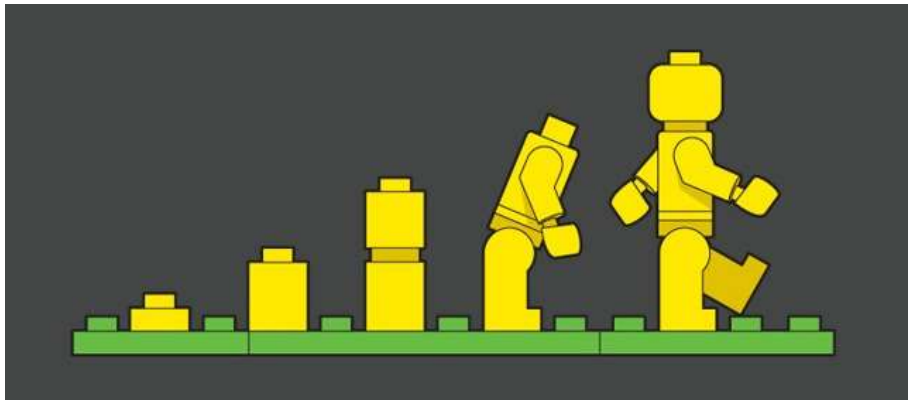
Version Control



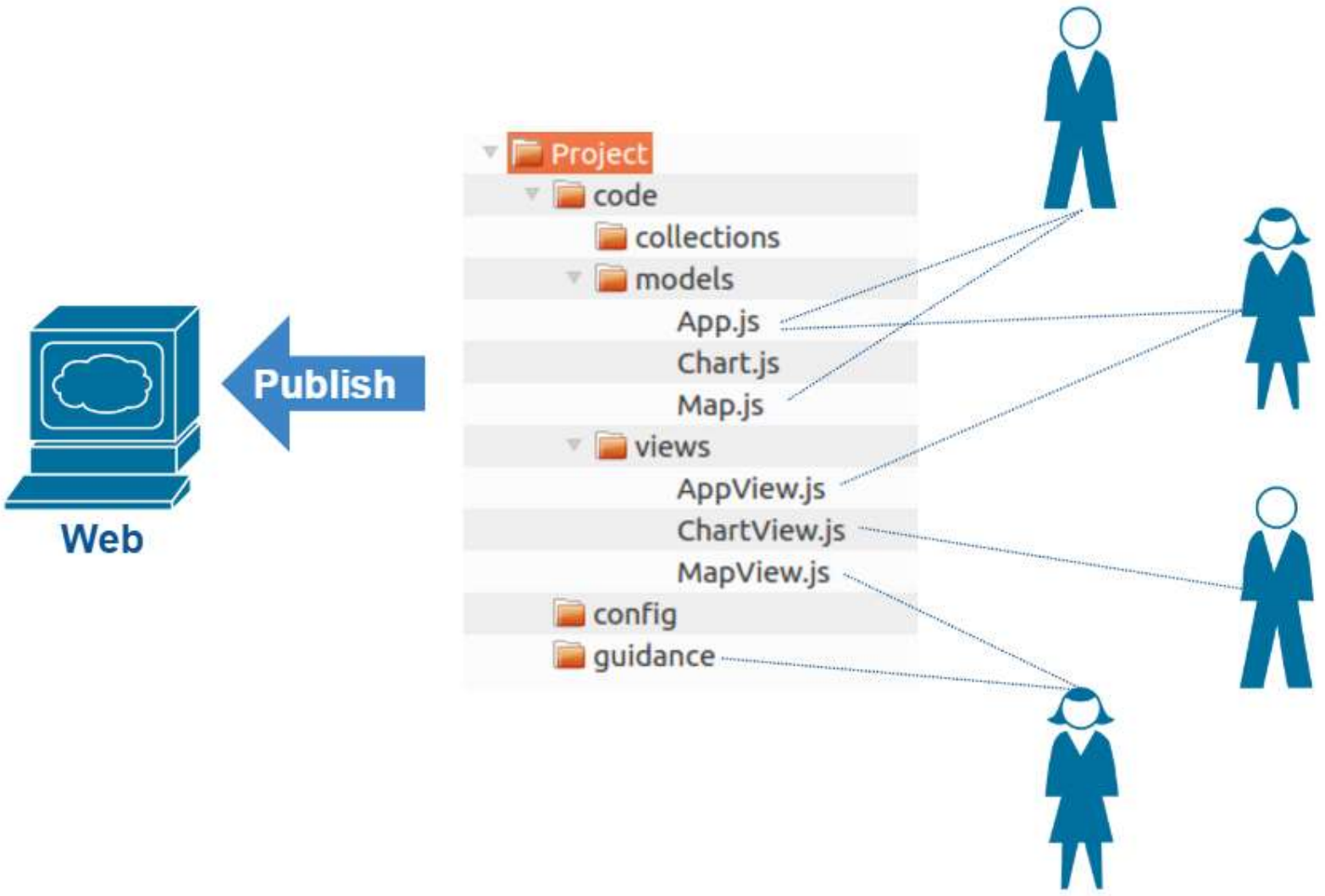
What you will learn today

- What version control is
- Why you need it (or maybe you don't?)
- *Session 1:* basic concepts
- *Session 2:* remote hosting
- *Session 3:* team working
- Up and running with tools and account

What is version control?

- *'Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later'*
- Great free book: https://git-scm.com/book/en/v2
- Excellent Atlassian tutorial: https://www.atlassian.com/git/tutorials/what-is-version-control

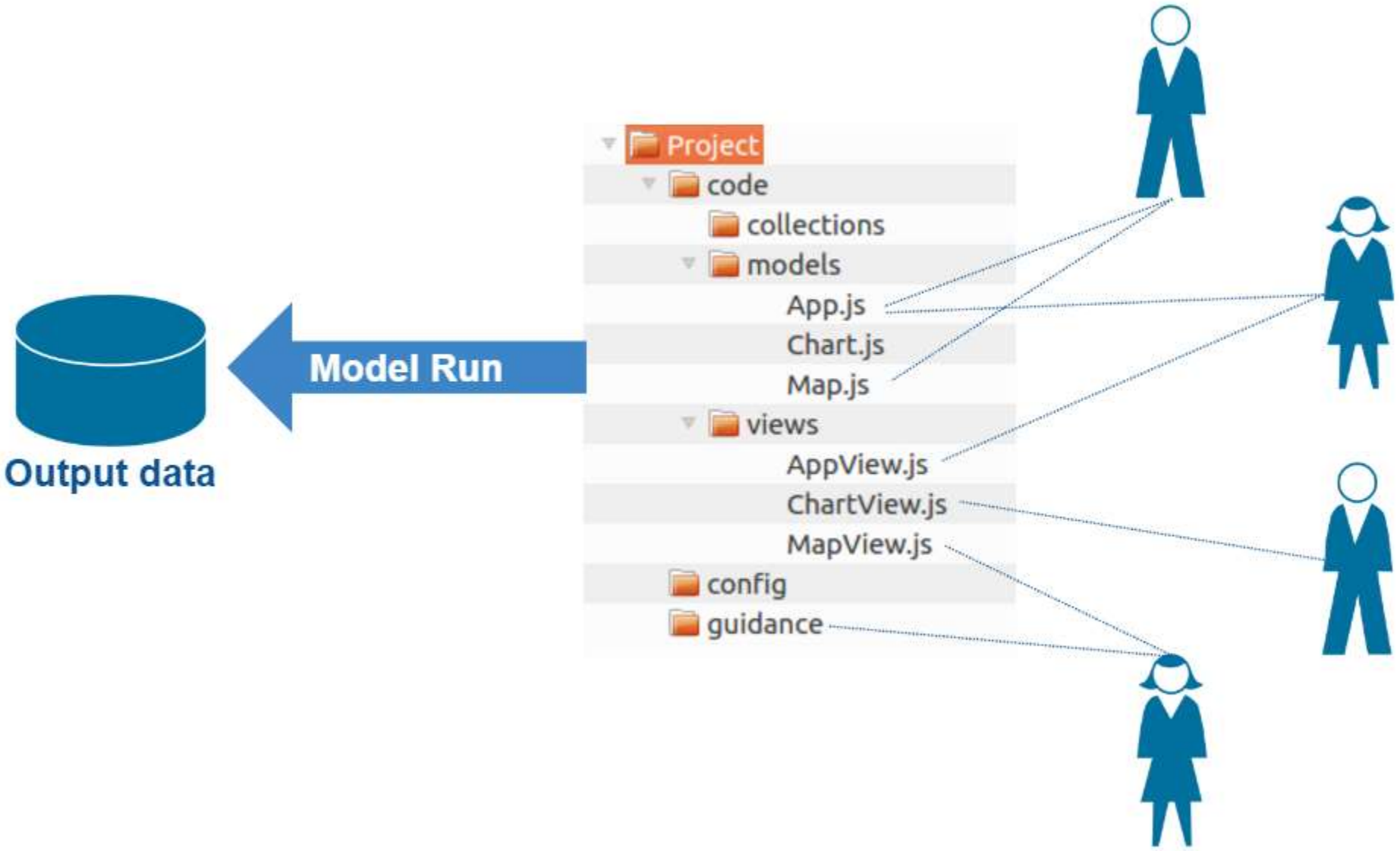
Typical scenario in our group



Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

Typical scenario with a model run?



What do you want from version control?

Recovery

Reproducibility

Track changes

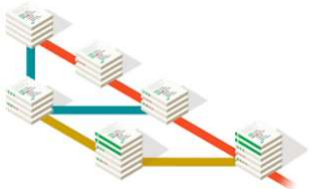
Audit and Traceability

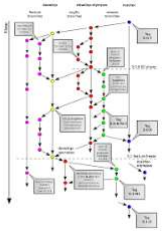
Funding requirement

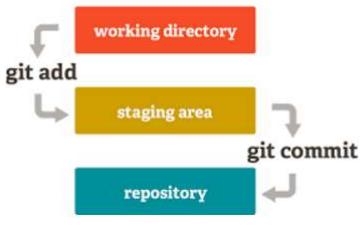
Safety and resilience

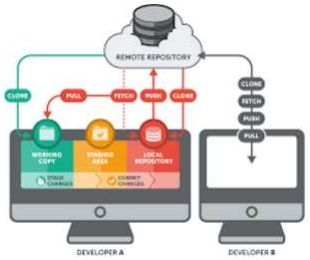
Collaboration

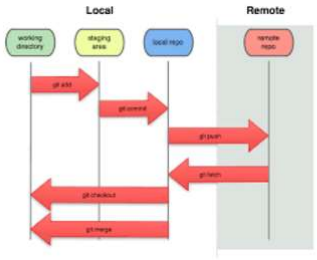
Experiments on code

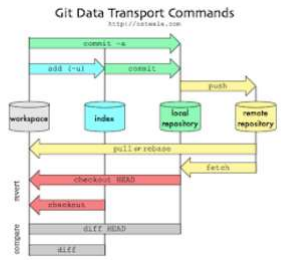



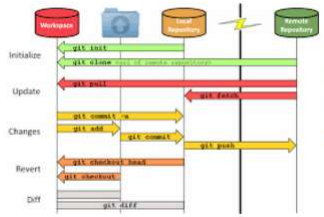
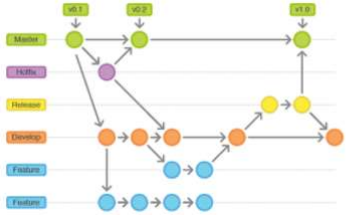




