GitHub workshop

Bastien Brugger
Ishan Mishra
Ryan Petersburg

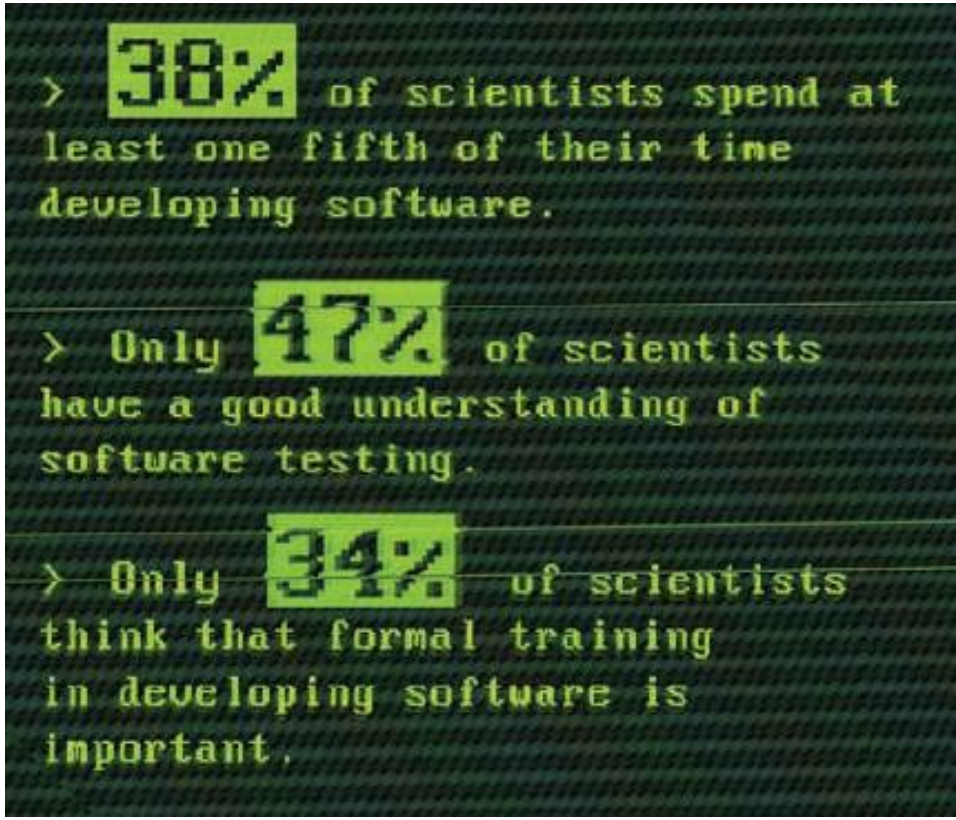
Day 3 – 2019/06/19

Cornell University

Merali (2010) *Nature 467, 775-777*

> *Problems created by bad documentation are further amplified when successful codes are modified by others to fit new purposes. The result is the bane of many graduate students or postdocs' life: the "**monster code**". Sometimes decades old, these codes are notoriously messy and become progressively more nightmarish to handle*

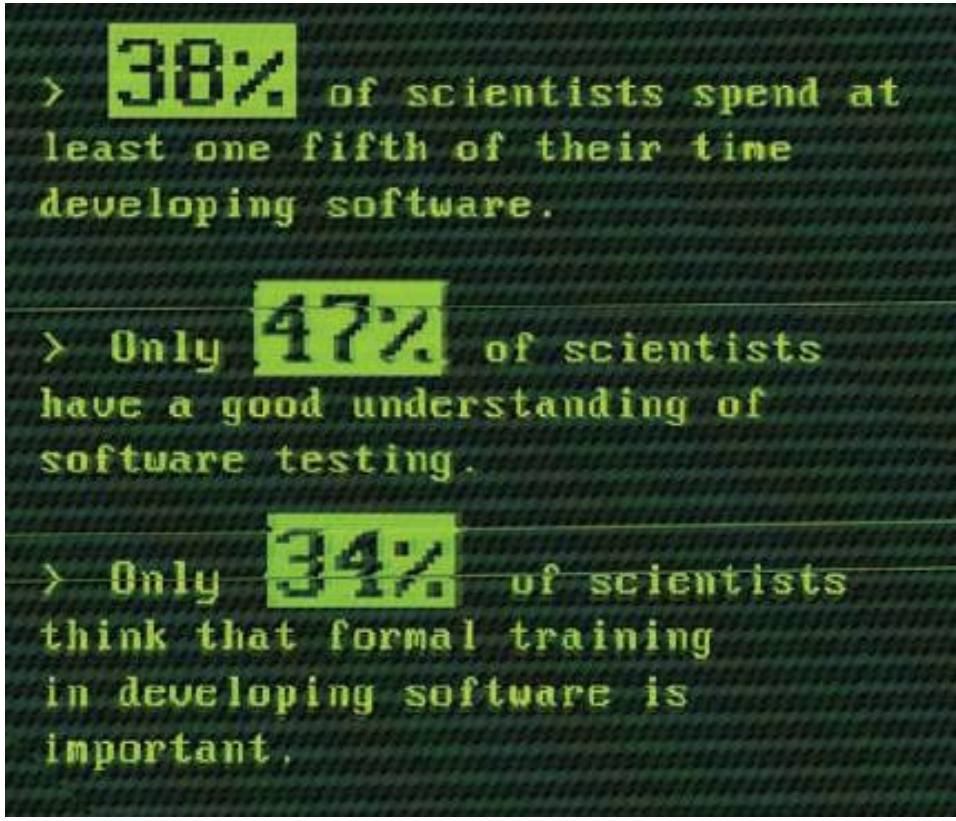
Merali (2010) *Nature 467, 775-777*



Merali (2010) *Nature 467, 775-777*

Five tips to make scientific code more robust:

- *Track your material*
- *Write testable software*
- *Test the software*
- *Encourage sharing of software*
- *Use a version-control system* (VCS)

