



<http://bit.ly/coderefinery-eunis19>



NORDIC E-INFRASTRUCTURE COLLABORATION

NelC's CodeRefinery Project

Bjørn Lindi, Radovan Bast, Thor Wikfeldt

Nordic e-Infrastructure Collaboration/(NTNU, UiT and KTH)

Part 1/3

- The Nordic e-Infrastructure Collaboration
- CodeRefinery
- What are we(CodeRefinery) doing?

Part 2/3

- The CodeRefinery Content and Format
- The background of the CodeRefinery Workshop participant
- Parallels between the research process and program development

Part 3/3

- Open Data and Open Software and the scientific process
- Are the tools used after a while?
- Recommendations/Conclusions



NORDIC E-INFRASTRUCTURE COLLABORATION

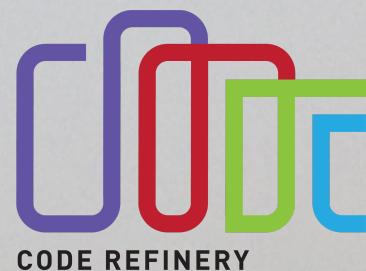
NeIC is hosted by NordForsk, which provides for and facilitates cooperation on research and research infrastructure across the Nordic and Baltic region.

In 2018 NeIC organized 11 project, of which CodeRefinery is one, with a project staff of 158 people from all Nordic countries and Estonia.

NeIC will be the coordinator of the European Open Science Cloud-Nordic and lead a Work Package on FAIR data, starting 1 September 2019.