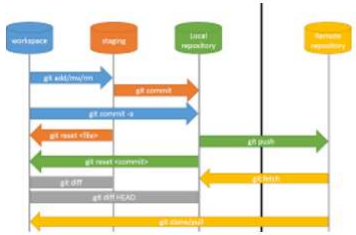




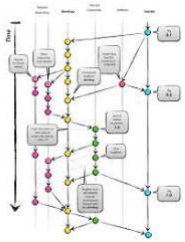











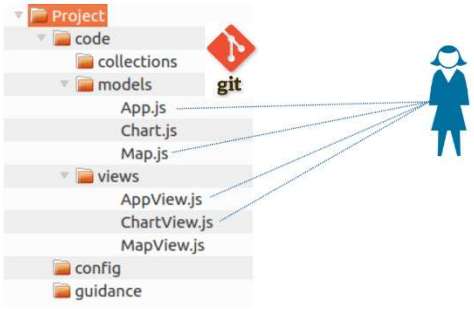

Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

Three learning goals in three sessions

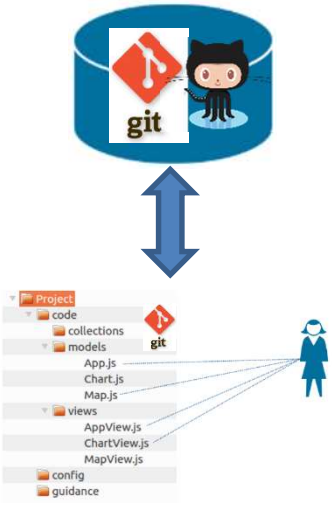
Session 1

Basic principles



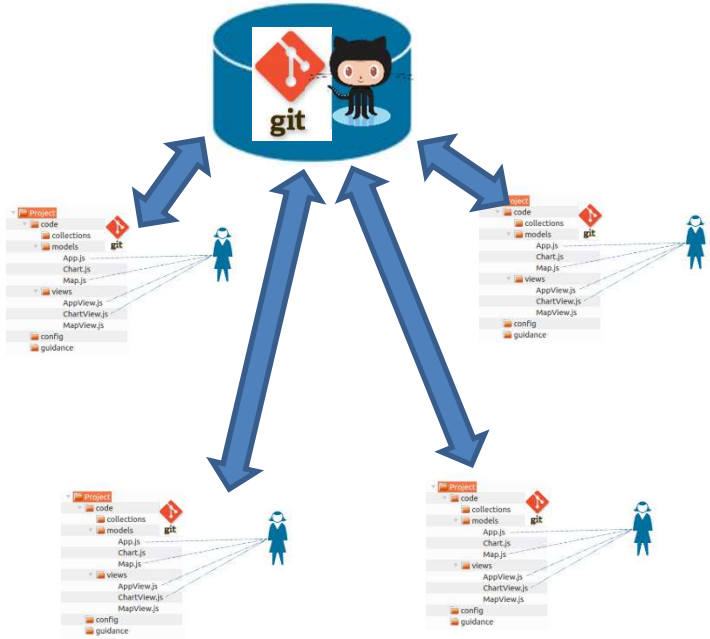
Session 2

Remote hosting



Session 3

Team working

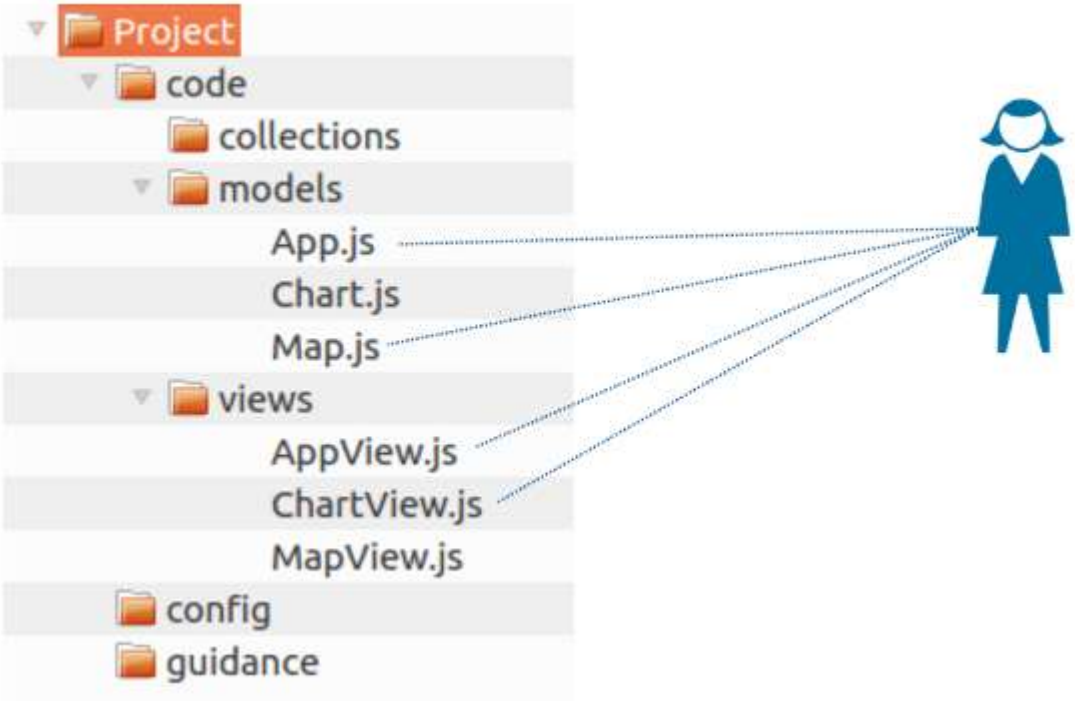


Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

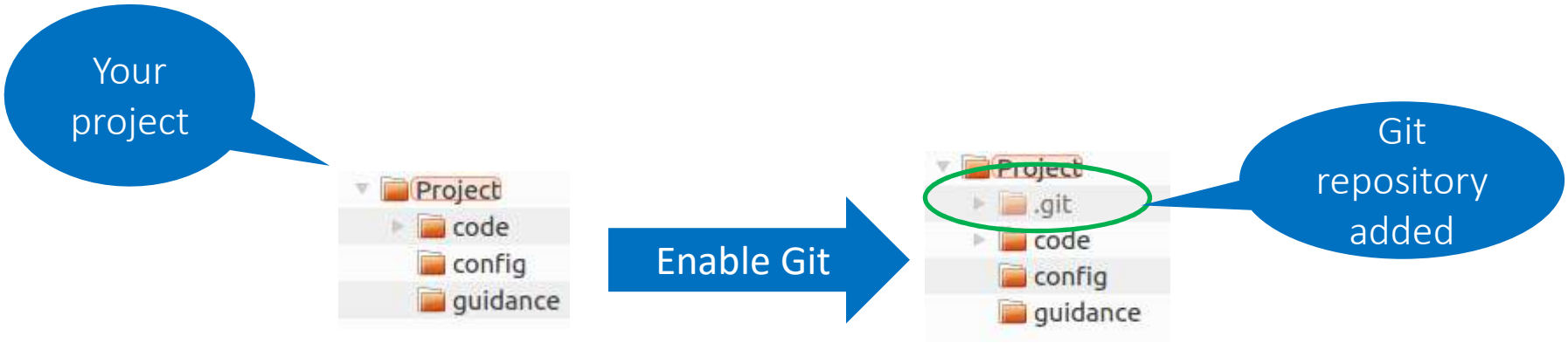
Session 1: Principles

- Version control a set of files on my local drive



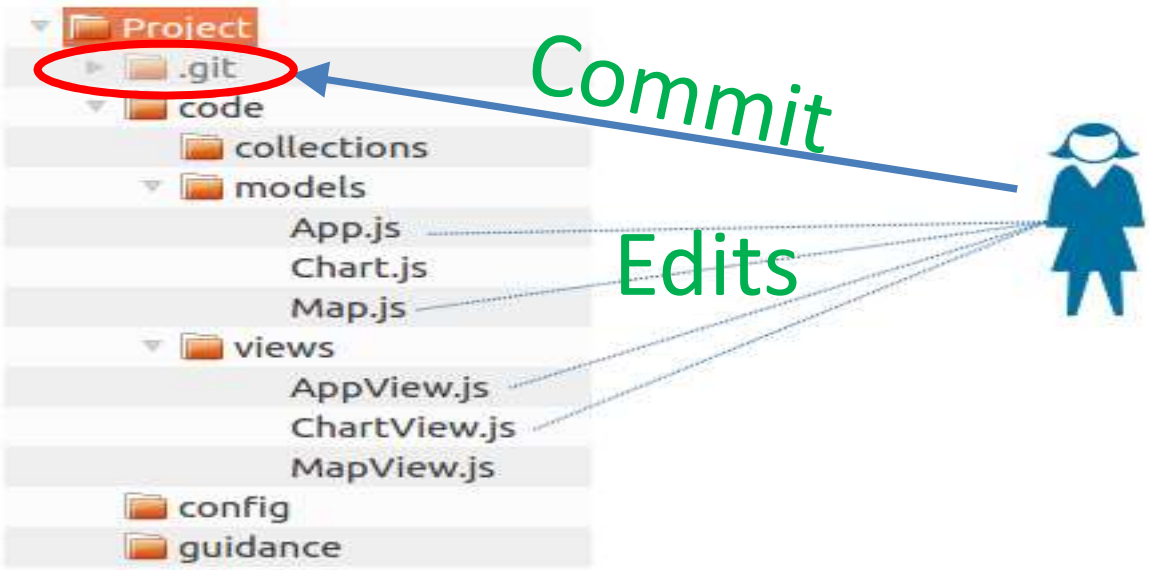
Session 1: Principles

Add Git to your project



Session 1: Principles

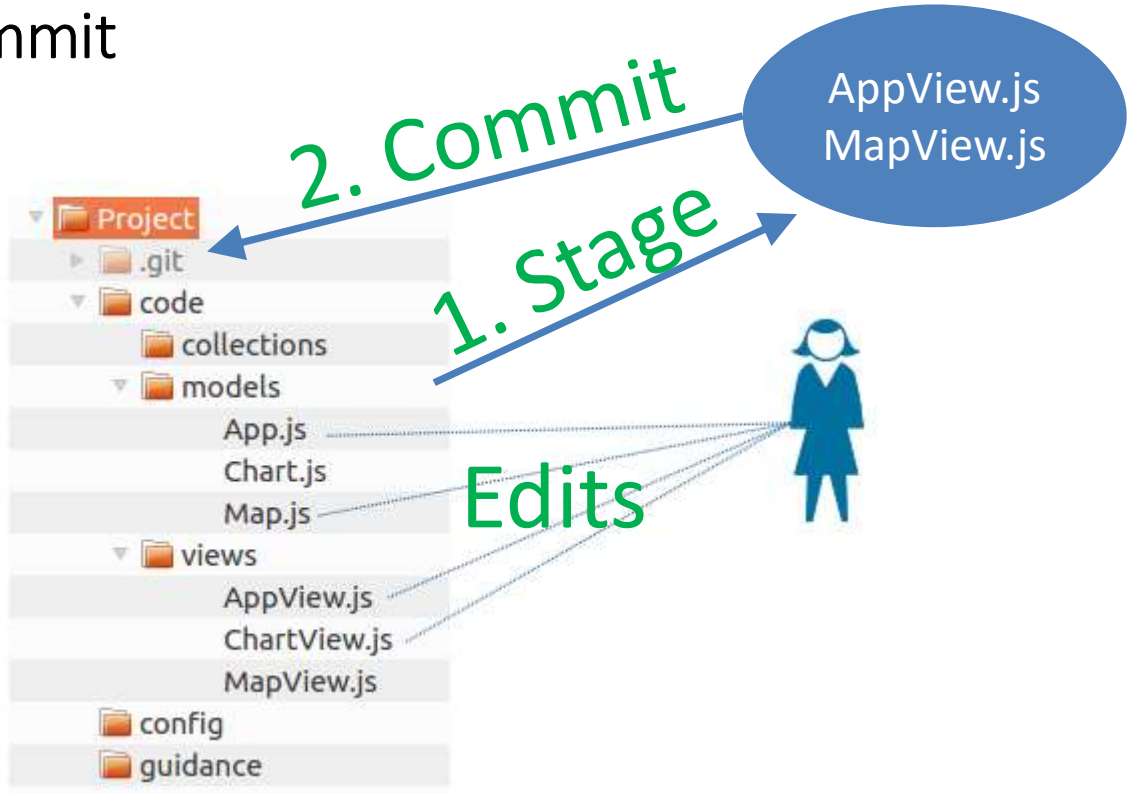
- After editing project files, save your changes to Git
- This is called **Committing**
- Git does **not automatically** save your changes
- Committing records who did what when



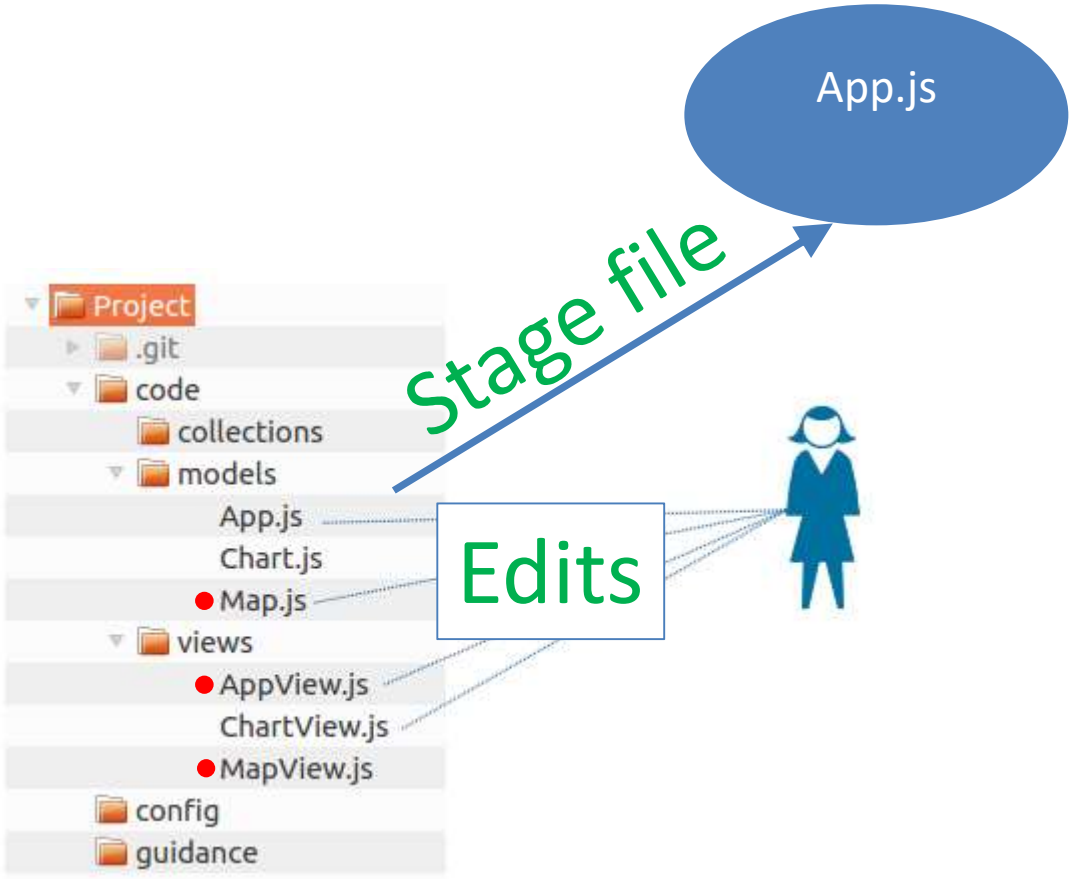
Session 1: Principles

Committing is a two step process:

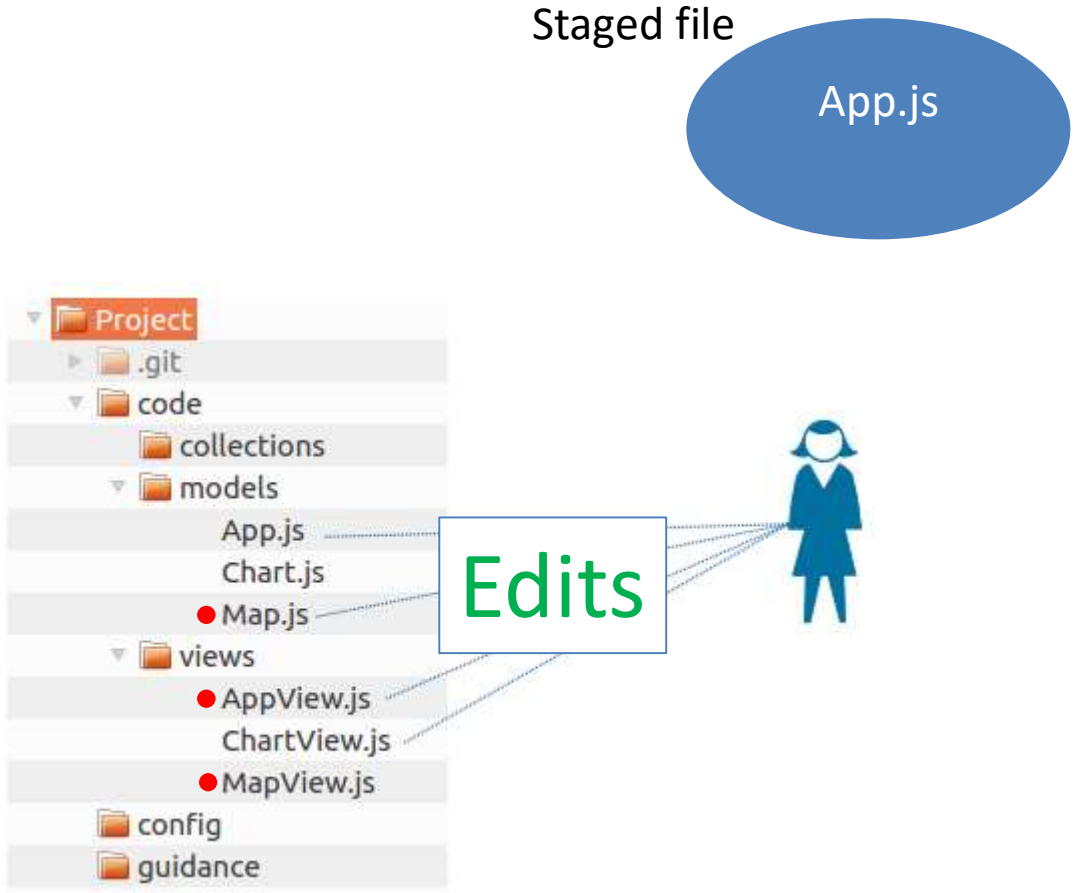
1. Choose the changes you want in the commit – called **Staging**
2. Do the **Commit**



