UNIVERSITY of WASHINGTON

Data Science UW Methods for Data Analysis

Git and Github
Extra Topics
Nick McClure

1. git commit

2. git push

3. git out

What is Git and why do we need it?

- Git is system that allows you and others to:
  - Simultaneously work on the same project.
  - Version control.
  - Backup all history of changes.
  - Share code.
  - Create different branches of your code: Main Branch, feature branches, development branch, research branch, …
- Scenarios that are useful:
  - If your computer died overnight, how soon could you get back to work?
  - If a malicious employee changes code or a bug is introduced, git can restore to prior points.
  - If a colleague wants to use your code, share your code, or do code review, git easily allows them do to that.

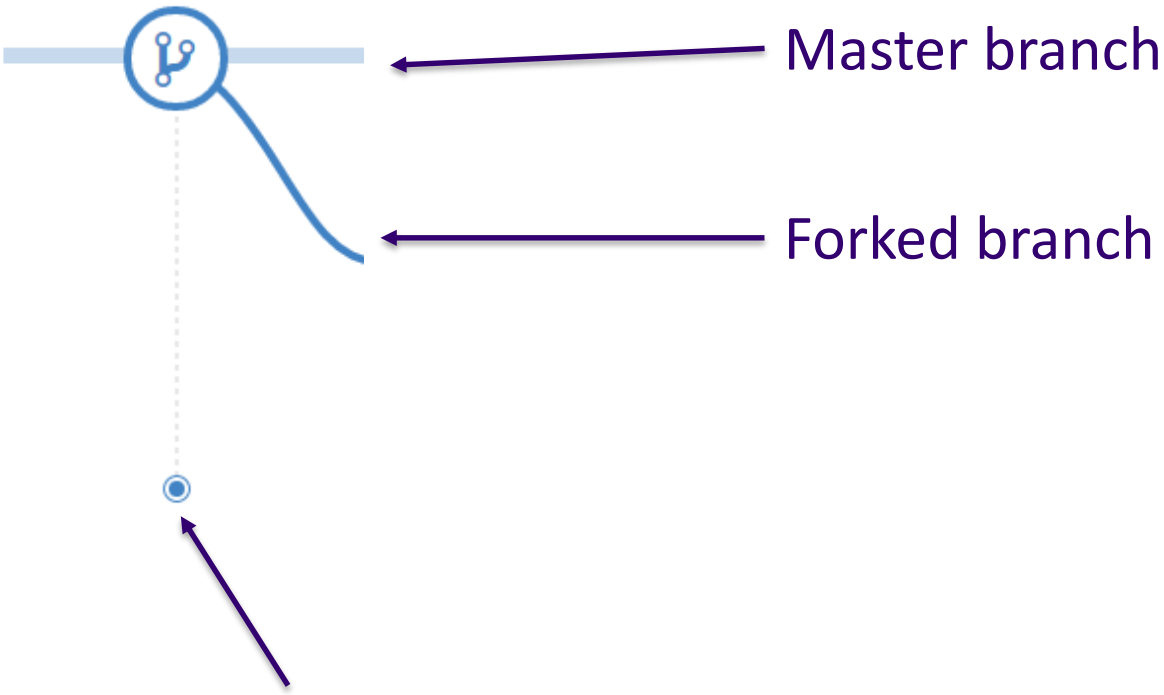
Git vs Github

- Git is a system for version control. You can have this system implemented privately, locally, or across networks.
- Github is a free online implementation of git.
  - This provides not only version control, but an online backup of code.

