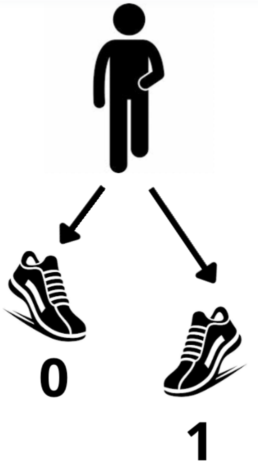
Classical Computing

- Bit = Binary Digity
- Values = 0 or 1

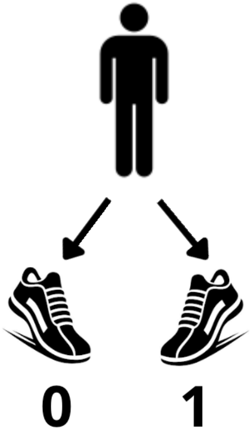
Quantum Computing

- Qubit = Quantum Bit
- Values = 0, 1, 0 and 1 at the same time

Classical Bit



Quantum Bit (Qubit)



LIPES

Applications

- **Simulations for complex quantum experiments**: complex chemistry, physics, and biology problems (SCHAETZ; MONROE; ESSLINGER, 2013)
- **Weather prediction**: forecast of natural disasters and allied complex tasks (FROLOV, 2017)
- **Post-quantum cryptography**: classic computer systems are safe with quantic computers attacks (BERNSTEIN *et al.*, 2017)
- **Quantum Finance**: optimize portfolios, model risks, and accelerate complex calculations (CANABARRO *et al.*, 2022)
- **Quantum Internet**: instant transmission of quantum information with lower latency and greater resistance to interference (CACCIAPUOTI *et al.*, 2020; UFF, 2024)
- **Quantum AI**: efficient processing of large amounts of data and the optimization of complex models exponentially faster (PERDOMO-ORTIZ *et al.*, 2018)
- Among others

QC Goal

- QC offers different solutions to computational problems and enables more efficient problem-solving than what is possible with classical computations (GYONGYOSI; IMRE, 2019)
  - Example: *while a problem with 30 variables would take around 17 minutes to solve, the same problem with 50 variables would take more than 35 years* (CANABARRO *et al.*, 2022)
- Quantum Mechanics Concepts
  - Estates superposition
  - Quantum entanglement

LIPES

3. Quantum Mechanics Concepts

Young or Elderly?

Answer the questionnaire

Observation

States superposition

$$|Person\rangle = \frac{1}{\sqrt{2}}(|Young\rangle + |Elderly\rangle)$$

P(Y) = 50% e P(E) = 50%

Coin superposition

shutterstock.com · 1317073241

$$|Coin\rangle = \frac{1}{\sqrt{2}}|Heads\rangle + \frac{1}{\sqrt{2}}|Tails\rangle$$

