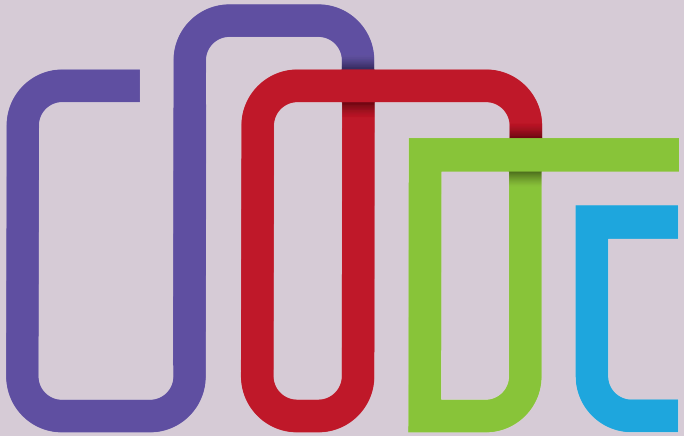


The CodeRefinery project for training in research software engineering

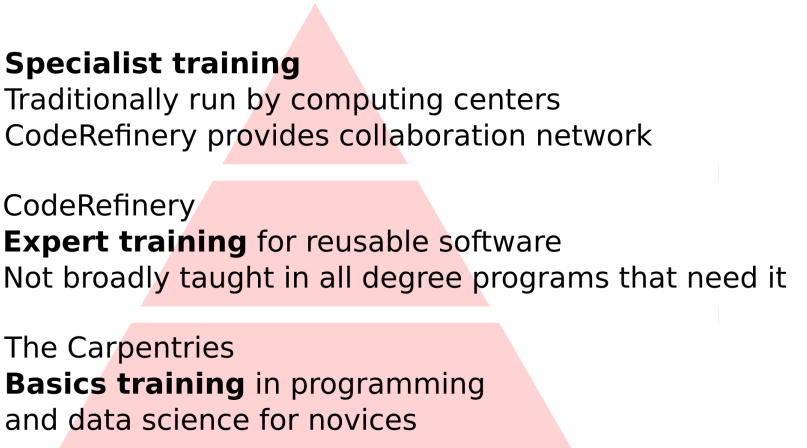
Johan Hellsvik, PDC center for high performance computing, KTH Royal Institute of Technology, Sweden

