CODE REFINERY

# Experiences from teaching basic RSE skills with CodeRefinery

Samantha Wittke, CSC - IT Center for Science, Finland

Radovan Bast, UiT - The Arctic University of Norway

RSECon 2023, Swansea

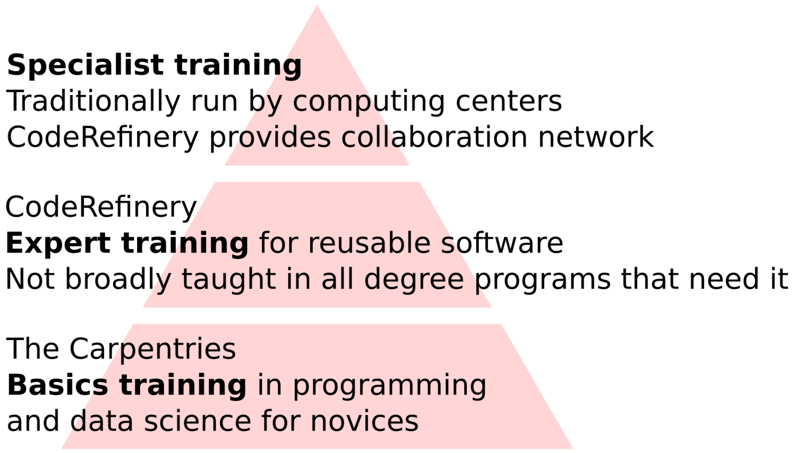
# Team and project: codérefinery.org

## 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

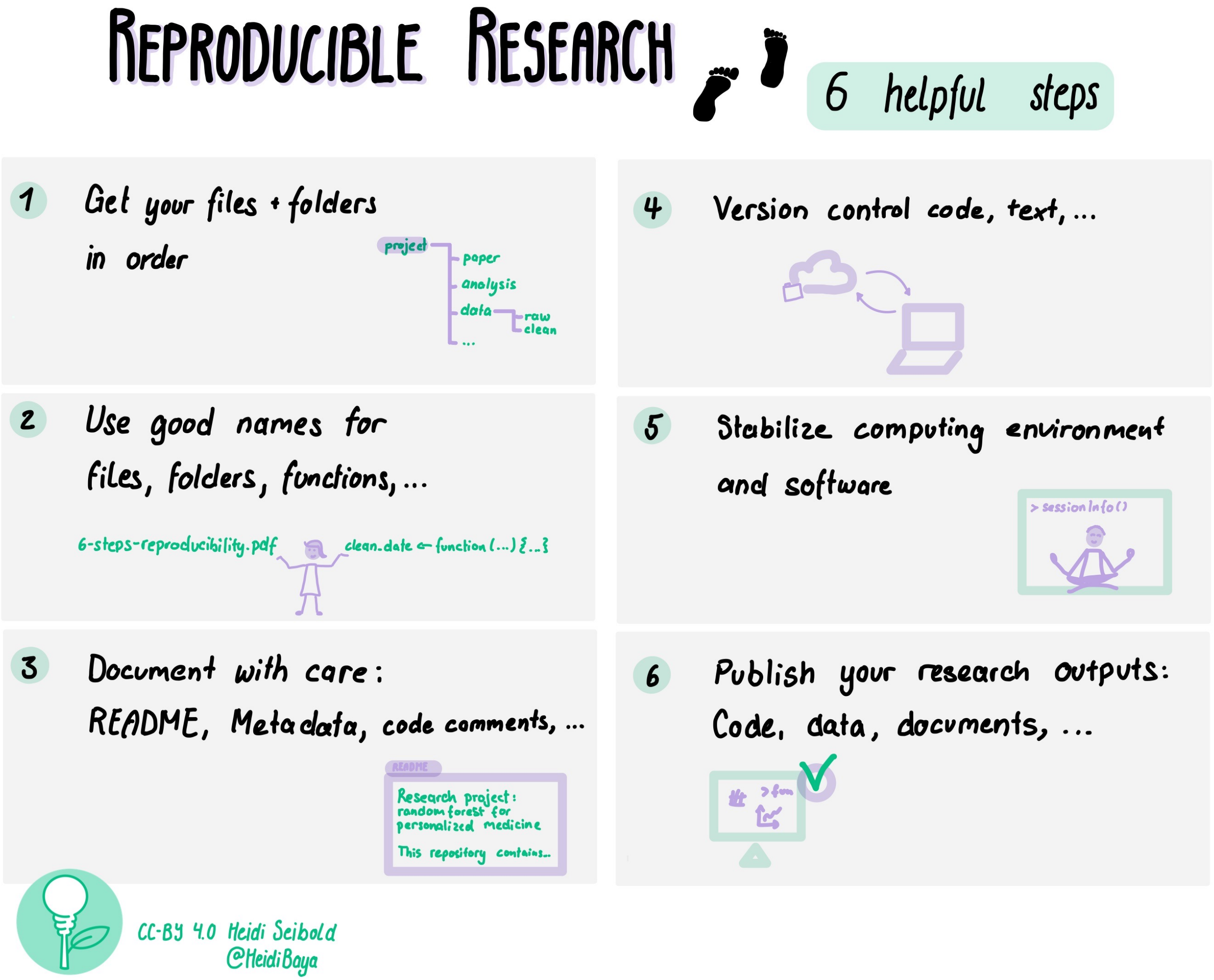
- We teach and co-organize
- Share lessons, video recordings, manuals
- All open source



**Specialist training**
Traditionally run by computing centers
CodeRefinery provides collaboration network

CodeRefinery
**Expert training** for reusable software
Not broadly taught in all degree programs that need it

The Carpentries
**Basics training** in programming
and data science for novices



# Reproducible Research

6 helpful steps

1 Get your files + folders in order

project
paper
analysis
data
raw
clean
...