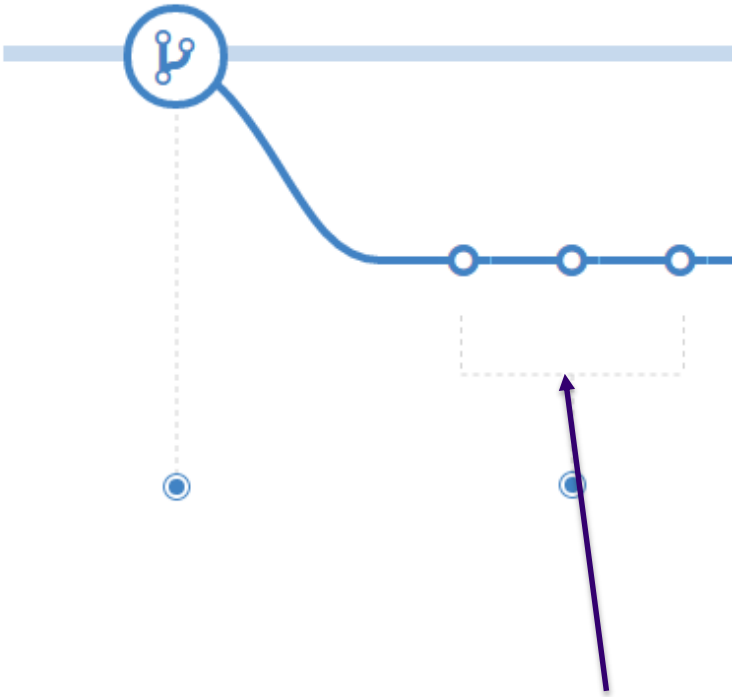
Github Terms

- Creating a Repo: Creating a new repository for work.
- Master branch: This is the final version that everyone will see and use in your repository.
- Creating a branch: Create a branch to try out new ideas without breaking the Master branch on your repo or other repos.
- Pull Request: Request that your branch be reviewed and pulled back (or merged) into the Master branch.
- Adding Commits: Keep track of your progress through commit points.
- Forking a Repo: Creates a copy for you to work on
- Cloning: Downloading a copy of a repo

