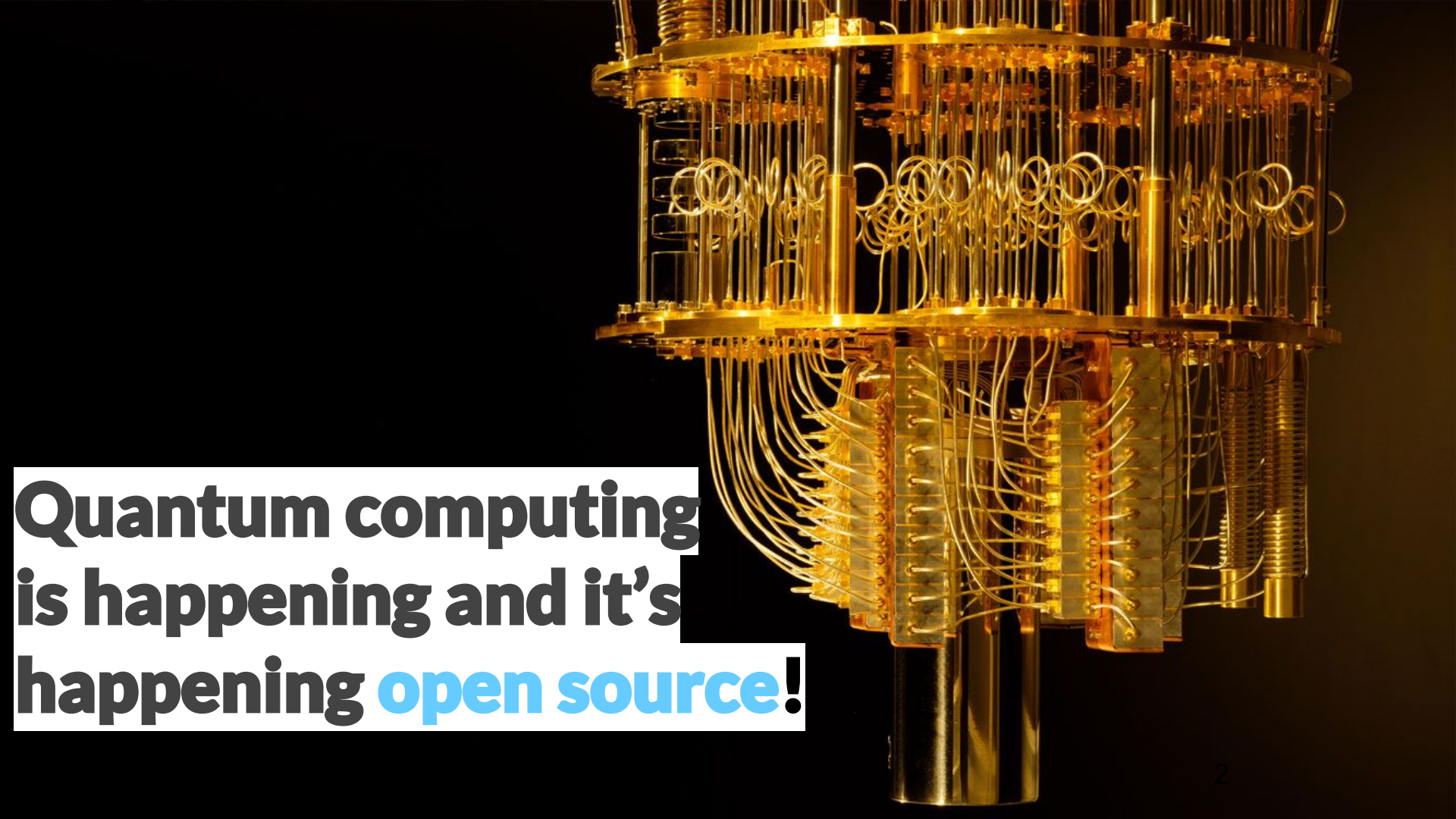
When open source meets quantum computing

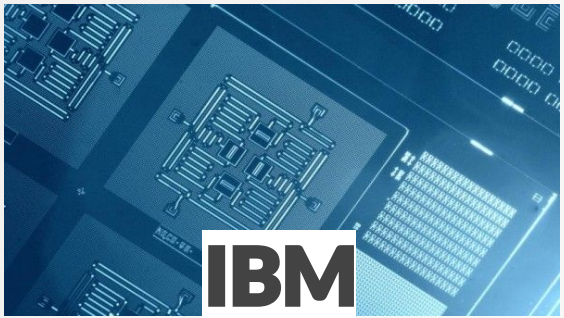
Mark Fingerhuth

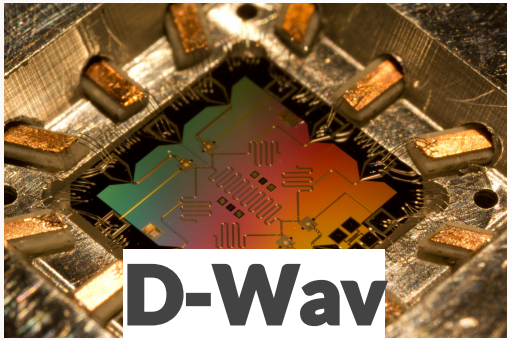
FOSDEM 2019
Brussels, Belgium
February 2-3, 2019

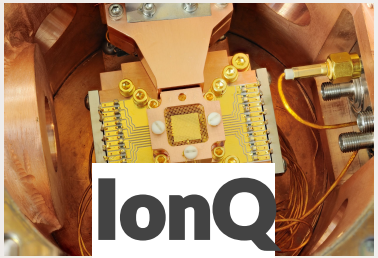

Quantum computing
is happening and it's
happening open source!

