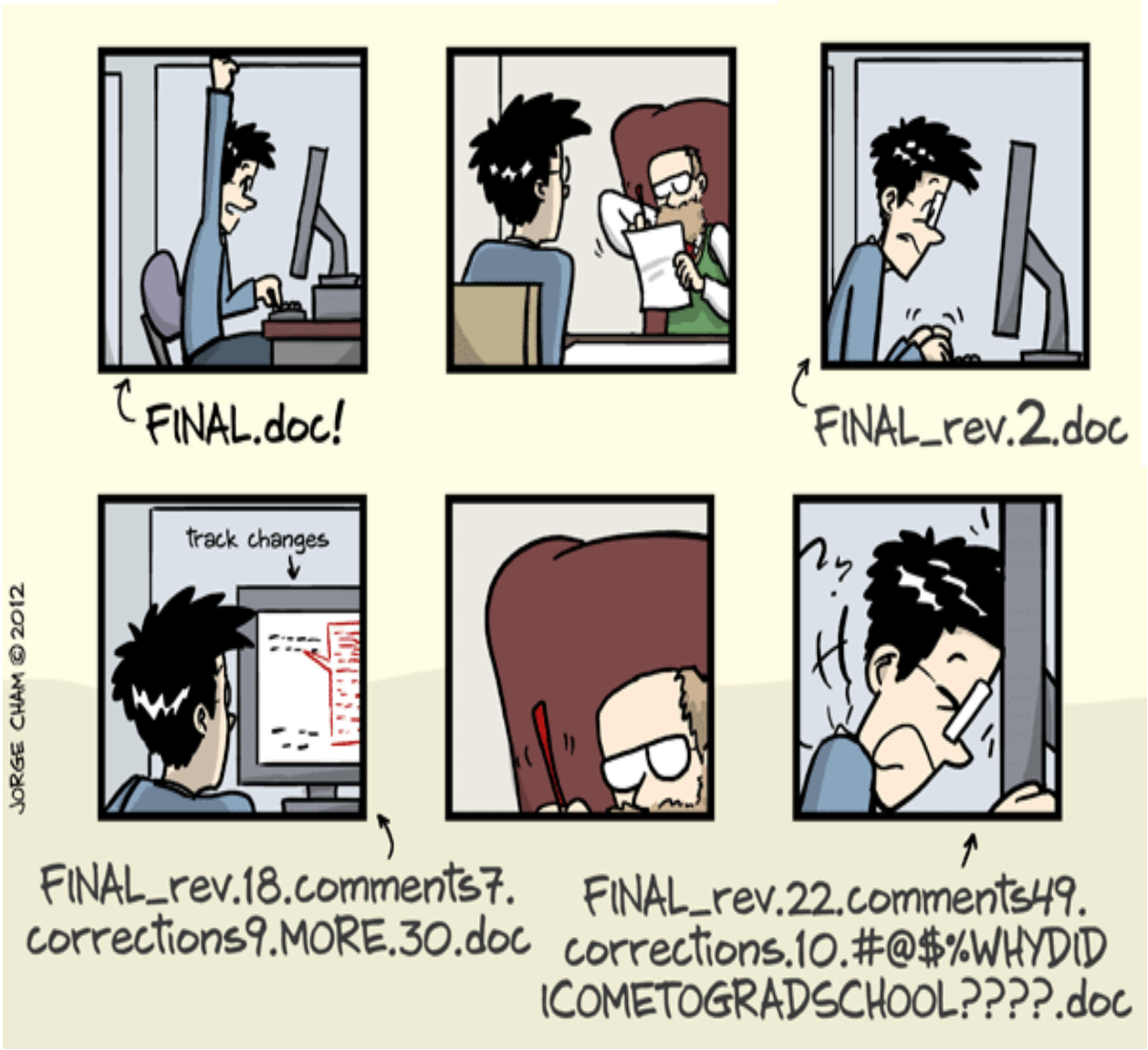
Fachhochschule Kiel
University of Applied Sciences

Tools and Programming for Data Science
Git and GitHub

Study Program Data Science
Prof. Dr. Tillmann Schwörer

Why Git and GitHub?



Why Git and GitHub?

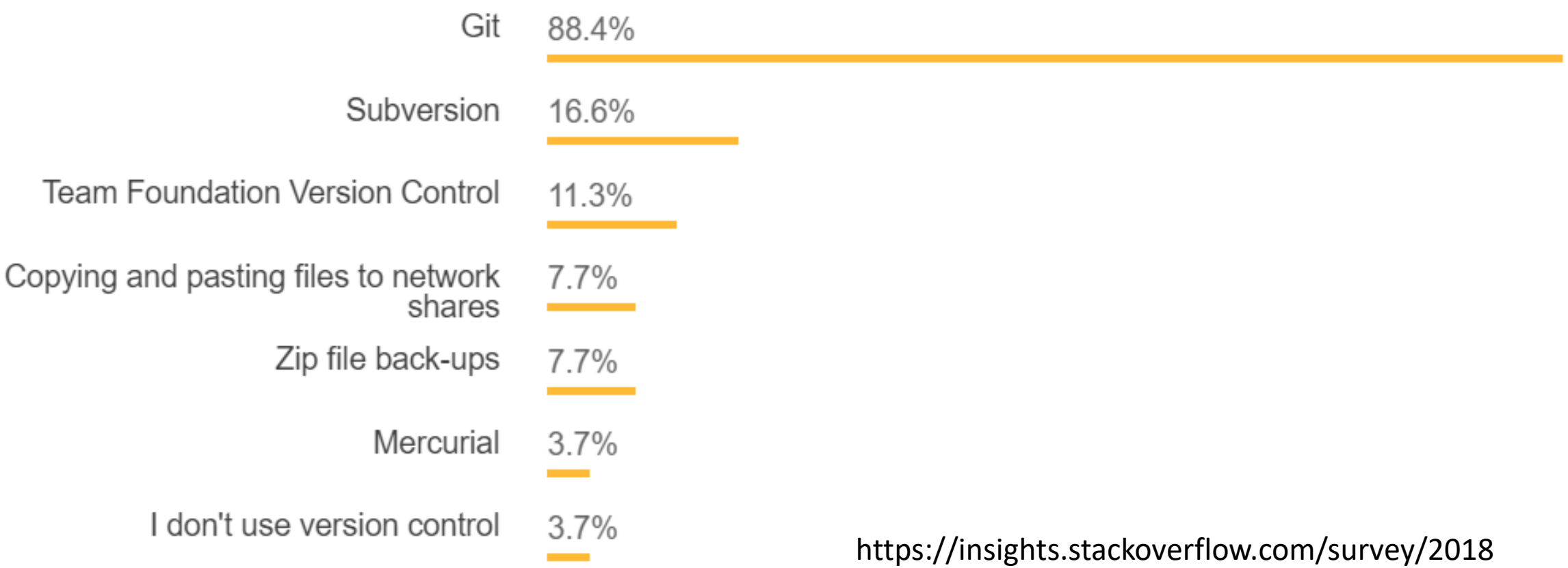
Fachhochschule Kiel
University of Applied Sciences

- **Backup**: undo changes, restore files, safely experiment
- **Transparency:** what was changed? by whom? when?
- **Collaboration:** work simultaneously with coauthors on the same project
- **Job applications:** showcase your version control and data science skills through your own GitHub repository