2 Use good names for files, folders, functions, ...

6-steps-reproducibility.pdf
clean.date <- function(...){...}

3 Document with care: README, Metadata, code comments, ...

README
Research project: random forest for personalized medicine
This repository contains...

4 Version control code, text, ...

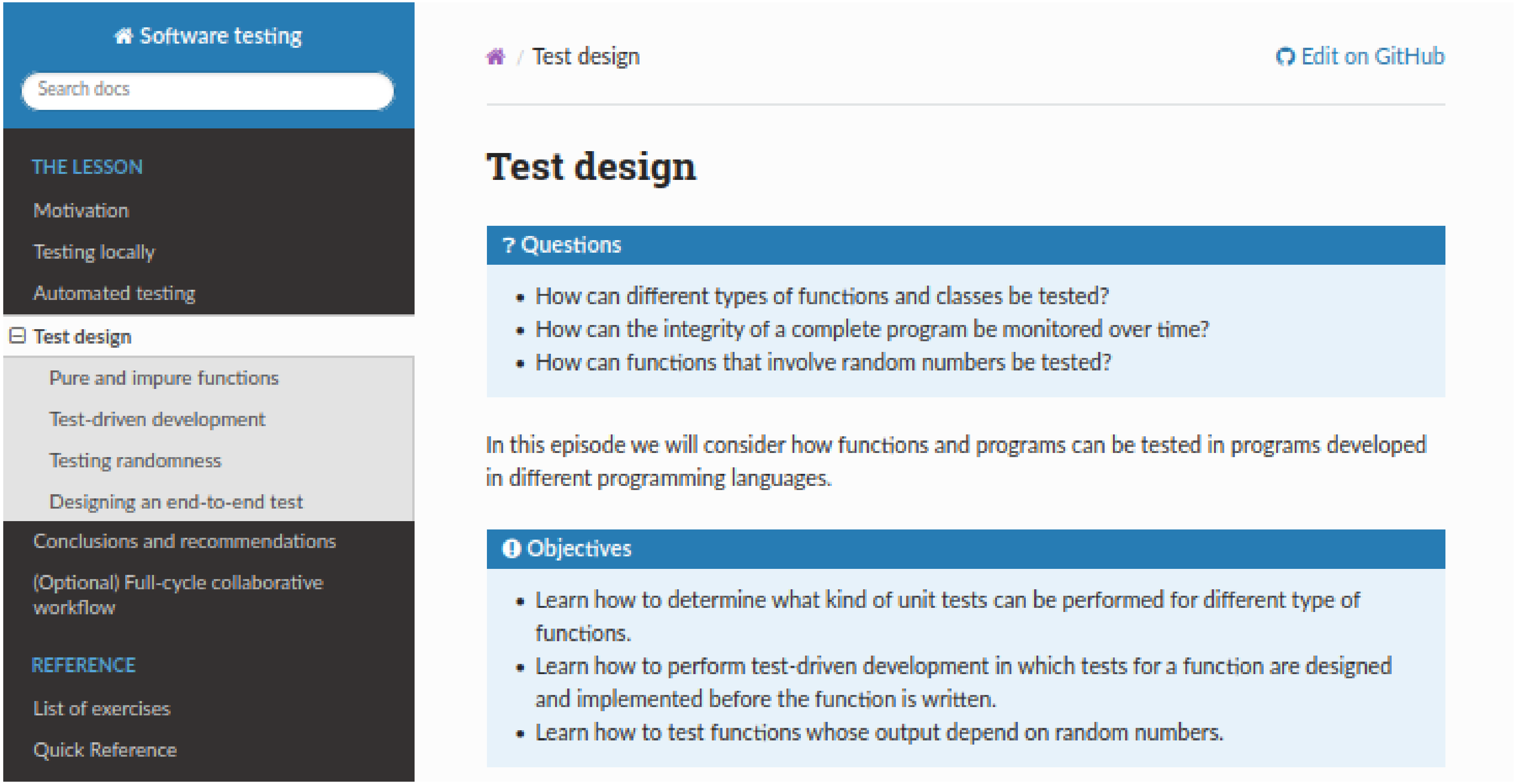
5 Stabilize computing environment and software