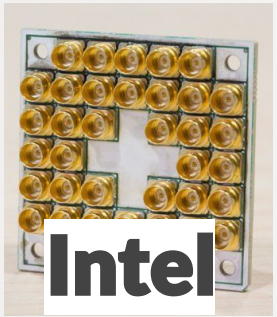
Plenty of quantum hardware ("QPUs") available today...

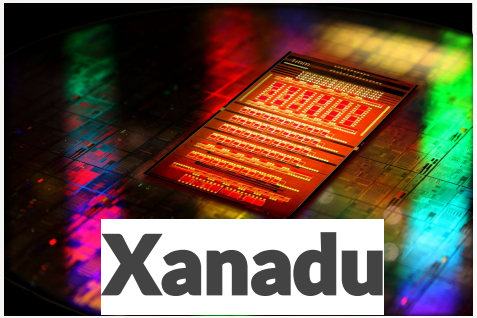
Google


IBM


D-Wave

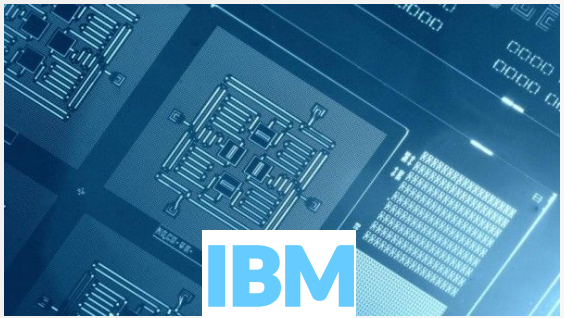

IonQ

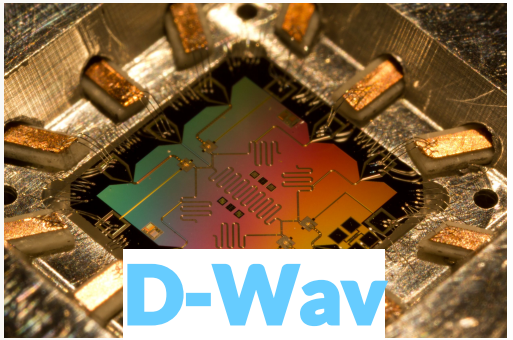

Intel

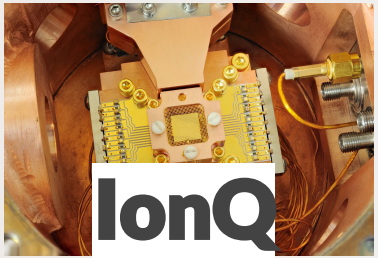

Xanadu

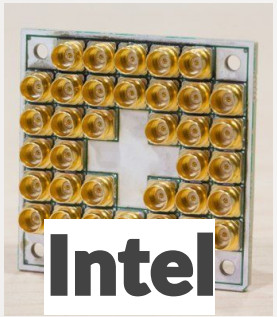

Rigetti

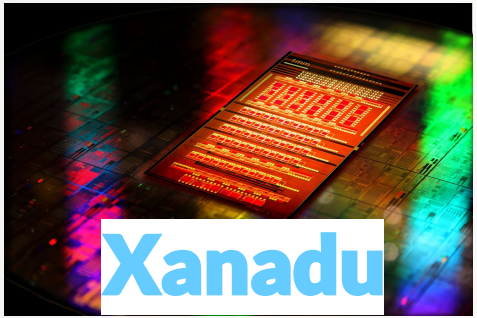
Plenty of quantum hardware ("QPUs") available today...

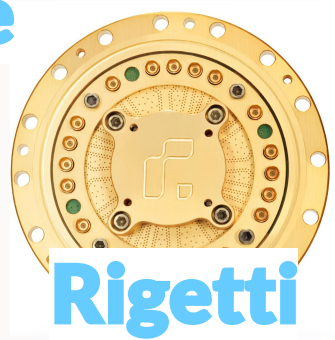












There even is a incubator program for quantum software startups!

CREATIVE DESTRUCTION LAB

Applications are open now!

In QC everyone is chasing exponential speedups over classical algorithms

Calculating the Thermal Rate Constant with Exponential Speed-Up on a Quantum Computer