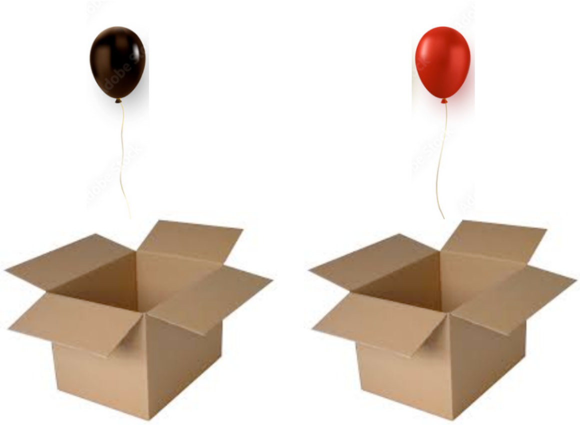
or simply

$$|Coin\rangle = \frac{1}{\sqrt{2}}(|Heads\rangle + |Tails\rangle)$$

Entanglement

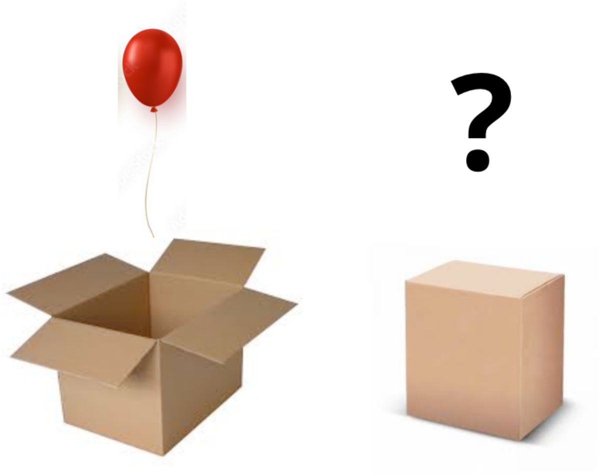
- In an entangled state, the properties of qubits are linked to each other, in spite of physical separation between them (GILL *et al.*, 2022)

Entanglement example



Entanglement example

Entanglement example



4. Quantum Computing

Quantum Computing

(GILL *et al.*, 2022)

QC is an emerging paradigm with the potential to **offer significant computational advantage** over conventional classical computing by **exploiting quantum-mechanical principles** such as superposition and entanglement

Algebraic Representation

$$|Coin\rangle = \frac{1}{\sqrt{2}}|Heads\rangle + \frac{1}{\sqrt{2}}|Tails\rangle$$

Algebraic representation

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \alpha, \beta \in \mathbb{C}$$

- α and β are **probabilities amplitudes**
- $|\alpha|^2 + |\beta|^2 = 1$
- Computational base:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Geometrical Representation

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{-i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

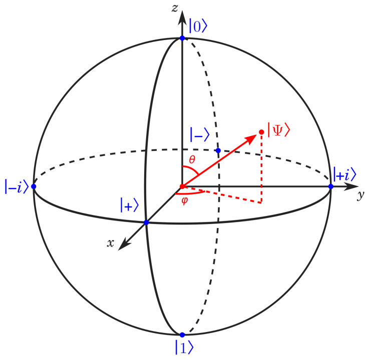


Figure: Bloch sphere

Access simulation

Bloch Sphere Example

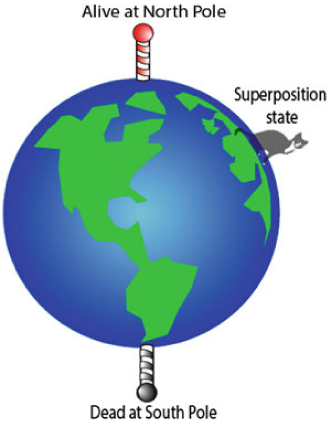


Figure: A cartoon of the Bloch sphere (HUGHES *et al.*, 2021)

Qubit States

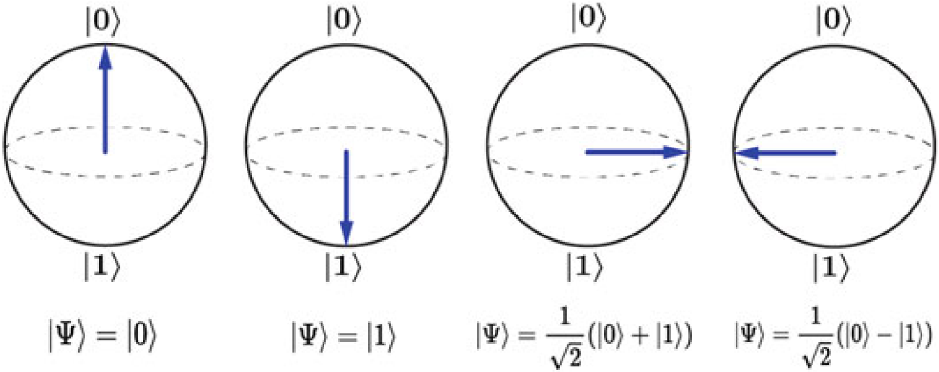
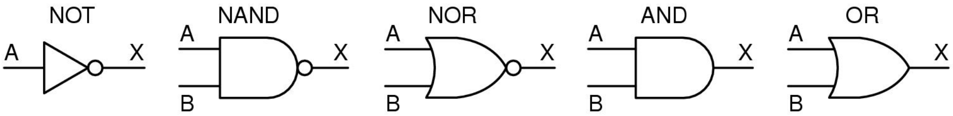


