

So you want to know something about Git

Maria Lehtivaara (CSC –IT Center for Science)



16.6.2021



CSC – Finnish expertise in ICT for research, education, culture and public administration

Aim of this presentation

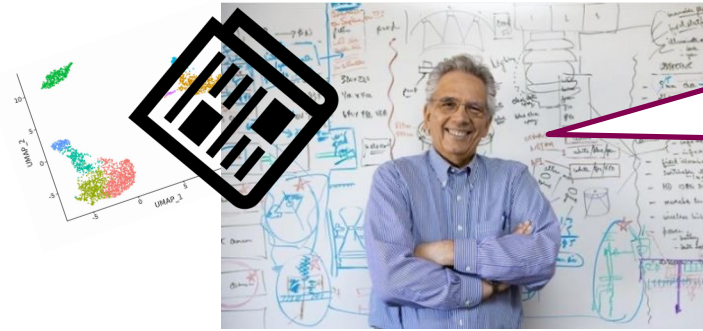
- Give a brief idea of how Git can be used in producing re-producible science
 - ➡ Hopefully you will:
 - recognise the tools & some terms, can map these to some context
 - think of possible use cases
 - know how to learn more

Disclaimers

- You are **not** a Git guru after this 20 mins 😞
 - I'm **not** a Git guru -just a humble user
- Git usage or git tools are **not** specifically maintained or supported by CSC. Except...
 - our manuals and training material
 - NEIC CodeRefinery collaboration
 - We are quite friendly thou 😊 servicedesk@csc.fi



Reproducibility: Bad vs better situation



Hey, can you check which parameters were used to get these results? Could we try this small modification?

- Typing commands directly on the command prompt
- Storing files in personal computer in random hierarchy
- Having the softwares installed (only) in your own computer



Ummm... I HAD some results here on my laptop... and I THINK this was the laptop where I had the softwares installed...

- Typing commands and comments to **version controlled files**, using **automated workflows**
- Storing files in (public/joined) **repositories** in cloud
- Having the softwares installed as **documented containers**

Sure! Let me just check my repository, and we can re-run the needed sections in these containers!



(...but PLEASE don't ask me again after 5 years...)

Git -what & why?

- Version control system
- Used by developers / coders, but useful to anyone who writes (code) or needs to **track changes to files** => **that's basically everyone, right!**
 - Tracks the changes you make to files
 - Allows to revert to specific versions (= recovery from "oopsies")
 - Makes collaboration easier (allow changes by multiple people to all be merged into one source)



Example:
TheBreadCode

P.S: Git was developed by Linus "Linux" Torvalds 2005, and it's free 😊

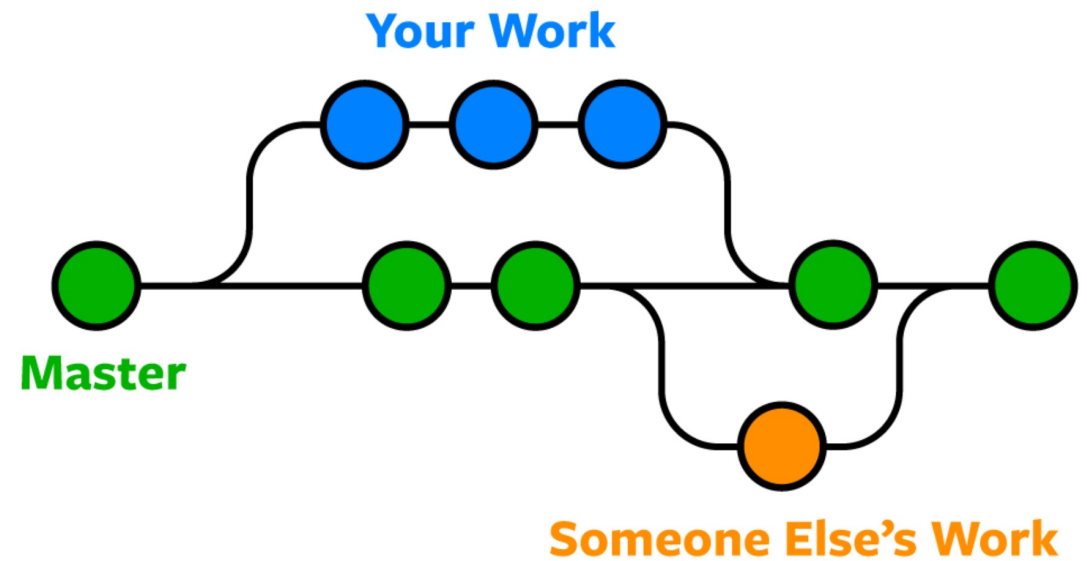
Source: <https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it>
Bread example: <https://github.com/hendricius/the-bread-code>