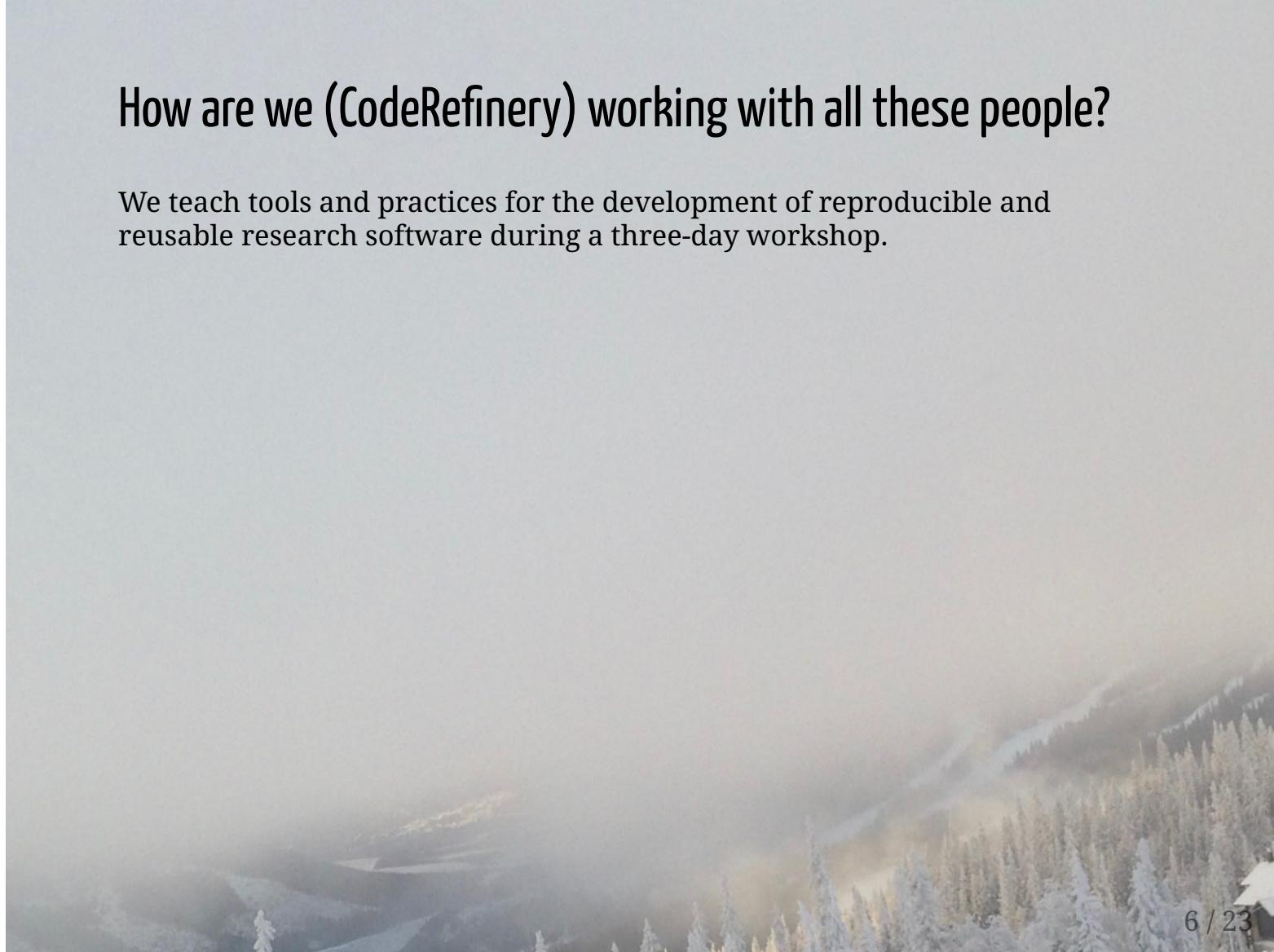
CodeRefinery

We are working with students, researchers, Research Software Engineers from all disciplines and national e-infrastructure partners to advance FAIRness of Software management and development practices so that research groups can collaboratively develop, review, discuss, test, share and reuse their codes.



How are we (CodeRefinery) working with all these people?

We teach tools and practices for the development of reproducible and reusable research software during a three-day workshop.