Figure: The state of a qubit is represented by an arrow on the Bloch sphere (HUGHES *et al.*, 2021)

Classic Logic Gates



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

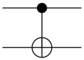
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

Quantum Logic Gates

Operator	Gate(s)		Matrix
Pauli-X (X)	X	⊕	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Pauli Gates

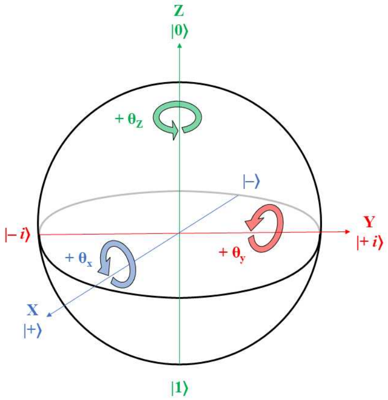


Figure: Access simulation

- **Pauli-X**: bit-flip

$$\sigma_x |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

- **Pauli-Z**: phase-flip

$$\sigma_z |1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = - |1\rangle$$

- **Pauli-Y**: bit-phase-flip

$$\sigma_y |0\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix} = i |1\rangle$$

Hadamard Gate

- Creating the quantum superposition:

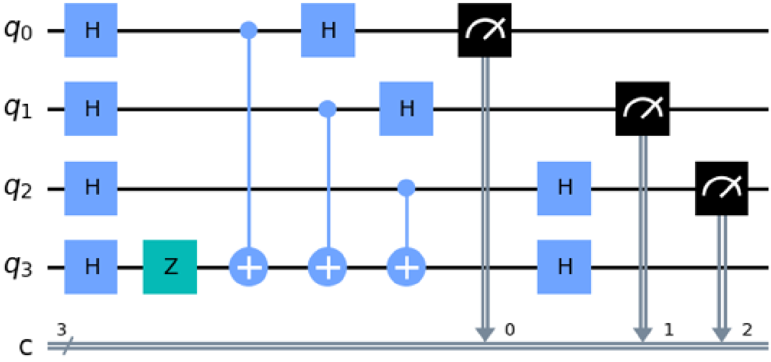
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Example:

$$H\left|1\right\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} =$$

$$\boxed{\frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) \text{ ou } \frac{(\left|0\right\rangle - \left|1\right\rangle)}{\sqrt{2}} \text{ ou } \left|-\right\rangle}$$

Quantum Circuits



Example (equation)

$$|\psi\rangle = CX \cdot (I\,|0\rangle \otimes H\,|0\rangle)$$

$$|\psi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \otimes \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \right] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ ou } \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$$

Example (circuit)

$$|\psi\rangle = CX \cdot (I\,|0\rangle \otimes H\,|0\rangle)$$

Example (qiskit code)

```
from qiskit import QuantumRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(2, 'q')
circuit = QuantumCircuit(qreg_q)

circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
```

Question 1

How to convert real problems into quantum algorithms?

Quantum Algorithms

Core computing algorithm	The name of algorithms	Applications	Potential application field
Quantum Fourier Transform (QFT)	Shor's algorithm	RSA decryption	Cryptography
	HHL	Inverse transform of a matrix	Machine learning
Grover's operator	Grover's algorithm	Search problem	Search in unsorted databases
Quantum-classical hybrid methods	Variational Quantum Eigensolver (VQE)	Eigensolver	New material finding
	Quantum Approximate optimization algorithm (QAOA)	Optimization	Financial industry, Satisfiability problems
Quantum adiabatic algorithm	Quantum Annealing algorithm	Optimization	Computing science, Machine learning, Financial industry