Git -what & why?

- The life of a document:



Video: <https://youtu.be/CvbLVVRzJF8>



Git, GitHub, GitLab –what's the difference?



- Git = version control system
 - On your own computer
 - Free
 - “the base”



- GitHub = web-based Git repository hosting service
 - In cloud, for sharing
 - Free when using public repos -> good for open source development
 - private repos for 3 members only!
 - Graphical web user interface & extra tools
 - Very popular



- GitLab = like GitHub, but:
 - Some terminology & workflow difference
 - Private repos free?
 - “Open core” source code
 - => **many organisations host their own GitLab**
 - Data stays more locally
 - Authentication
 - For example:
<https://coderefinery.org/repository/>

Sources: <https://blog.devmountain.com/git-vs-github-whats-the-difference/>
<https://blogs.helsinki.fi/thinkopen/versionhallinta-on-valttamaton-tyokalu-tutkimukselle/>

So... what is where?

Normal folder



Files



Your computer

Git repository



Files + revision history
for you



Your computer

Remote repository



Files + revision history
for everyone*
(= team, company, everyone...)



Cloud



archive &
reference
repositories

Check these:

<https://guides.github.com/activities/citable-code/>

<https://www.softwareheritage.org/save-and-reference-research-software/>

Source: <https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it>

Do I have... options?



- Yep, there are other **Version Control Systems (VCSs)**:

- Or “revision control”, “source code management”...