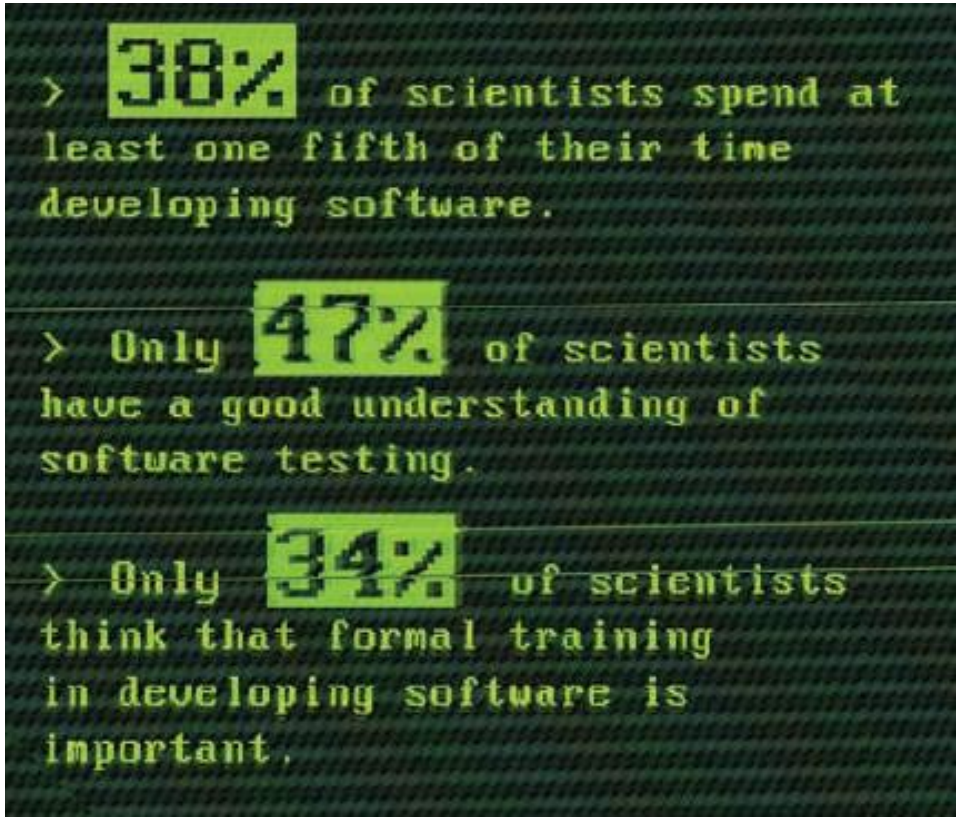


Merali (2010) *Nature 467, 775-777*

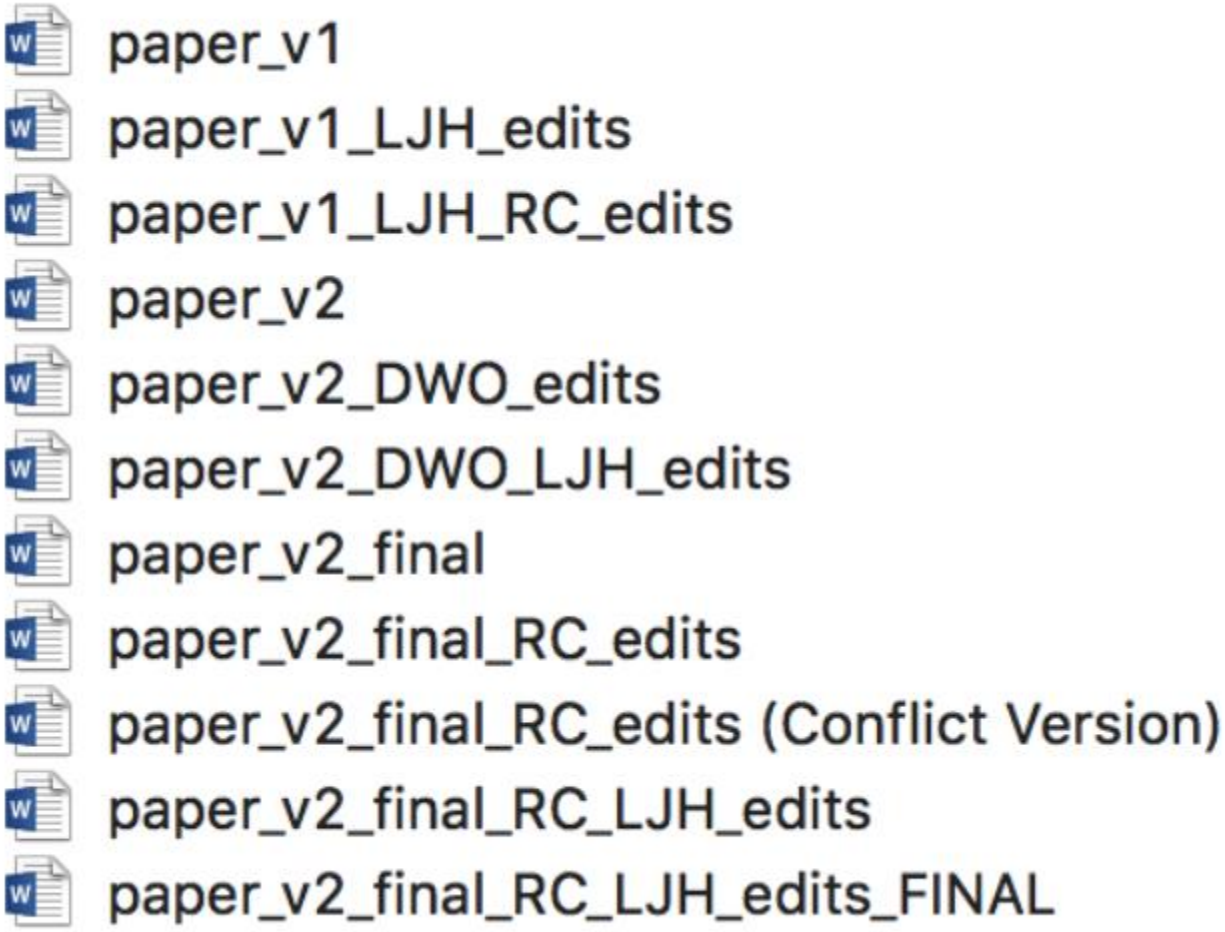
Five tips to make scientific code more robust:

- *Track your material*
- *Write testable software*
- *Test the software*
- *Encourage sharing of software*
- *Use a version-control system* (VCS)