Table: Major quantum algorithms and their possible applications (CHO *et al.*, 2021)

Quantum Algorithms Examples

Quantum Finance: portfolio optimization (CANABARRO *et al.*, 2022)

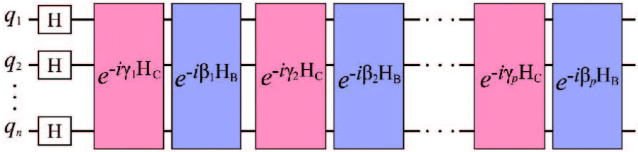


Figure: QAOA circuit implementation (CANABARRO *et al.*, 2022)

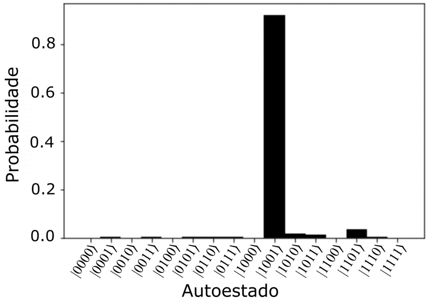


Figure: Histogram of probabilities distribution (CANABARRO *et al.*, 2022)

Quantum Algorithms Examples

Quantum Machine Learning (QML) (GOMES *et al.*, 2024)

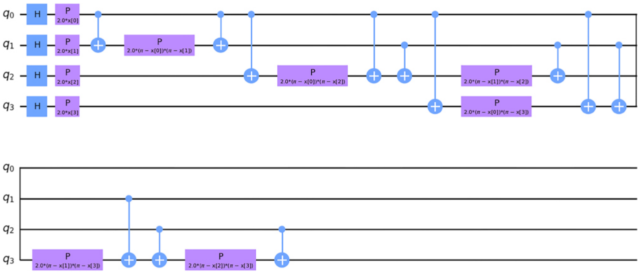


Figure: QML circuit (GOMES *et al.*, 2024)

```python
# Define o ansatz RealAmplitudes
ansatz = RealAmplitudes(num_qubits=n_qubits, reps=2)
ansatz.decompose().draw(output="mpl", style="clifford", fold=20)

# Define o otimizador COBYLA
optimizer = COBYLA(maxiter=50)

# Inicializa o classificador VQC
vqc = VQC(feature_map=feature_map, ansatz=ansatz, optimizer=optimizer)

# Treinamento do modelo VQC e teste
vqc.fit(X_train, y_train)
test_score = vqc.score(X_test, y_test)

# Fazendo previsões no conjunto de teste
y_pred = vqc.predict(X_test)

# Calculando as métricas no conjunto de teste
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Imprimindo as métricas
print(f"Acurácia: {accuracy:.2f}")
print(f"Precisão: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-Score: {f1:.2f}")
```

Figure: QML code (GOMES *et al.*, 2024)

LIPES

Quantum Algorithms Examples

Traveling Salesman Problem with QAOA (RUAN *et al.*, 2020)

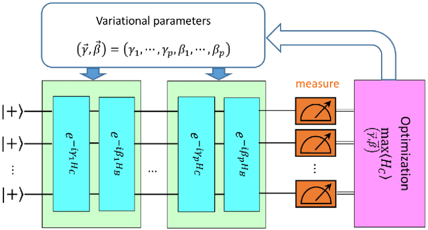


Figure: Schematic diagram of hybrid quantum-classical QAOA (RUAN *et al.*, 2020)

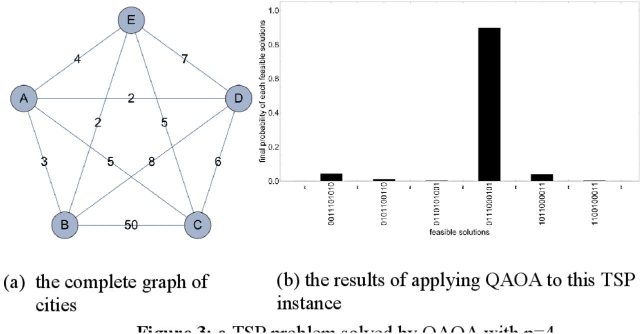


Figure: A TSP problem solved by QAOA with p=4 (RUAN *et al.*, 2020)

Question 2

How to manipulate quantum programs with the same ease and confidence of classic programs?