Github Work Flow

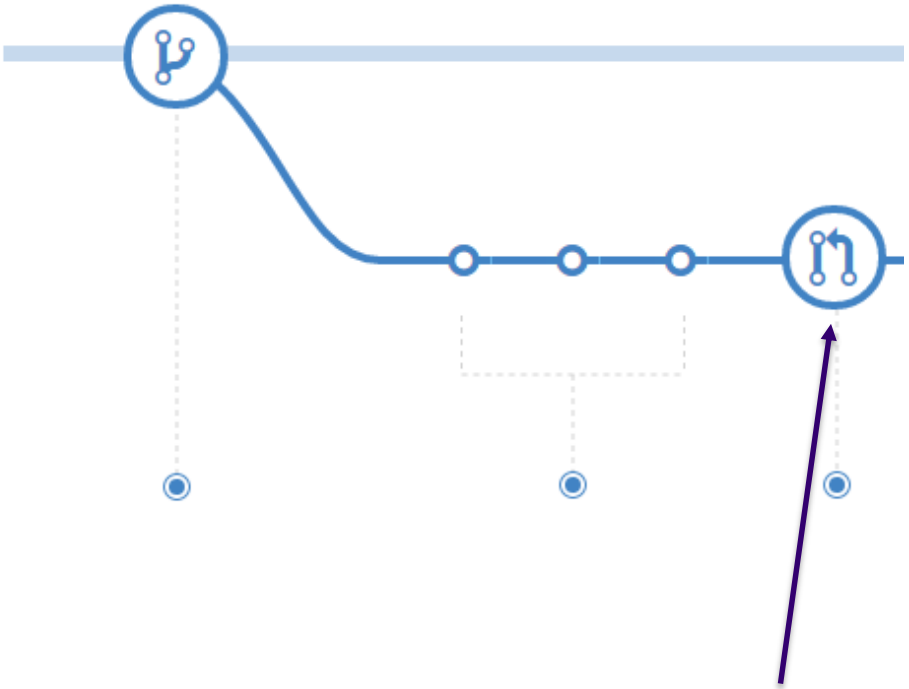


Github Work Flow



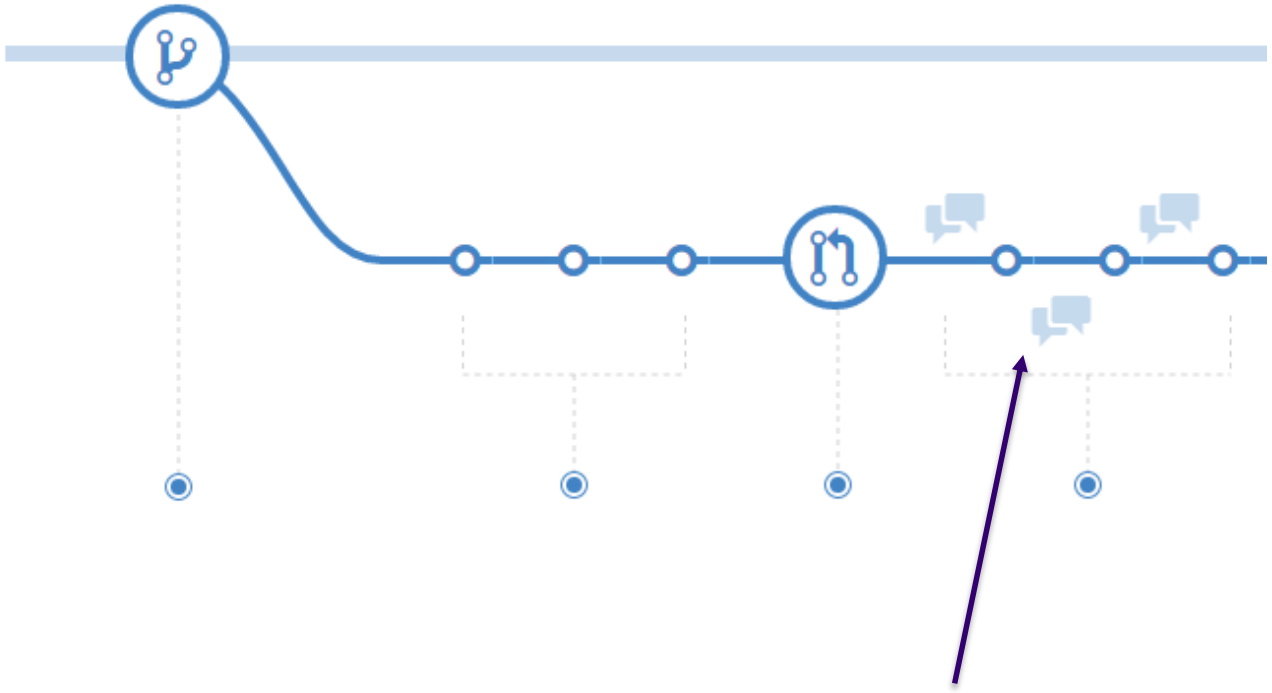
Add commits (changes)