Version control systems

Fachhochschule Kiel
University of Applied Sciences

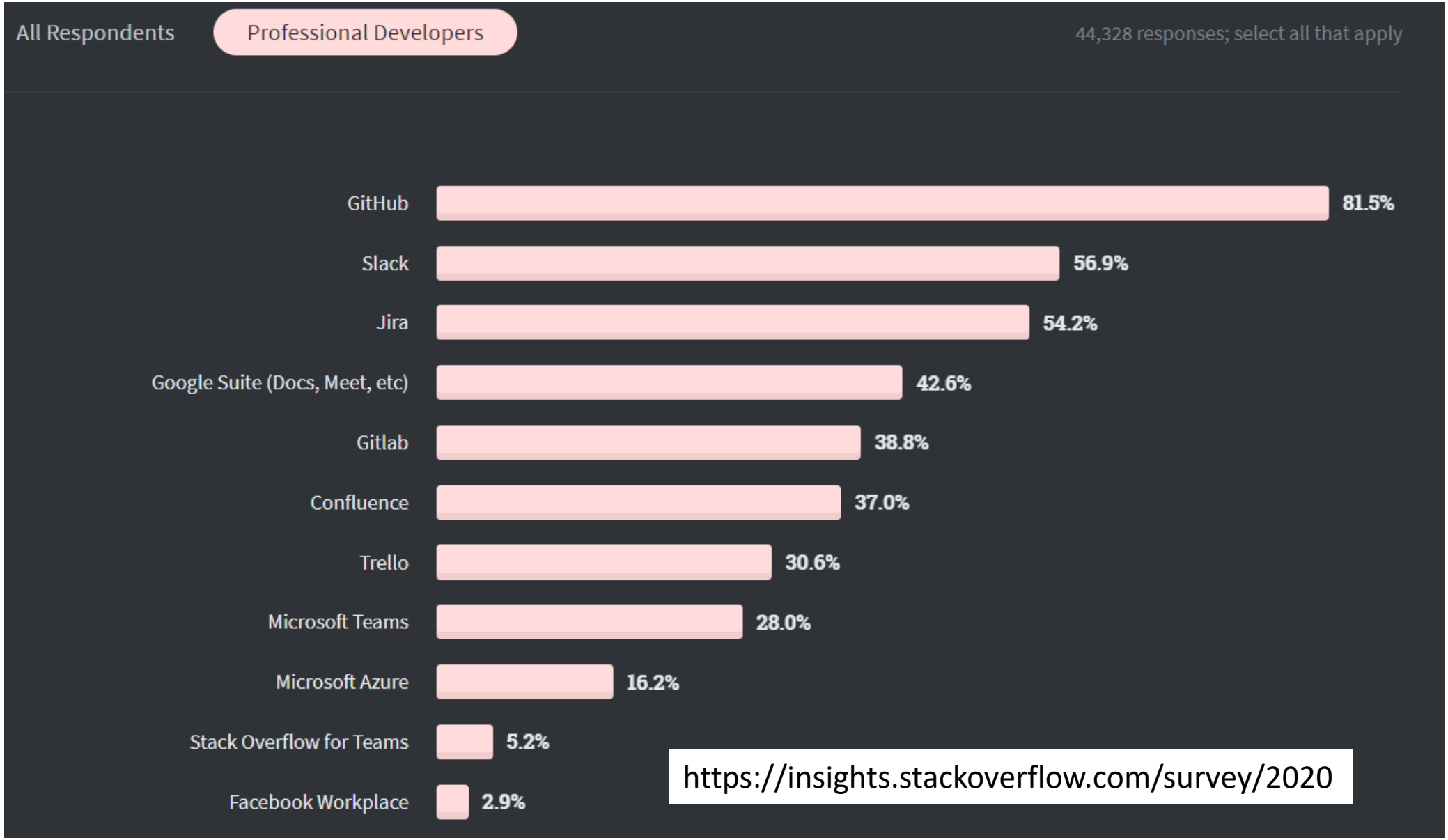
All Respondents | Professional Developers



https://insights.stackoverflow.com/survey/2018

Top Collaboration Tools

Fachhochschule Kiel
University of Applied Sciences



https://insights.stackoverflow.com/survey/2020

Example

Fachhochschule Kiel
University of Applied Sciences

Chaos Computer Club: Entdecke unsere Verfassung

Johannes Rau, Bundespräsident committed on 26 Jul 2002

1 parent dc4dd4d commit b17a4b2848dac62d37618ebc92c06cccce5385ff

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

2 020a.md

```
@@ -1,4 +1,4 @@
## Artikel 20a

- Der Staat schützt auch in Verantwortung für die künftigen Generationen die natürlichen Lebensgrundlagen im Rahmen der verfassungsmäßigen Ordnung durch die Gesetzgebung und nach Maßgabe von Gesetz und Recht durch die vollziehende Gewalt und die Rechtsprechung.
```

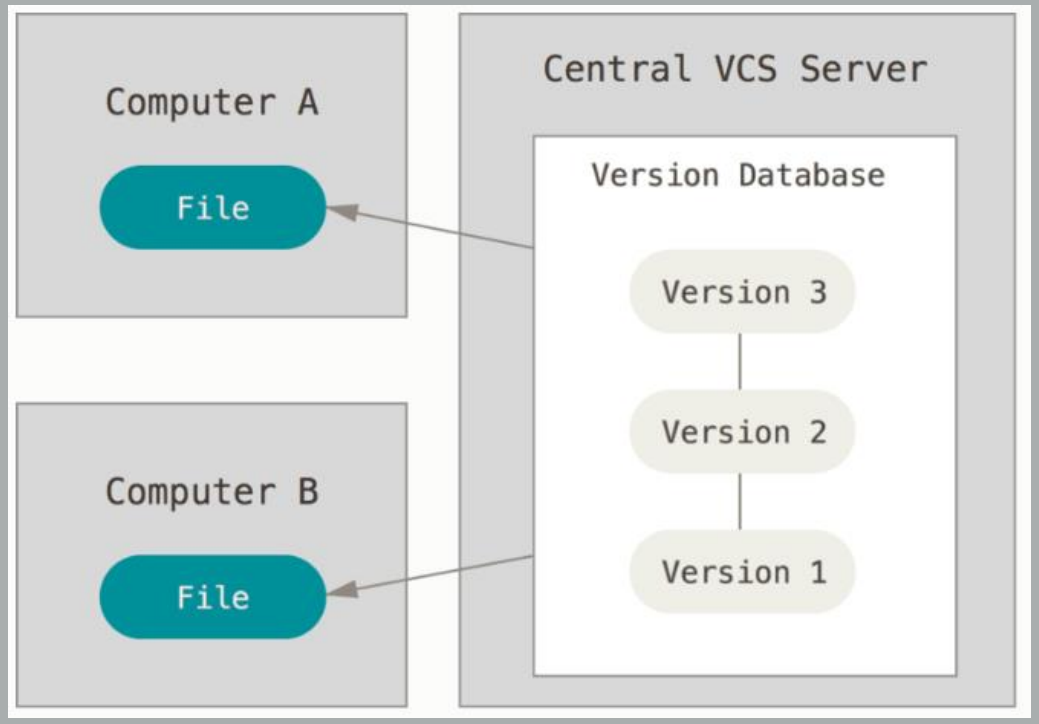
```
## Artikel 20a

+ Der Staat schützt auch in Verantwortung für die künftigen Generationen die natürlichen Lebensgrundlagen und die Tiere im Rahmen der verfassungsmäßigen Ordnung durch die Gesetzgebung und nach Maßgabe von Gesetz und Recht durch die vollziehende Gewalt und die Rechtsprechung.
```