5. Quantum Software Engineering

Quantum Software Engineering (QSE)

(ZHAO, 2021)

Quantum Software Engineering (QSE) is 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

- "**sound engineering principles**" to Quantum Software (QS) development
- the quantum software should be built **"economically"**
- the quantum software should be **"reliable"** and needs to work **"efficiently"** on quantum computers

QSE needs

- **Methods**
  - To provide the techniques for constructing the quantum software
  - Design of data structures, program architecture, algorithm procedure, coding, testing, and maintenance
- **Tools**
  - To provide automated or semiautomated support for these methods
- **Process**
  - To provide the glue that holds the methods and tools together and enables the rational and timely development of quantum software

Quantum Software Life Cycle

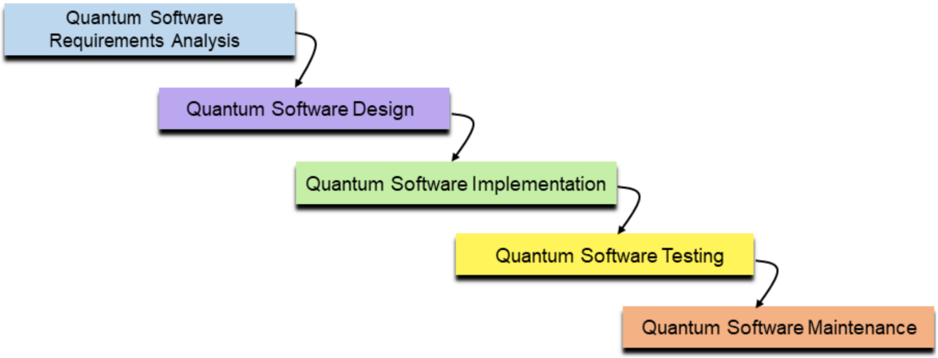


Figure: A quantum software life cycle (ZHAO, 2021)

QS Requirements Analysis

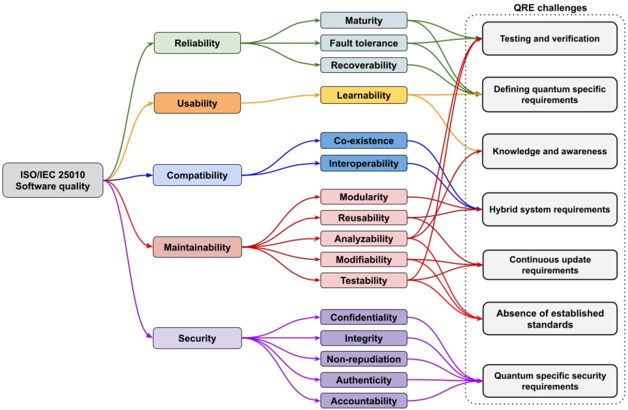


Figure: Quantum Requirements Engineering challenges (SEPúLVEDA *et al.*, 2024)

QS Design

- **Quantum Software Modelling**
  - *UML-Based Modelling language*: an approach to extending the UML to model quantum software systems (PéREZ-DELGADO; PEREZ-GONZALEZ, 2020)
  - *Generic Modelling Languages*: a preliminary conceptual model from the perspective of model-based engineering (ALI; YUE, 2020)
- **Quantum Software Specification**
  - Cartiere (2013) presented some work on defining a formal specification language for quantum algorithms
- **Modular Design of Quantum Systems**
  - Thompson *et al.* (2018) presented a formal framework to specify modularity in quantum systems
  - Sánchez and Alonso (2021) discussed the concept of module (next slide)