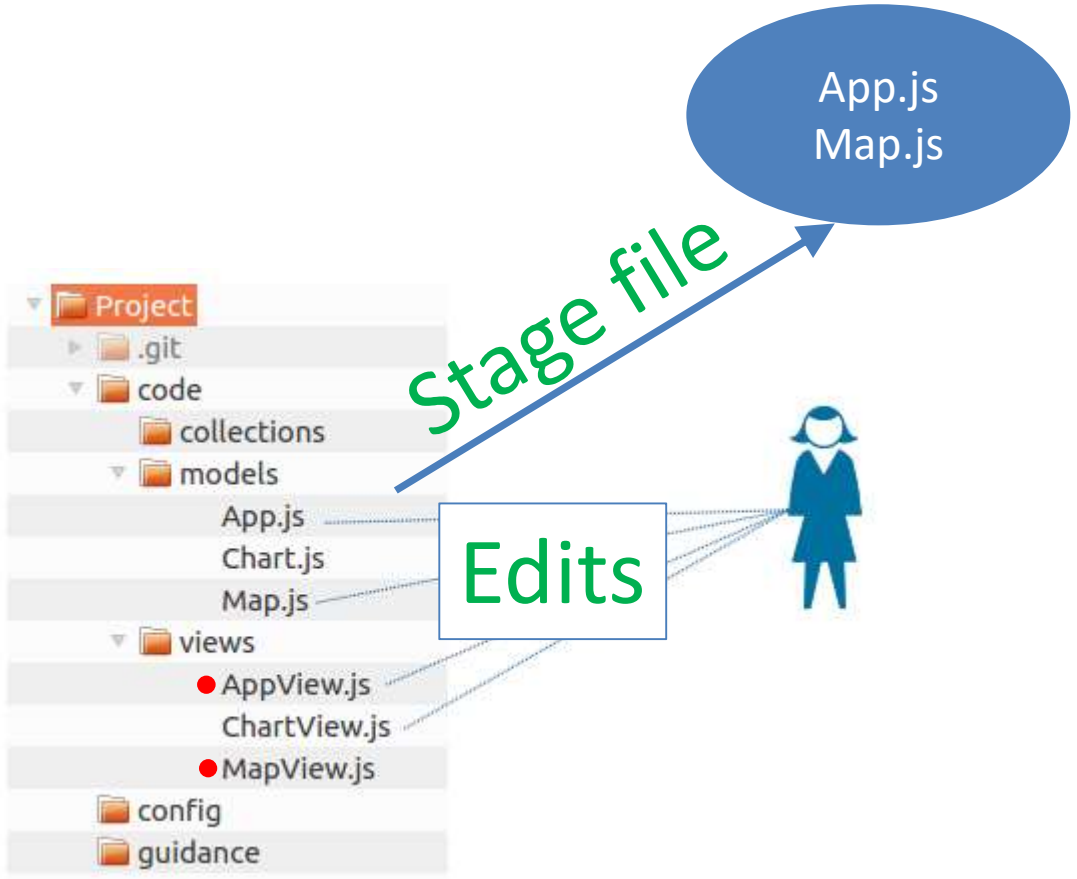
Session 1: Principles



Session 1: Principles



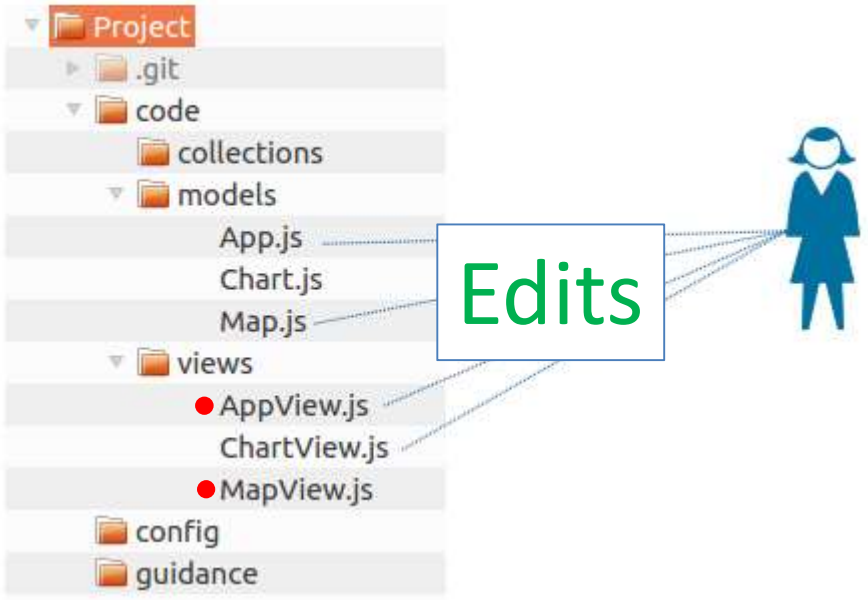
Session 1: Principles



Session 1: Principles

Staged files

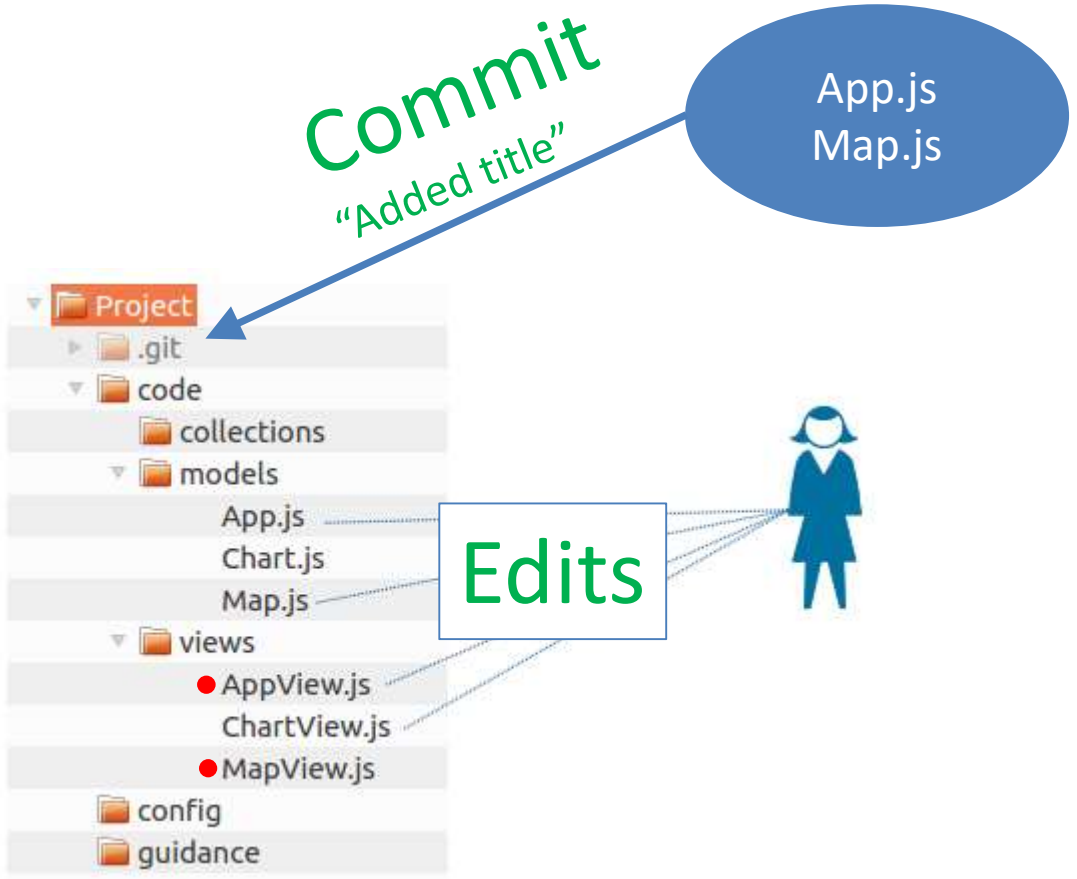
App.js
Map.js



Session 1: Principles

Commit log

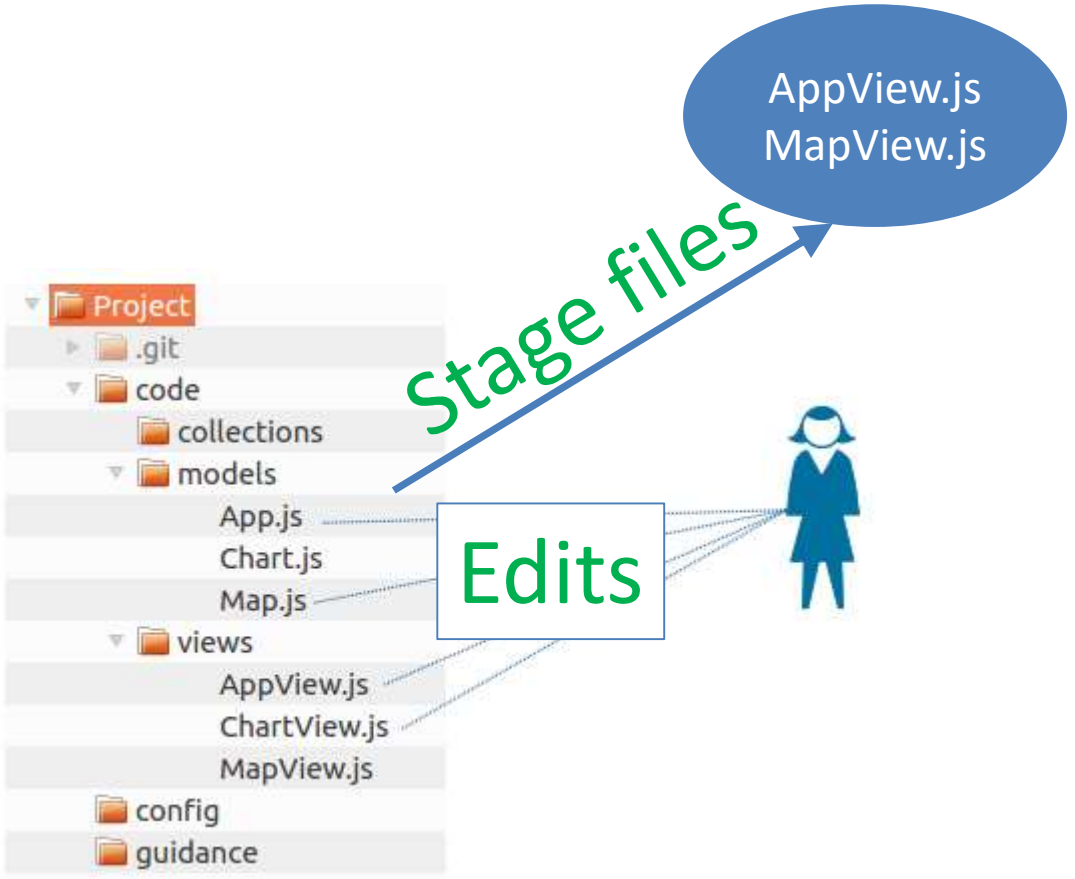
1 Added title *Jon 01-04-2017 13:42*



Session 1: Principles

Commit log

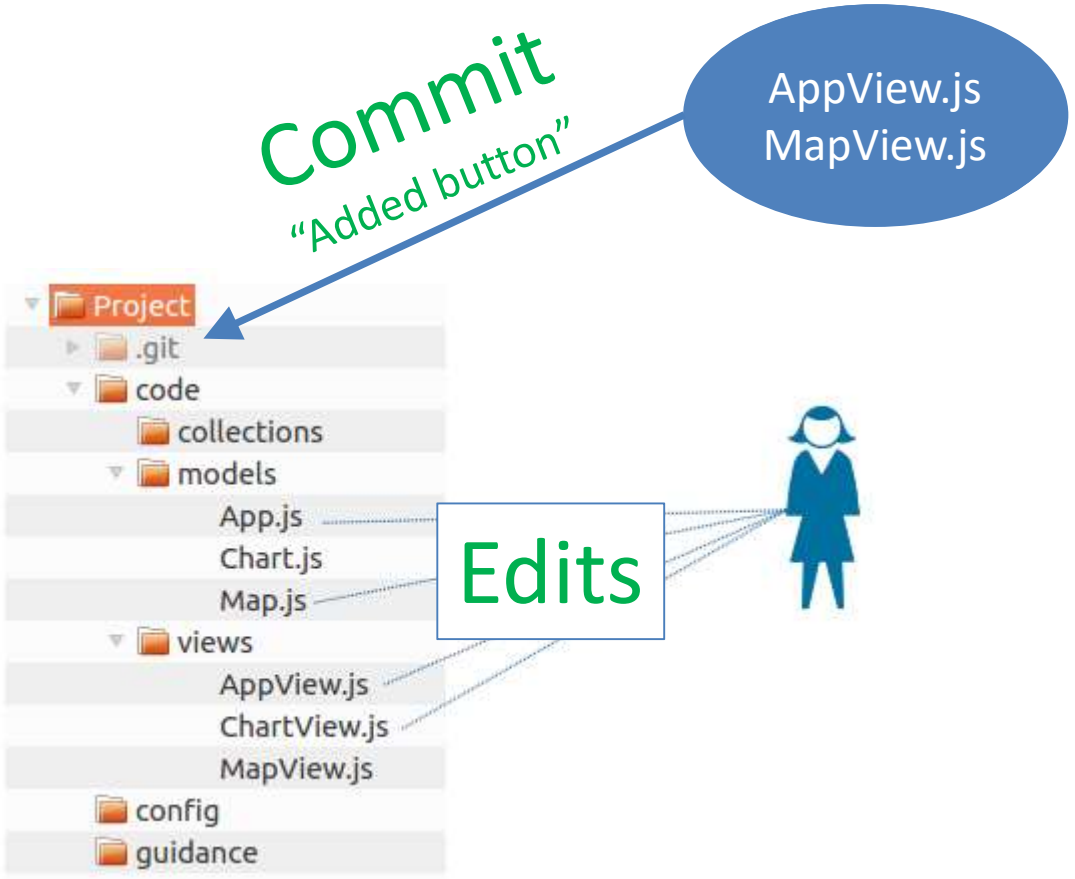
1 Added title *Jon 01-04-2017 13:42*



Session 1: Principles

Commit log

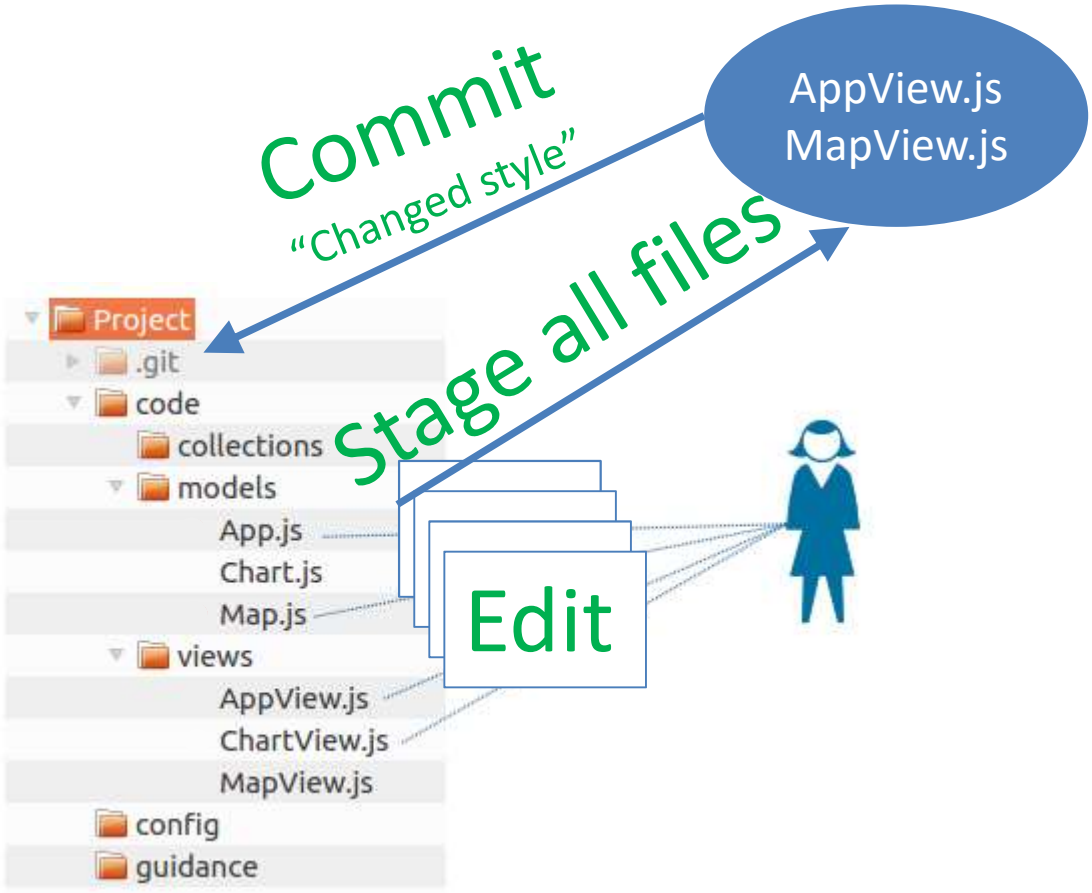
1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*



Session 1: Principles

Commit log

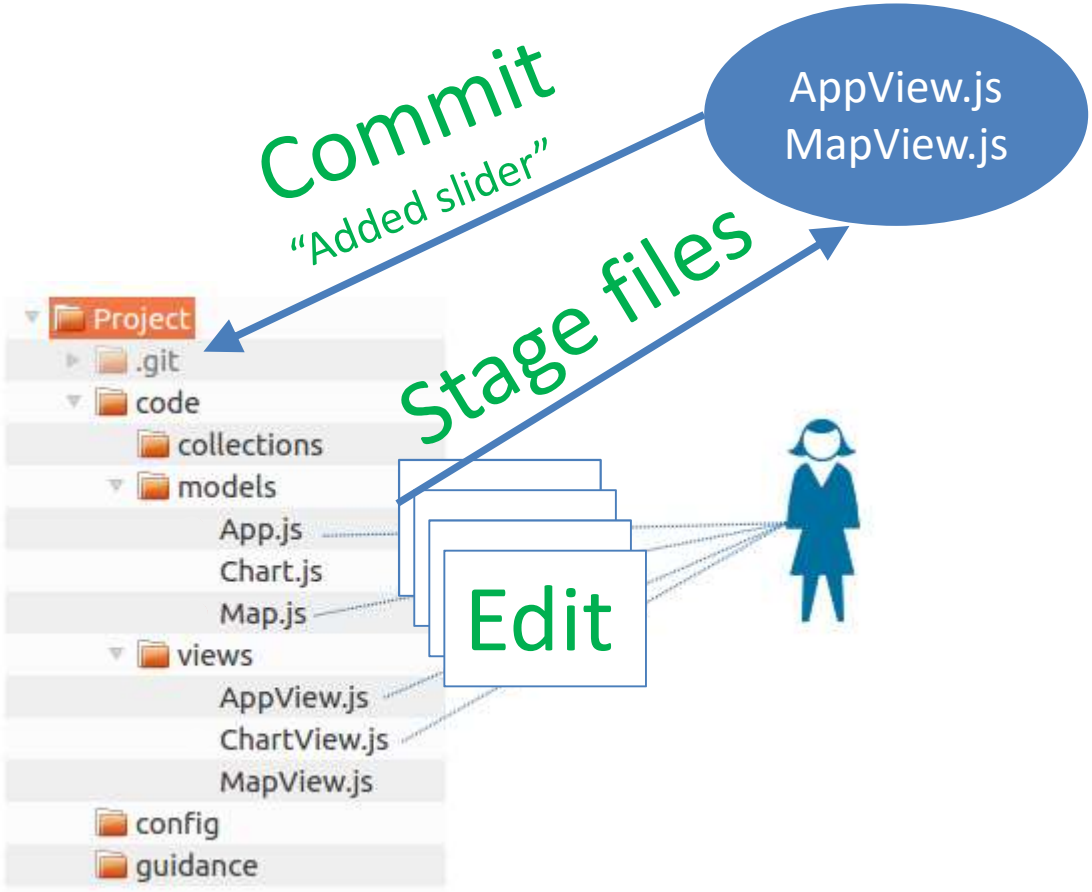
1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*



Session 1: Principles

Commit log

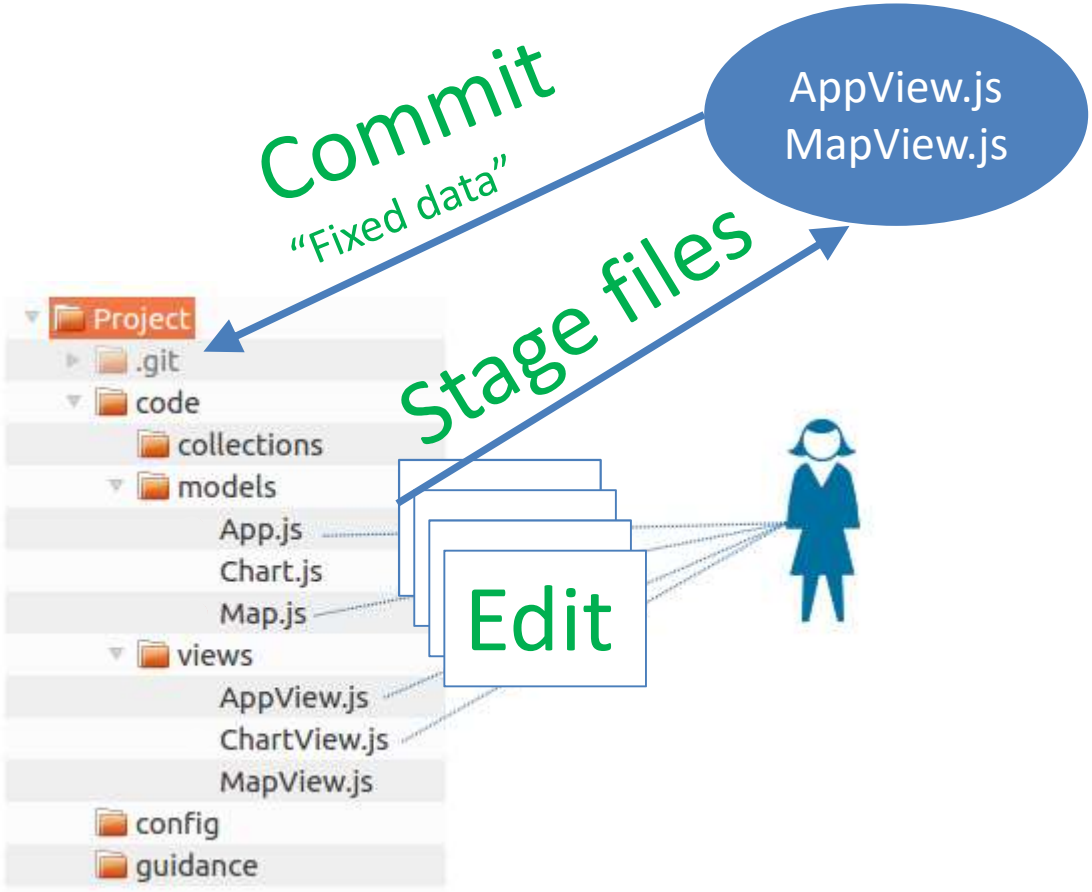
1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*