GitHub

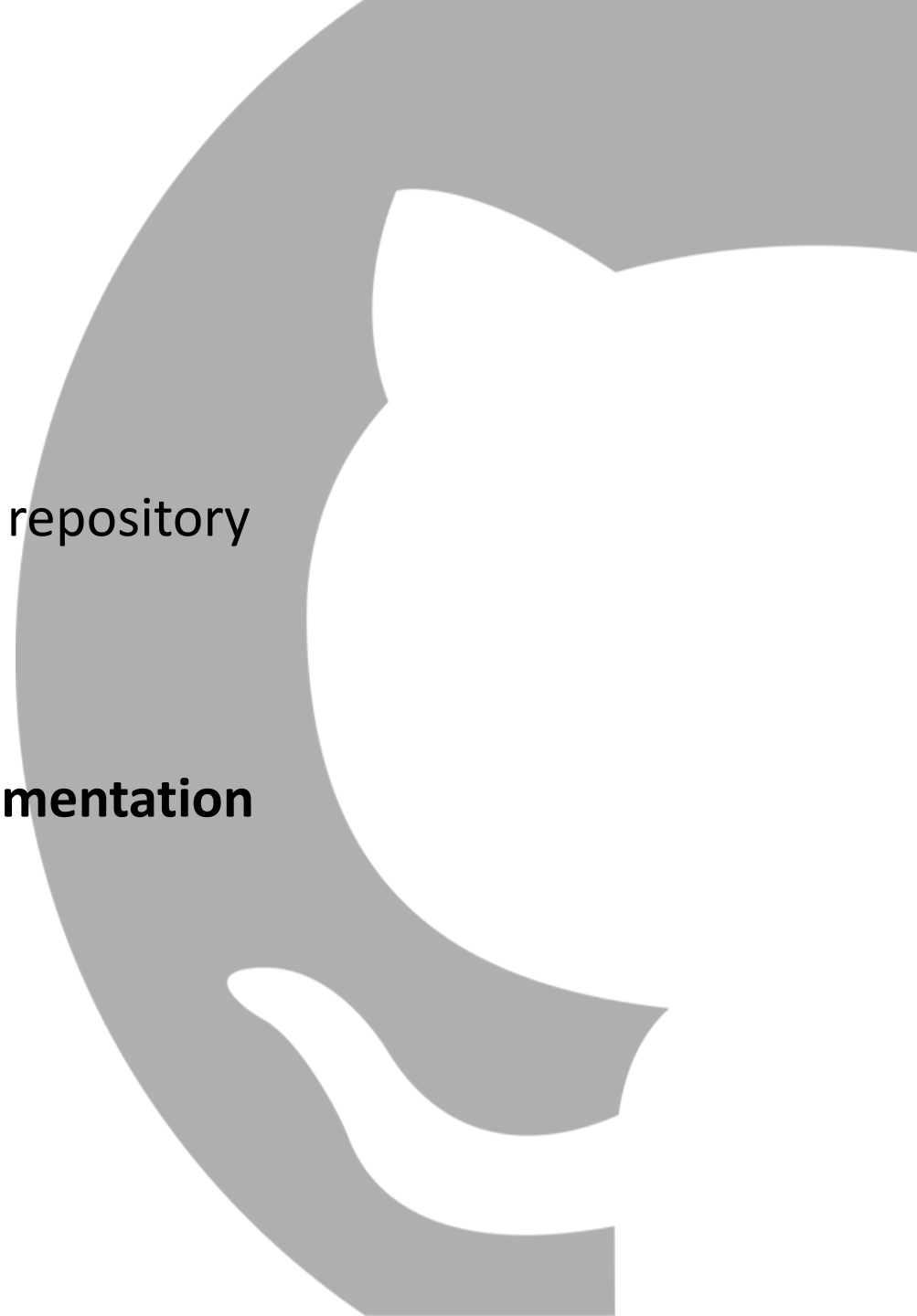


Motivation


paper_v1
paper_v1_LJH_edits
paper_v1_LJH_RC_edits
paper_v2
paper_v2_DWO_edits
paper_v2_DWO_LJH_edits
paper_v2_final
paper_v2_final_RC_edits
paper_v2_final_RC_edits (Conflict Version)
paper_v2_final_RC_LJH_edits
paper_v2_final_RC_LJH_edits_FINAL

Summary

1. What is **VCS** and why use it?
2. Creating and understanding your first **GitHub** repository
3. Managing an actual **test** case
4. Best practices for code **accessibility** and **documentation**
5. Creating your **website** with GitHub Pages
6. Other **resources**



Summary

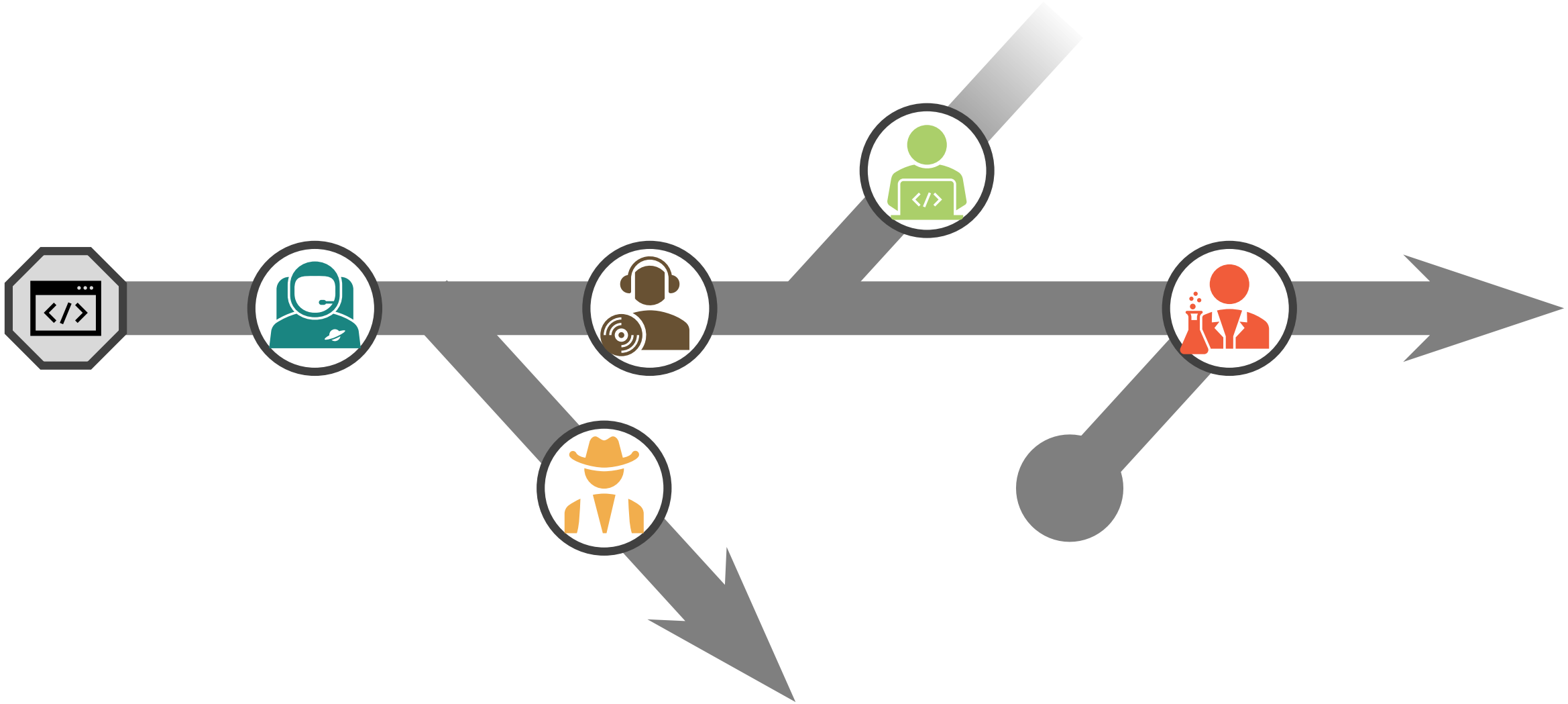
All material can be found at

`github.com/BastienBrugger/ERESV-github`

Disclaimer:

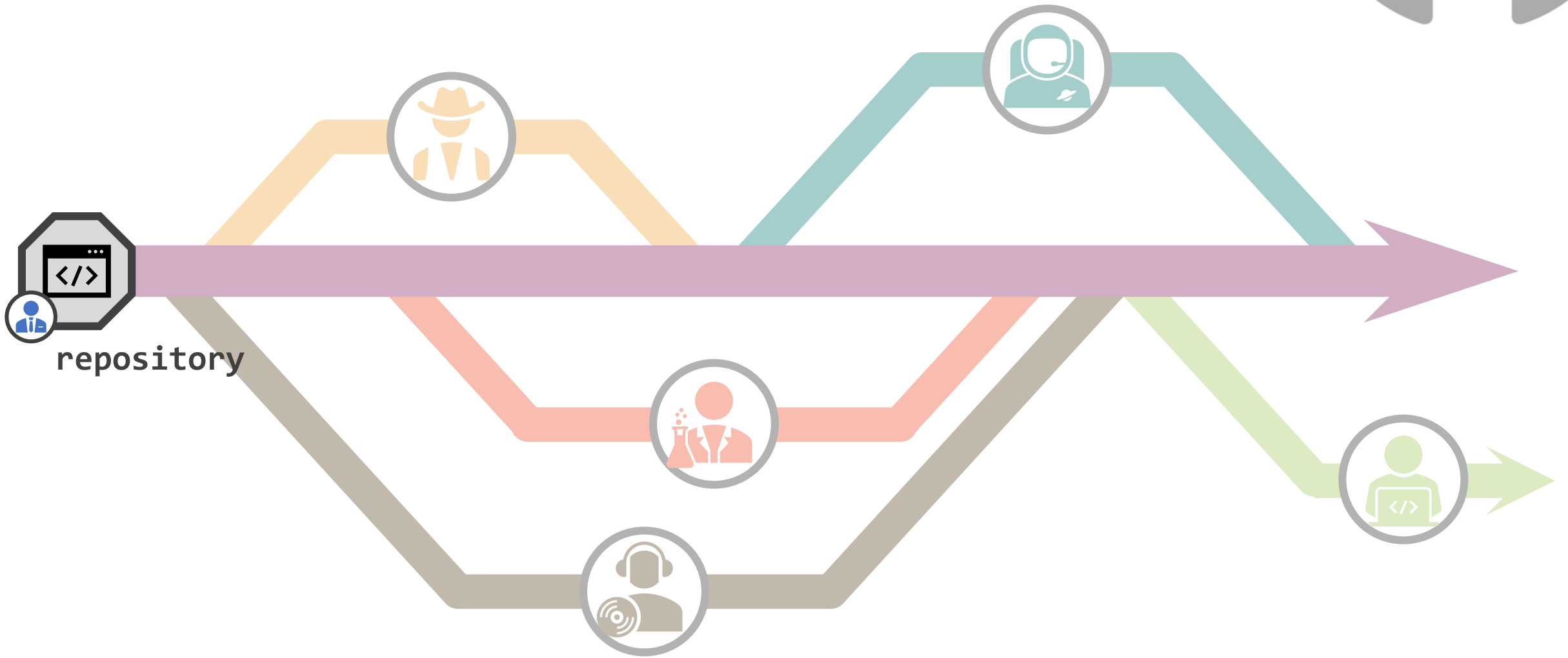
`Git`	*open-source VCS, the **tool** that manages your code history*
`GitHub`	*hosting **service** for `Git` repositories*