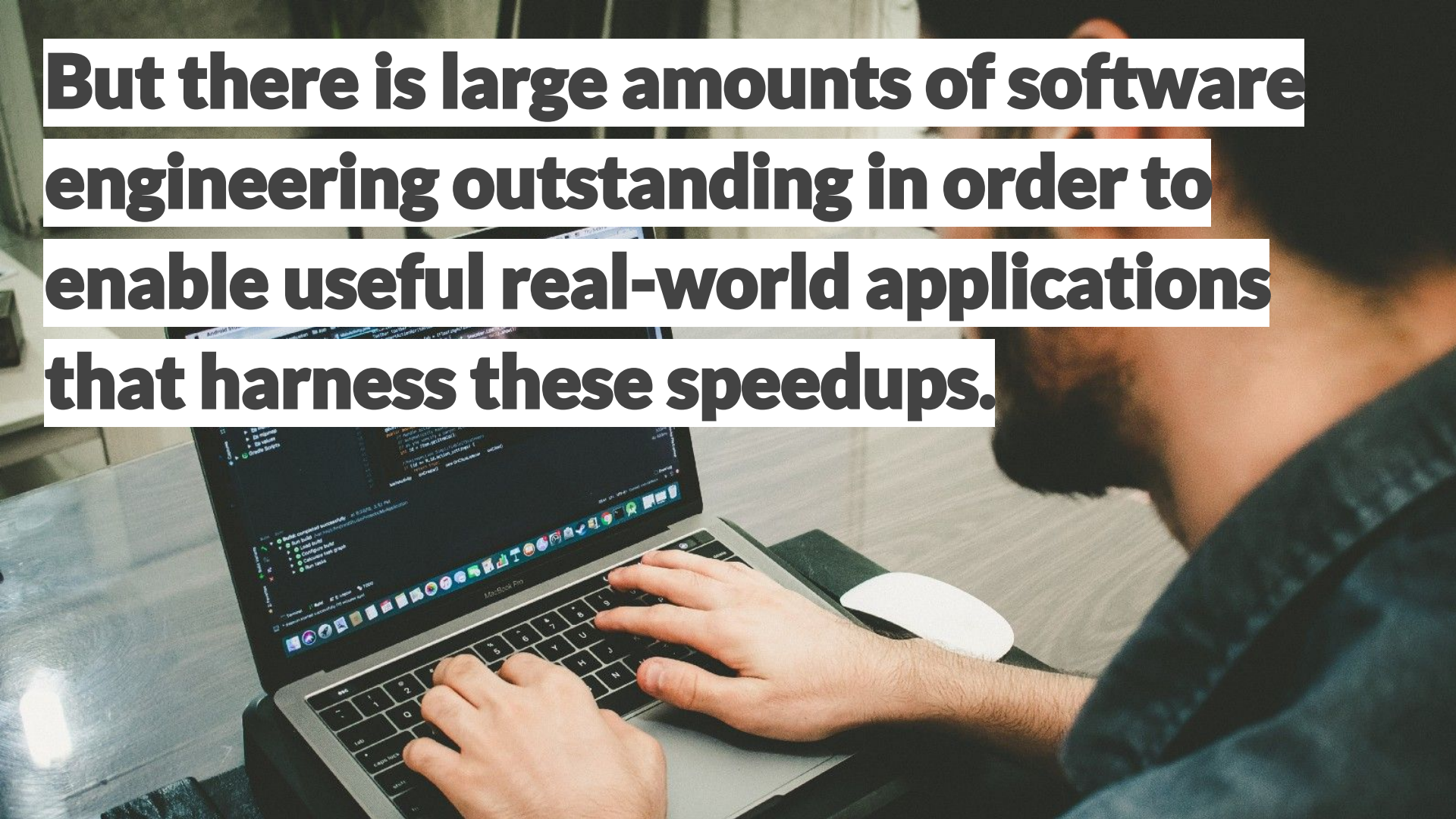
Daniel A. Lidar and Haobin Wang
Department of Chemistry, The University of California, Berkeley, CA 94720

Exponential algorithmic speedup by quantum walk

Andrew M. Childs,[1,*] Richard Cleve,[2,†] Enrico Deotto,[1,‡] Edward Farhi,[1,§] Sam Gutmann,[3,¶] and Daniel A. Spielman[4,**]

Eigenvector Approximation Leading to Exponential Speedup of Quantum Eigenvalue Calculation

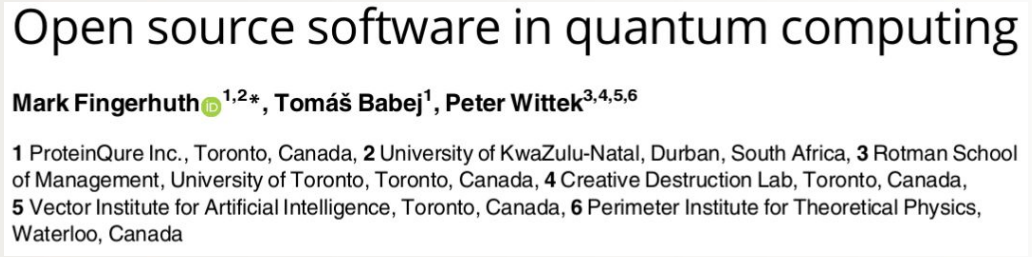
Peter Jaksch[*] and Anargyros Papageorgiou[†]


But there is large amounts of software engineering outstanding in order to enable useful real-world applications that harness these speedups.

There is already a mesmerizing diversity of quantum open source projects and there is a curated list:

https://github.com/qosf/os_quantum_software

We set out and did an extensive review of the current state of quantum open source software...



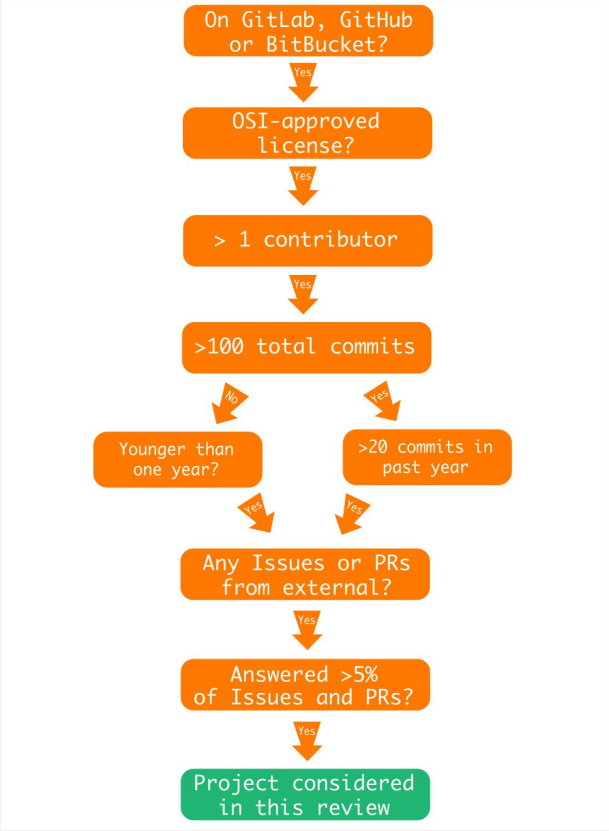
Open source software in quantum computing

Mark Fingerhuth[1,2]*, **Tomáš Babej**[1], **Peter Wittek**[3,4,5,6]

1 ProteinQure Inc., Toronto, Canada, **2** University of KwaZulu-Natal, Durban, South Africa, **3** Rotman School of Management, University of Toronto, Toronto, Canada, **4** Creative Destruction Lab, Toronto, Canada, **5** Vector Institute for Artificial Intelligence, Toronto, Canada, **6** Perimeter Institute for Theoretical Physics, Waterloo, Canada

and here is what we found.