6 Publish your research outputs: Code, data, documents, ...

CC-BY 4.0 Heidi Seibold @HeidiBaya

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

# Lessons



## Test design

**? Questions**

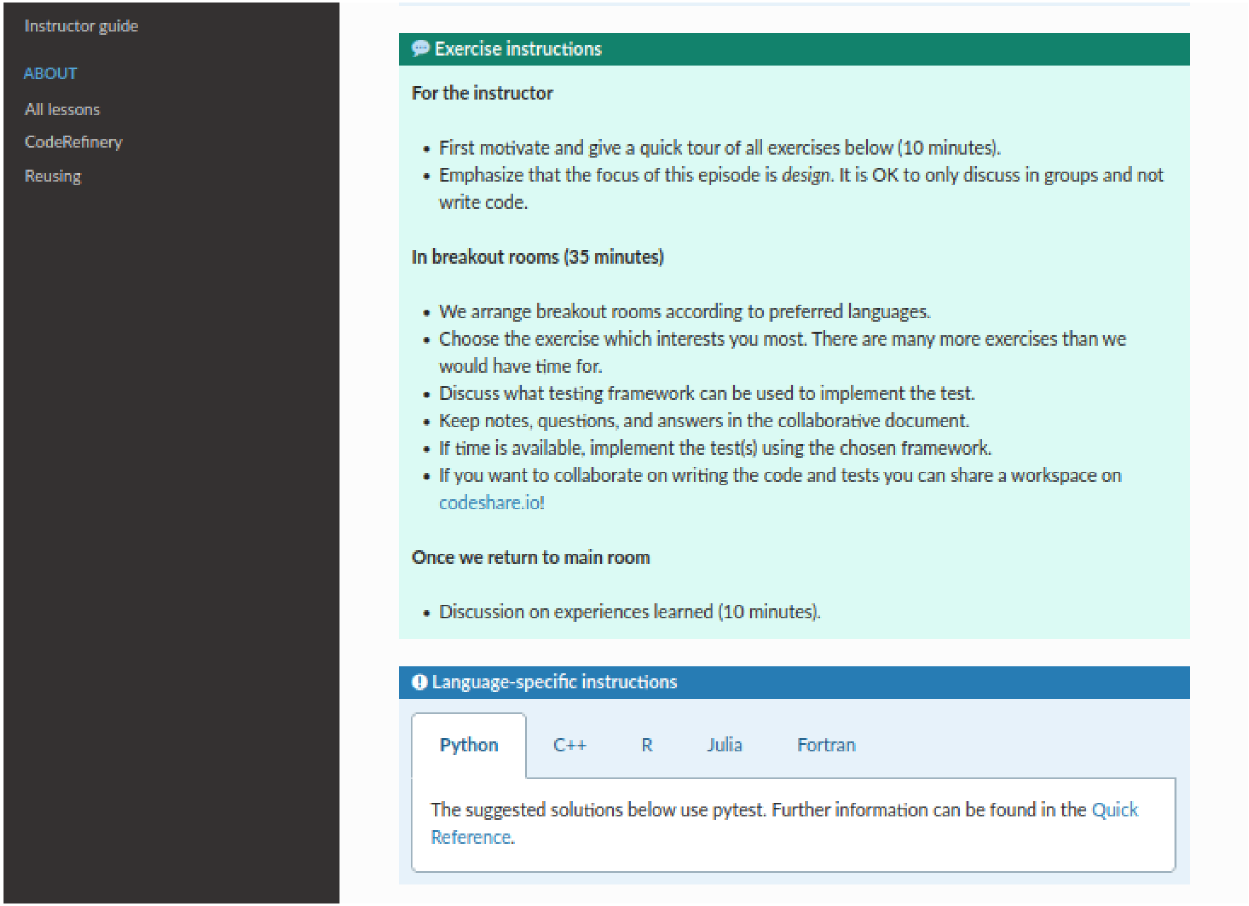
- How can different types of functions and classes be tested?
- How can the integrity of a complete program be monitored over time?
- How can functions that involve random numbers be tested?

In this episode we will consider how functions and programs can be tested in programs developed in different programming languages.

**Objectives**

- Learn how to determine what kind of unit tests can be performed for different type of functions.
- Learn how to perform test-driven development in which tests for a function are designed and implemented before the function is written.
- Learn how to test functions whose output depend on random numbers.

# Lessons



Exercise instructions

**For the instructor**

- First motivate and give a quick tour of all exercises below (10 minutes).
- Emphasize that the focus of this episode is *design*. It is OK to only discuss in groups and not write code.

**In breakout rooms (35 minutes)**

- We arrange breakout rooms according to preferred languages.
- Choose the exercise which interests you most. There are many more exercises than we would have time for.
- Discuss what testing framework can be used to implement the test.
- Keep notes, questions, and answers in the collaborative document.
- If time is available, implement the test(s) using the chosen framework.
- If you want to collaborate on writing the code and tests you can share a workspace on codeshare.io!