Talk at The HPC Training Ecosystem in Europe workshop – October 2024

Team and project: coderefinery.org

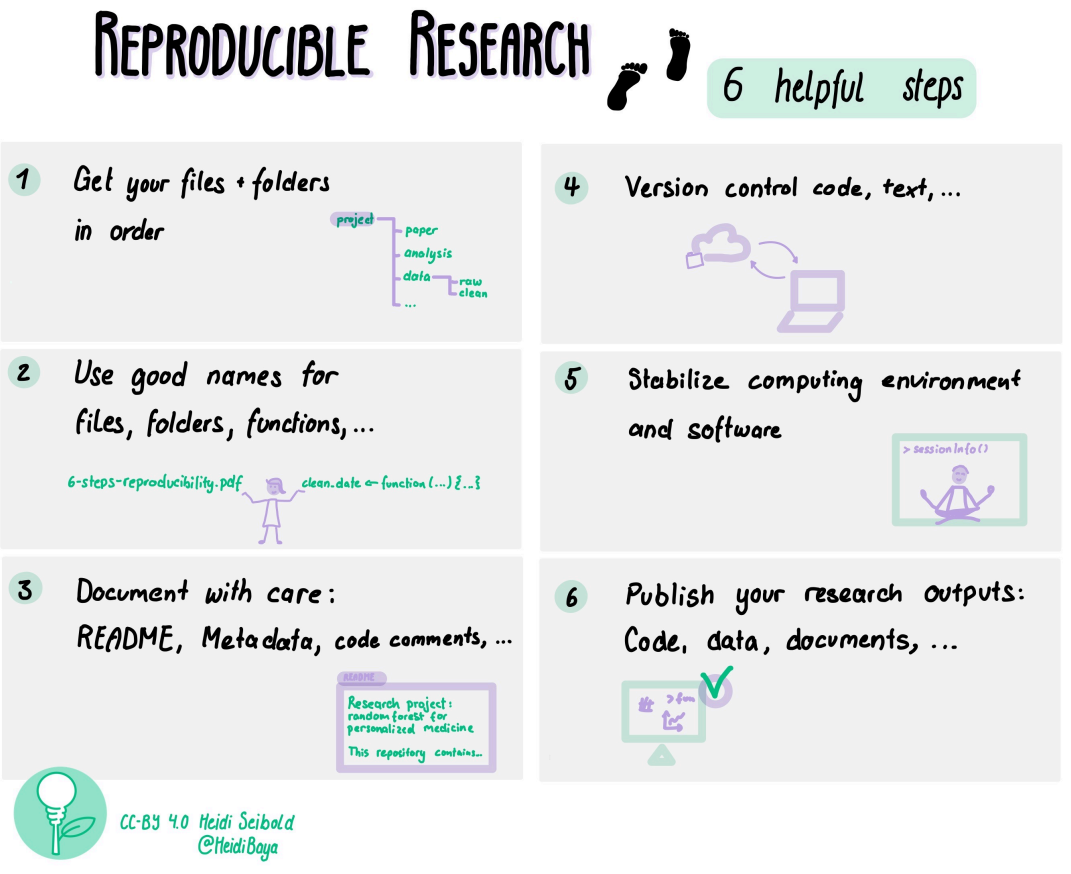
What we are

- A hub for FAIR research software practices
- Since 2016, now phase 3 until 2025
- Currently funded by NeIC
- Training network
- Community



What we do

- We teach and co-organize
- Share lessons, video recordings, manuals
- All open source



[Heidi Seibold, CC-BY 4.0, https://twitter.com/HeidiBaya/status/1579385587865649153]

Similar projects: UNIVERSE-HPC, DIGITAL RESEARCH ACADEMY, INTERSECT, and probably many more ...

Available lesson material

- Introduction to version control: Git and GitHub for own projects
- Collaborative version control: Branching, pull/merge requests, forks, and collaboration.
- Reproducible research: Reproducible dependencies, environments, and computational steps.
- Social coding and open software: Software and data licensing and software citation.
- How to document your research software
- Reusable and reproducible Jupyter notebooks
- Automated testing: Motivation, test design, and tools.
- Modular code development: Organizing projects as they grow from one screen-full to larger.

Tested in 10 online and 29 in-person workshops

- We reach over 500 persons/year
- Over 30 instructors/speakers
- Over 100 helpers/ exercise leaders

Lessons

We use Sphinx/sphinx-lesson to build our lessons from Markdown.