Session 1: Principles

Commit log

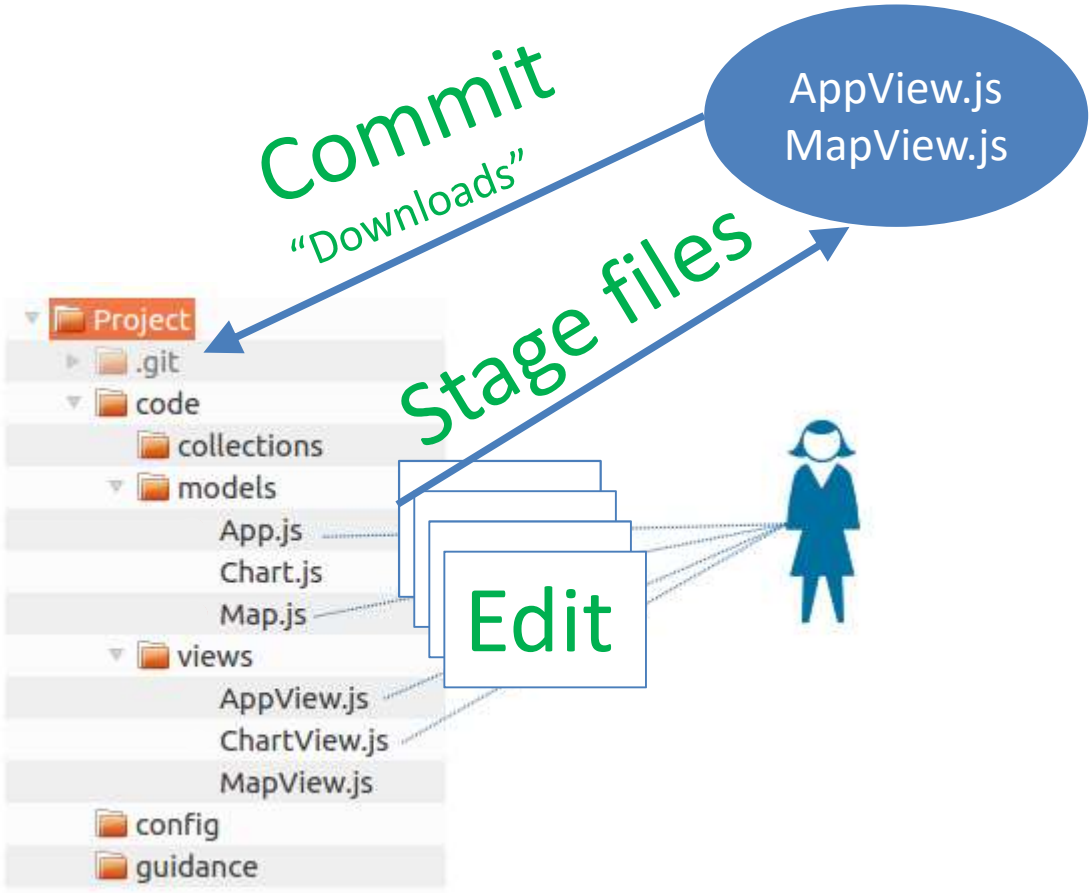
1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*
5 Fixed data *Jon 03-04-2017 10:15*



Session 1: Principles

Commit log

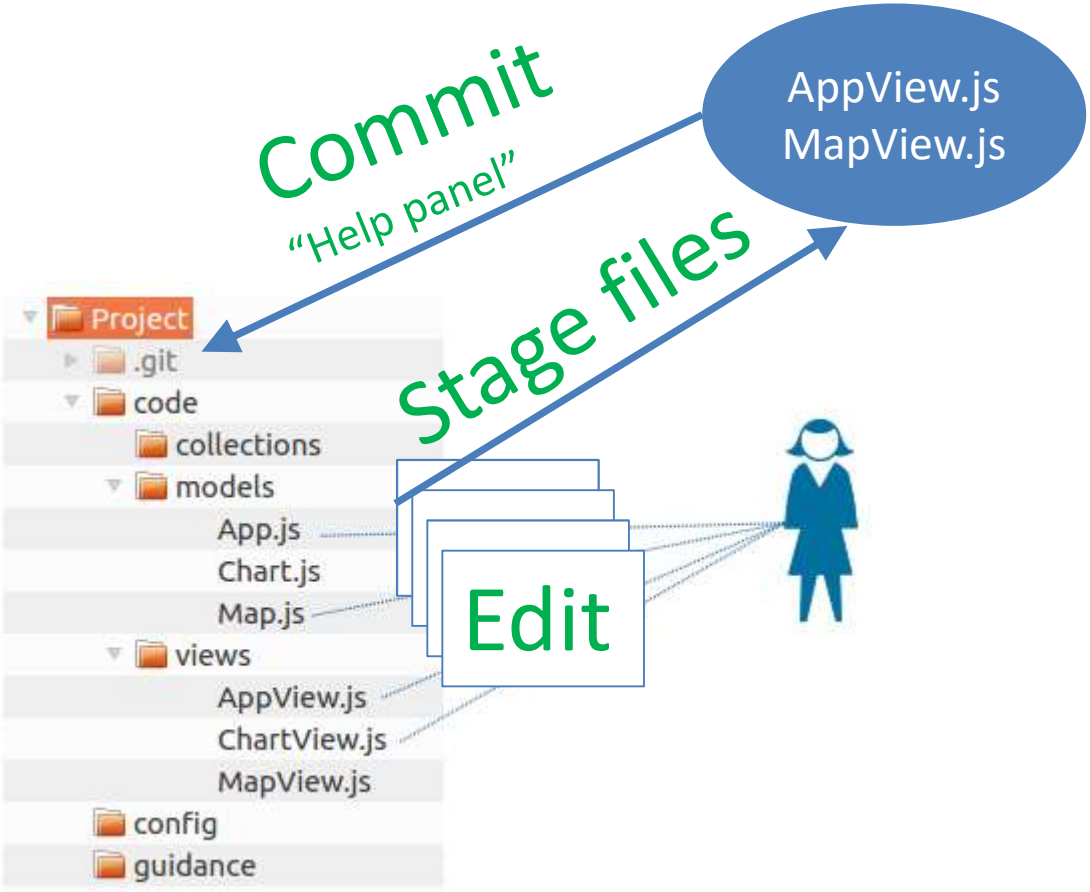
1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*
5 Fixed data *Jon 03-04-2017 10:15*
6 Downloads *Jon 03-04-2017 10:15*



Session 1: Principles

Commit log

1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*
5 Fixed data *Jon 03-04-2017 10:15*
6 Downloads *Jon 03-04-2017 10:15*
7 Help panel *Jon 03-04-2017 13:27*



Session 1: Principles

Commit log

1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*
5 Fixed data *Jon 03-04-2017 10:15*
6 Downloads *Jon 03-04-2017 10:15*
7 Help panel *Jon 03-04-2017 13:27*

- Snapshot of files at each commit
- Who did what when
- Checkout previous version

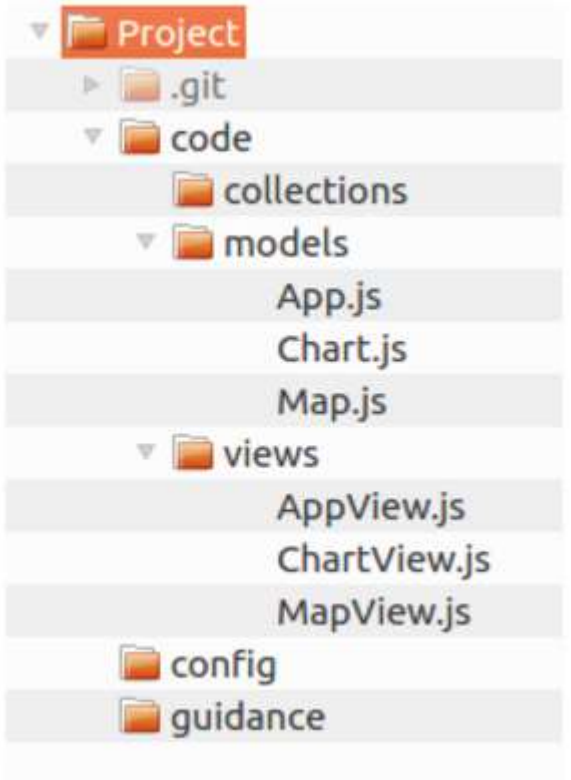
Session 1: Principles

Commit log

1 Added title *Jon 01-04-2017 13:42*
2 Added button *Jon 01-04-2017 14:47*
3 Changed style *Jon 01-04-2017 15:23*
4 Added slider *Jon 03-04-2017 08:32*
5 Fixed data *Jon 03-04-2017 10:15*
6 Downloads *Jon 03-04-2017 10:15*
7 Help panel *Jon 03-04-2017 13:27*