LIPES

QS Reuse

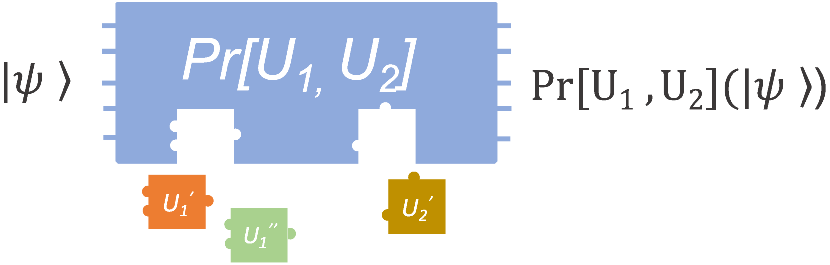


Figure: A modular approximation for quantum computing (SÁNCHEZ; ALONSO, 2021)

QS Implementation

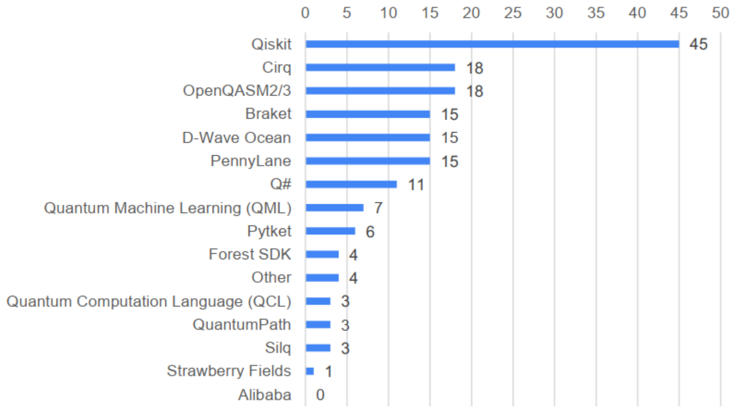


Figure: Quantum programming languages or toolkits used (JIMENEZ-NAVAJAS *et al.*, 2024)

QS Testing

- How to define testing coverage criteria of quantum software?
- How to automatically and efficiently generate test cases for quantum software?
- How to evaluate the test data quality for quantum software?
- How to test quantum software regressively?

(ZHAO, 2021)

QS Maintenance

- How to understand the existing quantum software?
- How to modify the existing quantum software?
- How to re-validate the modified quantum software?

(ZHAO, 2021)

6. Conclusion

Question 3

Is it necessary to adapt all SE to the quantum program paradigm?

- State of the practice
- QSE Education

7. References

References I

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**. New York, NY, USA: Association for Computing Machinery, 2020. (APEQS 2020), p. 14–21. ISBN 9781450381000.

BERNSTEIN, D. J. *et al*. Post-quantum rsa. In: LANGE, T.; TAKAGI, T. (Ed.). **Post-Quantum Cryptography**. Cham: Springer International Publishing, 2017. p. 311–329. ISBN 978-3-319-59879-6.

CACCIAPUOTI, A. S. *et al*. Quantum internet: Networking challenges in distributed quantum computing. **IEEE Network**, v. 34, n. 1, p. 137–143, 2020.

CANABARRO, A. *et al*. Quantum finance: um tutorial de computação quântica aplicada ao mercado financeiro. **Revista Brasileira de Ensino de Física**, Sociedade Brasileira de Física, v. 44, p. e20220099, 2022. ISSN 1806-1117. Available: <https://doi.org/10.1590/1806-9126-RBEF-2022-0099>.

CARTIERE, C. R. **Quantum software engineering: bringing the classical software engineering into the quantum domain**. Thesis (Doctorate) — Master's Thesis, University of Oxford, Department of Computer Science . . . , 2013.

References II

CHO, C.-H. *et al.* Quantum computation: Algorithms and applications. **Chinese Journal of Physics**, v. 72, p. 248–269, 2021. ISSN 0577-9073. Available: <https://www.sciencedirect.com/science/article/pii/S0577907321001039>.

FERNANDES, F. A. **VisAr3d-Dynamic: uma abordagem para apoiar a compreensão do comportamento dinâmico de software por meio de realidade virtual**. Master thesis (Master degree) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2017.