How VCS works

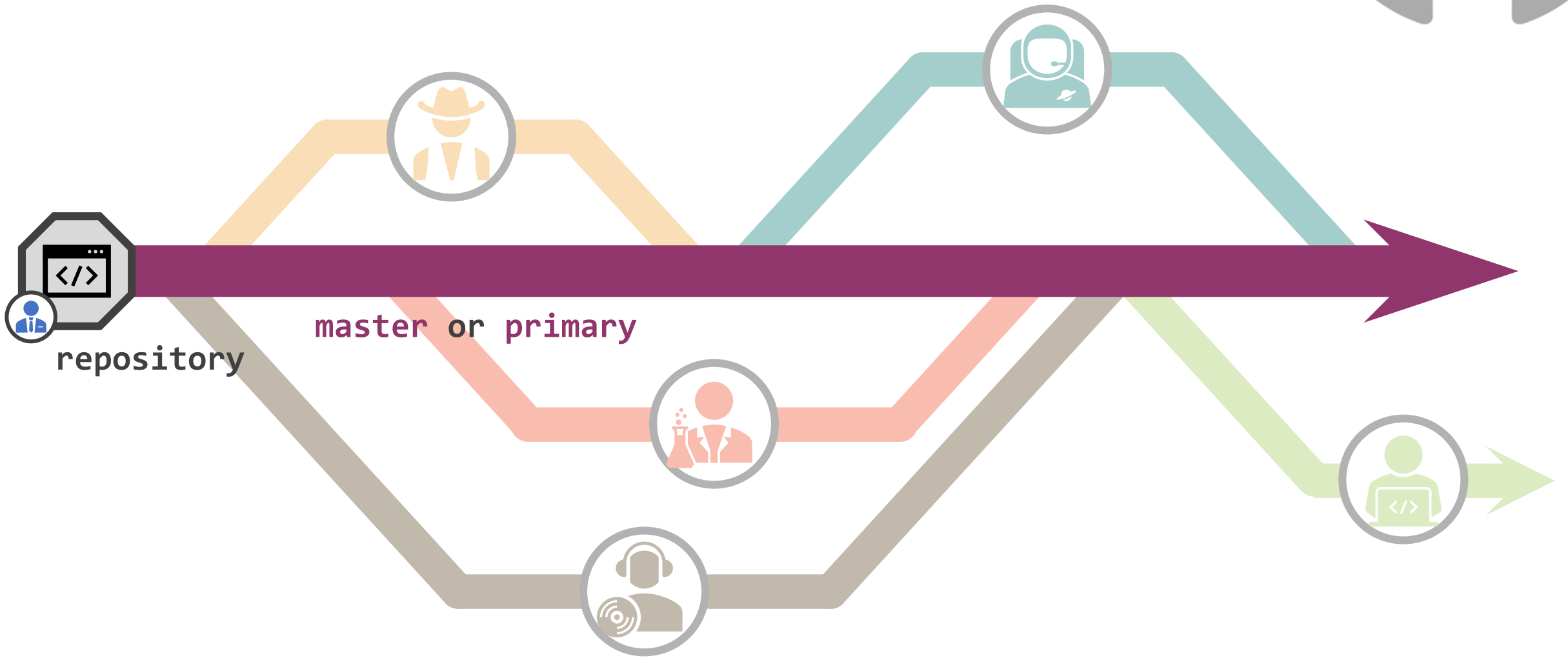


How VCS works

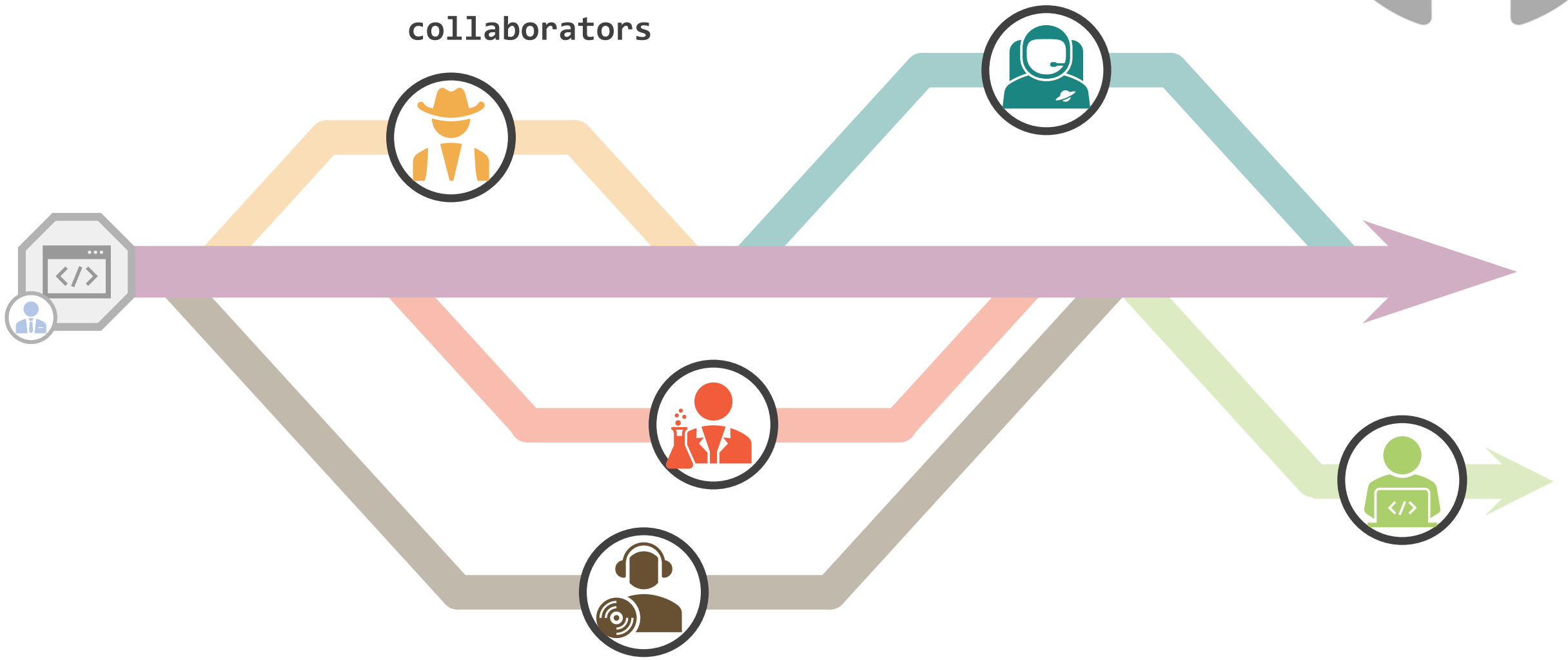
How GitHub works



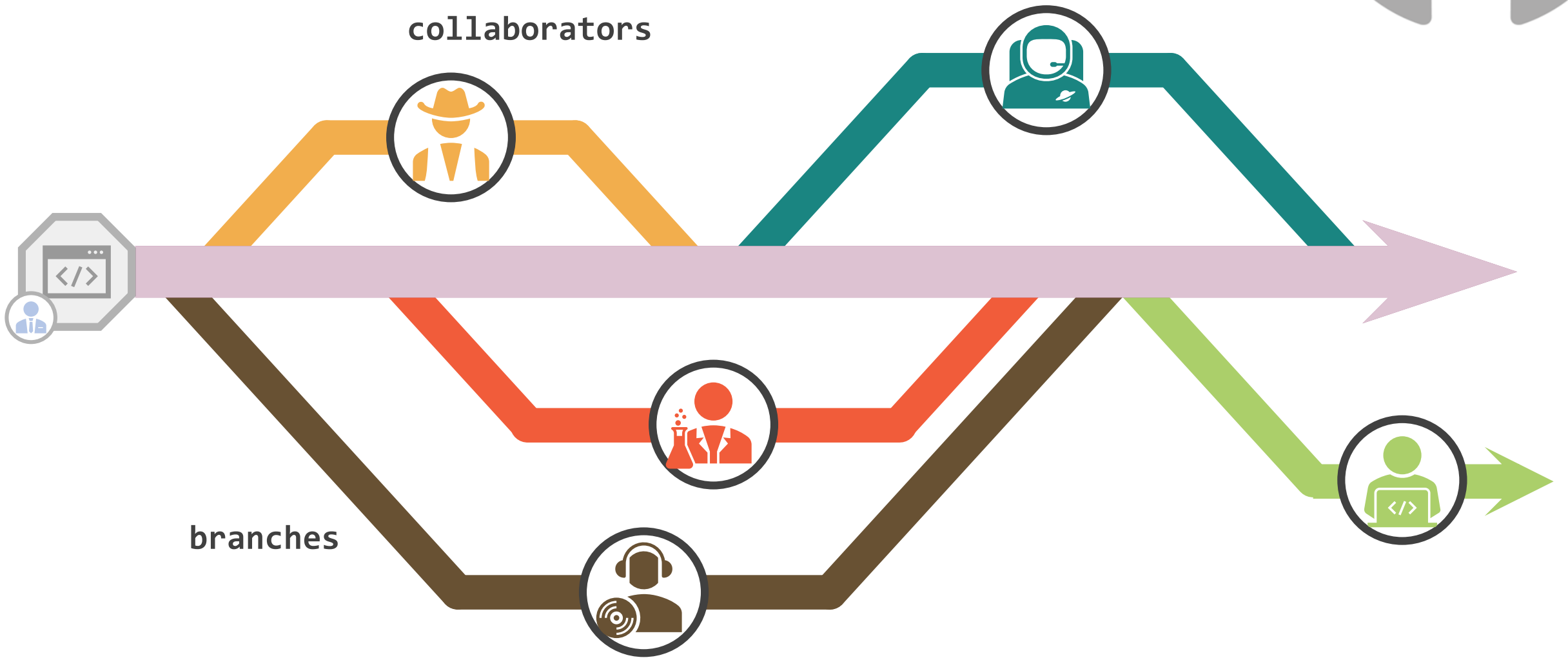
How GitHub works



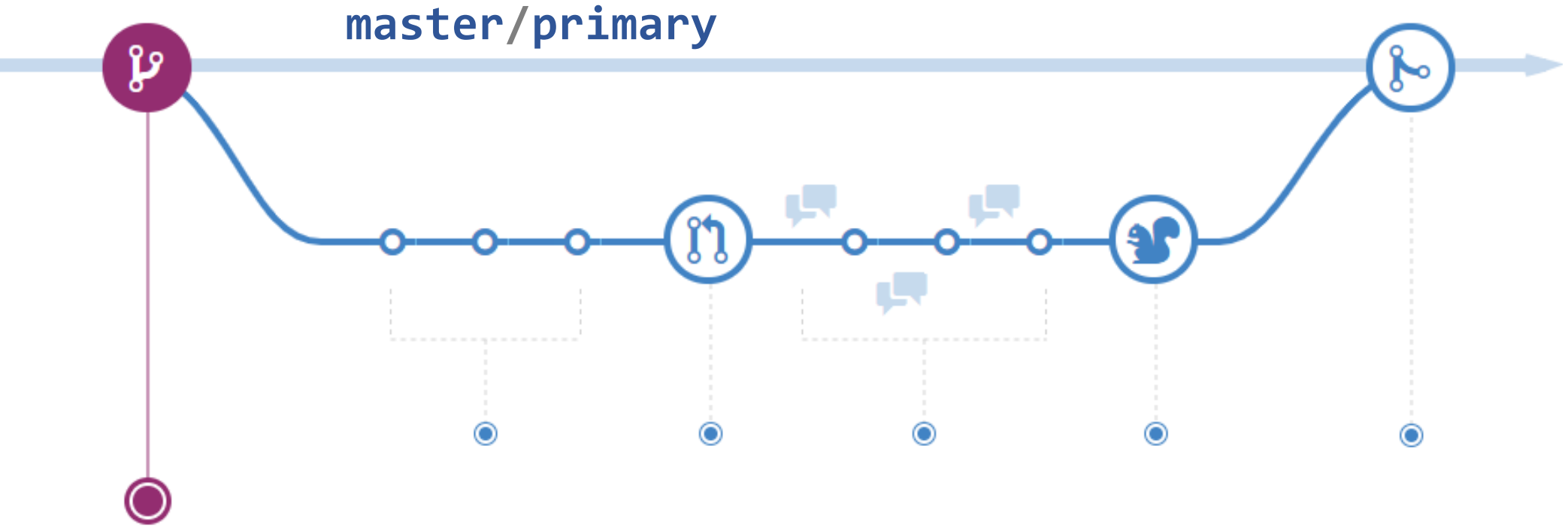
How GitHub works



How GitHub works



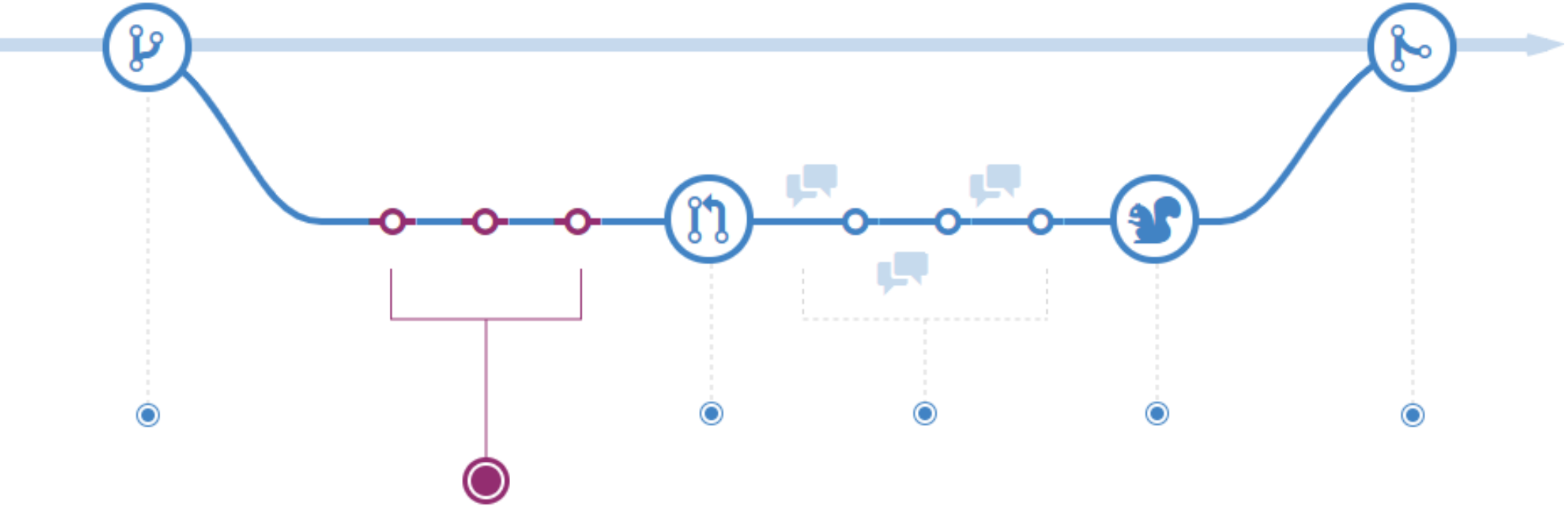
GitHub flow



Create a **branch**

*Copy files in an environment where you can **experiment** new ideas, without affecting the* **`master`**

GitHub flow