Quantum project selection criteria



- We checked >60 quantum software repositories

26 quantum software projects were selected based on these criteria.

Main reason for exclusion was contributor count and lack of external interest.

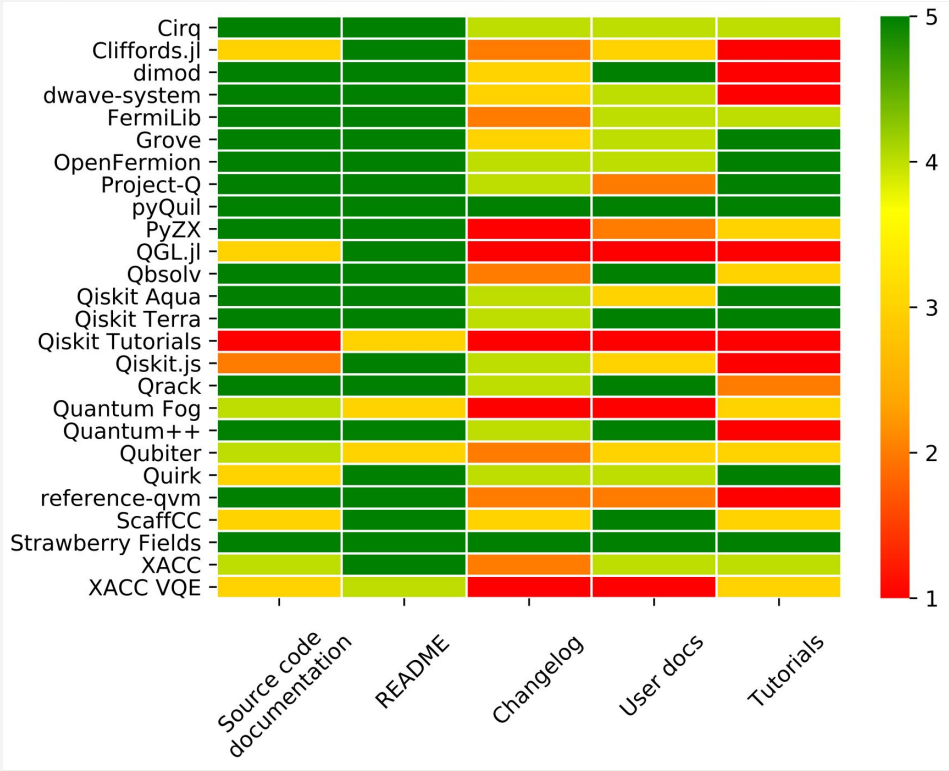
Results: Open source licences

- >90% of the projects used permissive (open) licences
- 65% of the projects chose the permissive Apache-2.0 licence
- Only two projects are released under copyleft licences

Community seems open with respect to commercial use of their software.

Results: Documentation analysis

- Most projects have good source code documentation and README files
- **Major shortcomings** in changelogs, extensive per-feature user documentation
- **Lack of hands-on tutorials** that show application of the software



Static analysis of quantum software

Writing high-quality code and testing is crucial.

- 23 out of 26 projects used automated test suites
- Average code coverage was found to be 75%
- Median code coverage was 87% (standard: >85%)

```
def test_text_cswap_reversebits(self):
    """ CSwap drawing with reversebits. """
    expected = '\n'.join(["                    ",
                          "q_2: |0>─X──X──■──",
                          "         │  │  │  ",
                          "q_1: |0>─X──■──X──",
                          "         │  │  │  ",
                          "q_0: |0>─■──X──X──",
                          "                  "])
    qr = QuantumRegister(3, 'q')
    circuit = QuantumCircuit(qr)
    circuit.cswap(qr[0], qr[1], qr[2])
    circuit.cswap(qr[1], qr[0], qr[2])
    circuit.cswap(qr[2], qr[1], qr[0])
    self.assertEqual(str(_text_circuit_drawer(circuit, reversebits=True)), expected)

def test_text_cu3(self):
    """ cu3 drawing. """
    expected = '\n'.join(["                                                  ",
                          "q_0: |0>──────────────────────■──────────────────",
                          "                             │                  ",
                          "q_1: |0>─U3(1.5708,1.5708,1.5708)──────────────",
                          "                                                ",
                          "q_2: |0>──────────────────────────────────────■──",
                          "                                              │  ",
                          "                                              "])
    qr = QuantumRegister(3, 'q')
    circuit = QuantumCircuit(qr)
    circuit.cu3(pi / 2, pi / 2, pi / 2, qr[0], qr[1])
    circuit.cu3(pi / 2, pi / 2, pi / 2, qr[2], qr[0])
    self.assertEqual(str(_text_circuit_drawer(circuit)), expected)
```

Static analysis of quantum software

Responding to issues and pull requests is important for building a healthy ecosystem.

- Issues and pull requests without answer for 30 days are considered ignored
- We measured *attention rate* for all projects
  AR = 1- [unanswered I & PRs]/[total I & PRs]

Static analysis of quantum software

Responding to issues and pull requests is important for building a healthy ecosystem.

- Issues and pull requests without answer for 30 days are considered ignored
- We measured *attention rate* for all projects
  AR = 1- [unanswered I & PRs]/[total I & PRs]

Commercial backed project are not better at replying to issues and pull request than community efforts!