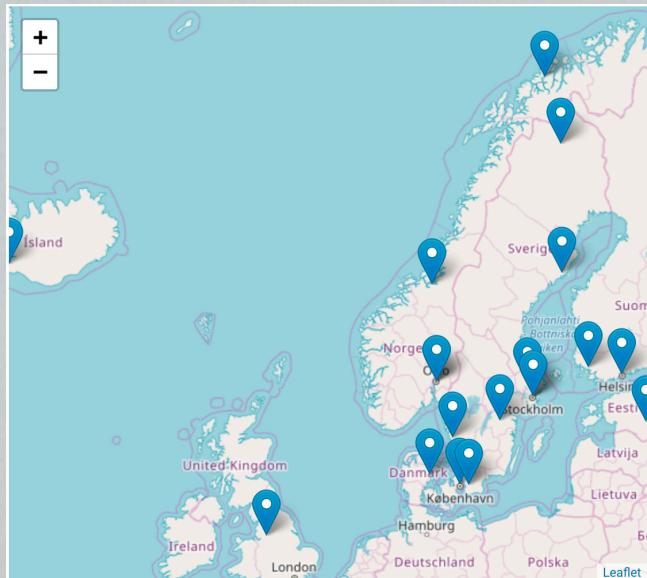
How are we (CodeRefinery) working with all these people?

We teach tools and practices for the development of reproducible and reusable research software during a three-day workshop.

The content of the workshop is

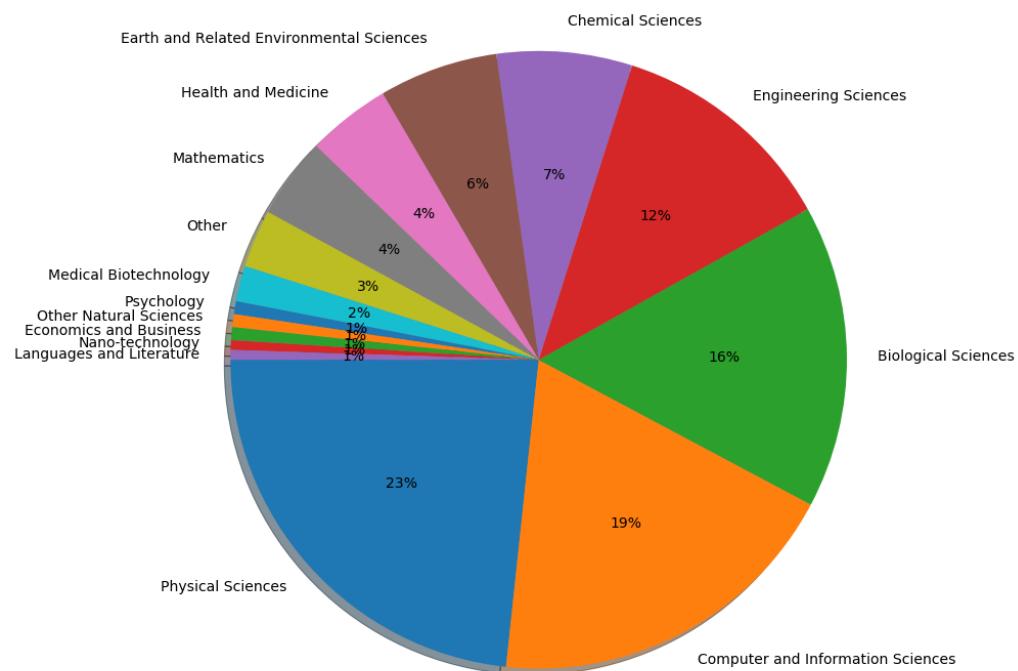
- Basic and collaborative Git
- Git branch design
- Code documentation
- Automated testing
- Jupyterlab/ Jupyter Notebooks
- Integrated Development Environments
- Building portable code with CMake
- Social coding and open software
- Modular code development
- Reproducible research