Clarifying terms

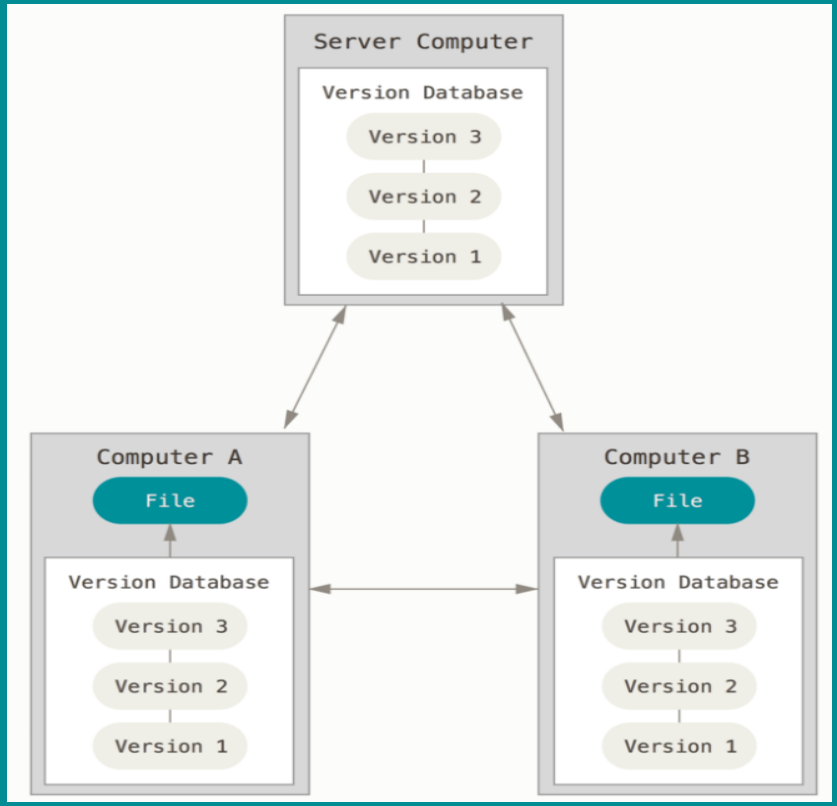
Fachhochschule Kiel
University of Applied Sciences

- **Version control system** (VCS): keeps track and manages changes in your / your team's documents
- **Git:** an open source, distributed version control system (VCS)
- **GitHub:** a platform for hosting and collaborating on Git repositories

Version control systems



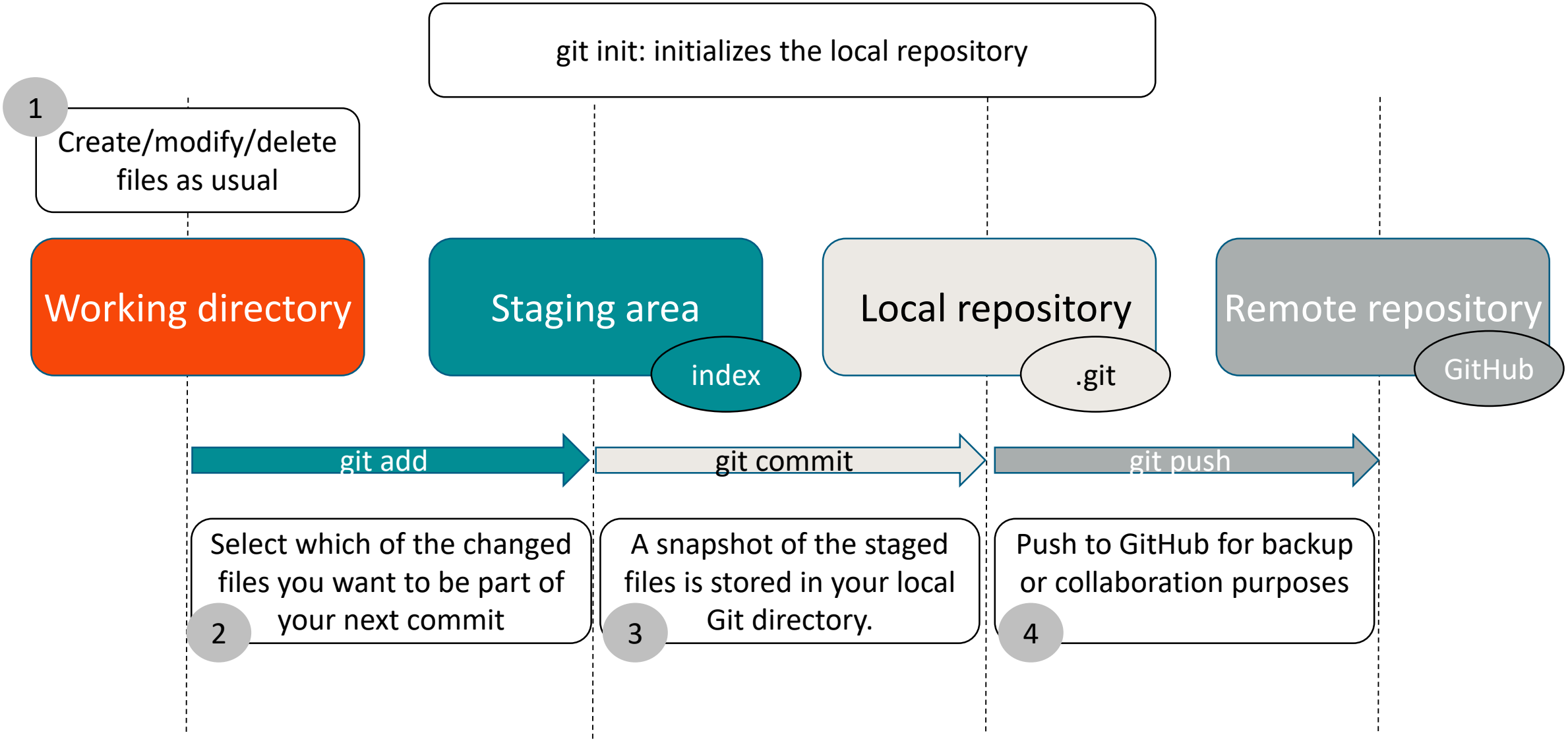
Centralized: single files are pulled and pushed from/to central repository (e.g. **Subversion**)



Distributed: Each client has full copy of entire repository (→ **git**)