Results: Community analysis

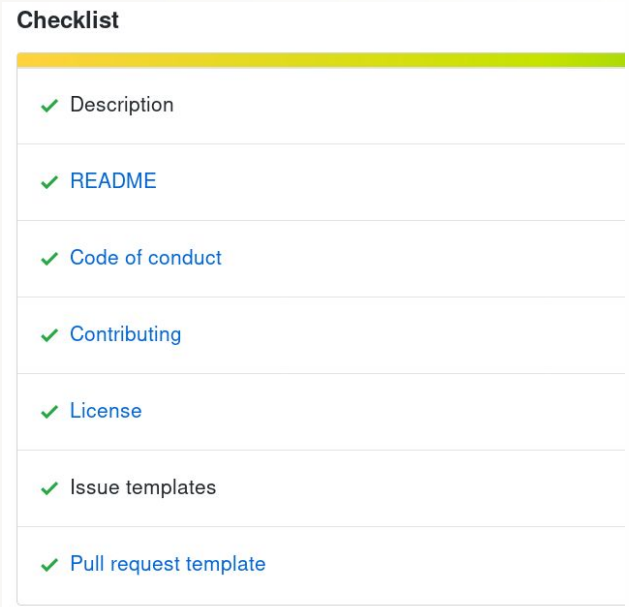
- **GitHub's community profile** is a quantitative measure for best practices within a project
- Only 4 projects scored 7/7 on this metric.

Code review is important to maintain quality code.

- 10 out of 26 projects do **not review the code** of core contributors.

Support & discussion channels:

- We identified a **lack of developer-centric discussion forums** to drive design decisions
- Qiskit is the only project with a **public roadmap**



Screenshot of GitHub community profile

This type of research gets outdated quickly...

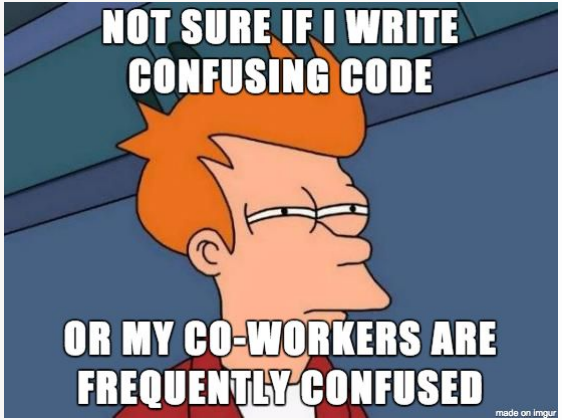
This type of research gets outdated quickly...

...so we decided to automate the evaluation process and continuously publish the results online:

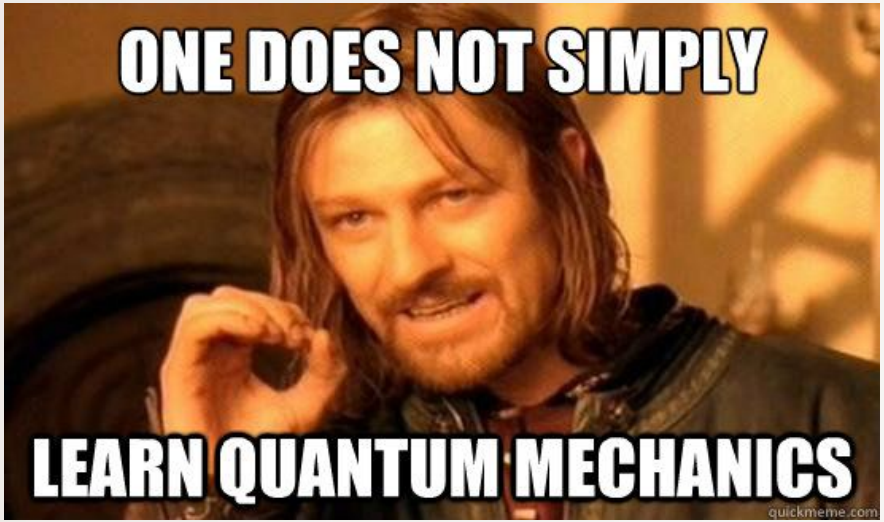
https://qosf.org

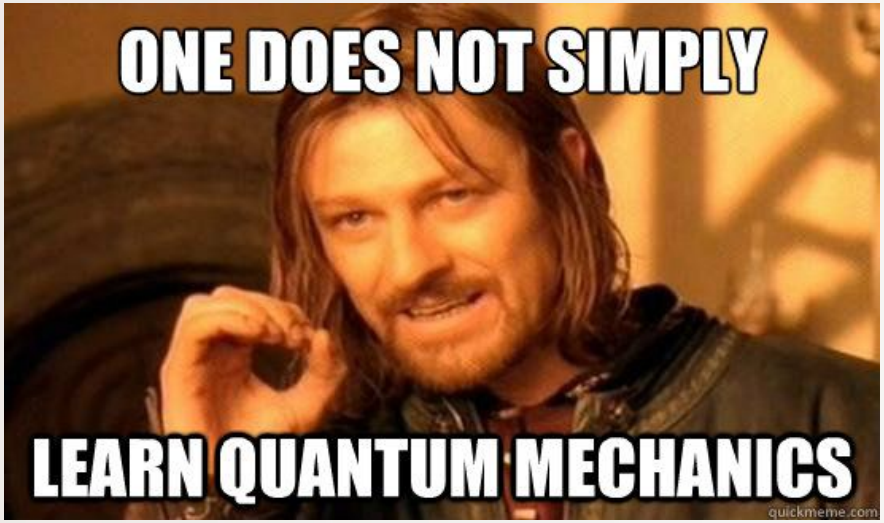
Physicists are great with equations but not necessarily with code.