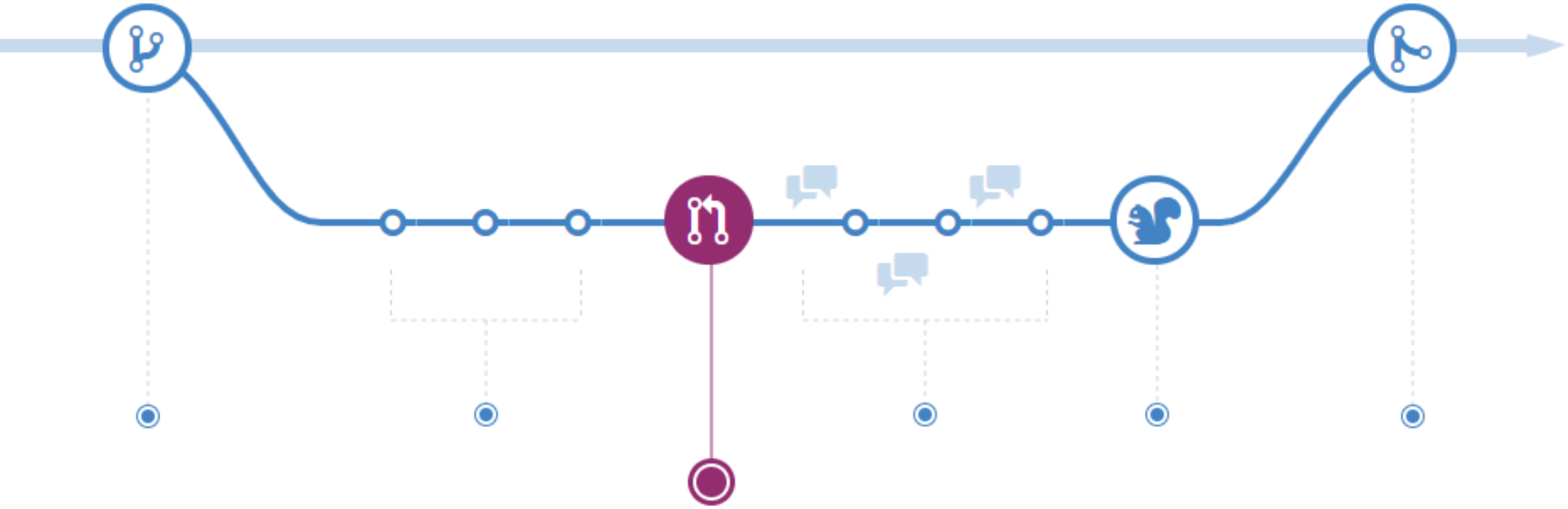


Add **`commits`**

Adding, editing or deleting files

*Creates a **transparent history** of your work: each* **`commit`** *has an associated message*

GitHub flow

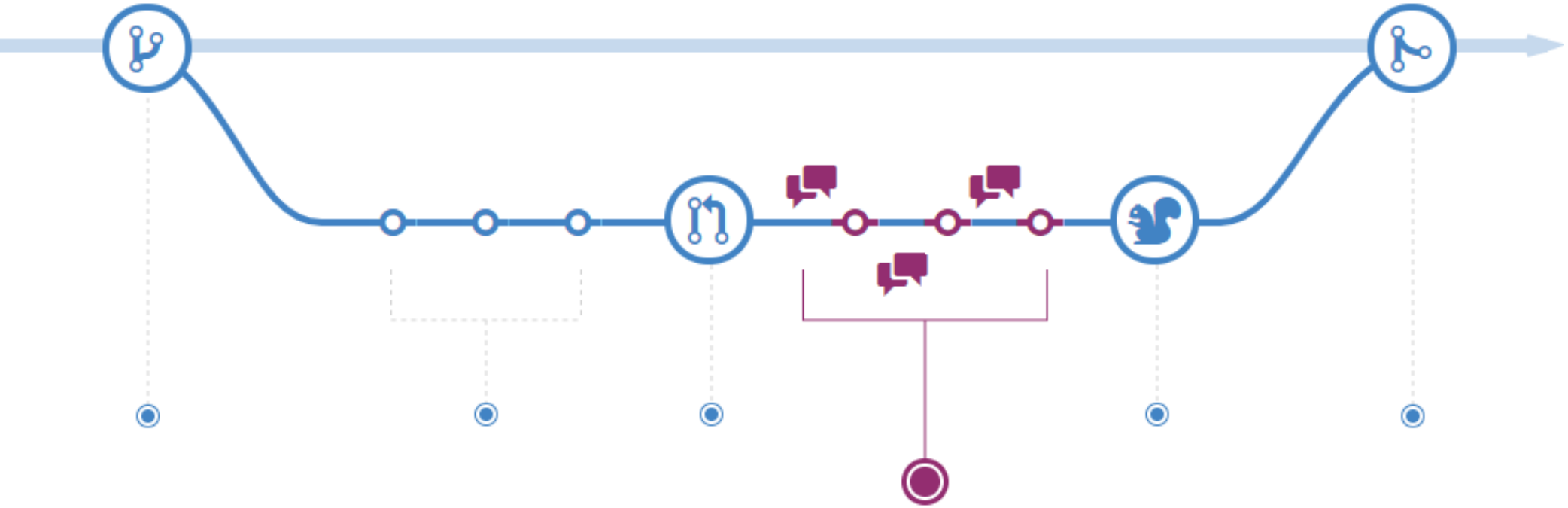


Open a `pull request`

Show your changes to other **`collaborators`** *and initiate discussion*

When you're ready to ***add your work****, when you want to* ***share ideas****, when you're stuck and* ***need help****…*