Github Work Flow



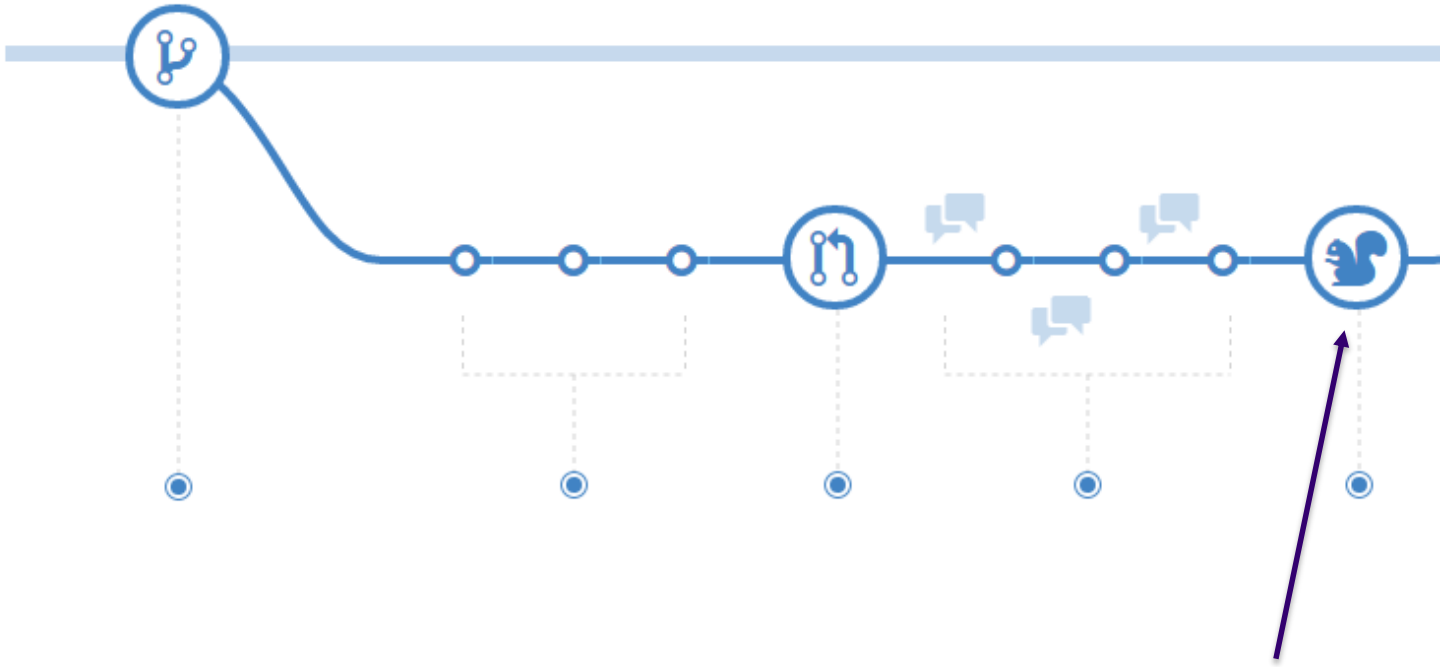
Submit (Open) a pull request

Github Work Flow



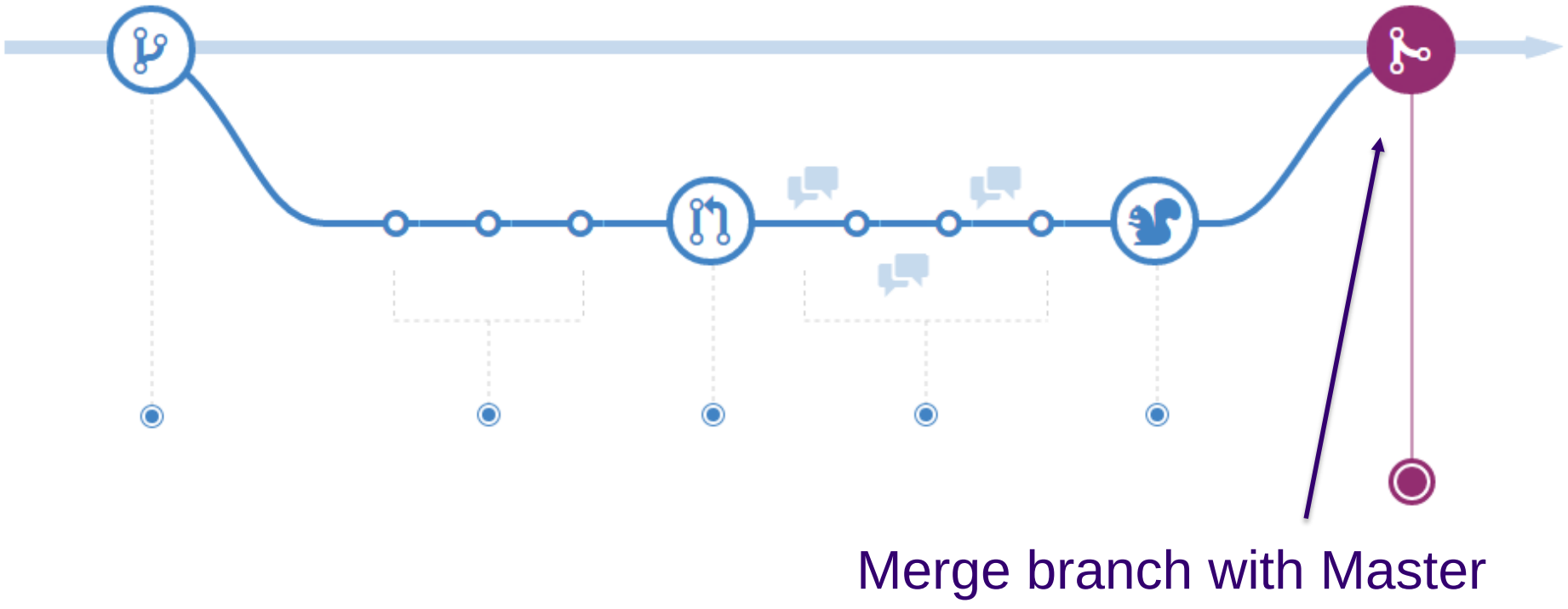
Discuss and review code changes (Diff)