Checkout

All files as they were at commit 3



Working copy

Session 1: Principles

1 Working copy

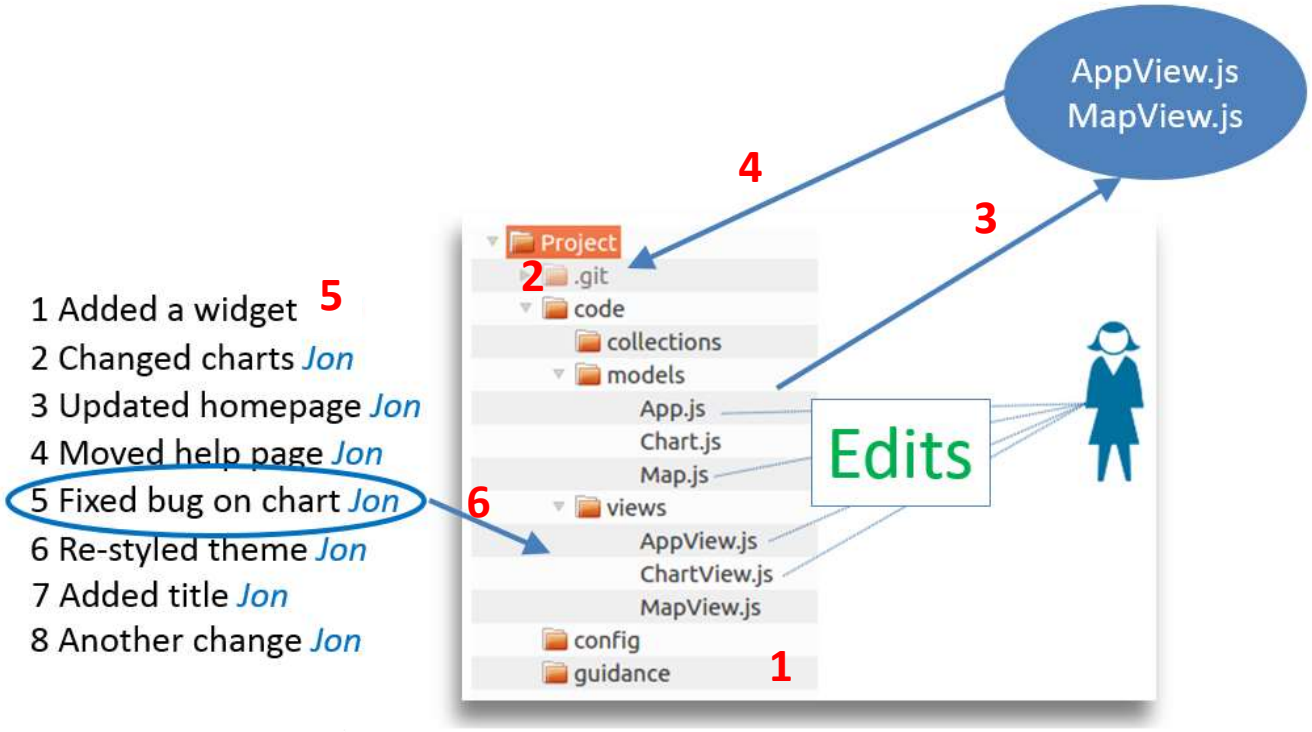
2 Git repository

3 Stage files

4 Commit files

5 Commit log

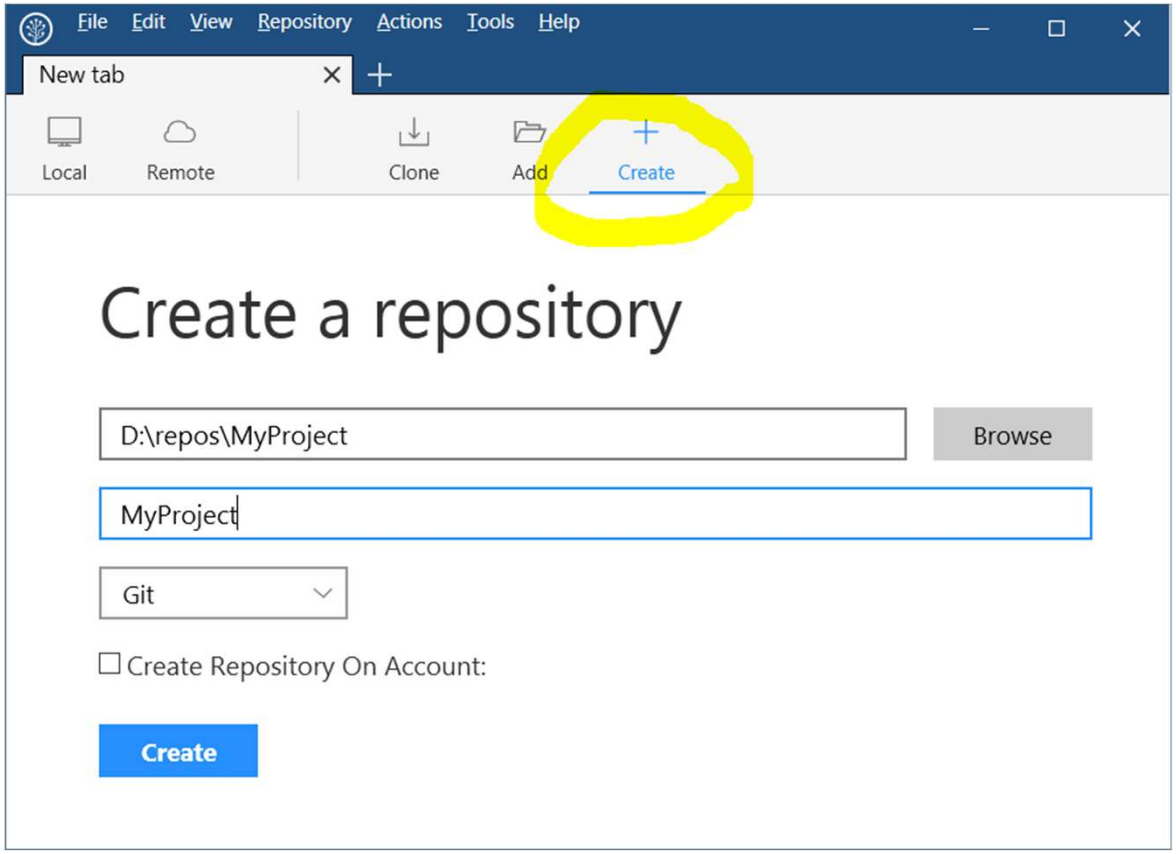
6 Checkout



Session 1: Principles

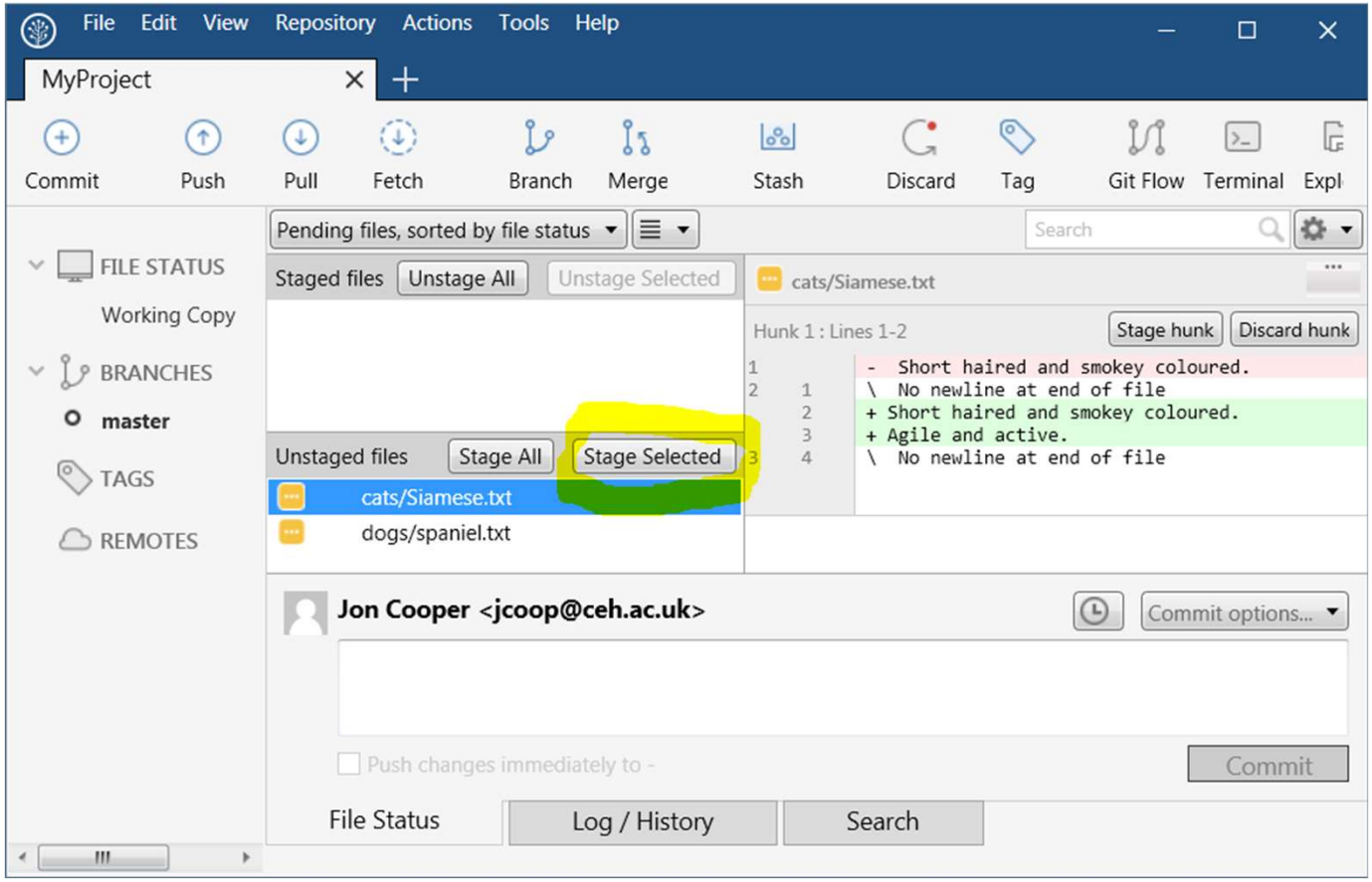
All exercises use SourceTree

https://nerc-ceh.github.io/version_control/install_sourcetree_windows



Session 1: Principles

Example of selecting files to stage in SourceTree

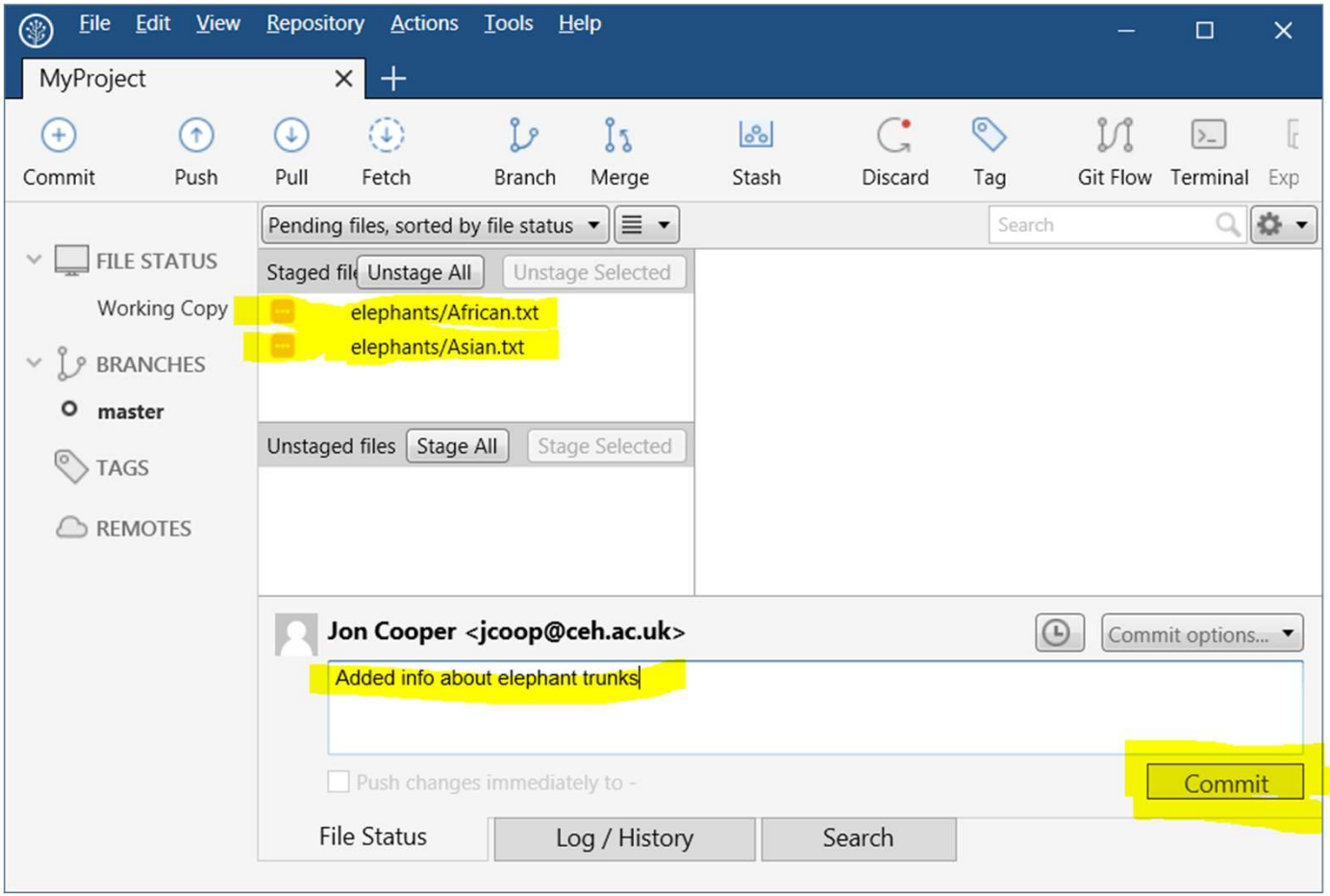


Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

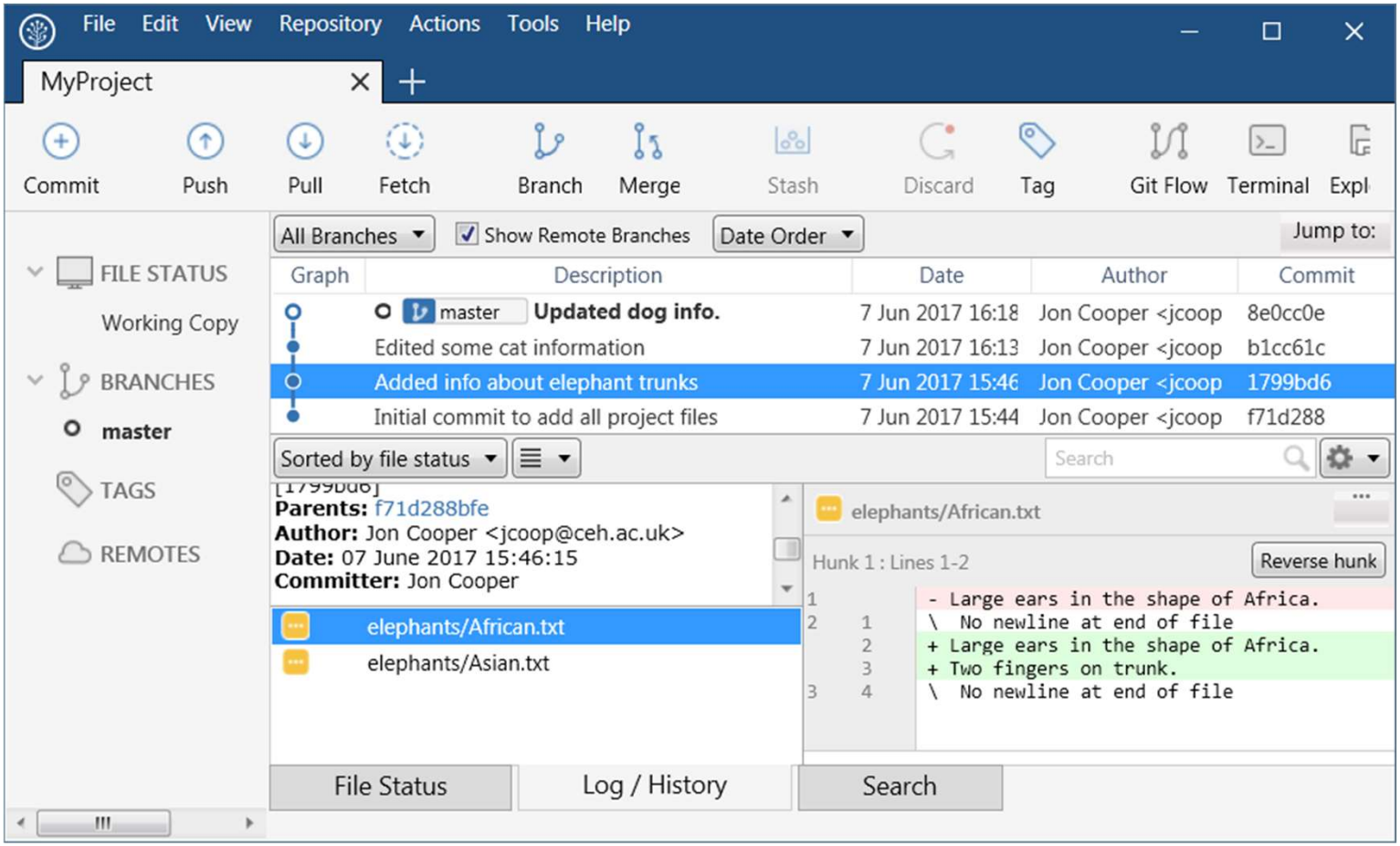
Session 1: Principles

Example of committing staged files



Session 1: Principles

Example of viewing the commit log



Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

Session 1: Principles

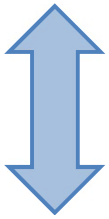
Exercise 1

https://nerc-ceh.github.io/version_control/exercise1

- Enable Git on a folder
- Review the status of files in the folder
- Repeatedly edit files and add your changes to Git
- Review changes
- Checkout a previous version of your changes
- Apply a tag to a specific version for future reference
- Exclude files from version control