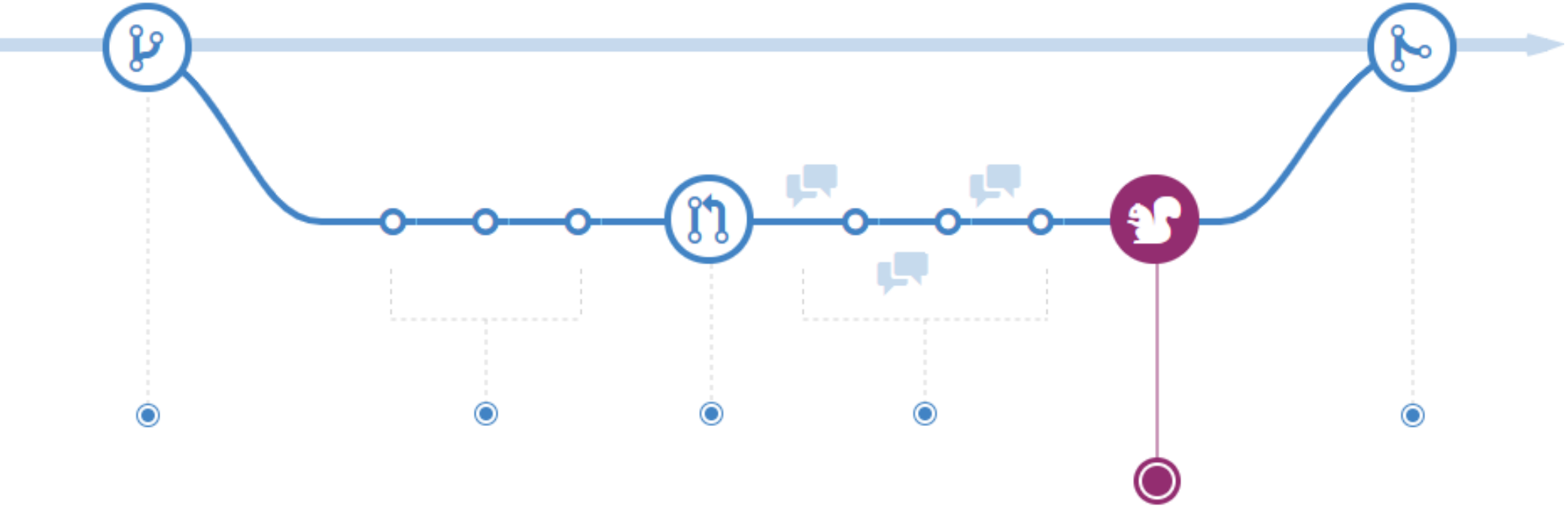
GitHub flow



Discuss and review code

Check if everything is fine

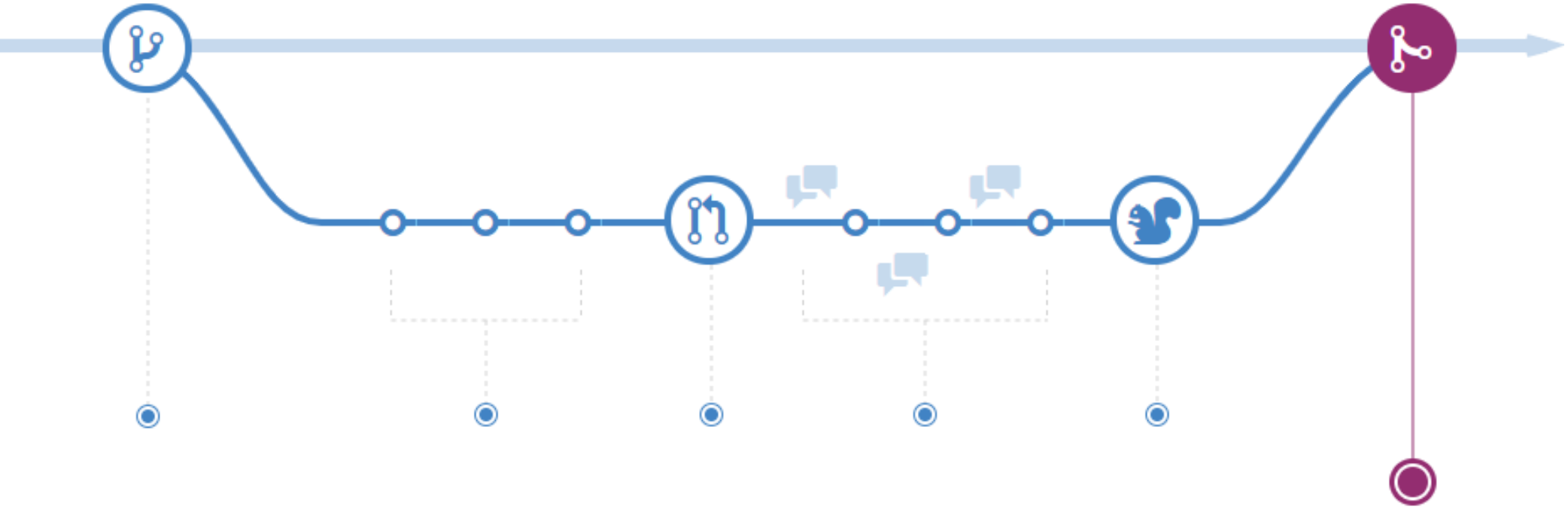
GitHub flow



Deploy

For final testing

GitHub flow



Merge

Merge your code into the **`master`** *branch*

Record is preserved

How GitHub works

`clone` a repository

Copy the files on your computer

No GitHub account required

Can be used offline

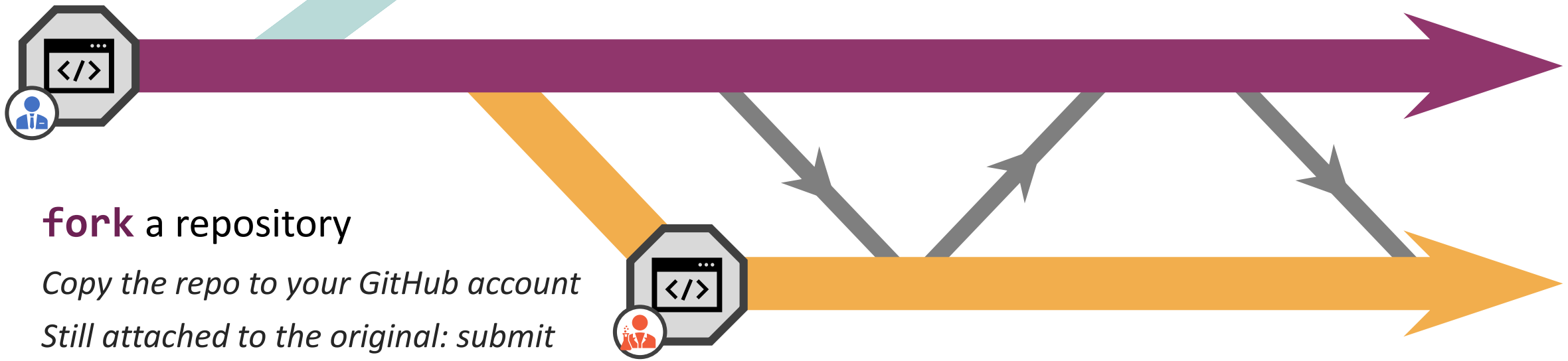
How GitHub works

clone a repository

Copy the files on your computer

No GitHub account required

Can be used offline



fork a repository

Copy the repo to your GitHub account

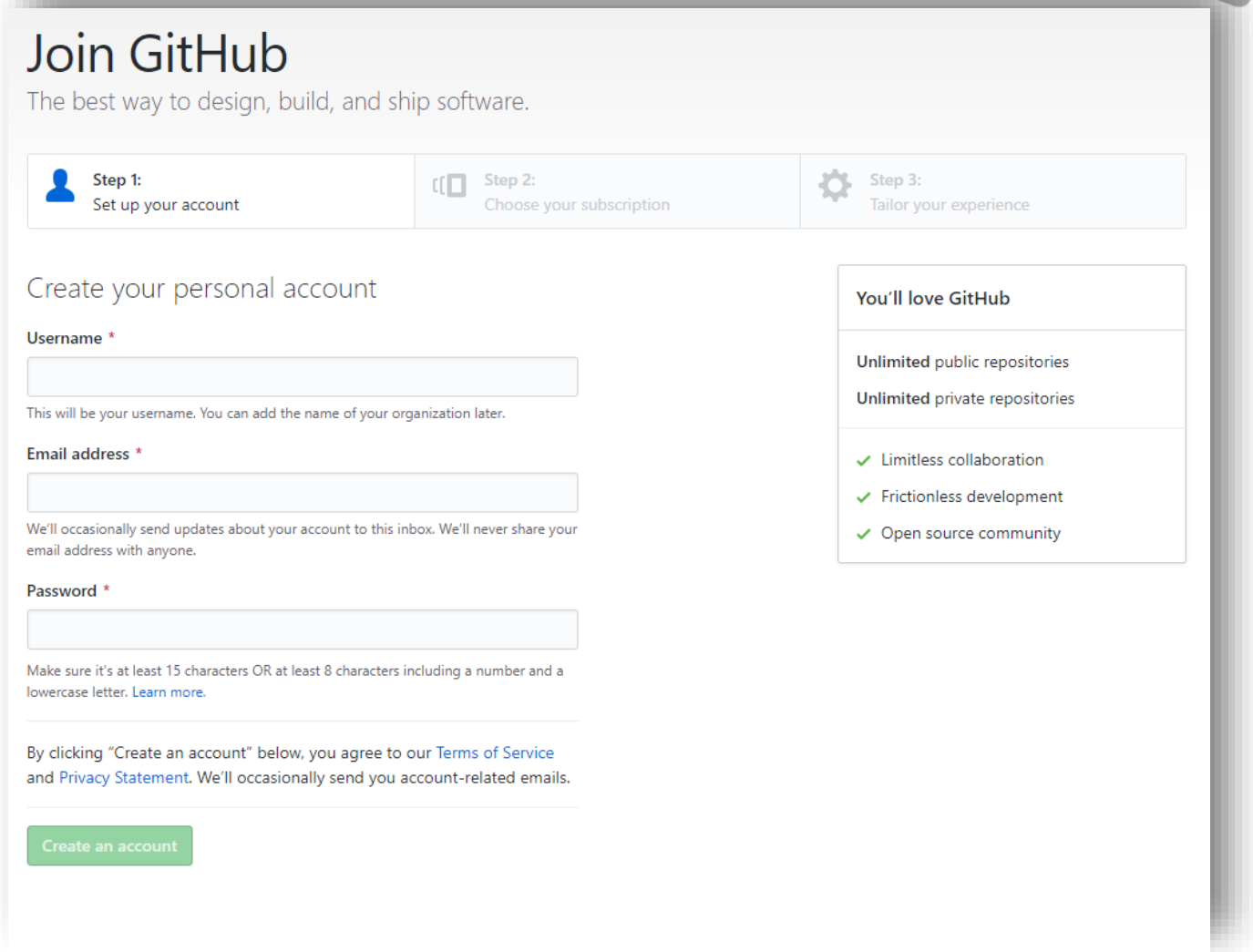
Still attached to the original: submit pull requests to it, import updates from it

Setting up

A new account

Go to

github.com/join



Setting up

Your dashboard