The CodeRefinery Workshops have been arranged all over the Nordic region

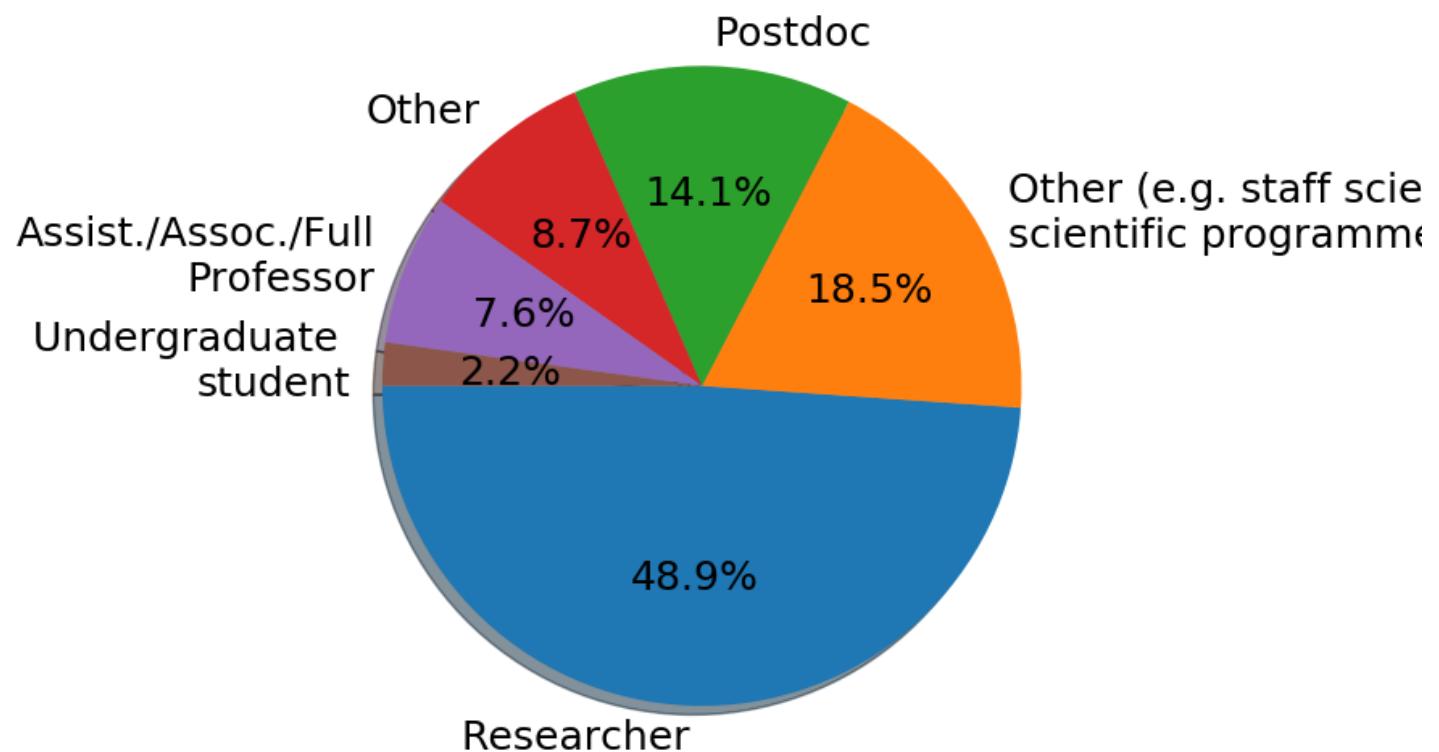


- <https://coderefinery.org>
- <https://coderefinery.org/workshops/>

The Workshops participants comes from many disciplines

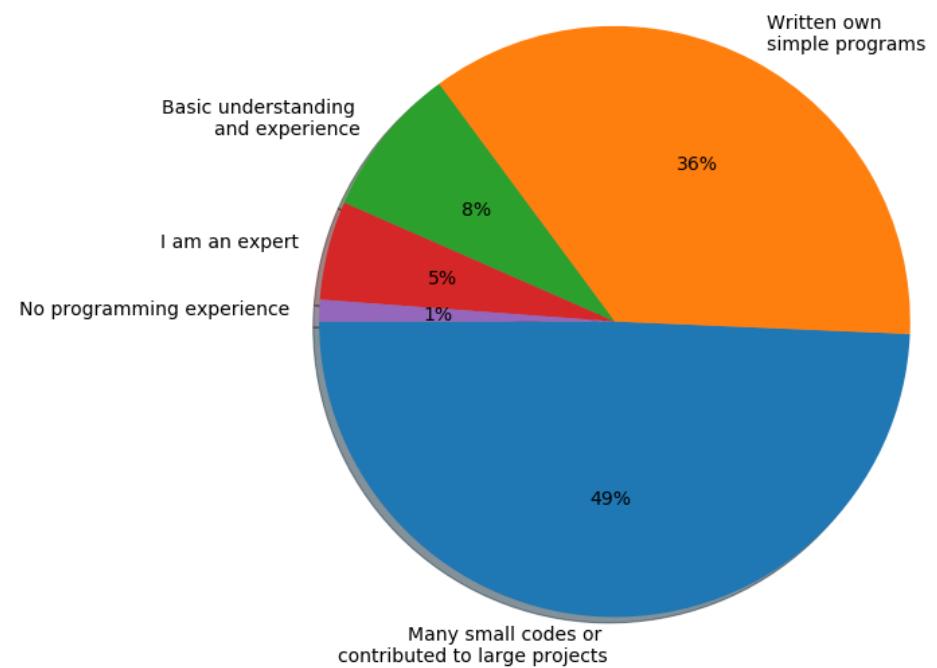


The Workshops participants have a varied background



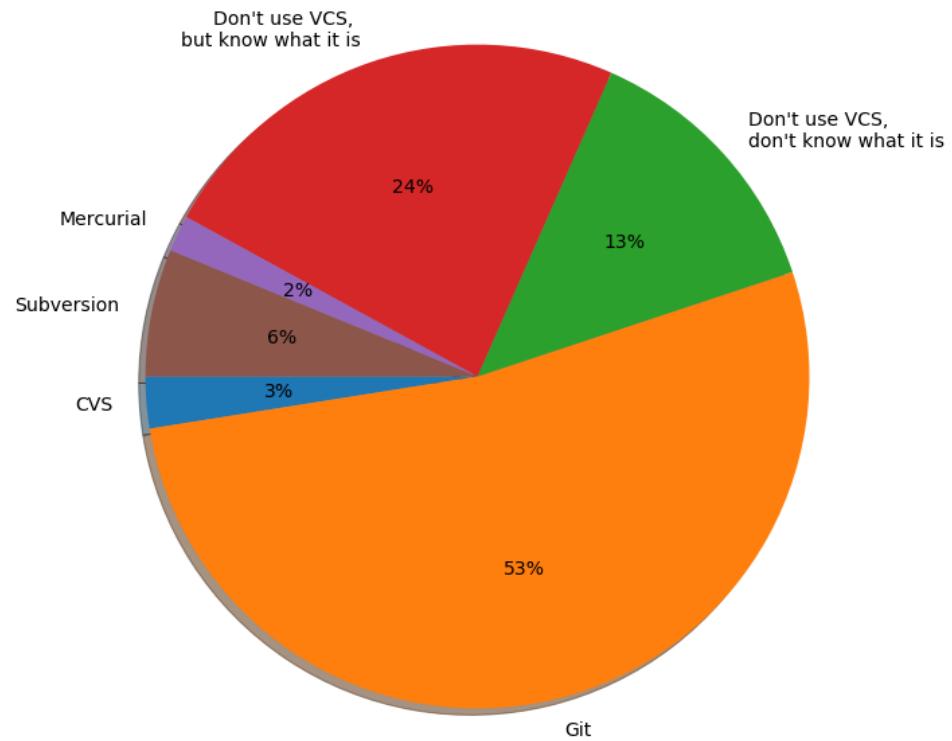
Programming experience is very varied

How would you describe your programming experience?