Testing in a nutshell

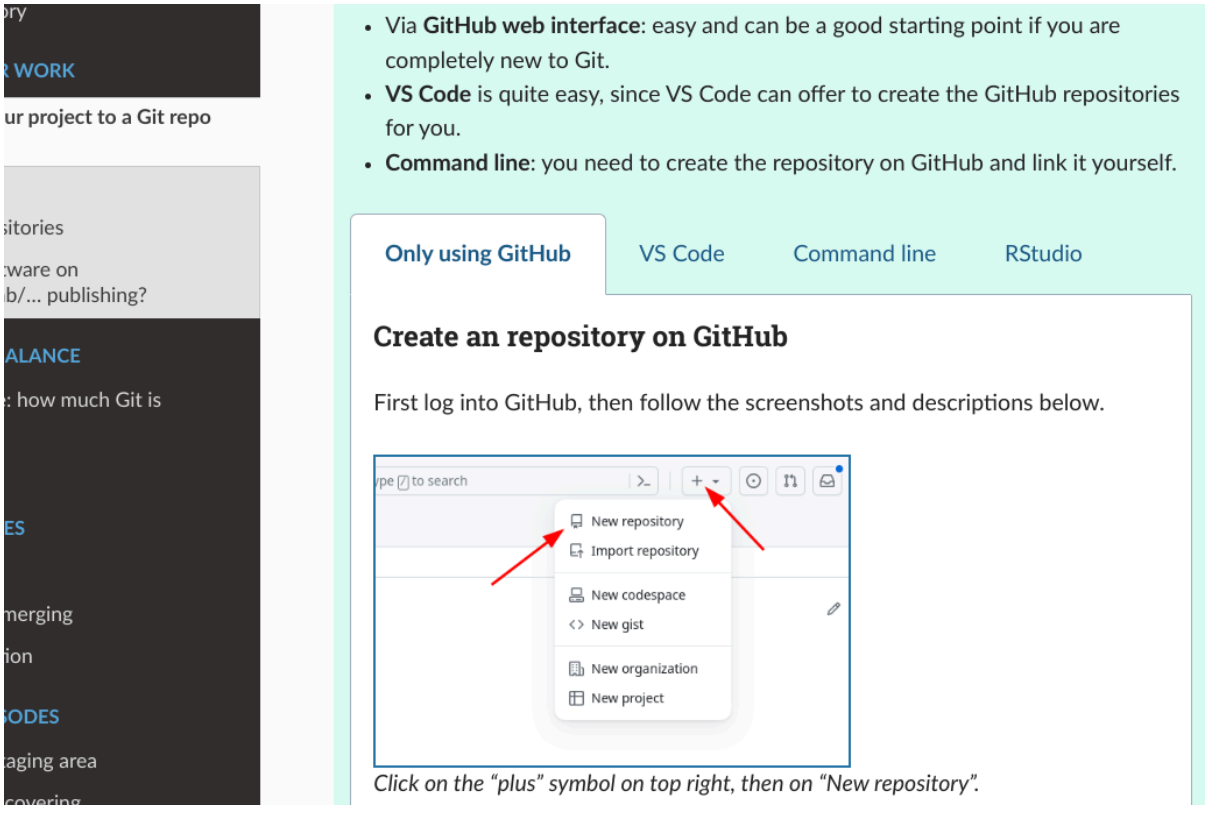
In software tests, expected results are compared with observed results in order to establish accuracy. Why are we not comparing directly all digits with the expected result?:

Python | C++ | R | Julia | Fortran

```python
def fahrenheit_to_celsius(temp_f):
    """Converts temperature in Fahrenheit
    to Celsius.
    """
    temp_c = (temp_f - 32.0) * (5.0/9.0)
    return temp_c

# This is the test function: `assert` raises an error if something
# is wrong.
def test_fahrenheit_to_celsius():
    temp_c = fahrenheit_to_celsius(temp_f=100.0)
    expected_result = 37.777777
    assert abs(temp_c - expected_result) < 1.0e-6
```

Another example: Git lesson



You can try our [lesson template]

Collaborative document: Markdown

- Interactive, anonymous, parallel, async
- New question every 1-2 minutes!
- ASCII-graph feedback

```
### [Exercise: Practice creating commits and branches](https://coderefinery.github.io/git-
intro/commits/#exercise)

How is it going?
- done: oooooooooooooo
- not trying: oo
- had problems: ooo
- everything was good: oooooooo

- When committing, there is an option to make a branch. How is that different than that we n
started with creating a new branch?
  - It's essentially the same. Creating a branch first is just closer to what you would norma
do if you don't edit on the web-interface but on your local machine.

- In step (6) Compare the brances, where do I access the compare changes view through the UI
not through the URL)?
  - On GitHub you can add a "/compare" behind the URL and from there it is easier to compare
branches and commits. We need to add this to the solution
    - I opened an issue in the lesson repository: https://github.com/coderefinery/git-
intro/issues/457
```

We publish Q&A for each workshop: [Example]

Connection to high-performance computing

- Many partners with HPC involvement
- Aalto University very active in inviting others to join HPC kickstart
- Tuesday Tools & Techniques for High Performance Computing
- ENCCS instructor training -> Materials based on CR train-the-trainer
- NRIS (Norway) and LUMI (EuroHPC-JU) have adopted our training approach and material as template
- We started sharing tools across borders at https://github.com/nordichpc

What we have learned

About motivating/teaching

- Teaching isn't a lecture anymore. It's more like a live TV production, which *can* be as interactive as people in a room.
- Co-teaching is a great way to onboard, get better quality, and reduce stress
- Good enough practices better than perfect practices not applied
- Instead of "good for others": "good for your future you and as side effect good for others"

What we have learned

About scaling

- "bring your own classroom" seems to be a way to scale
- Installation instructions and on-boarding become more important
- We don't "see" classrooms -> feedback mechanism in Q&A doc
- Make exercises longer to give classrooms the chance to interact