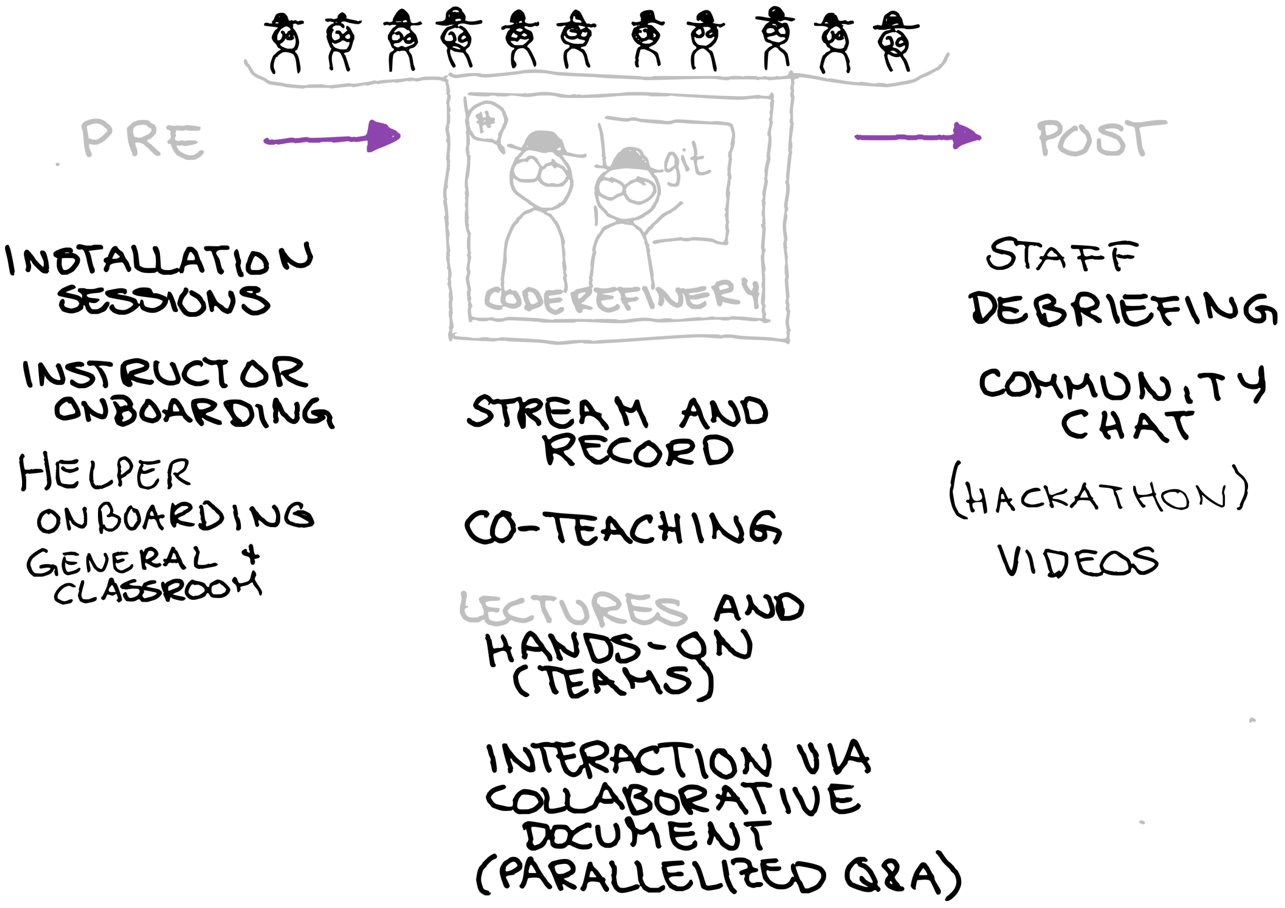
**Once we return to main room**

- Discussion on experiences learned (10 minutes).

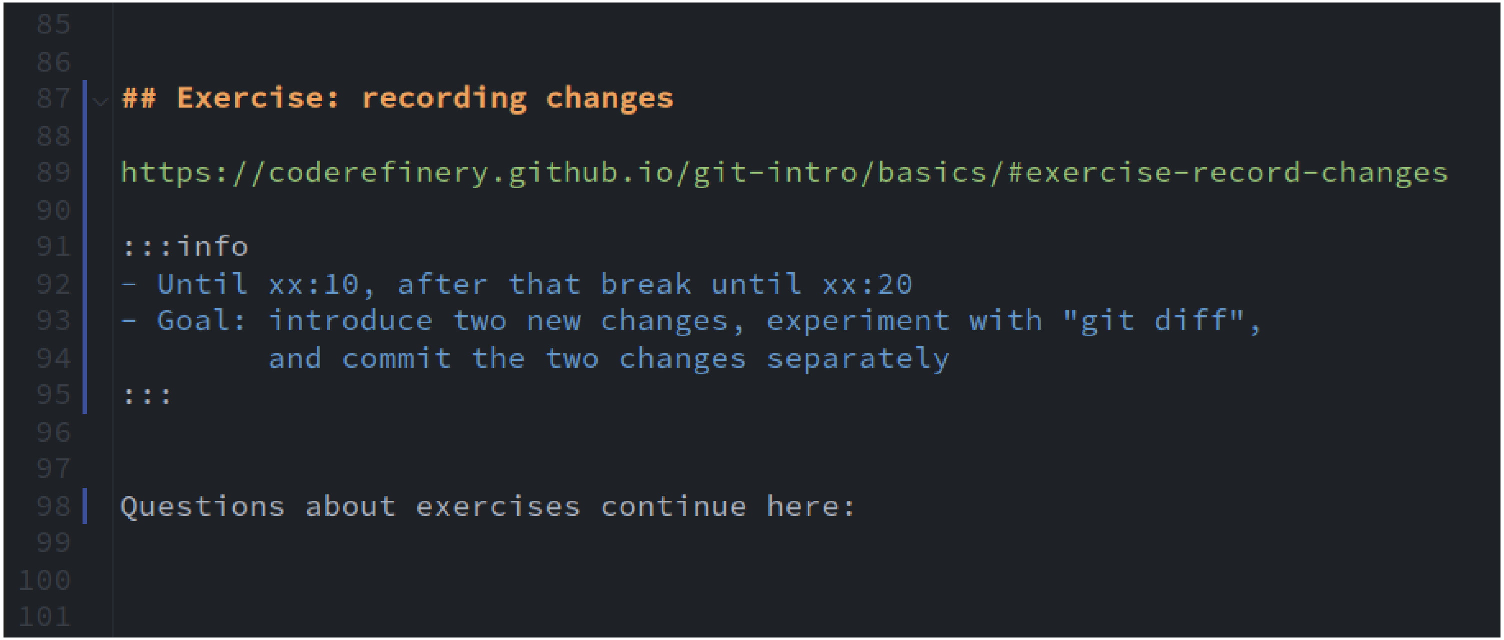
Language-specific instructions

The suggested solutions below use pytest. Further information can be found in the Quick Reference.