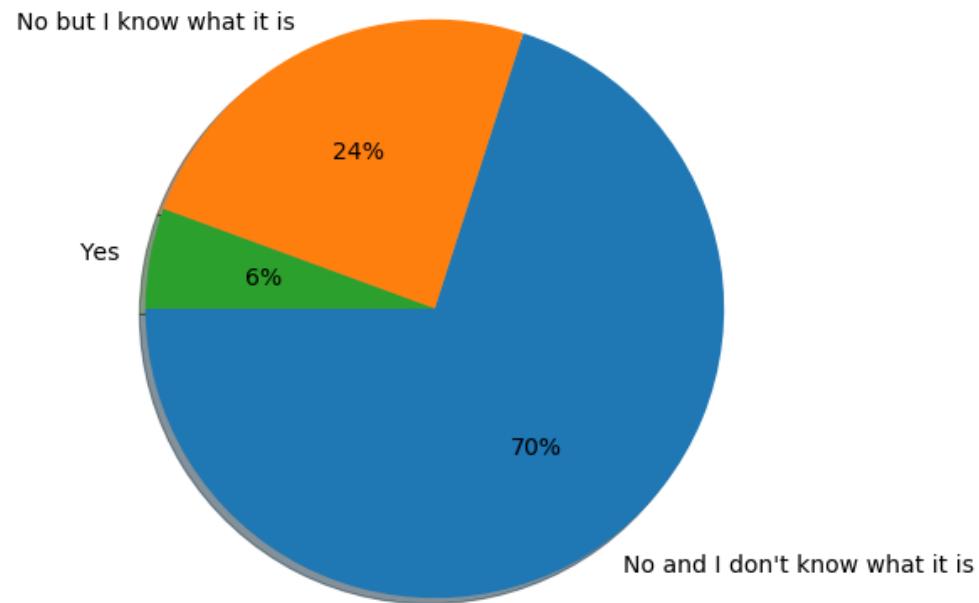
The Workshop participants exposure to version control

Are you using version control? If yes, which?



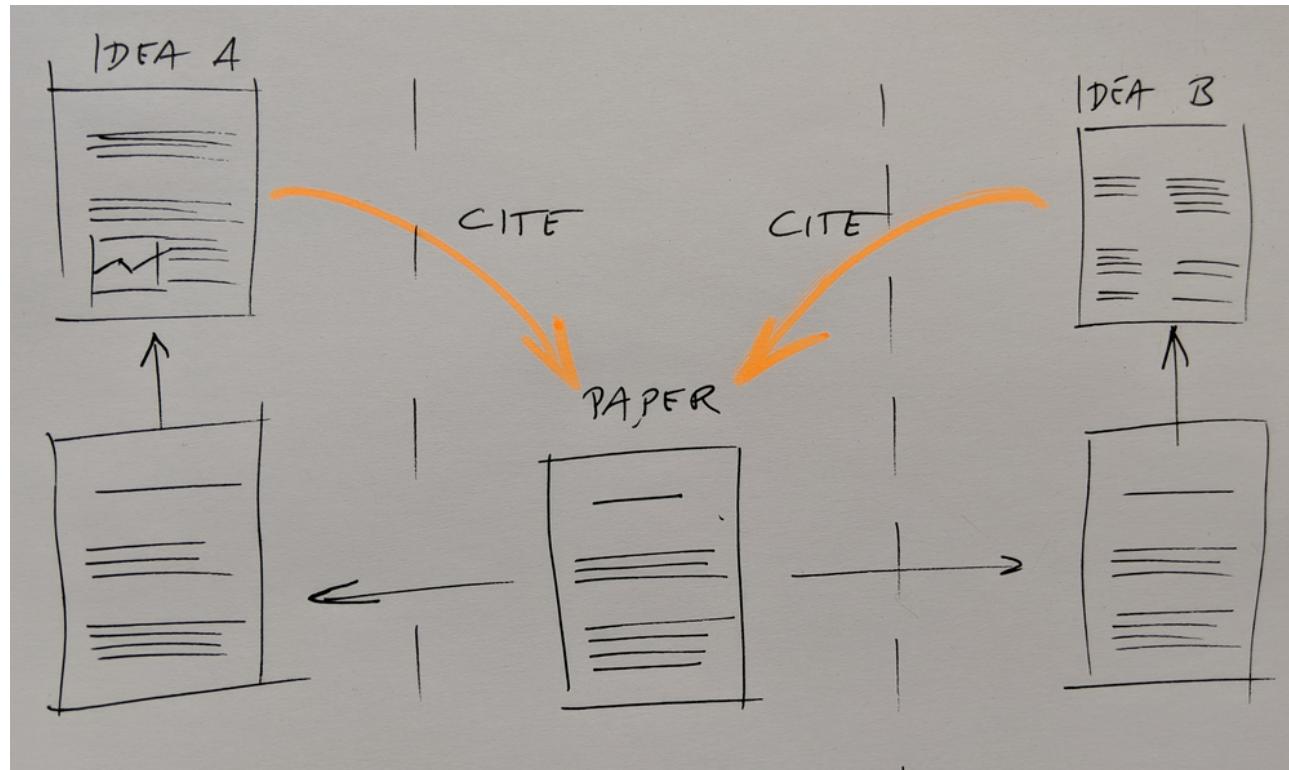
The Workshop participants experience with code review

Are you employing code review in your programming project(s)?