Session 2: Remote repository

Github.com



Others are available

- Gitlab
- Stash
- Bitbucket
- etc…

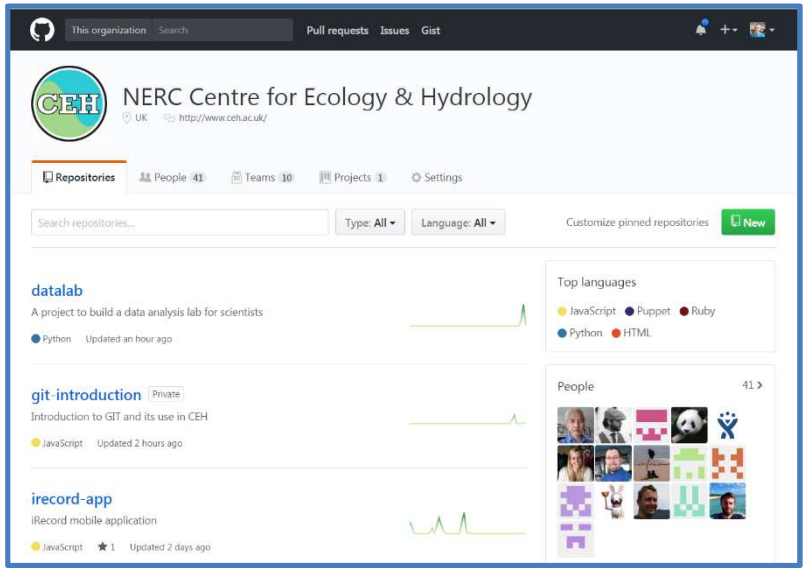
Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

Session 2: Remote repository

- Register with Github.com
- Free for public and open source projects
- Join the NERC-CEH organisation on Github for free CEH *private* projects

https://github.com/NERC-CEH



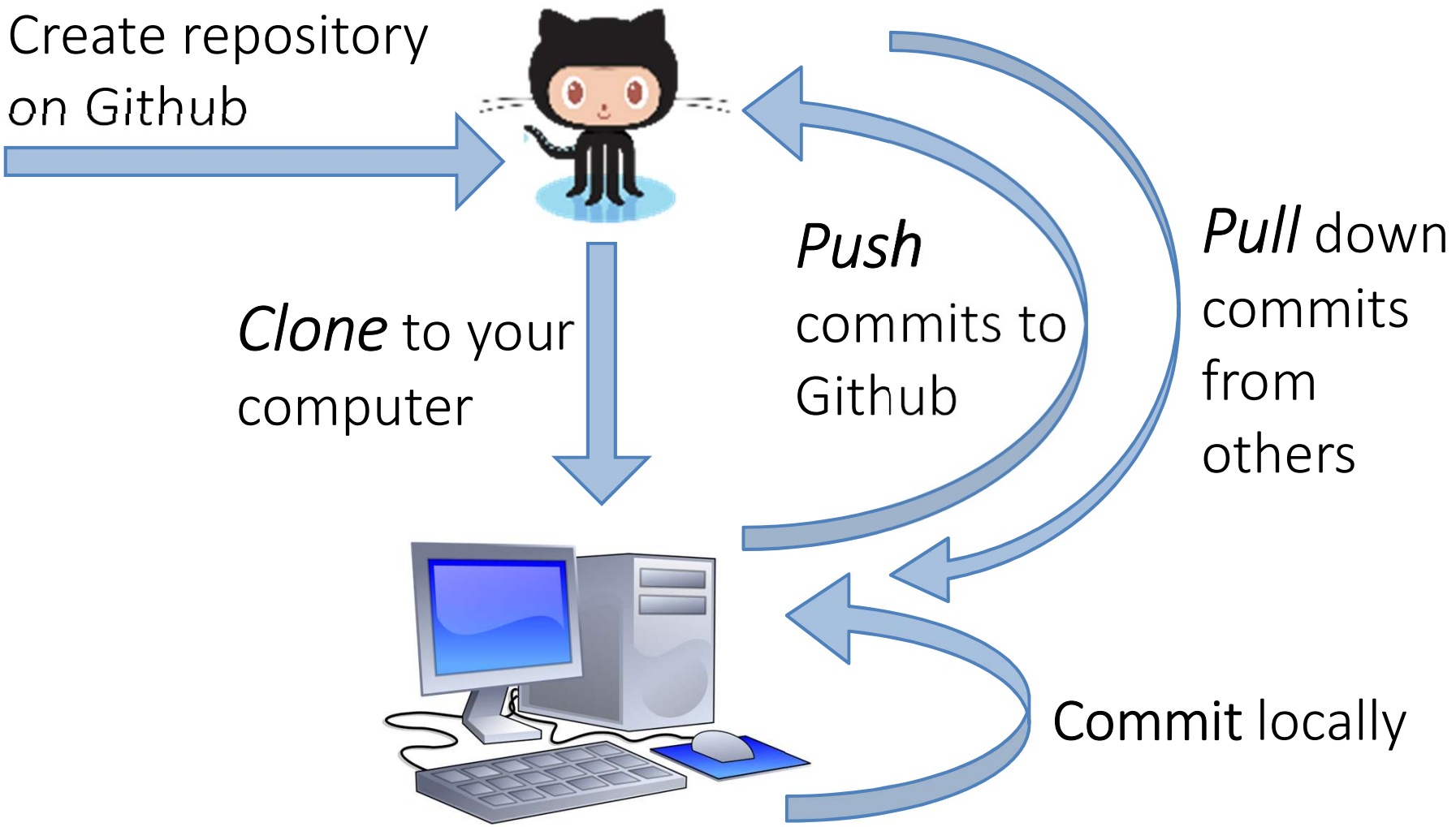
Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

Session 2: Remote repository

- Create empty repository on Github
- Download a local copy – called *Cloning*
- Work locally just as you did in session 1, committing your changes
- Periodically synchronise with Github using *Push* and *Pull*

Session 2: Remote repository



Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT

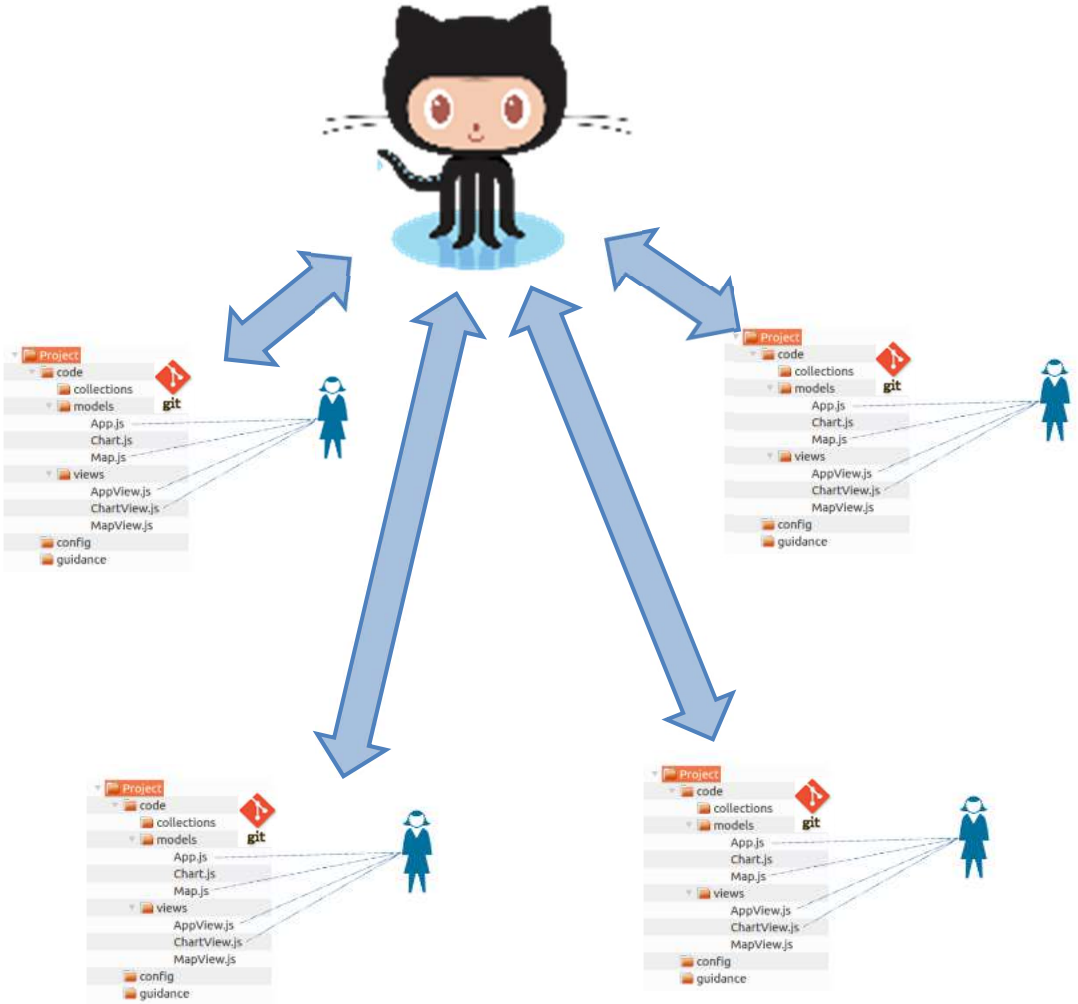
Session 2: Remote repository

Exercise 2

https://nerc-ceh.github.io/version_control/exercise2

- Register with Github
- Join the NERC-CEH organisation
- Create a repository in the NERC-CEH organisation
- Clone it on local machine and commit changes
- Push repository state to Github

Session 3: Team working



Session 3: Team working

- Collaborating in Github
- Branching and merging
- Sharing changes
- Conflict resolution
- Pull requests and quality control

Session 3: Team working

Add collaborators to your private Github repository

- Send invitation to collaborators via Github
- Collaborators have *Push* (write) access to repository
- More details here:

https://help.github.com/articles/inviting-collaborators-to-a-personal-repository

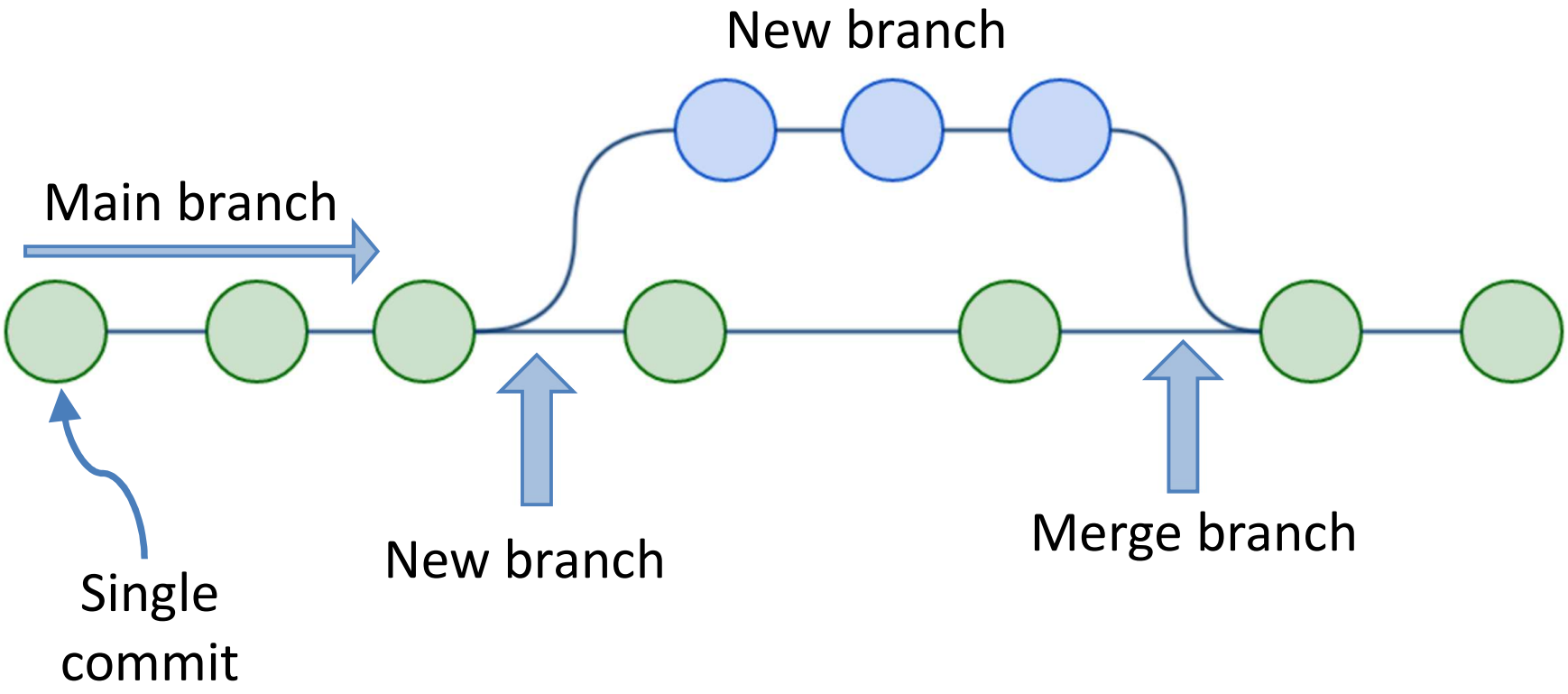
Session 3: Team working

Branching and merging

- A *Branch* is a parallel line of development in repository
- Keeps work off main branch until it is ready
- *Merging* is the process of putting the changes on that branch back into the main branch