# Workshop setup

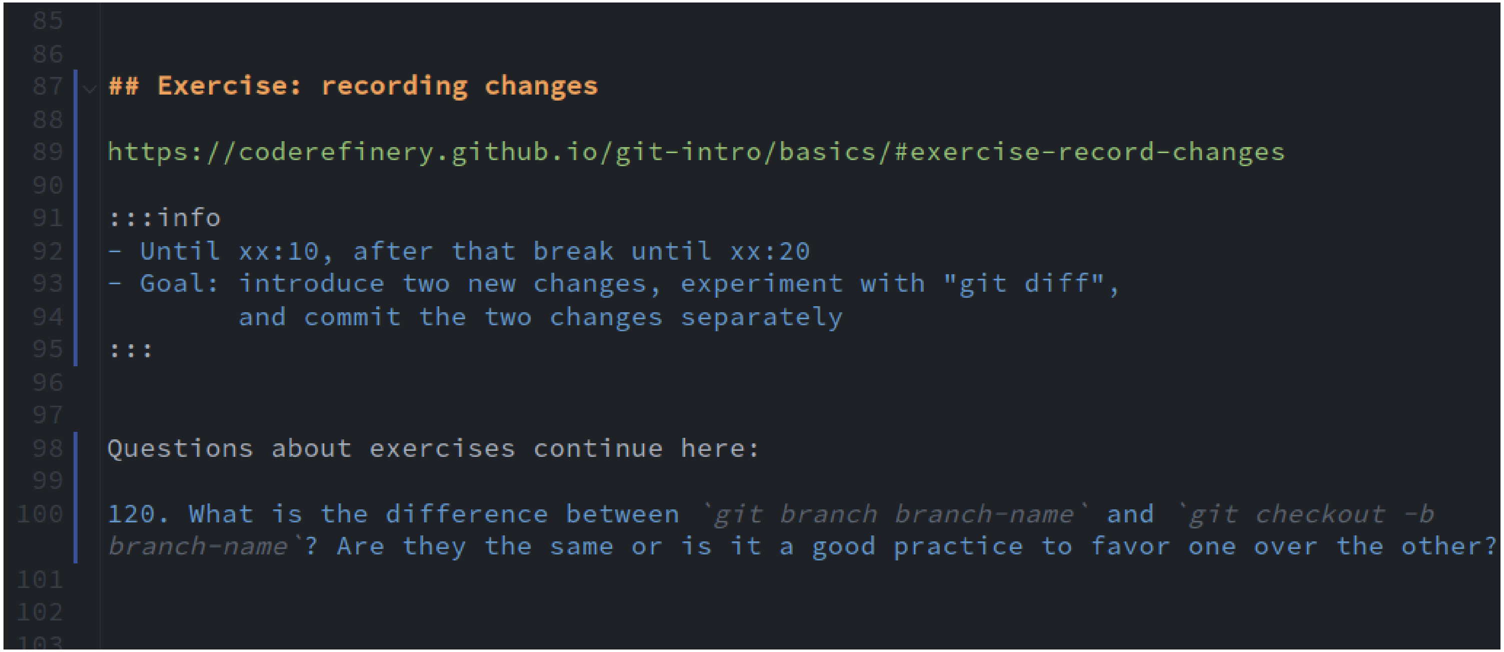


# Collaborative document: Markdown



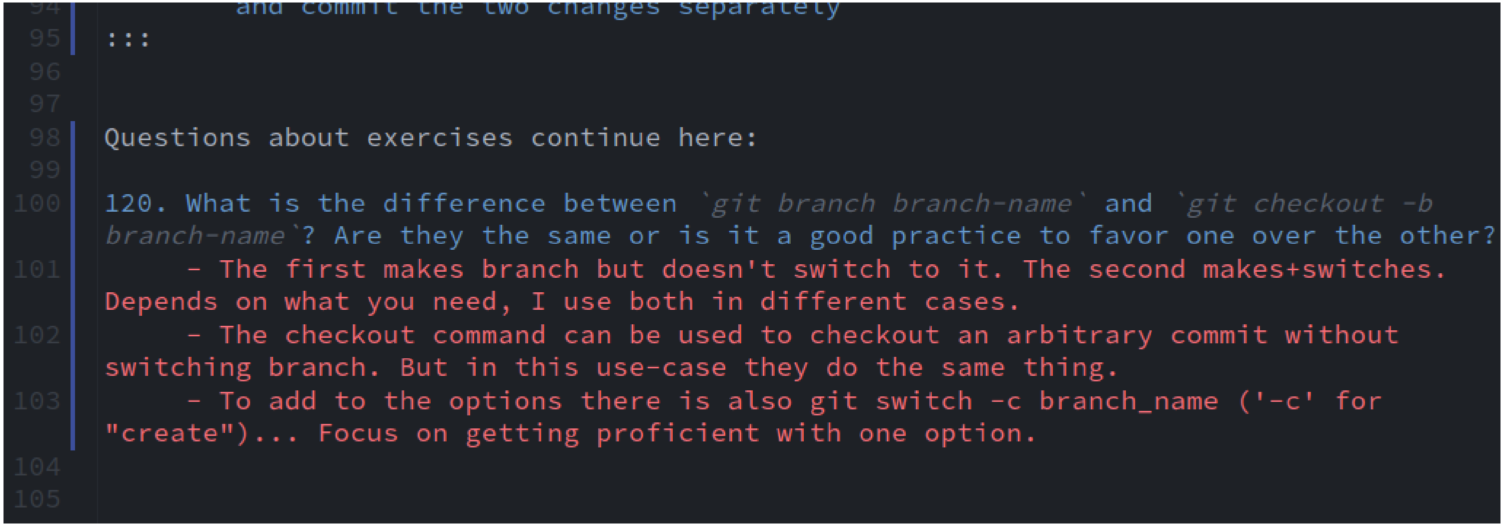
```


## Exercise: recording changes

https://coderefinery.github.io/git-intro/basics/#exercise-record-changes

:::info
- Until xx:10, after that break until xx:20
- Goal: introduce two new changes, experiment with "git diff",
        and commit the two changes separately
:::


Questions about exercises continue here:



```

# Interactive, anonymous, parallel, async



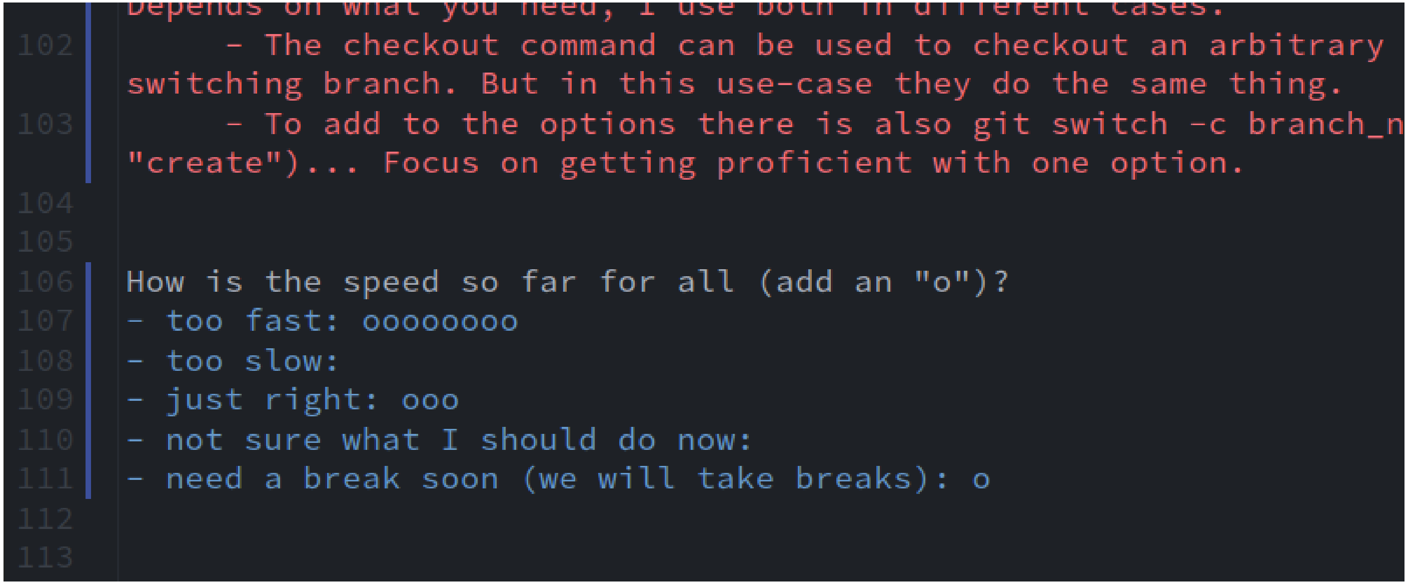
```


## Exercise: recording changes

https://coderefinery.github.io/git-intro/basics/#exercise-record-changes

:::info
- Until xx:10, after that break until xx:20
- Goal: introduce two new changes, experiment with "git diff",
        and commit the two changes separately
:::


Questions about exercises continue here:

120. What is the difference between `git branch branch-name` and `git checkout -b branch-name`? Are they the same or is it a good practice to favor one over the other?
```

# New question every 1-2 minutes!



```
        and commit the two changes separately
:::


Questions about exercises continue here:

120. What is the difference between `git branch branch-name` and `git checkout -b branch-name`? Are they the same or is it a good practice to favor one over the other?
     - The first makes branch but doesn't switch to it. The second makes+switches. Depends on what you need, I use both in different cases.
     - The checkout command can be used to checkout an arbitrary commit without switching branch. But in this use-case they do the same thing.
     - To add to the options there is also git switch -c branch_name ('-c' for "create")... Focus on getting proficient with one option.
```

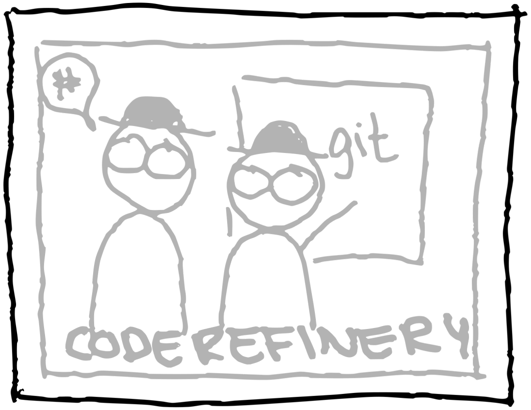
# ASCII-graph feedback



```
Depends on what you need, I use both in different cases.
     - The checkout command can be used to checkout an arbitrary
switching branch. But in this use-case they do the same thing.
     - To add to the options there is also git switch -c branch_n
"create")... Focus on getting proficient with one option.


How is the speed so far for all (add an "o")?
- too fast: oooooooo
- too slow:
- just right: ooo
- not sure what I should do now:
- need a break soon (we will take breaks): o
```

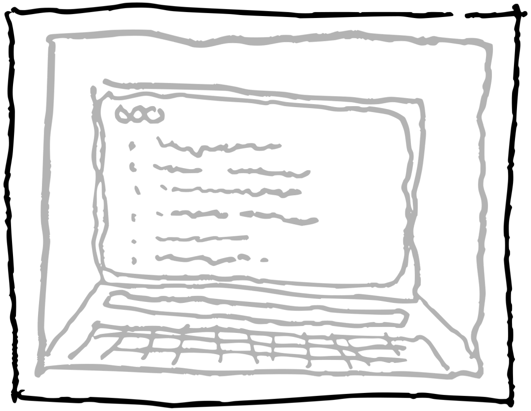
We publish Q&A for each workshop: Example