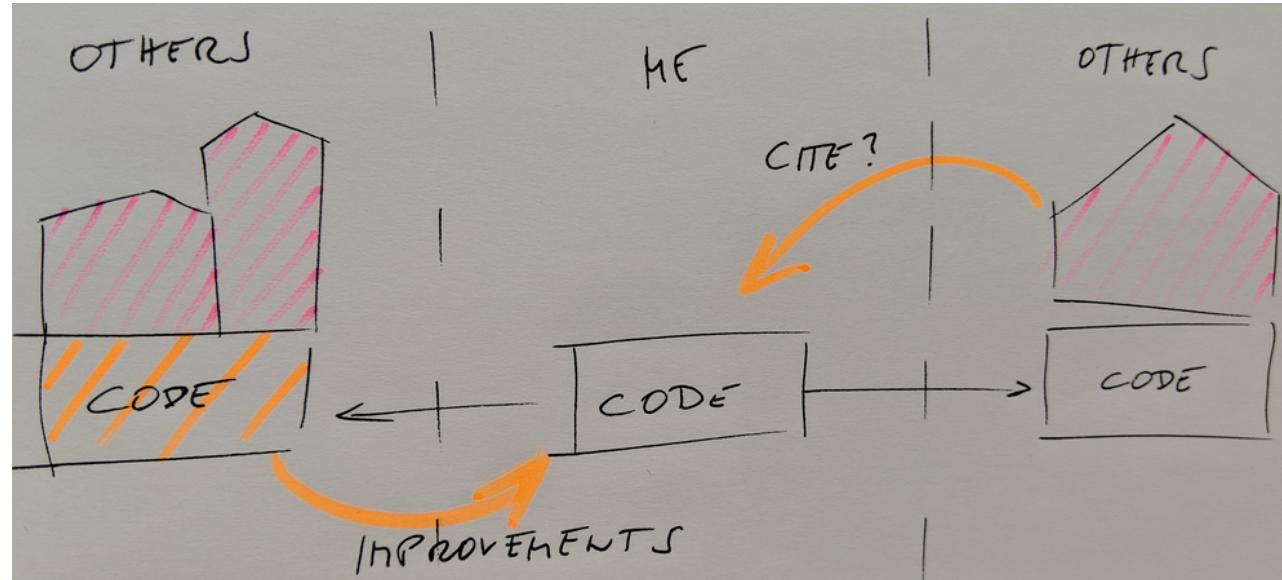


The Workshop participants experience with automated testing

Peer review of paper is a established principle



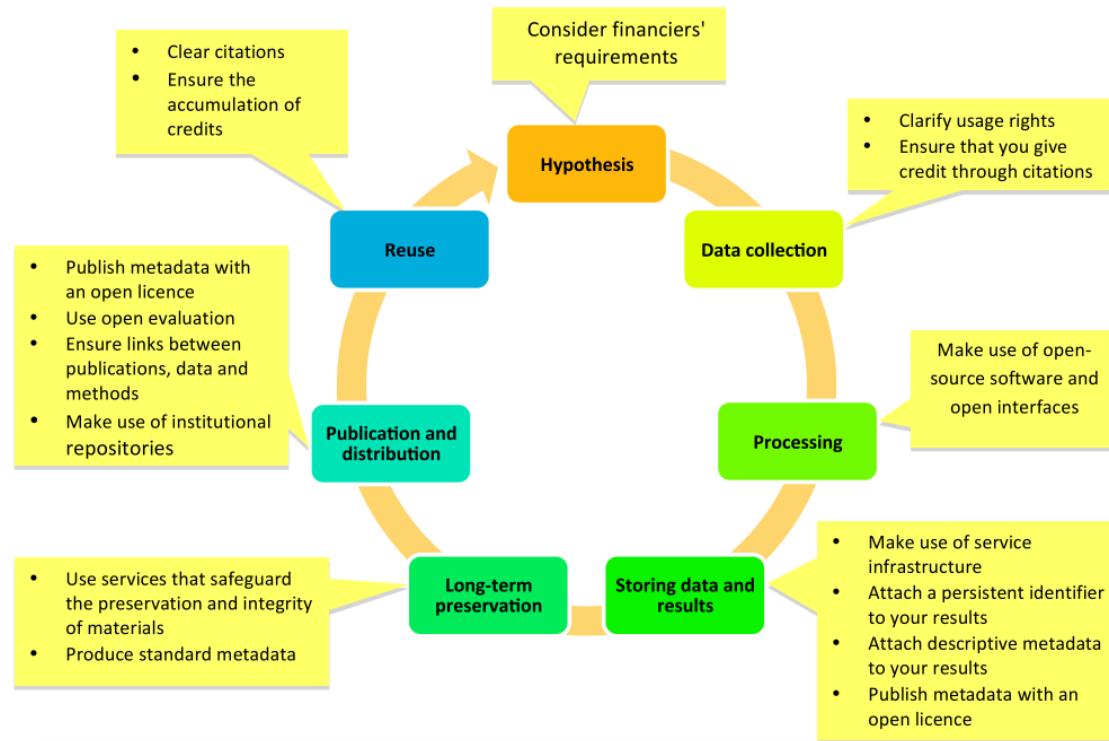
Peer review of code is very close, but much less common