FERNANDES, F. A. **MetaSEE: An Approach to Enable the Metaverse-based Software Engineering Education**. 279 p. Thesis (Doctorate) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2023.

FROLOV, A. V. Can a quantum computer be applied for numerical weather prediction? **Russian Meteorology and Hydrology**, v. 42, n. 9, p. 545–553, 2017. ISSN 1934-8096. Available: <https://doi.org/10.3103/S1068373917090011>.

GILL, S. S. *et al.* Quantum computing: A taxonomy, systematic review and future directions. **Software: Practice and Experience**, v. 52, n. 1, p. 66–114, 2022. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3039>.

GOMES, N. D. *et al.* Introdução ao aprendizado de máquina quântico, suas aplicações e vantagens. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 46, p. e20240230, 2024.

LIPES

References III

GYONGYOSI, L.; IMRE, S. A survey on quantum computing technology. **Computer Science Review**, v. 31, p. 51–71, 2019. ISSN 1574-0137. Available: <https://www.sciencedirect.com/science/article/pii/S1574013718301709>.

HUGHES, C. *et al.* **Quantum Computing for the Quantum Curious**. [S.l.]: Springer Nature, 2021. 150 p. ISBN 9783030616014.

JIMENEZ-NAVAJAS, L. *et al.* Quantum software development: a survey. **Quantum Information and Computation**, v. 24, n. 7 and 8, p. 0609–0642, 2024.

PERDOMO-ORTIZ, A. *et al.* Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. **Quantum Science and Technology**, IOP Publishing, v. 3, n. 3, p. 030502, jun 2018. Available: <https://dx.doi.org/10.1088/2058-9565/aab859>.

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**. New York, NY, USA: Association for Computing Machinery, 2020. (ICSEW'20), p. 442–444. ISBN 9781450379632. Available: <https://doi.org/10.1145/3387940.3392183>.

RUAN, Y. *et al.* The quantum approximate algorithm for solving traveling salesman problem. **Computers, Materials & Continua**, 2020. Available: <https://api.semanticscholar.org/CorpusID:219040815>.

References IV

SCHAETZ, T.; MONROE, C. R.; ESSLINGER, T. Focus on quantum simulation. **New Journal of Physics**, IOP Publishing, v. 15, n. 8, p. 085009, aug 2013. Available: <https://dx.doi.org/10.1088/1367-2630/15/8/085009>.

SEPúLVEDA, S. *et al.* Systematic review on requirements engineering in quantum computing: Insights and future directions. **Electronics**, v. 13, n. 15, 2024. ISSN 2079-9292. Available: <https://www.mdpi.com/2079-9292/13/15/2989>.

SáNCHEZ, P.; ALONSO, D. On the definition of quantum programming modules. **Applied Sciences**, v. 11, n. 13, 2021. ISSN 2076-3417. Available: <https://www.mdpi.com/2076-3417/11/13/5843>.

THOMPSON, J. *et al.* Quantum plug n' play: modular computation in the quantum regime. **New Journal of Physics**, IOP Publishing, v. 20, n. 1, p. 013004, jan 2018. Available: <https://dx.doi.org/10.1088/1367-2630/aa99b3>.

UFF. **Rede de internet quântica integra instituições de ensino e pesquisa no Rio**. 2024. Acessado em: 29 jan. 2025. Available: <https://www.uff.br/16-09-2024/rede-de-internet-quantica-integra-instituicoes-de-ensino-e-pesquisa-no-rio/>.

ZHAO, J. **Quantum Software Engineering: Landscapes and Horizons**. 2021. Available: <https://arxiv.org/abs/2007.07047>.

Thank You Very Much

Quantum Computing and Software Engineering Superposition: $\frac{1}{\sqrt{2}}(|QC\rangle + |SE\rangle)$

Prof. DSc. Filipe Fernandes

filipe.fernandes@ifsudestemg.edu.br

LIPES
Laboratório de Inovação, Pesquisa e Engenharia de Software