# Participating as a learner



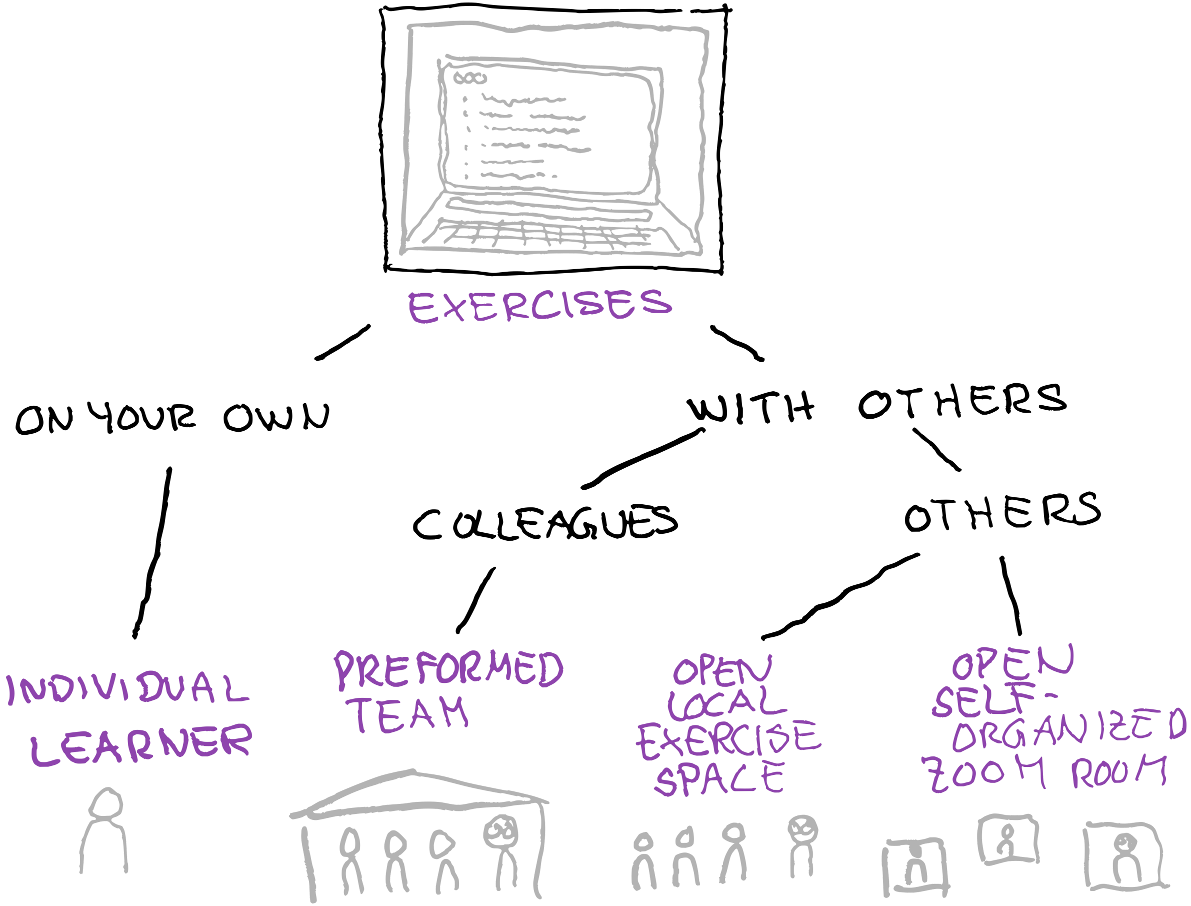
LIVE STREAM +
COLLABORATIVE
DOCUMENT

+



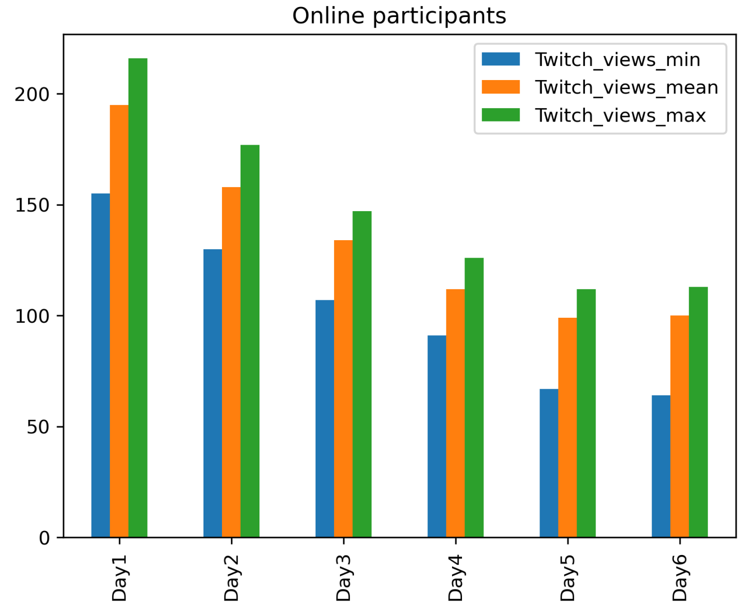
EXERCISES

# Exercise options

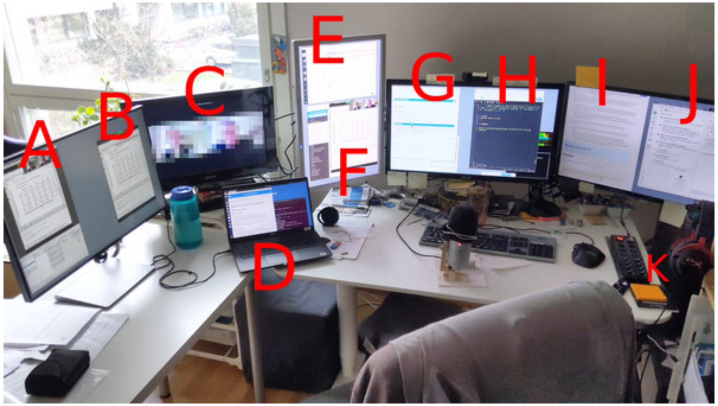


# Workshop stats

- 7 online and 28 in-person workshops
- We reach over 500 persons/year
- Over 30 instructors/speakers
- Over 100 helpers/ exercise leaders



# Collaboration across funding borders



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

neic NORDIC E-INFRASTRUCTURE COLLABORATION   A! Aalto University   NRIS   ENCCS EuroCC National Competence Centre Sweden   AARHUS UNIVERSITY   CSC   National Academic Infrastructure for Supercomputing in Sweden NAISS