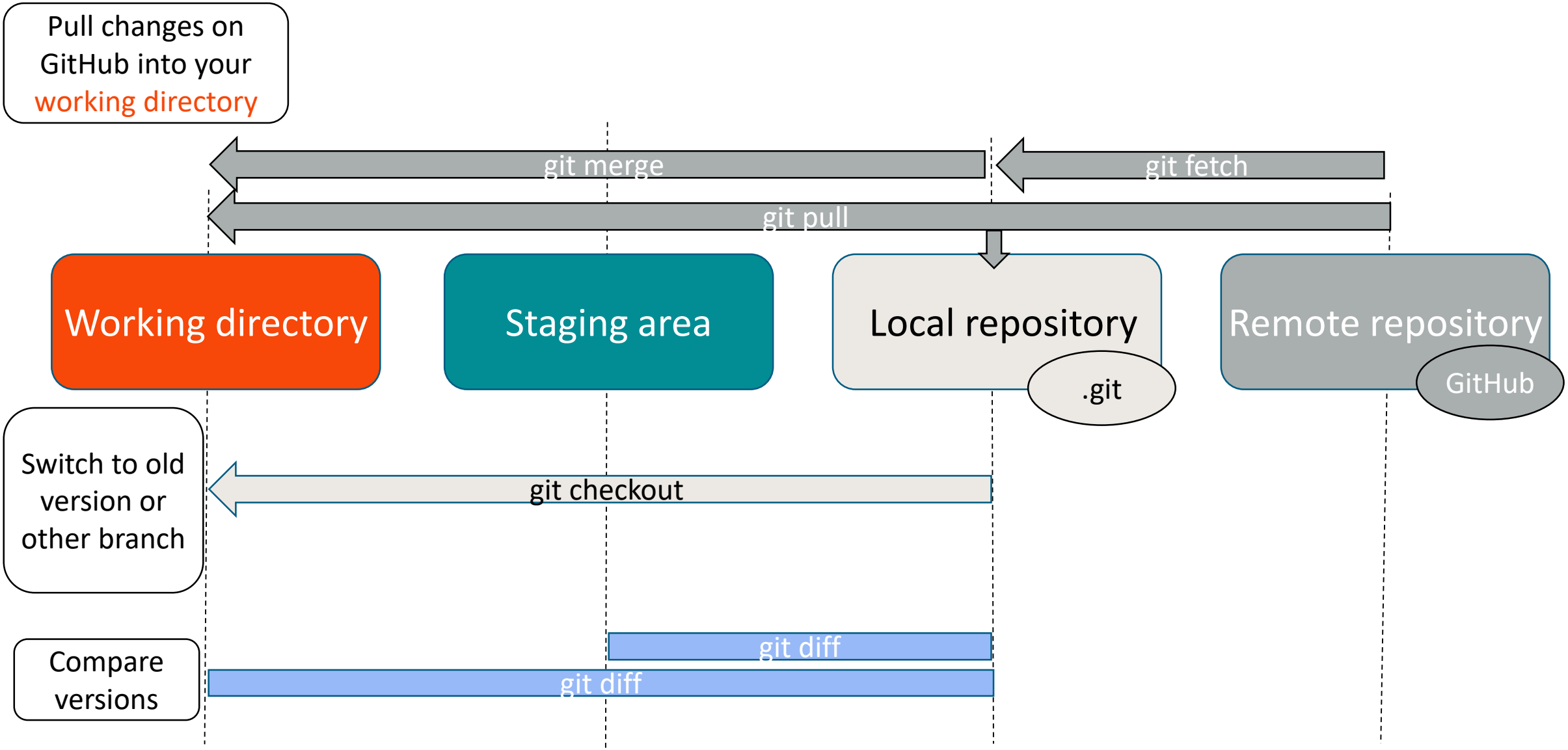
Important concepts / commands

Fachhochschule Kiel
University of Applied Sciences



Important concepts / commands

Fachhochschule Kiel
University of Applied Sciences



Undo commits

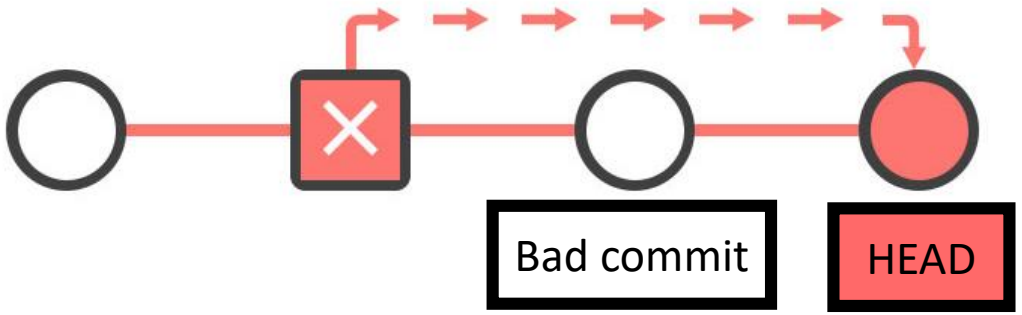
Fachhochschule Kiel
University of Applied Sciences

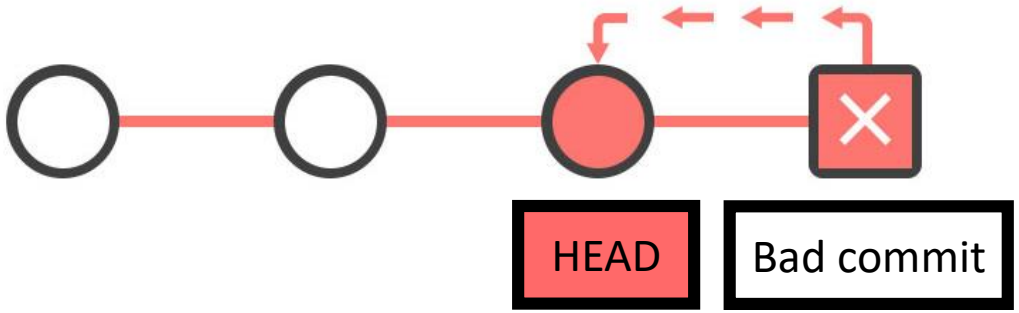
<u>Revert</u> (*#bad-commit*)

- Adds a new commit which reverses the „bad commits“
- Leaves the commit history intact
- Can be safely done, even if you have already pushed to GitHub

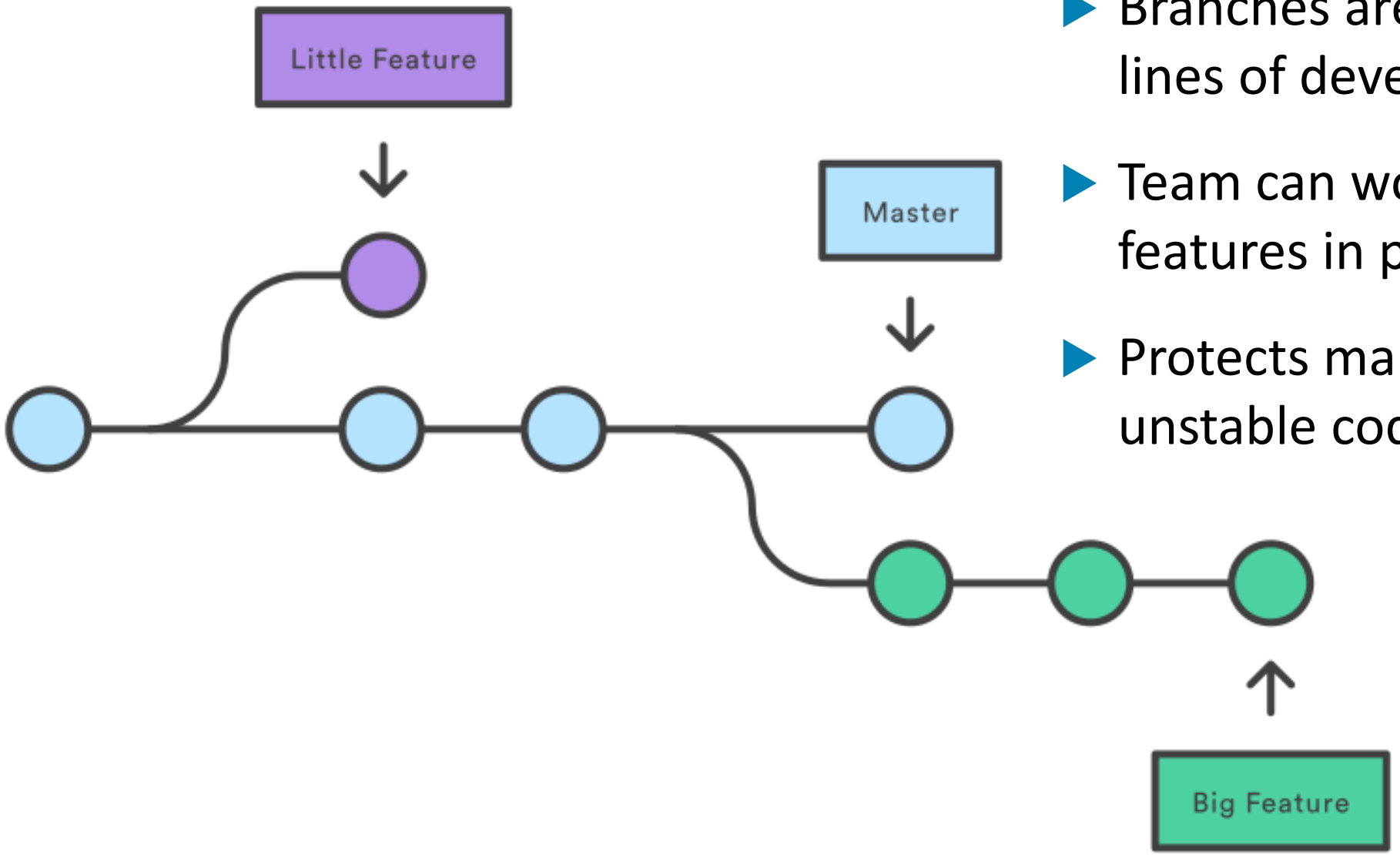
<u>Reset</u> (*#last-good-commit*)

- Discard the „bad commit“
  - <u>Mixed</u> (default): moves HEAD to the desired old commit, but keeps changes in working directory
  - Hard: entirely resets to desired old commit (danger!)
- Use Reset only for local changes!