Github Work Flow



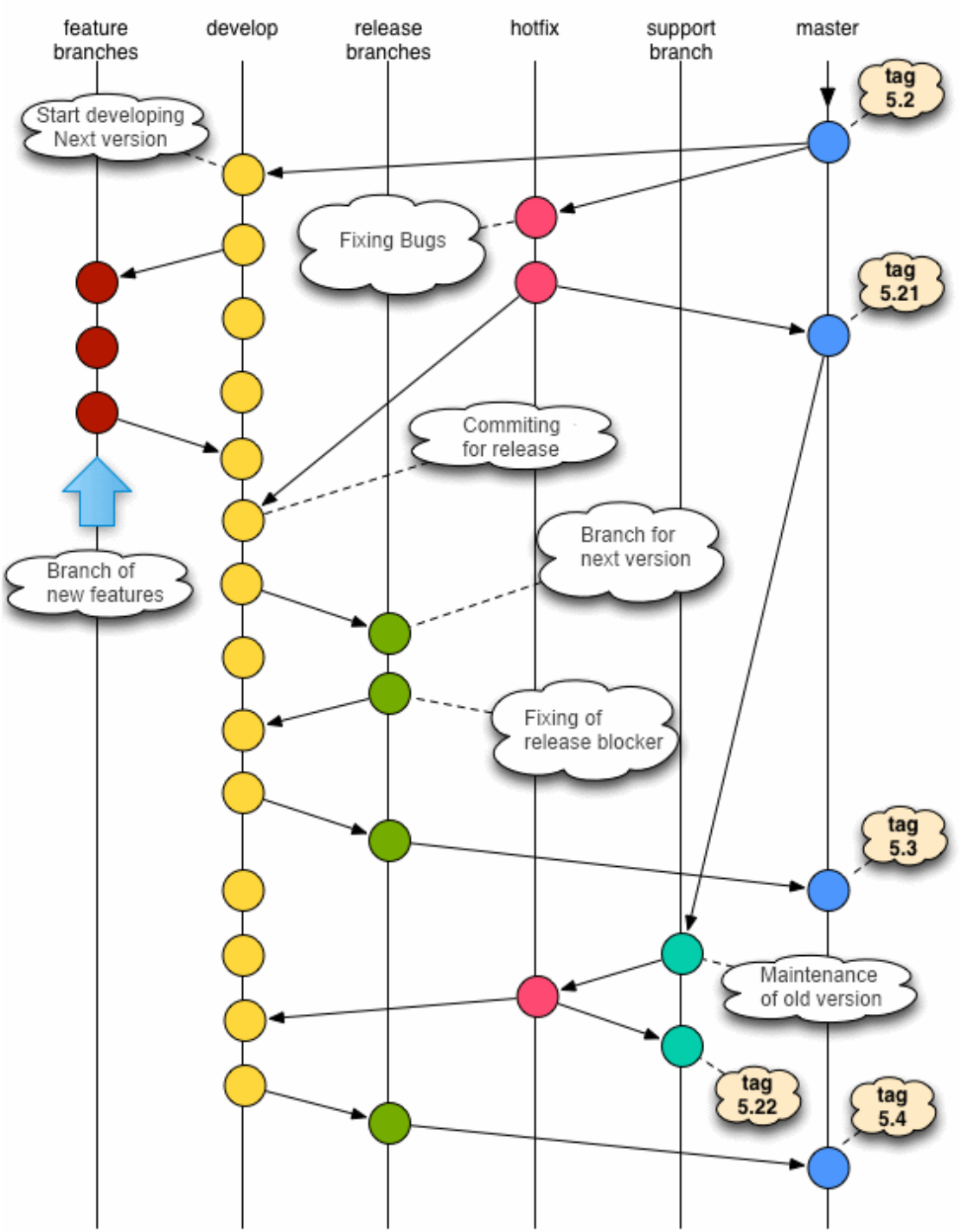
Check unit tests

Github Work Flow



Merge branch with Master

Workflows can get complicated:



Getting Started with Github

- Github makes it nice to start and create Repositories via a user interface.
  - https://desktop.github.com/
- Linux (and Mac) come with git preinstalled.
- When using git on the command line/locally, you must add the github repo as an ‘origin’: (one time command)

```
git remote add origin https://github.com/username/repo.git
```

- Then if we make changes remotely, to sync online:

```
git add .
git commit –m “Updated Lecture 1”
git push origin master
```

Ignoring some Local Files

- Sometimes we don't want to sync some local files or folders:
  - *.Rdata, *.Rhistory, *.pptx, *.ppt, solutions/*,…
- We add these entries to a '.gitignore' file in our git folder locally. Git will then ignore such files.