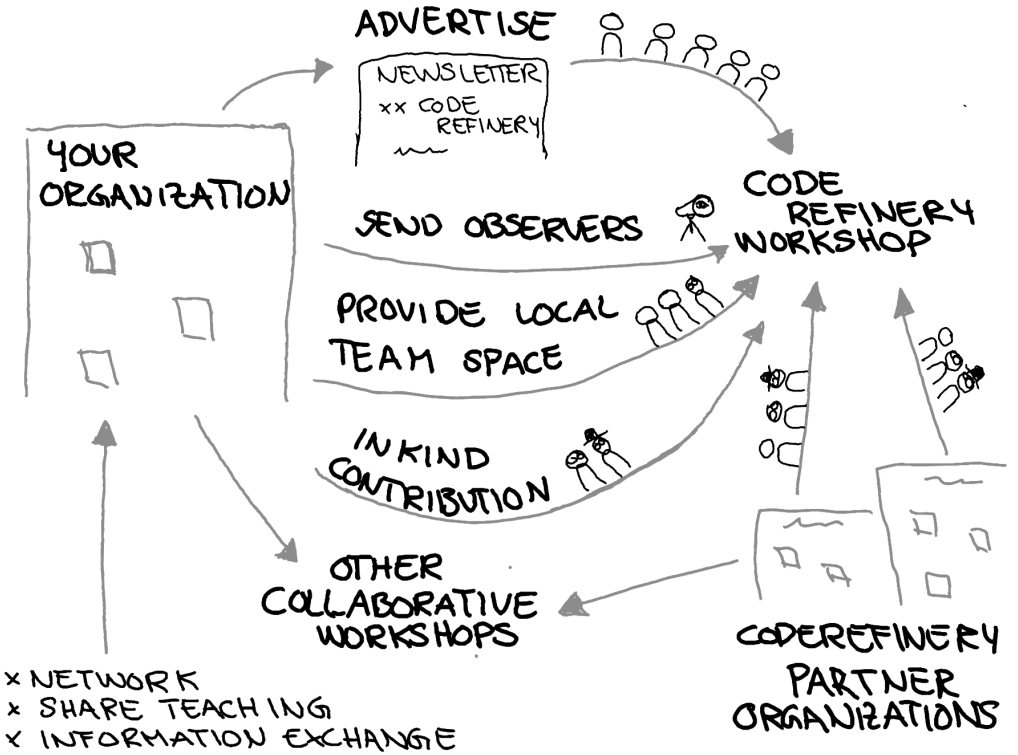
Future: Community project

- Communicate value for volunteers and organizations
- Research groups send their students to us instead of creating isolated material
- More collaboration with similar projects ("helper exchange program")
- Governance is community-driven

Teaching format

- Continue large-scale workshops
- Support local events
- More asynchronous content coupled with online events ("flipped classroom approach")

How you or your organization can participate



- Join our next workshop autumn 2024; follow our newsletter to get involved
- Tell your students and researchers about it
- Send one or more exercise teams or join as observer
- Use our material and give feedback

What is in it for you?

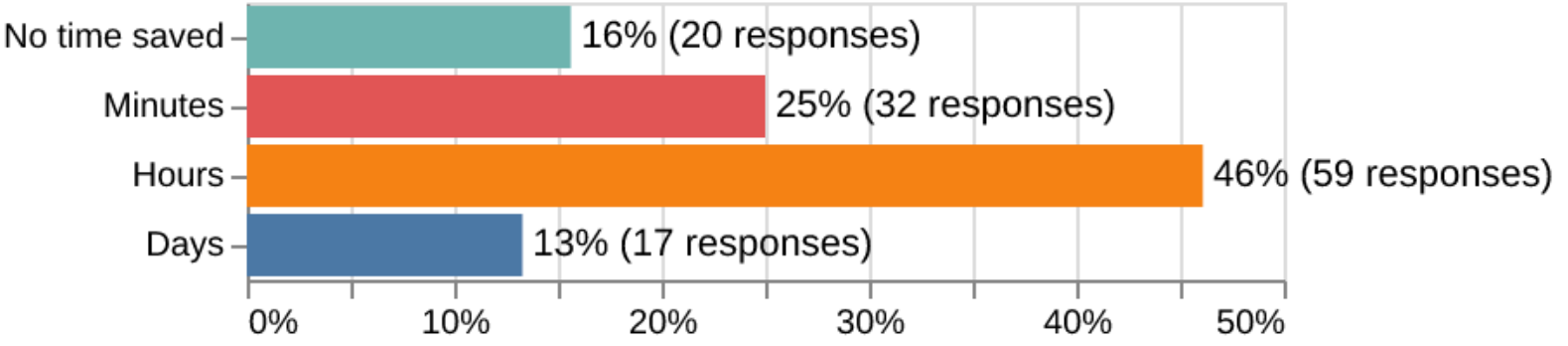
- Joining is easier than organizing: It is easier to bring 10% to an event than to organize the 100% yourself
- Material exchange: let's not reinvent the wheel
- Train-the-trainer: we can help you to get started
- Community as test-bed: let's try out new ideas together