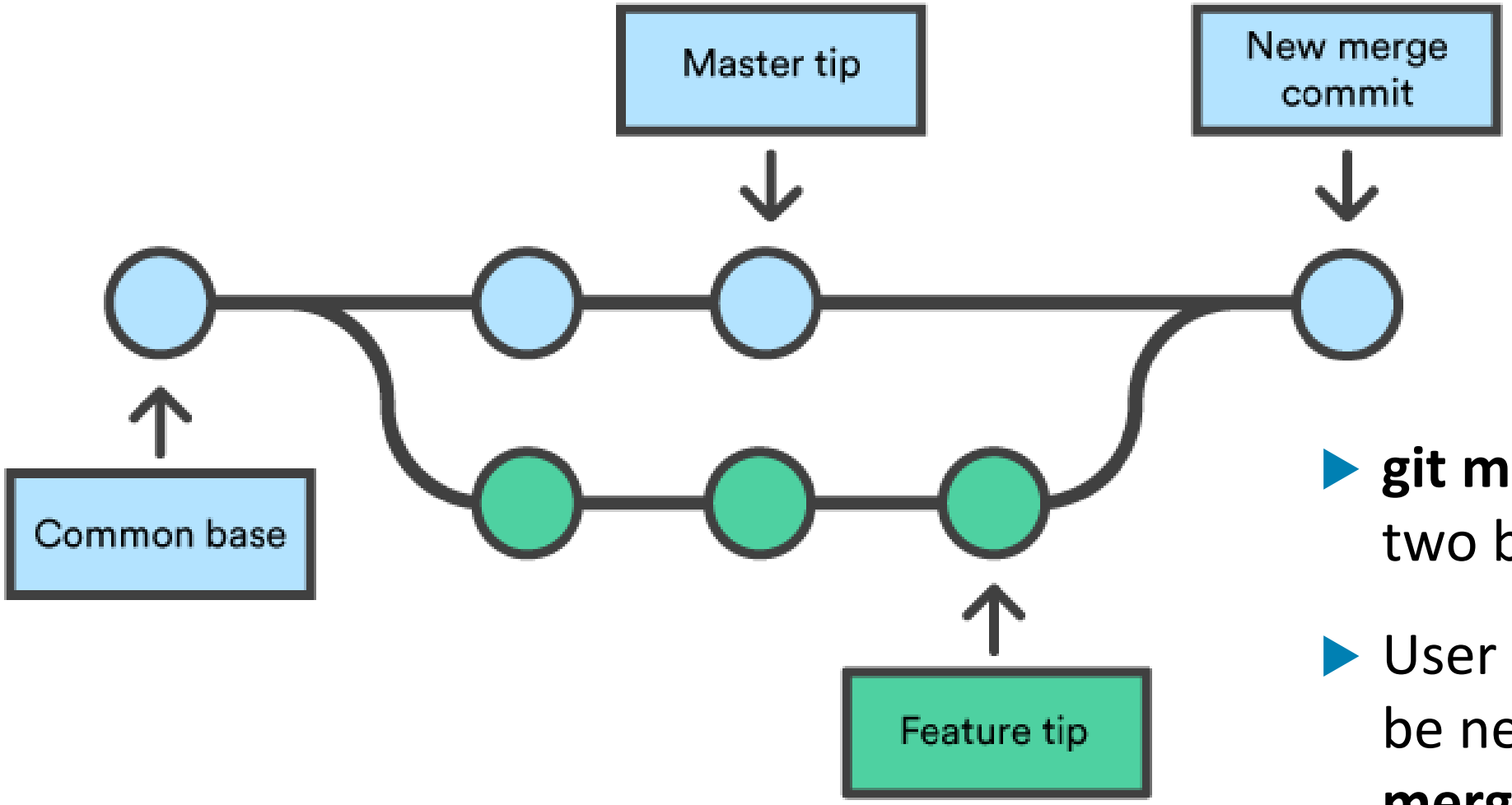


Branching



- Branches are independent lines of development
- Team can work on separate features in parallel
- Protects main branch against unstable code

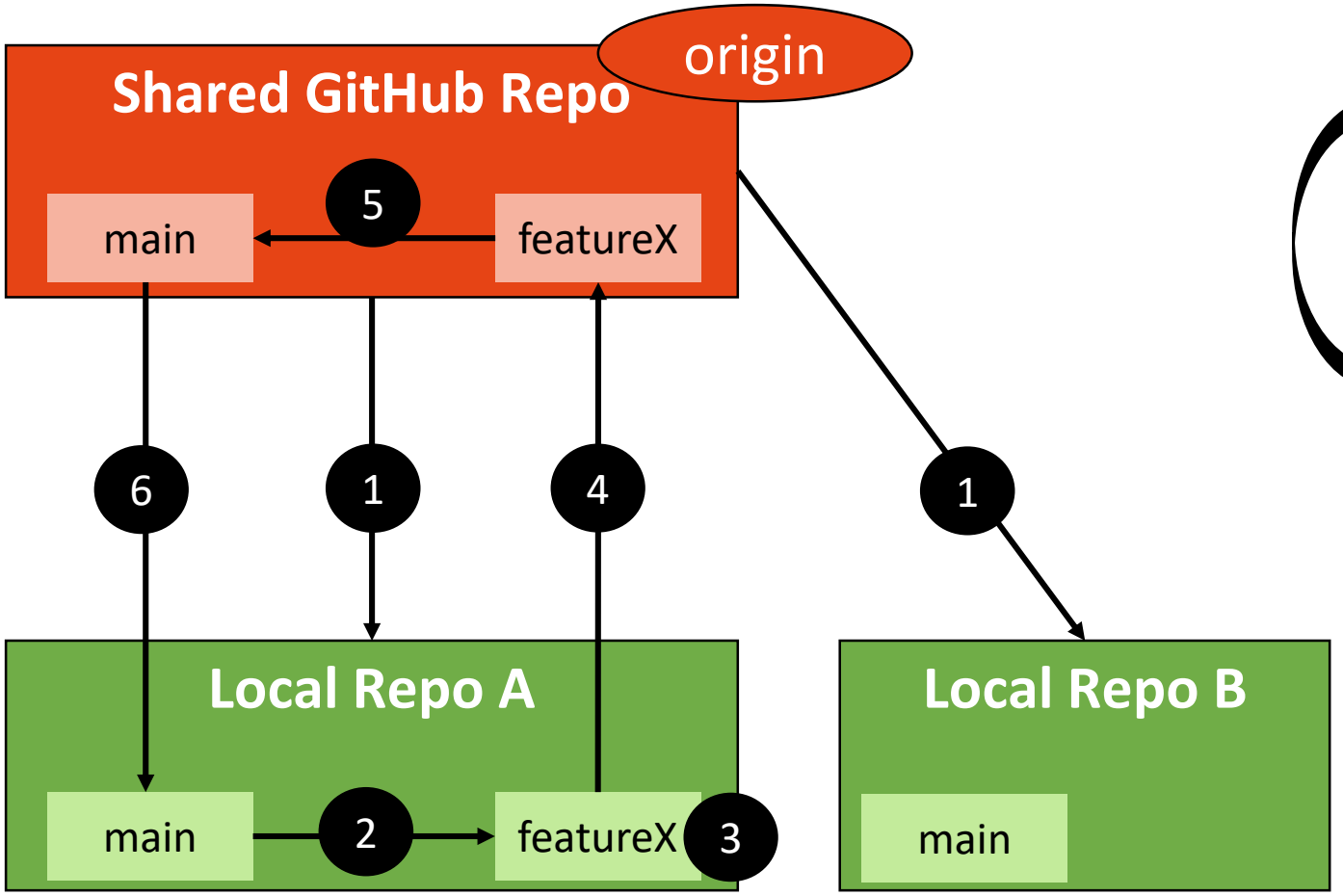
Merging



- **git merge** combines two branches into one.
- User intervention may be needed to resolve **merge conflicts**