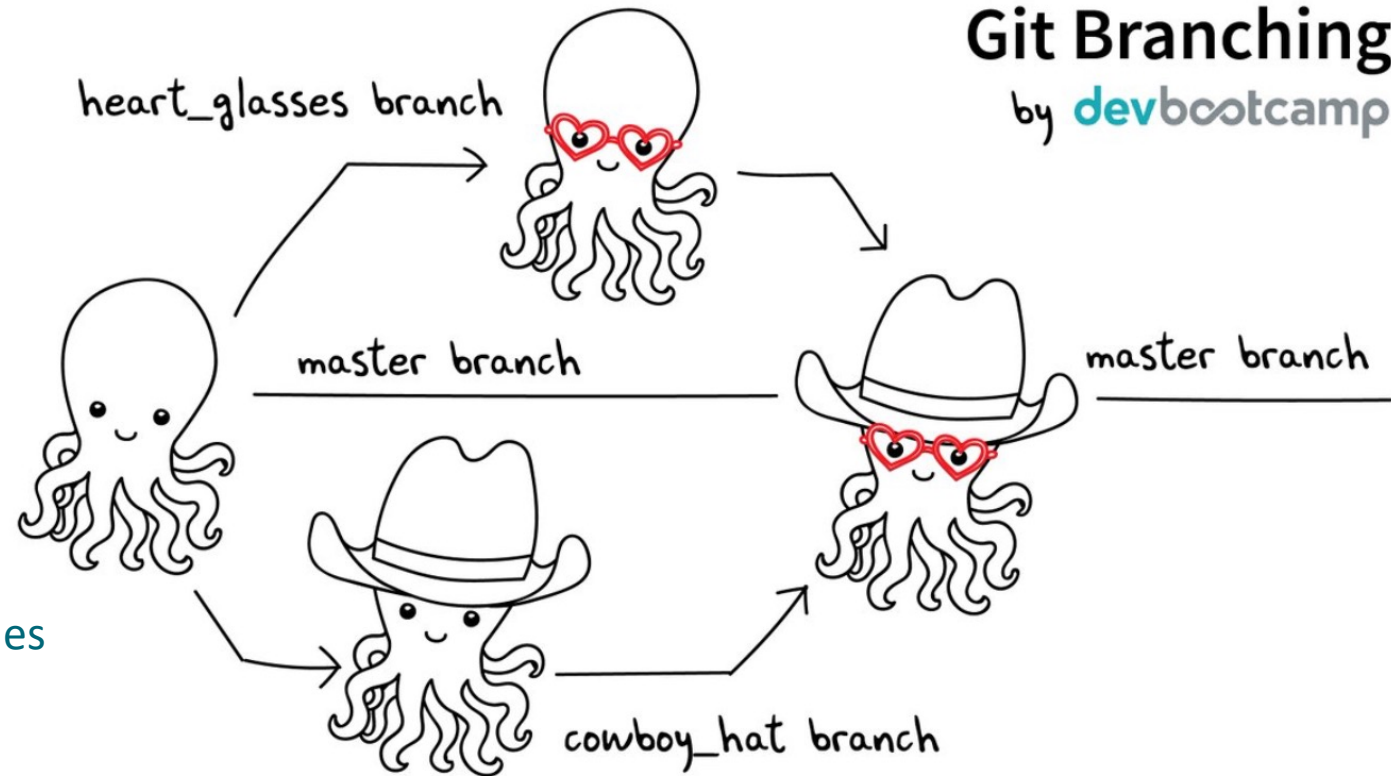
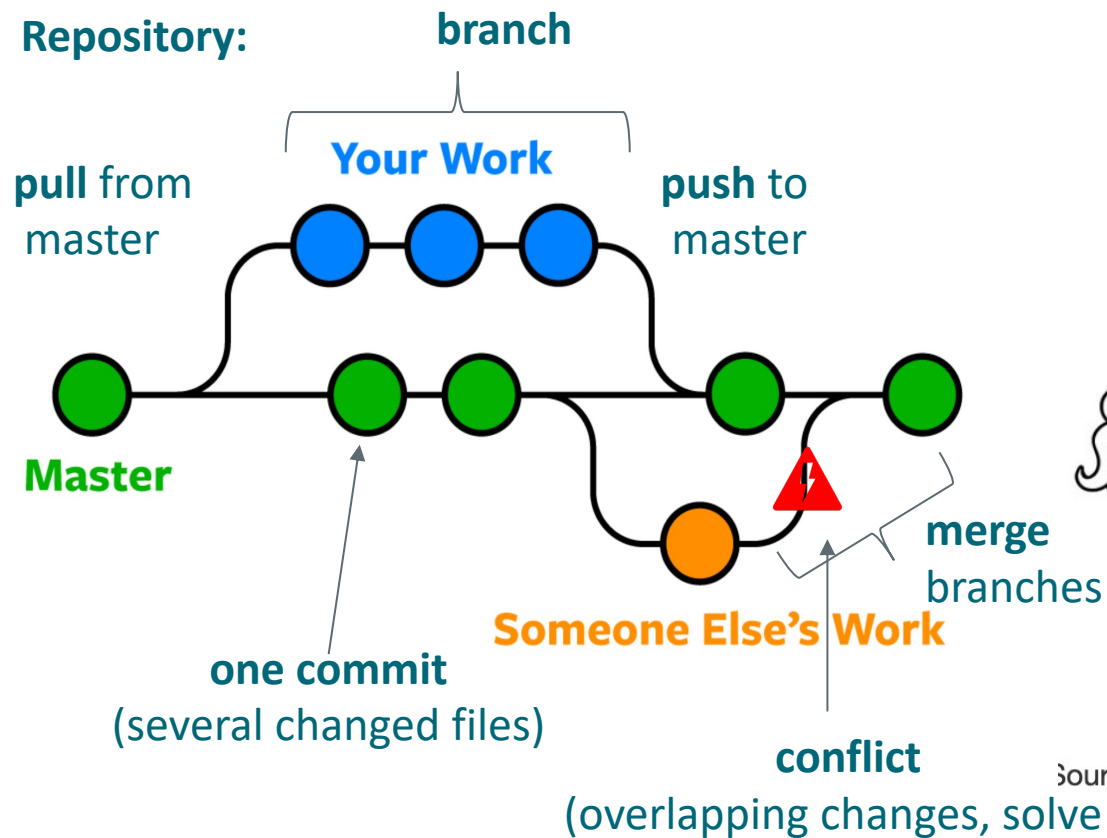
PERFORCE