We try to make it easier to join

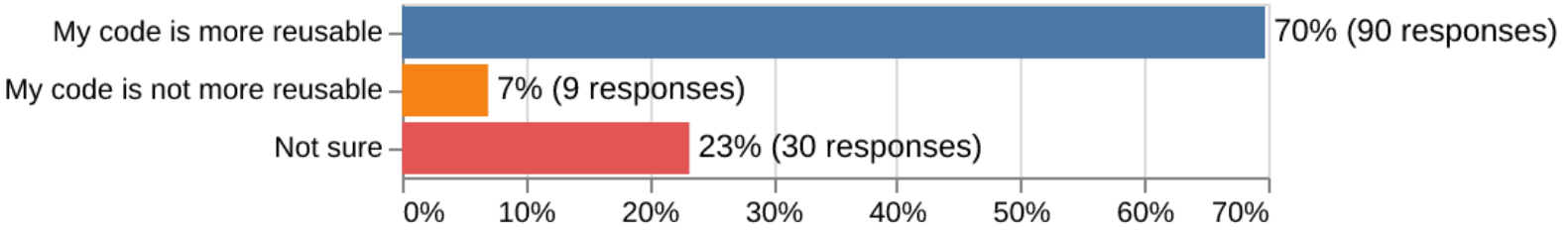
- Chat with us: https://coderefinery.zulipchat.com (ask questions about coding or learn about new tools)
- Onboarding manual
- Blog
- Newsletter and chat digest
- X/Twitter
- Mastodon
- Support e-mail

Results from post-workshop survey 2024

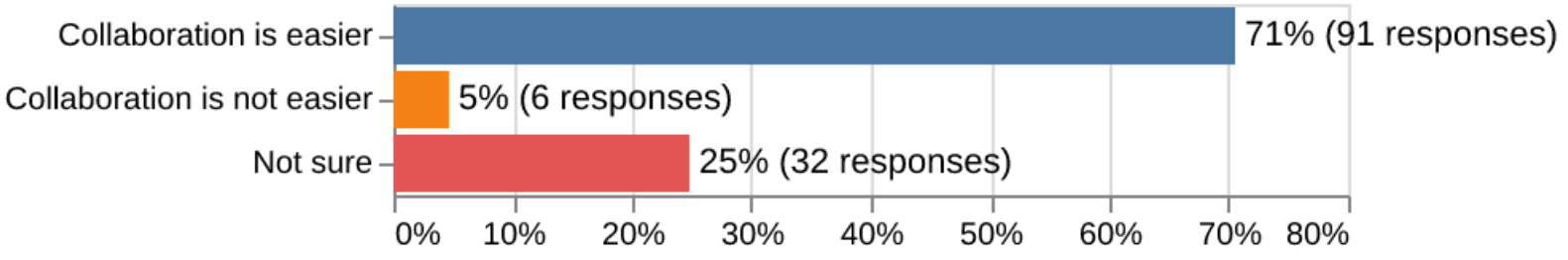
- Sent out to workshop participants from 2022 and 2023
- 129 answers