Feature Branch Workflow

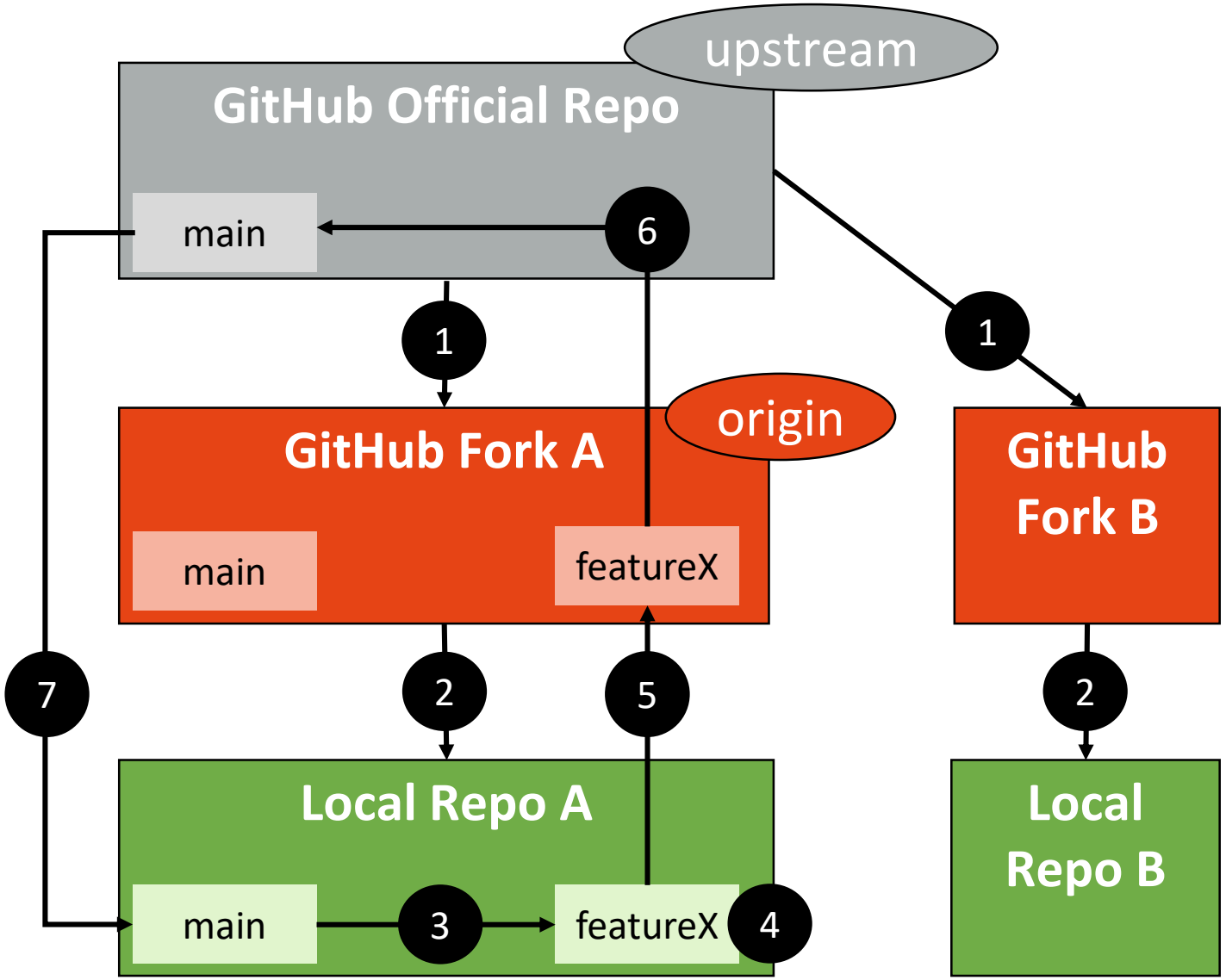
Company Setting



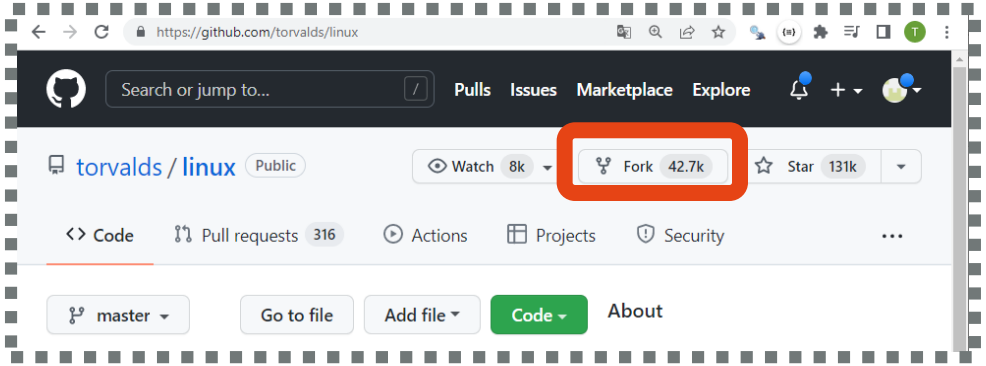
1. git clone <url>
2. git branch featureX
   git checkout featureX
3. edit, git add, git commit
4. git push
5. Pull Request → git merge
6. git pull

Forking Workflow

Open Source Setting

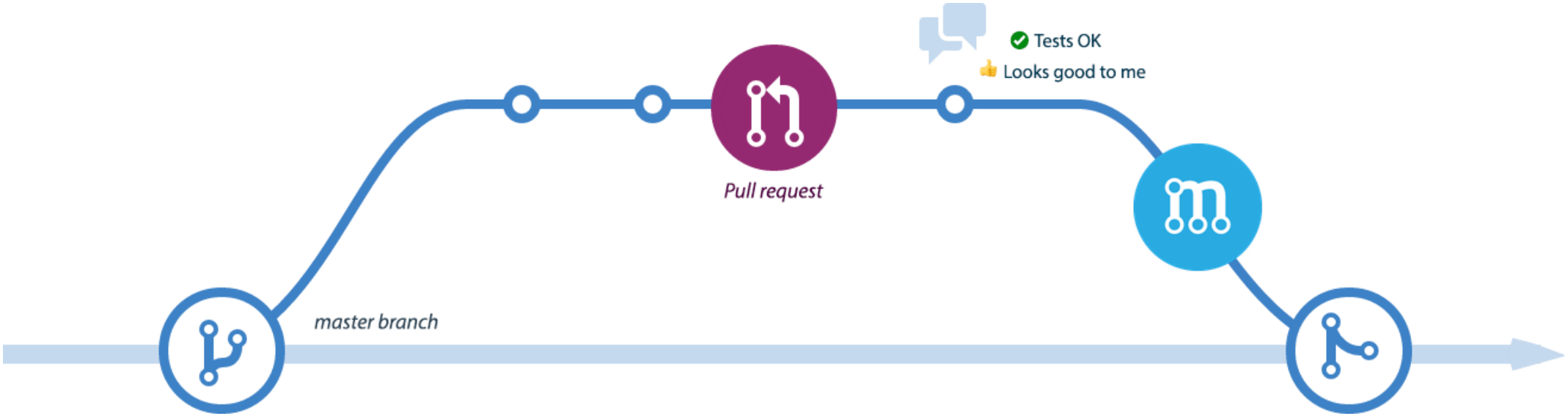


1. Fork
2. git clone <url>
3. git branch featureX
   git checkout featureX
4. edit, git add, git commit
5. git push
6. Pull Request → git merge
7. (git remote add upstream <url>)
   git pull upstream