- Yep, there are other **repository hosting services**:



Some Git-terminology

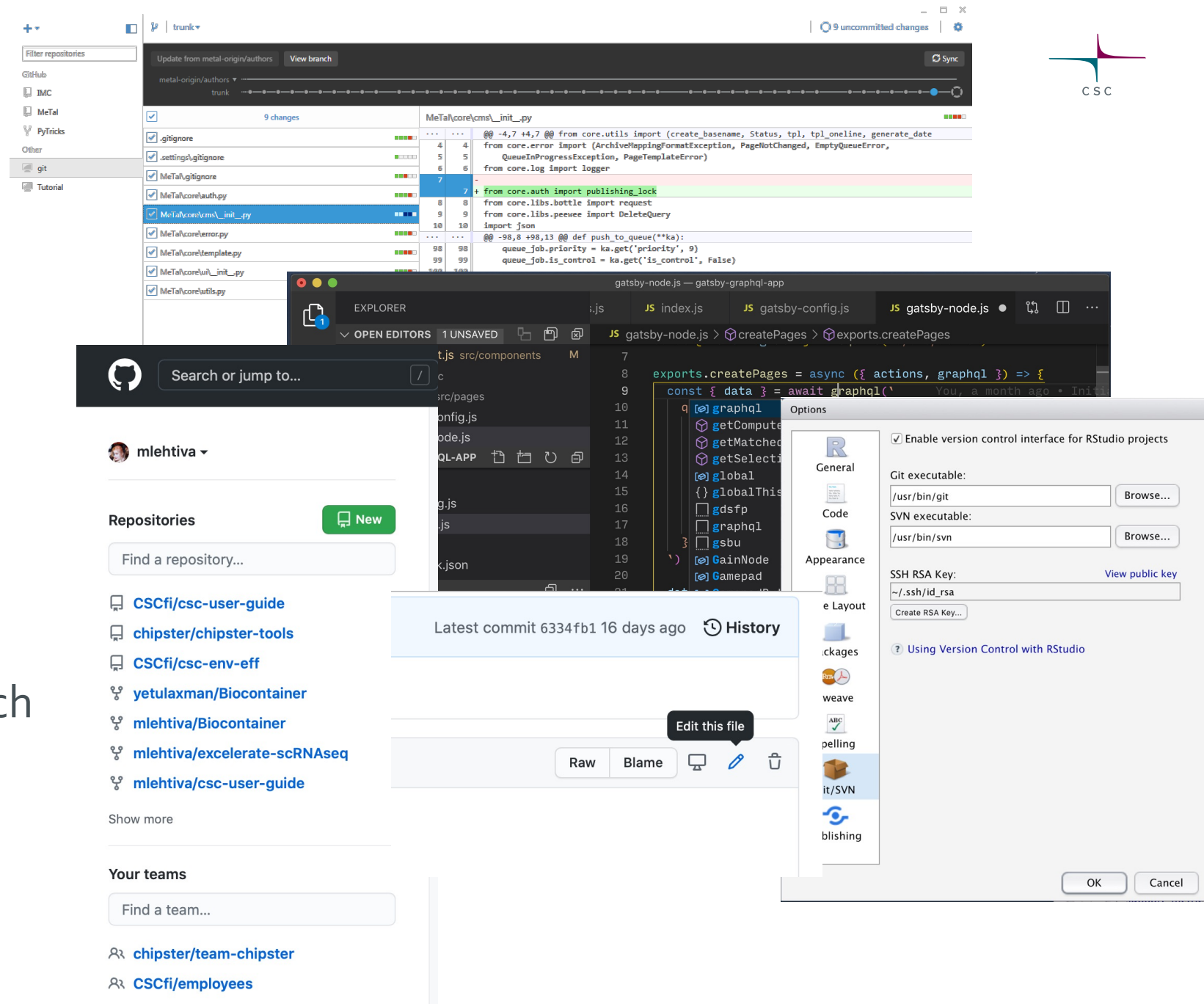
- Some terms you might run into (over-simplified! ⚡)