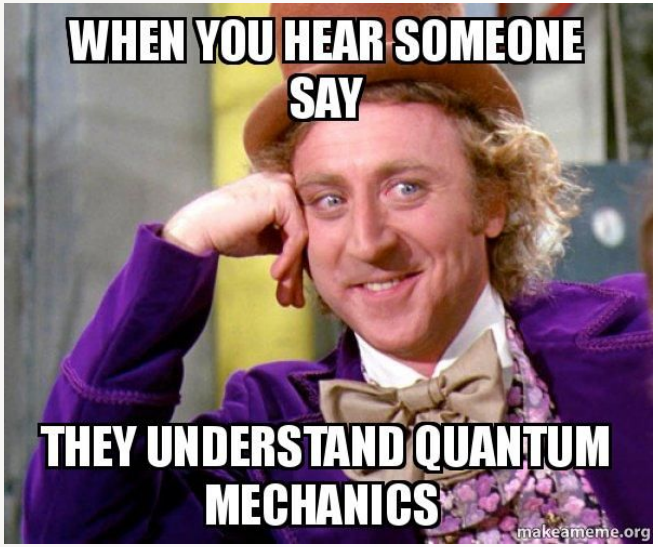
ONE BAD PROGRAMMER CAN EASILY CREATE TWO NEW JOBS A YEAR

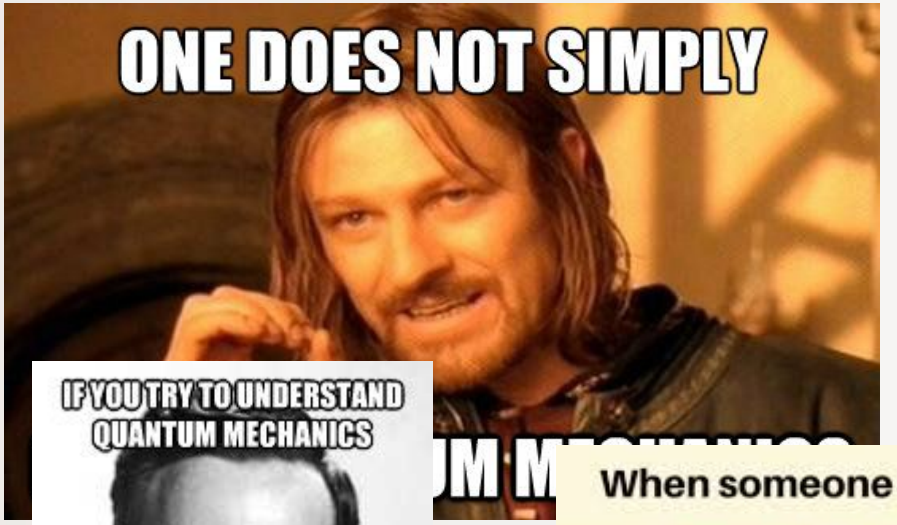


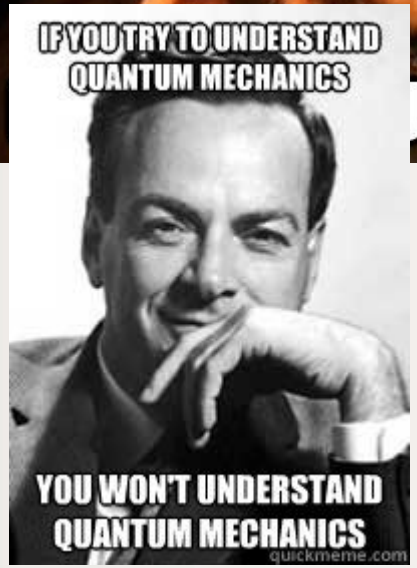
The field **needs skilled software engineers like you** to come in and help build the software for future quantum computers!