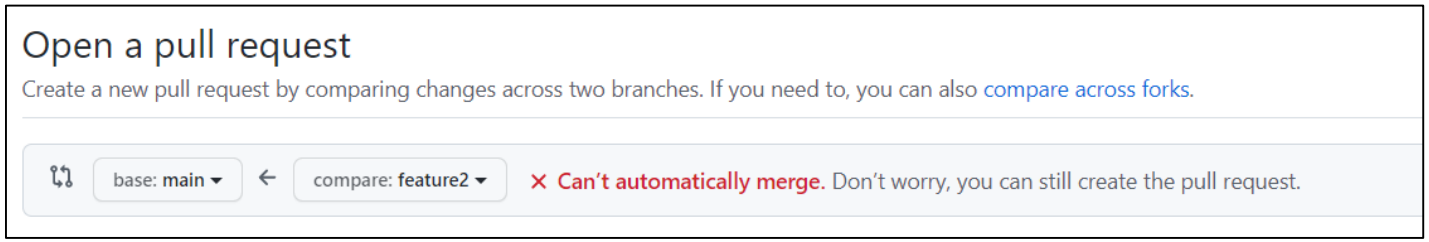
Pull Request

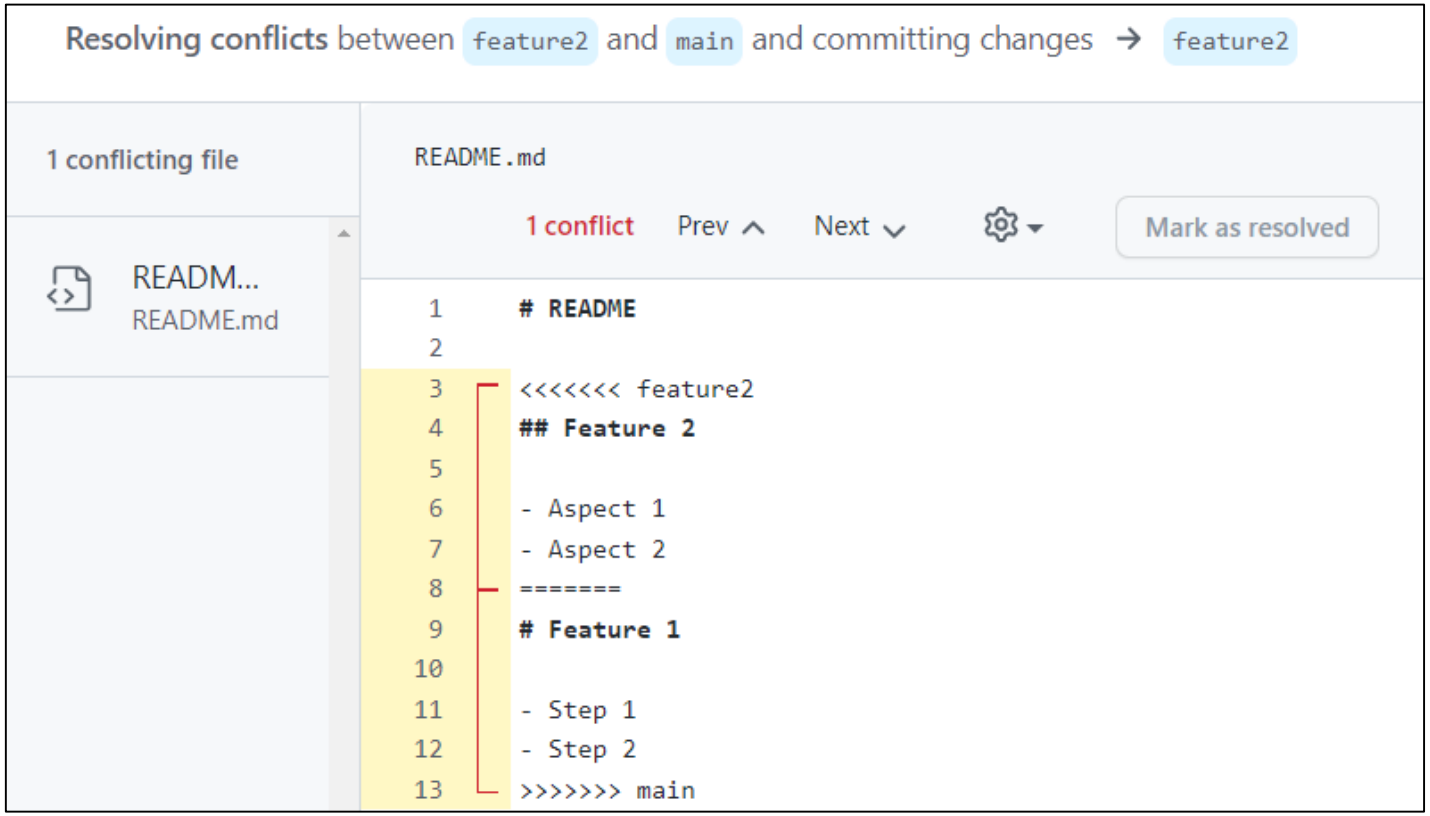
- **A Pull request is a GitHub feature**, not a git command: you request to pull changes from your feature branch ("merge request" on Gitlab)
- May involve multiple iterations of discussions, code reviews, and follow-up commits, before the commit is merged into the main branch



Pull Request with a merge conflict

Fachhochschule Kiel
University of Applied Sciences





Your changes cannot be automatically merged into the main code base

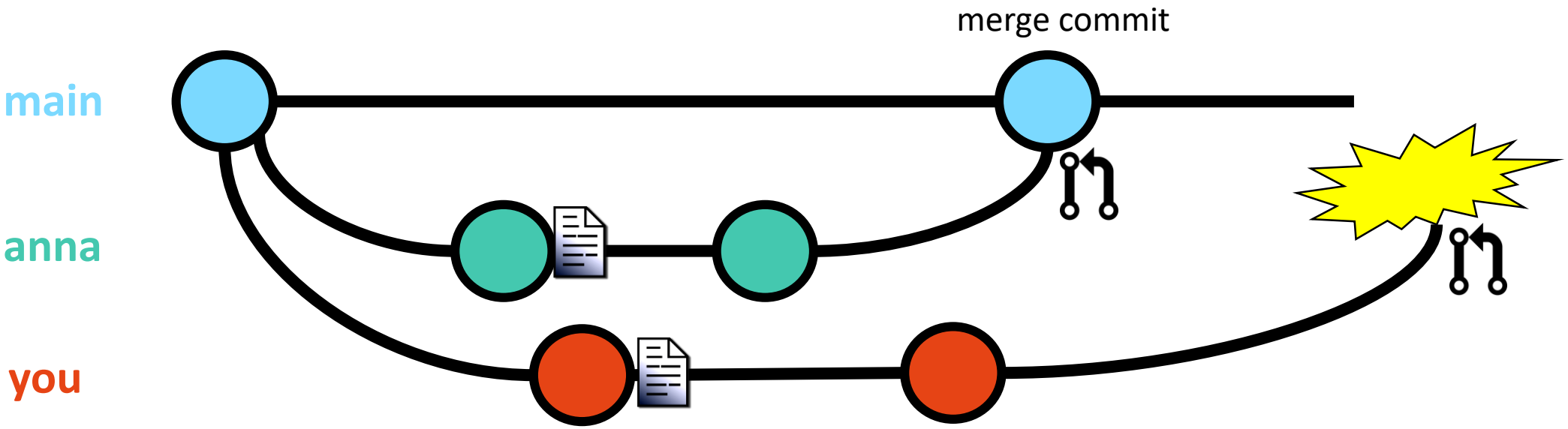
→ **Must be resolved manually**

- Which changes do you want to keep?
- In which order?