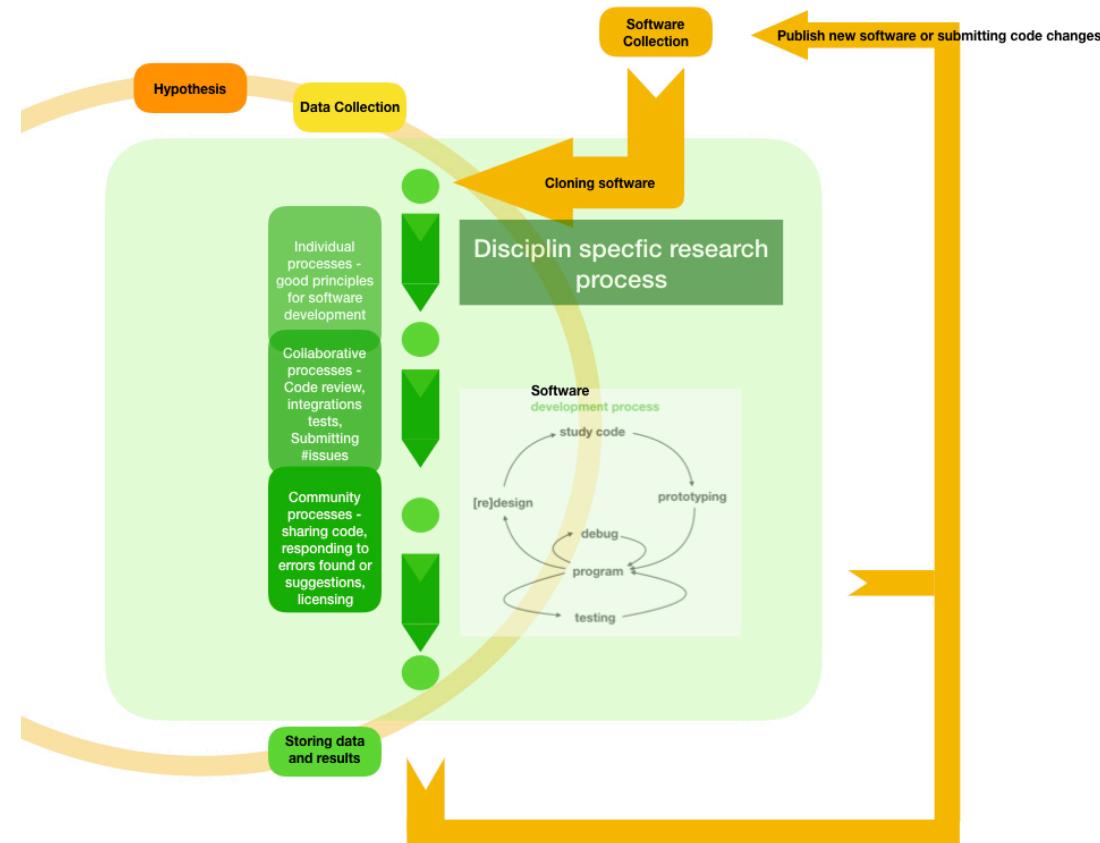
Simulations and analysis with untested software do not constitute science

"Before relying on a new experimental device, an experimental scientist always establishes its accuracy. A new detector is calibrated when the scientist observes its responses to known input signals. The results of this calibration are compared against the expected response. **An experimental scientist would never conduct an experiment with uncalibrated detectors** - that would be unscientific. So too, simulations and analysis with untested software do not constitute science." (copied from [Testing and Continuous Integration with Python](#), created by Kathryn Huff, see also the Testing chapter in [Effective Computation In Physics](#) by Anthony Scopatz and Kathryn Huff)