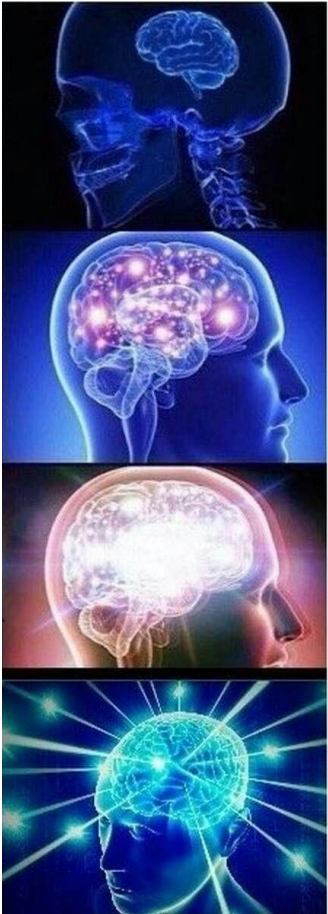
source: https://twitter.com/jay_gee/status/703360688618536960

Tools for Git

- Command line (=terminal)
- GitHub
- Handy graphical GUIs to your own computer
 - GitHub Desktop, Visual Studio Code, Sourcetree, Fork...
- Directly from RStudio...
- Lot's of great easy-to-approach material online!
 - <https://coderefinery.org/lessons/>



Some use cases –have you considered these?



- Code development
 - Developing analysis pipeline or tool
- Data analysis
 - “What did I do again? Did I test that already? Can you continue this PhD students work?”
 - Publications link to Git repos containing everything needed to repeat the analysis
- Writing
 - Anything in plain text* format: article, blog post, notes... (* Markdown, LaTeX...)
- Guides, learning materials...
 - CSC’s user guides: docs.csc.fi
 - CSCs course materials: <https://github.com/csc-training>



Feel free to use and develop!

Workflow sneak peak: editing our user guide

At docs.csc.fi:

- Home
- ▶ Accounts
- ▶ Computing
- ▶ Cloud
- ▶ Data
- ▶ Applications
- ▶ Support

Running your first job on Puhti

Logging in

To prepare and run your jobs, you first need to log in to Puhti. You can use either command line application or a special terminal program. Command line applications come standard on most operating systems. Terminal programs may need to be installed separately, but they typically offer more options on things like font size, copy-paste etc.

On Linux or macOS open a terminal. In Windows 10 open Powershell. Give command:

```
ssh yourcscusername@puhti.csc.fi
```

Where yourcscusername is the username you got from CSC

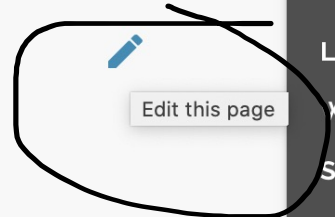


Table of Contents

- Logging in
- What is Puhti?
- Software environment
- Planning your job
 - How many cores can my application use?
 - How much memory does my application need?
 - How much time should I reserve?
- Running your job



Browser address bar: github.com

Navigation icons: back, forward, home, search, etc.

System tray icons: WhatsApp, Slack, etc.

GitHub navigation bar: Search or jump to... Pull requests Issues Marketplace Explore

User profile: +, profile picture, Editing...