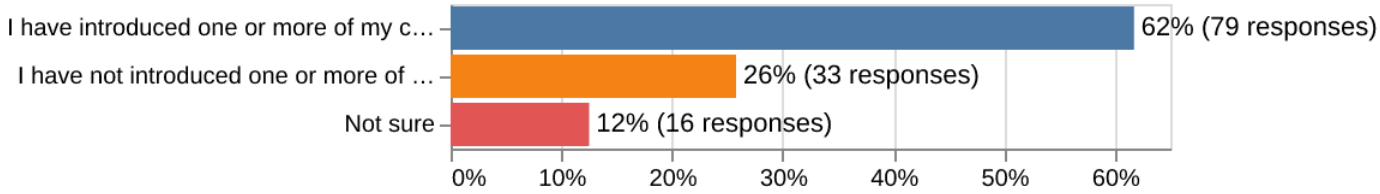
In your estimate, how much time per month have you saved as a result of attending a CodeRefinery workshop?



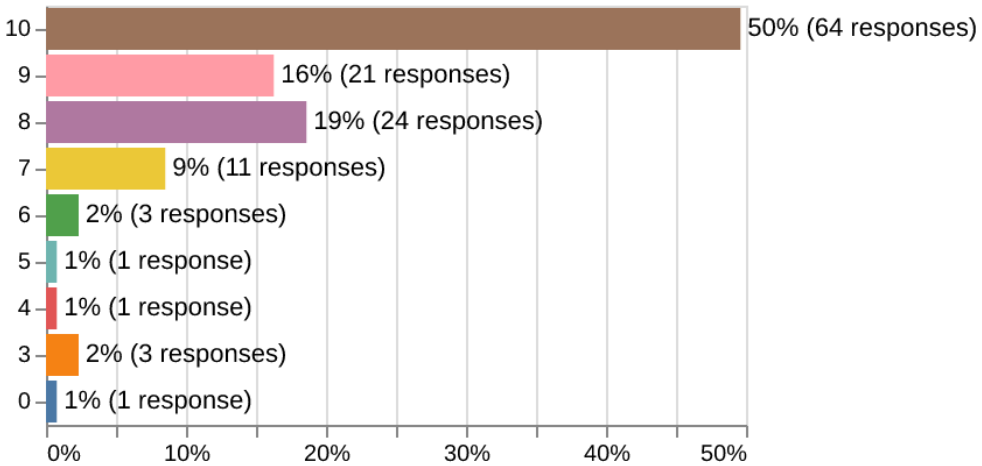
After attending the workshop, would you judge your code to be more reusable or not more reusable?



After attending the workshop, has it become easier or not for you to collaborate on software development with your colleagues and collaborators?



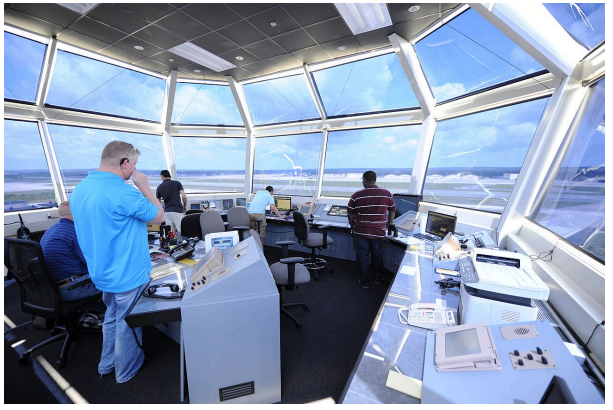
Have you introduced one or more of your colleagues to new tools or practices as a result of the workshop?



How likely is it that you would recommend CodeRefinery workshop to a friend or colleague?

0 means definitely not. 10 means definitely yes.

Collaboration across funding borders



0.9 FTE (2 persons) + 10 persons in-kind + volunteers

Co-advertize and co-organize with us

Thank you for your attention!

Credits and license

Text

- All text: CodeRefinery project, CC-BY 4.0

Images

- Slide 3: H. Seibold, "6 helpful steps for reproducible research", CC-BY 4.0
- Slide 12: S. Wittke
- Slide 18: ATC tower, P. R. Miller, CC-BY 2.0
- Slide 18: Monitor setup, R. Darst
- Slide 18: Logos, (c) respective organizations
- All other images: CodeRefinery project, CC-BY 4.0