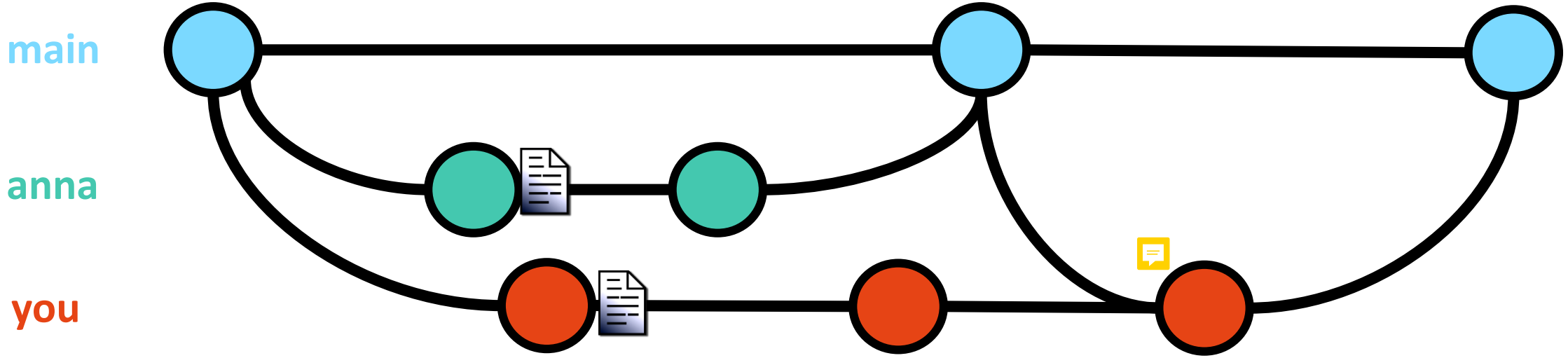
Conflicts can be resolved

- on Github
- or locally

Merge conflict

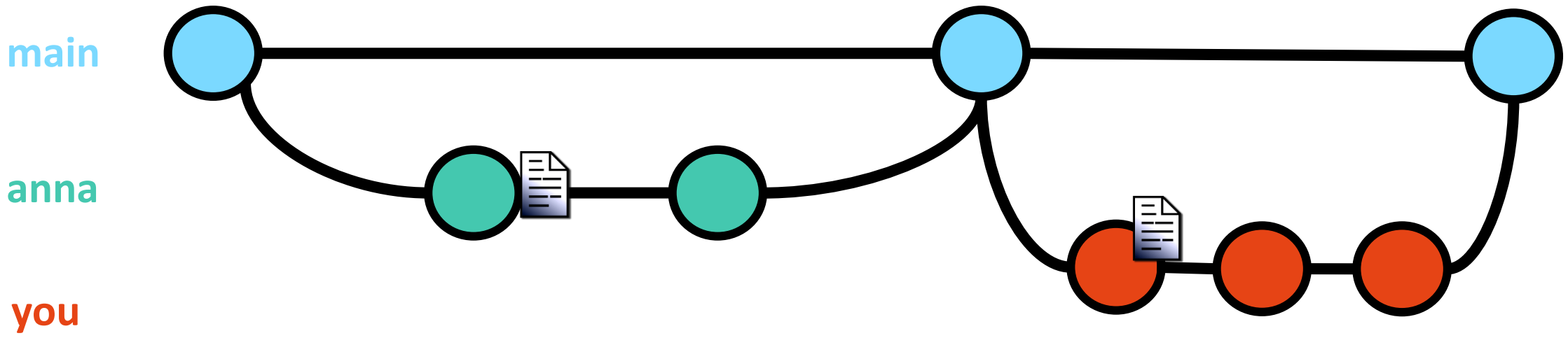


Dealing with a merge conflict (1)



Option 1: Before opening the pull request, check for changes in the official version and **merge** those changes into your branch (dealing manually with the conflicts)

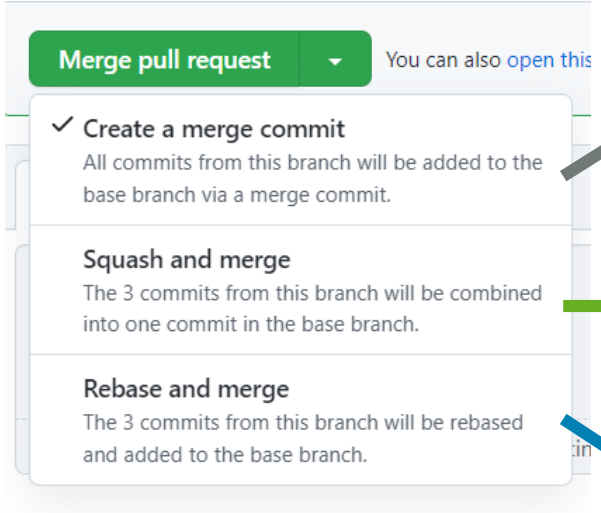
Dealing with a merge conflict (2)



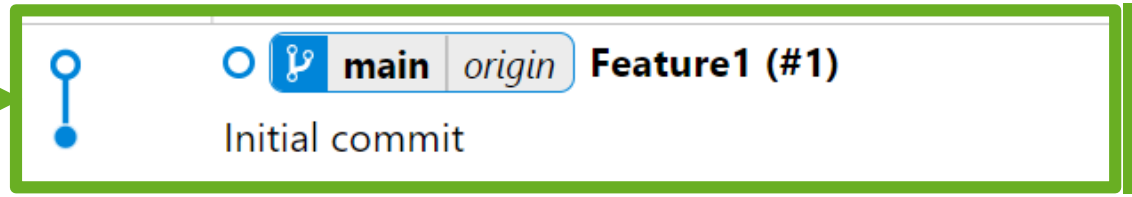
Option 2: Before opening the pull request, check for changes in the official version and **rebase** your changes on top of the current state of the main branch (dealing manually with the conflicts)

Pull Request: 3 ways of merging

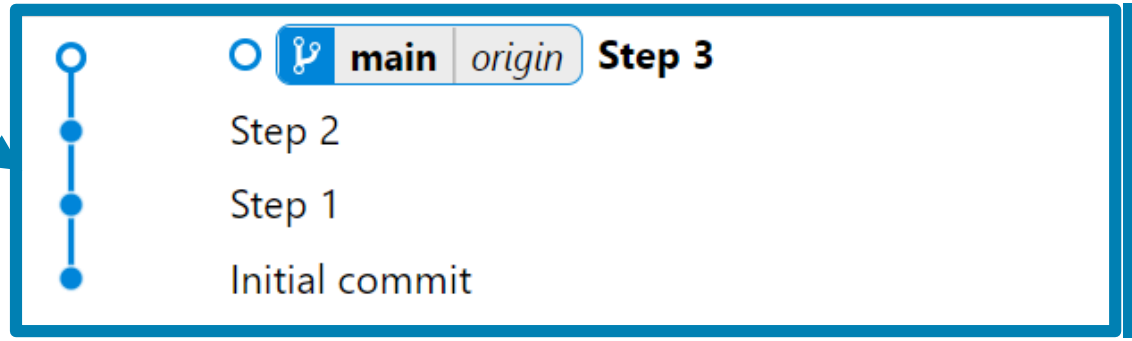
Fachhochschule Kiel
University of Applied Sciences



+ Truthful history of actions
- May pollute project history



+ Short and linear project history
+ Pull requests can still be identified
- Actual single commits are not logged anymore



+ Linear project history
+ Single commits are all part of the history
- Feature branches / pull requests cannot be identified any more

Further aspects

Fachhochschule Kiel
University of Applied Sciences

- **.gitignore:** ensure that certain files not tracked by Git remain untracked.
  - e.g. password files (.env), .ipynb_checkpoints, very large binary files
  - Github provides standard gitignore files for Python and other languages
  - If you want to ignore a file that is already checked in, you must untrack the file before you add a rule to ignore it.
- **Large files**
  - GitHub limits the size of files allowed in repositories to 100 MB
  - To track files beyond this limit, you can use Git Large File Storage (2 GB file size limit)
  - Git LFS stores references to the file in the Github repository. The actual file is stored separately (up to 2 GB file size limit).

Further Resources

Fachhochschule Kiel
University of Applied Sciences

- Tutorial on Git (Workflows) (available in German)
- Git reference book (available in German)
- Git commands: cheat sheet
- 10-minute reads mostly on GitHub topics (by GitHub)
- Glossary of Git and GitHub terms (by Github)
- Tutorial for R users