CSCfi / csc-user-guide

Watch 20 Star 5 Fork 45

Code Issues 8 Pull requests 8 Actions Projects Wiki Security Insights

csc-user-guide / docs / support / tutorials / biojobs-on-puhti.md

Cancel changes

Edit file Preview

Spaces 4 Soft wrap

```

1 # Running your first job on Puhti
2
3 ## Logging in
4
5 To prepare and run your jobs, you first need to log in to Puhti. You
6 can use either command line application or a special terminal
7 program. Command line applications come standard on most operating
8 systems. Terminal programs may need to be installed separately, but they
9 typically offer more options on things like font size, copy-paste etc.
10
11 On Linux or macOS open a terminal. In Windows 10 open Powershell. Give command:
12
13 ```text
14 ssh yourcscusername@puhti.csc.fi
15 ```

```

type type type

Commit -> branch + pull request -> review -> accept pull request + merge



Commit changes



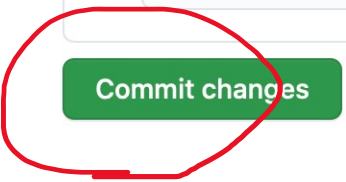
Added a link to connecting tutorial



Link to "Using CSC HPC environment efficiently" course material tutorial about connecting added.

- You can't commit to `master` because it is a **protected branch**.
- Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

`mlehtiva-improve-bio101-tutorial`



Commit changes Cancel



Review required

At least 1 approving review is required by reviewers with write access. [Learn more.](#)

[Show all reviewers](#)



1 pending reviewer



All checks have passed

3 successful checks

[Show all checks](#)



Merging is blocked

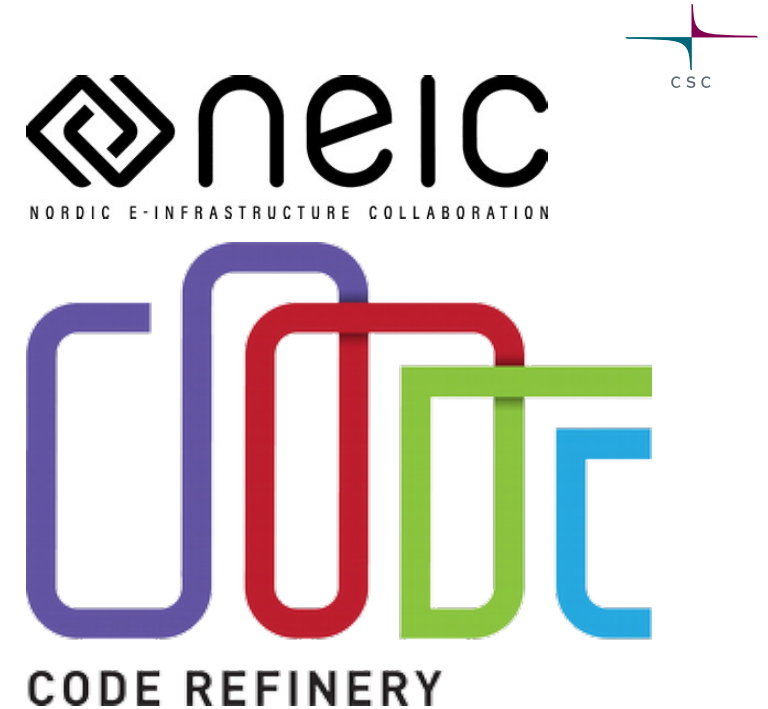
Merging can be performed automatically with 1 approving review.



More info –where?



- CSC services: servicedesk@csc.fi
- Your organisations own IT support
 - They might have some system in use, like GitLab
 - in HY for example, version.helsinki.fi
- NEIC CodeRefinery (NEIC <3 CSC)
 - Workshops: Next in fall
 - May workshop materials: <https://coderefinery.github.io/2021-05-10-workshop/>
 - Learning materials: <https://coderefinery.org/lessons/>
 - Try for example:
 - Reproducible research: <https://coderefinery.github.io/reproducible-research/>
 - Git: <https://coderefinery.github.io/git-intro/>
 - Chat: <https://coderefinery.zulipchat.com> (if you are interested in teaching or improving the material for example)
 - Repository: <https://coderefinery.org/repository/> (GitLab-based, in Denmark)



Some learning curve in the beginning -patience 😊