Co-advertize and co-organize with us

TU Delft   netherlands eScience center   VU VRIJE UNIVERSITEIT AMSTERDAM

# 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

- 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

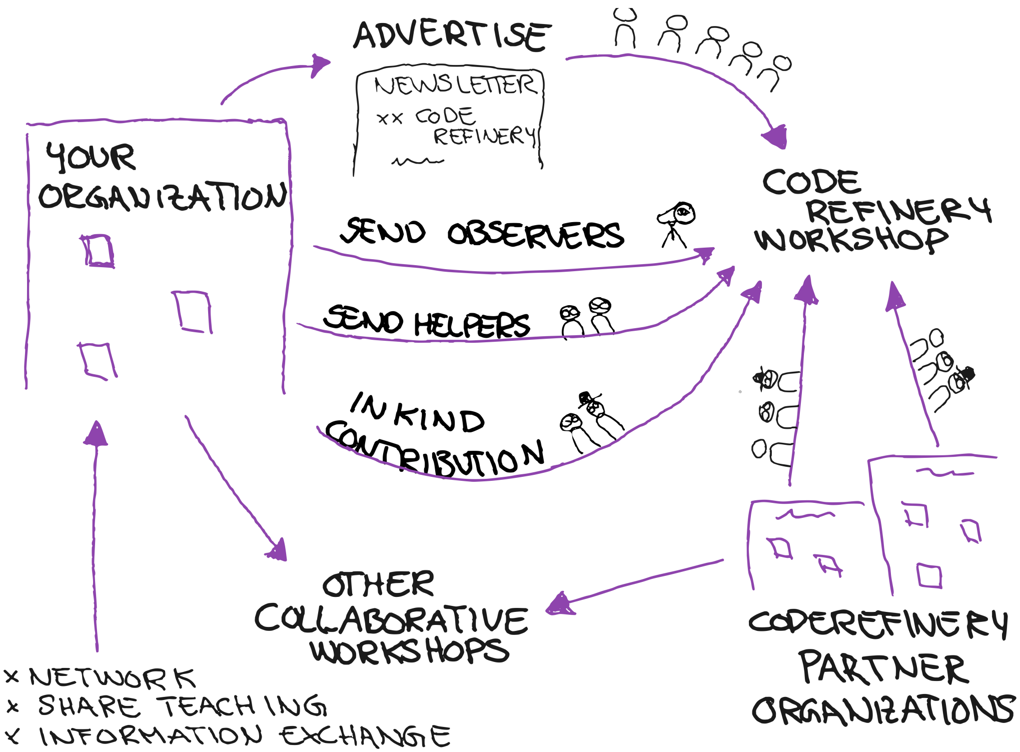
## Future: Organization

- 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")
- Towards non-profit organization so that we can participate in funding applications

## Teaching format

- Continue large-scale workshops
- Support local events
- More asynchronous content coupled with online events

# How you or your organization can participate



- **Join our next workshop** September 19-21 and 26-28, 2023: https://coderefinery.github.io/2023-09-19-workshop/
- Tell all your students and researchers to watch
- Send one or more **exercise teams** or **join as observer**
- Use our material and give feedback

# Credits and license

## Text

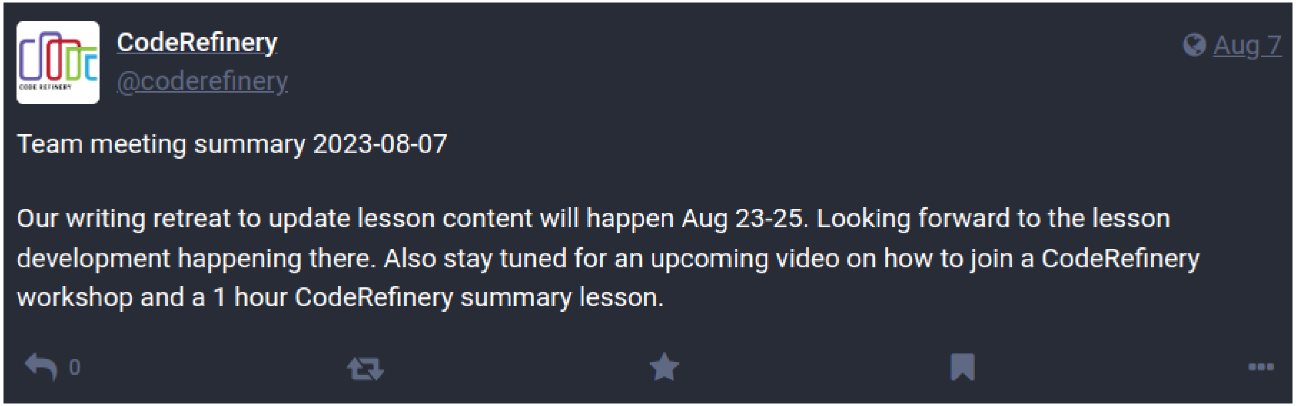
- All text: CodeRefinery project, CC-BY 4.0

## Images

- Slide 3: H. Seibold, "6 helpful steps for reproducible research", CC-BY 4.0
- Slides 6, 11, 12, 18: S. Wittke
- Slide 14: ATC tower, P. R. Miller, CC-BY 2.0
- Slide 14: Monitor setup, R. Darst
- Slide 14: Logos, (c) respective organizations
- All other images: CodeRefinery project, CC-BY 4.0

# We try to make it easy to join

- **Join our next workshop** September 19-21 and 26-28, 2023: https://coderefinery.github.io/2023-09-19-workshop/
- **Chat with us**: https://coderefinery.zulipchat.com (ask questions about coding or learn about new tools)
- Blog, Newsletter, Twitter, Mastodon, Support



CodeRefinery
@coderefinery
Aug 7

Team meeting summary 2023-08-07

Our writing retreat to update lesson content will happen Aug 23-25. Looking forward to the lesson development happening there. Also stay tuned for an upcoming video on how to join a CodeRefinery workshop and a 1 hour CodeRefinery summary lesson.

# Nordic RSE Unconference 2023

## Theme: "💎 Hidden gems and paper cuts 🩹"

October 25 - 26, online, afternoon (CET)

https://nordic-rse.org/events/2023-online-unconference/