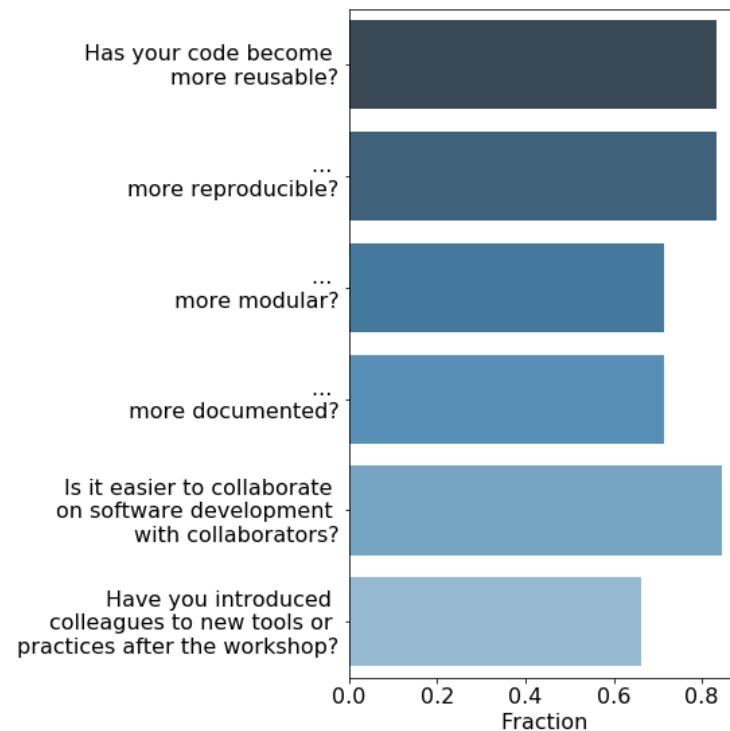
The Open Science Research life cycle



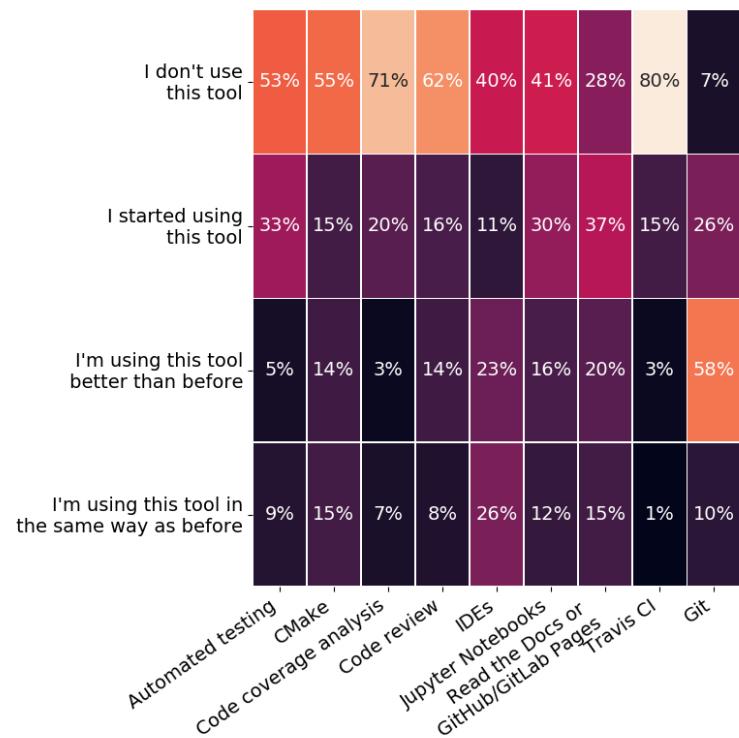
A closer look at the Process part of the Research life cycle



Has the Workshops participants' software become more useable?

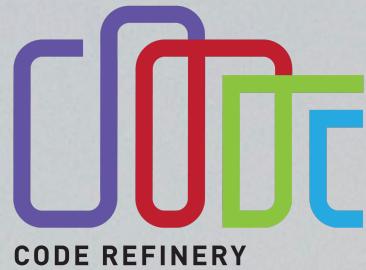


Are the Workshops participants using the different tools CodeRefinery has introduced?



CodeRefinery recommendations

- Encourage students/staff to take training.
- Support training activities; take part in the Software Carpentries and CodeRefinery; contribute to a network of instructors.
- Encourage sharing and open source.
- Promote FAIR-principles, not only by talks and presentations - use the principles in pilot projects.
- FAIR principles apply both to humans and machines - data and software must be FAIR for both.



<https://www.coderefinery.org>

- This talk is presented with cicero:https://cicero.readthedocs.io/en/latest/getting_started.html
- PDF and source is on https://github.com/blindij/talk_eunis19