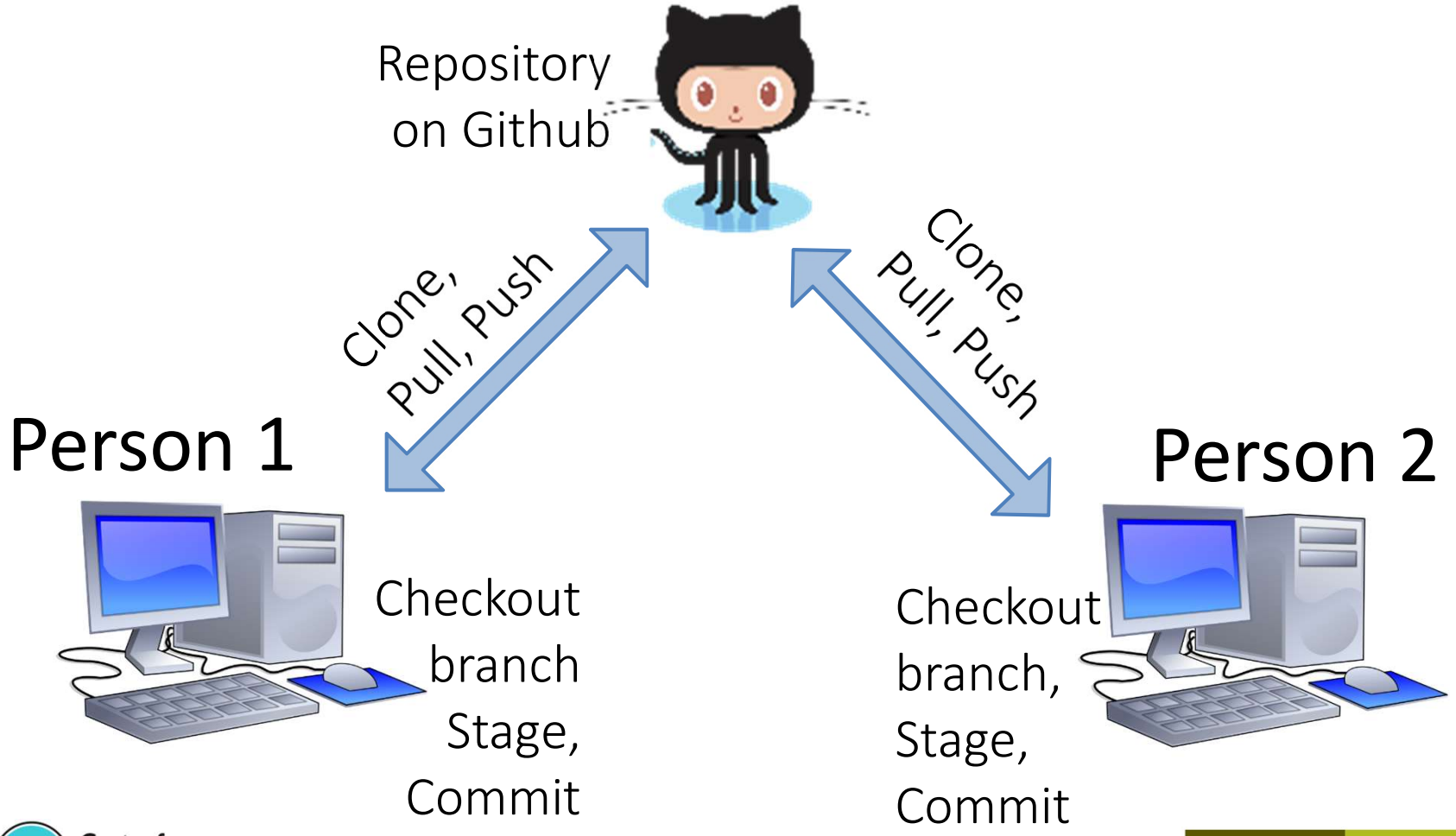
Session 3: Team working

Branching and merging



Session 3: Team working

Multiple people accessing one Github repository



Session 3: Team working

Conflict resolution

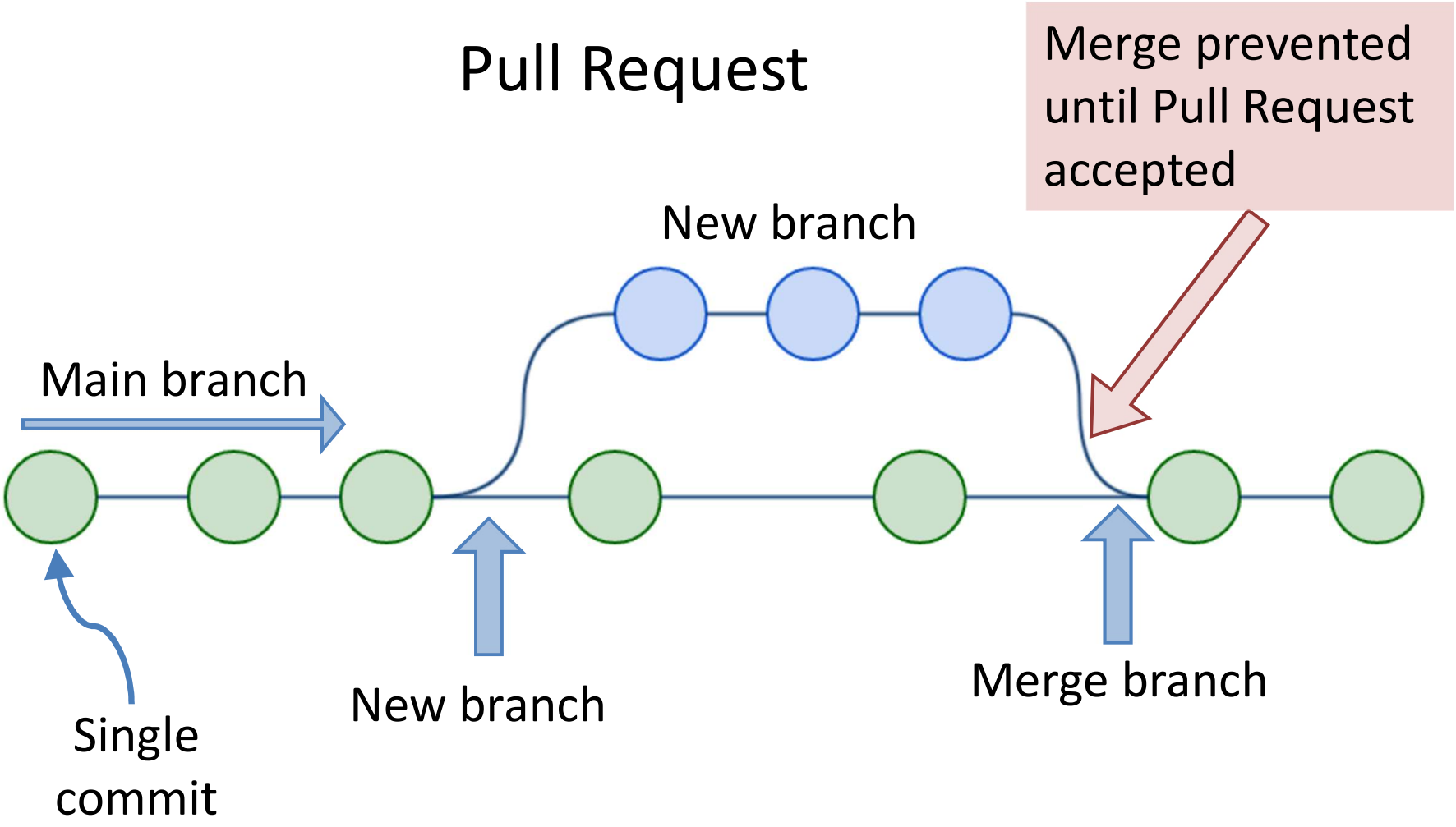
- Two collaborators edit the same text in the same file
- Person 1 Pushes their change to Github
- Person 2 Pulls down changes and gets a *Conflict*
- Person 2 cannot Push to Github until it is resolved
- Conflict must be edited and marked as *Resolved*
- Demonstrated in exercise
- Merge tools available to help

Session 3: Team working

Pull Request

- Pull Request is a Quality Assurance step around a Merge
- Lets collaborators know a branch is ready to be merged into the main branch
- Changes can be reviewed, discussed, altered, etc
- Branch merged only when Pull Request is accepted

Session 3: Team working



Session 2: Team working

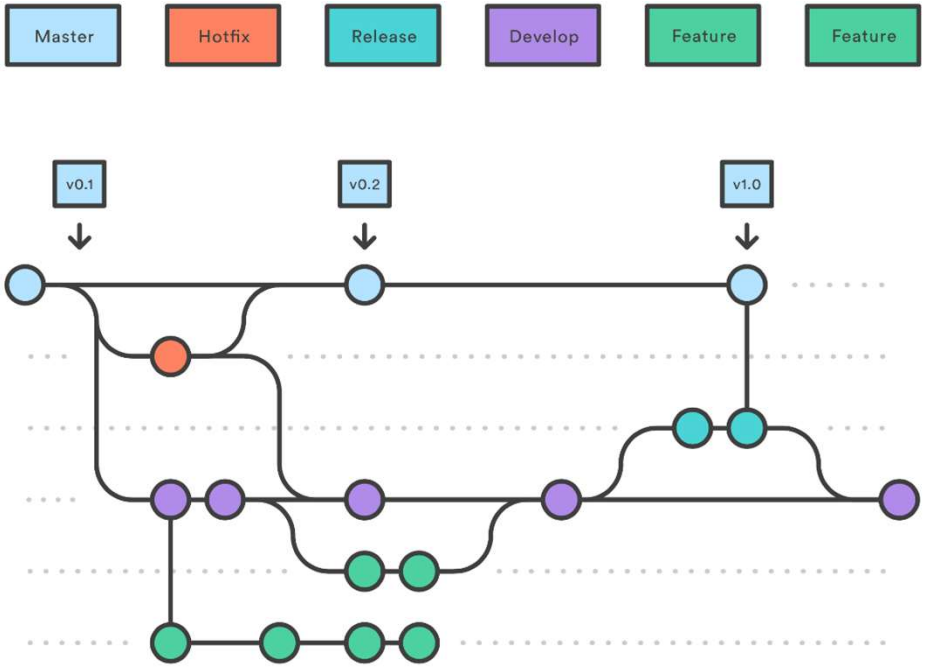
Exercise 3

https://nerc-ceh.github.io/version_control/exercise3

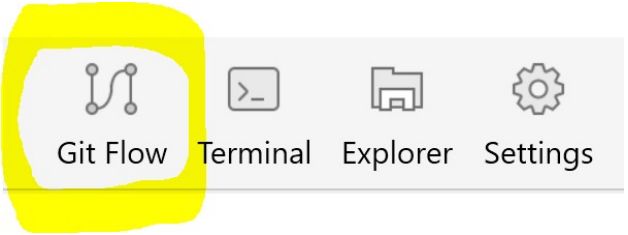
- Invite collaborators to your repository
- Work as a team on a new branch
- Create and resolve conflicts
- Issue and accept a pull request

Workflows

- Gitflow



- Available in SourceTree:



More than just version control

- Integrates with issue management
  - Branches matched to issues
- Workflows triggered by commits, pull requests, etc
  - Automates unit and integration tests
  - Automates deployment to servers
- Open sourcing
  - Promotes collaboration and quality
  - Increases reputation

SVN to GIT

To move from SVN to GIT without keeping history

1. Create new repository in github.com/NERC-CEH (exercise 2 step 2)
2. Clone it to your local drive (exercise 2 step 3)
3. Use SVN Export to get clean copy of your code from SVN into your new local git repository
4. Use git add, git commit and git push to get all those files into your github.com repository
5. Give your team members access to your new github repository (exercise 3 step 2)
6. Team members can now clone your repository and they are up and running in git

Cheat sheet

git clone

git checkout

git add

git commit

git pull

git push

git branch

git merge

Git repository in a mess?

1. Rename the root folder
2. Clone a new copy from github.com
3. With original files to help, make edits to files in new repo
4. Delete original when happy

Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL

NERC SCIENCE OF THE ENVIRONMENT