ONE DOES NOT SIMPLY
LEARN QUANTUM MECHANICS
quickmeme.com


ONE DOES NOT SIMPLY
LEARN QUANTUM MECHANICS
quickmeme.com

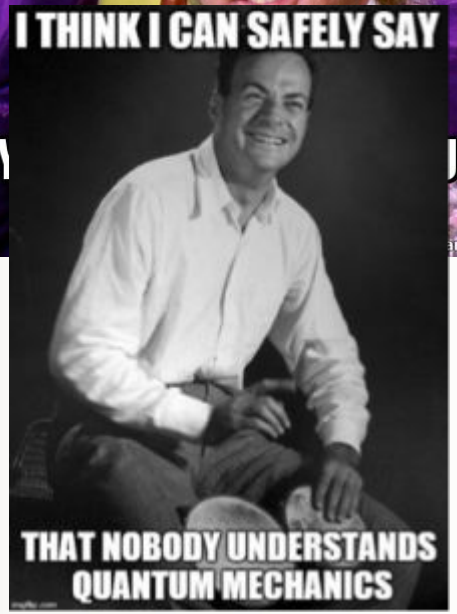

WHEN YOU HEAR SOMEONE SAY
THEY UNDERSTAND QUANTUM MECHANICS
makeameme.org

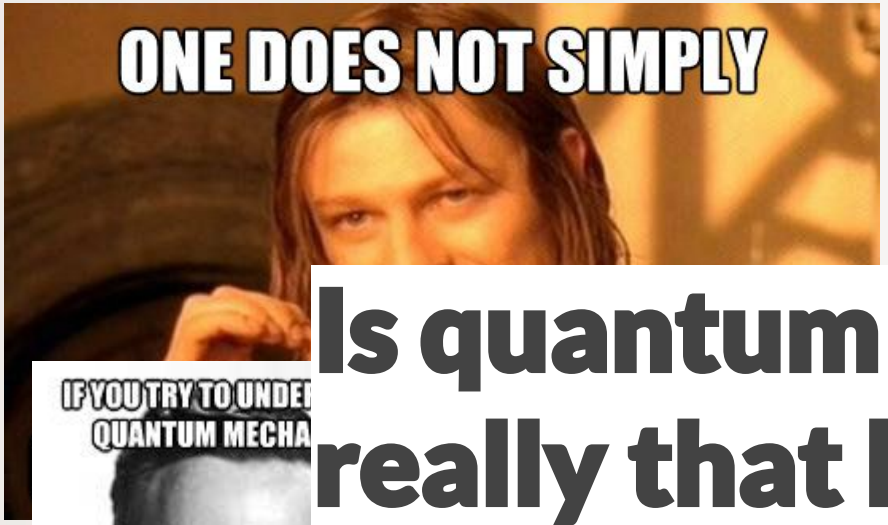

ONE DOES NOT SIMPLY


IF YOU TRY TO UNDERSTAND QUANTUM MECHANICS
YOU WON'T UNDERSTAND QUANTUM MECHANICS
quickmeme.com

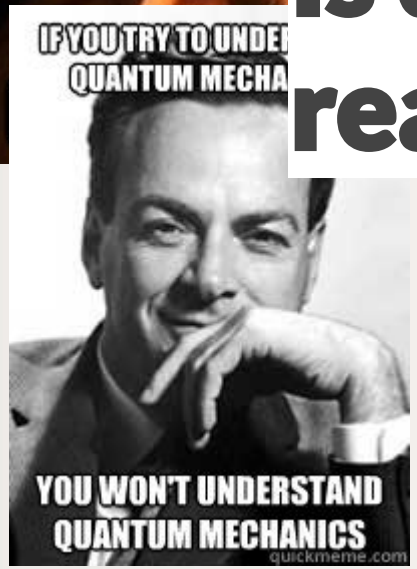

When someone says
Quantum mechanics is easy.

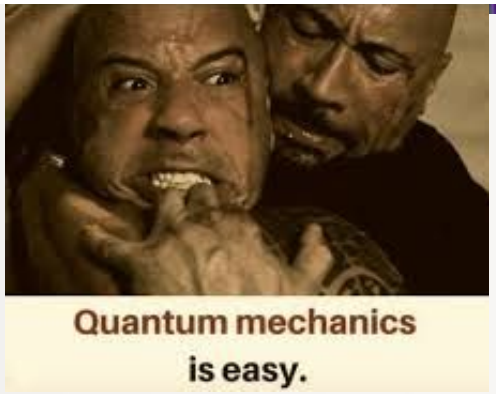

WHEN YOU HEAR SOMEONE SAY


I THINK I CAN SAFELY SAY
THAT NOBODY UNDERSTANDS QUANTUM MECHANICS



Is quantum mechanics really that hard?





You can learn how to program a QPU in less than day!

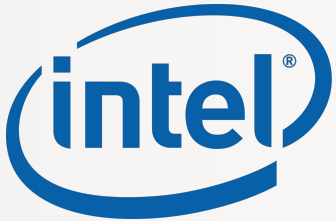
We are hosting a quantum workshop @ FOSDEM tomorrow!

https://qosf.org/fosdem-19-qc-workshop

The elephant in the quantum room

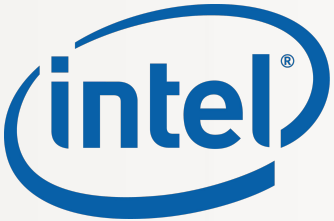
The elephant in the quantum room

Microsoft



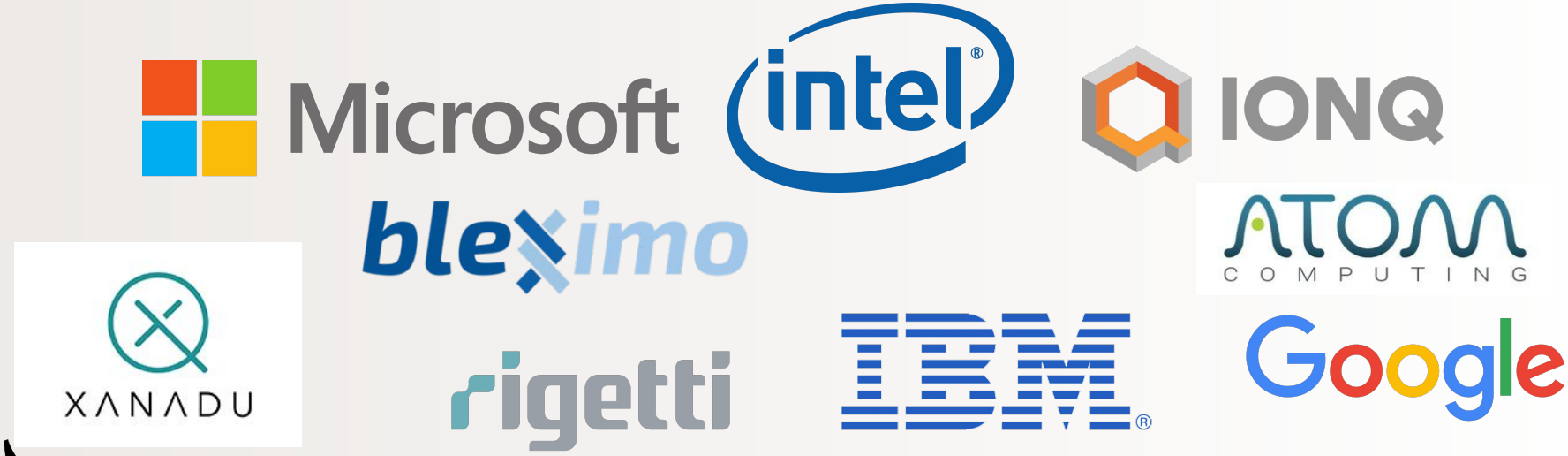
The elephant in the quantum room

Microsoft



IONQ

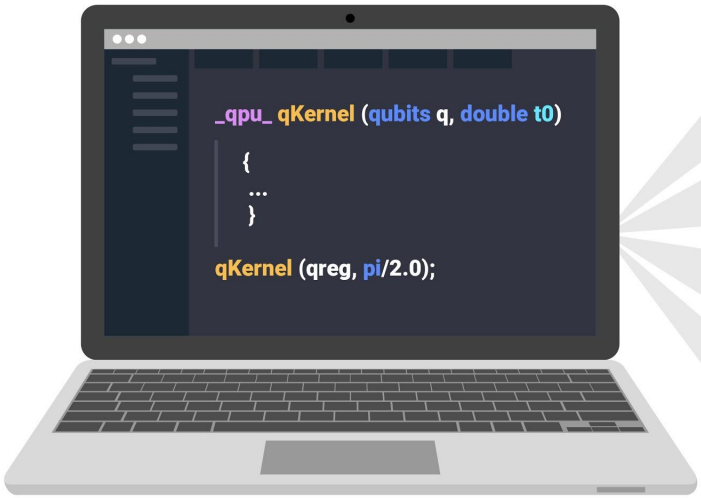
The elephant in the quantum room



standardization?

Unifying frameworks are key

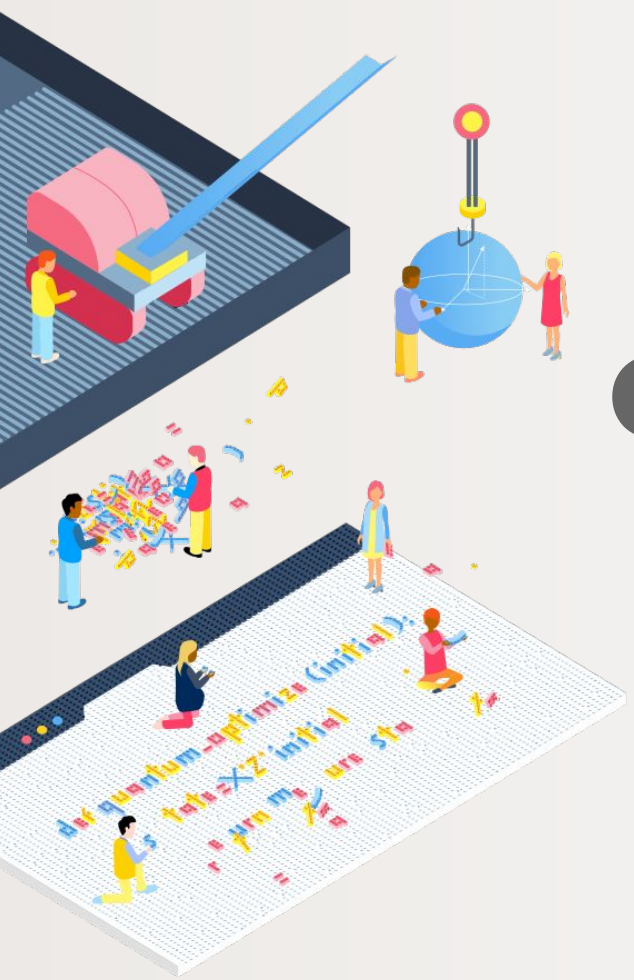
Eclipse XACC



- IBM QPU
- Rigetti QPU
- D-Wave QPU
- TNQVM Simulator

\+ ProjectQ

https://github.com/eclipse/xacc



Quantum Open Source Foundation - QOSF

Quantum Open Source Foundation (QOSF)

"Supporting the development and standardization of open tools for quantum computing."

This is a community effort between academia and industry and

we're looking for advocates, advisors, contributors, partners and funding!

Conclusions

- With NISQ devices becoming available, **quantum software engineering** is becoming increasingly important
- Abundance of academic and commercially-backed open source projects
- **Most projects lack proper documentation** making it hard for newcomers to start with quantum software engineering
- **Lack of public development roadmaps** with most decision being made internally
- **Shortage of quantum compiler projects** (either proprietary or within full-stack libraries)

Conclusions

- With NISQ devices becoming available, **quantum software engineering** is becoming increasingly important
- Abundance of academic and commercially-backed open source projects
- **Most projects lack proper documentation** making it hard for newcomers to start with quantum software engineering
- **Lack of public development roadmaps** with most decision being made internally
- **Shortage of quantum compiler projects** (either proprietary or within full-stack libraries)

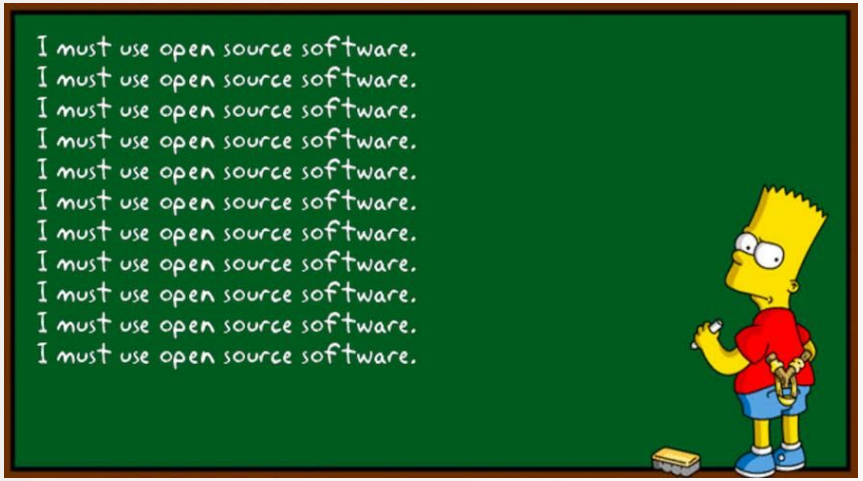
Overall, there is a **lack of standardization** with each hardware player developing their own full- stack solution. Both, users and developers, would profit immensely if **open standards were developed**.

There is a second FOSDEM day on quantum open source tomorrow!

- Community-built quantum software
- Quantum applications built on top of today's frameworks
- Quantum hackathon

Find the Sunday schedule for the quantum dev room here:

https://qosf.org/fosdem/



Thanks for listening and see you tomorrow to hack quantum!

Find the Sunday schedule for the quantum dev room here:
https://qosf.org/fosdem/