Main hub for your activities

Public

profile info, repositories, contributions…

The Octocat

octocat

Follow

PRO

GitHub

San Francisco

octocat@github.com

http://www.github.com/blog

Block or report user

Overview | Repositories 8 | Projects 0 | Stars 3 | Followers 2.7k | Following 9

Popular repositories

Spoon-Knife

This repo is for demonstration purposes only.

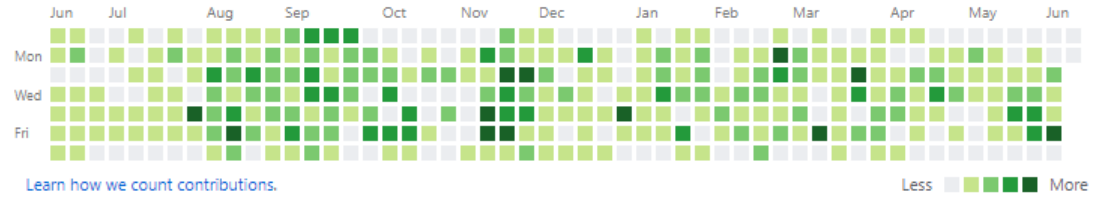
HTML ★ 10.1k 104k

Hello-World

My first repository on GitHub!

★ 1.5k 1.3k

1,587 contributions in the last year

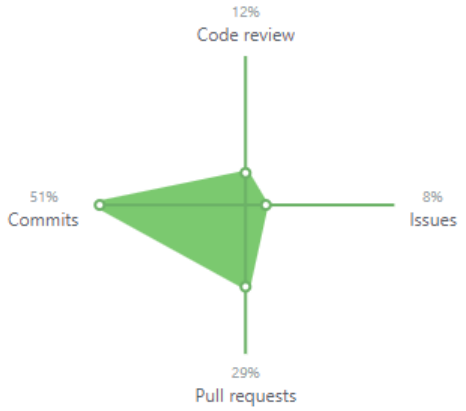


2019 2018 2017 2016 2015 2014 2013 2012 2011 2010 2009 2008

@octobox @ipfs @ipfs-shipyard More

Activity overview

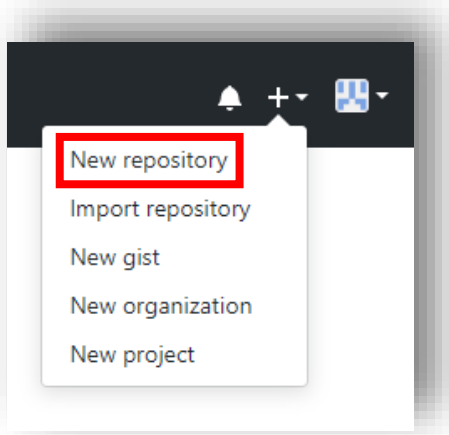
Contributed to octobox/octobox, 24pullrequests/24pullrequests, ipfs/package-managers and 5 other repositories



Setting up

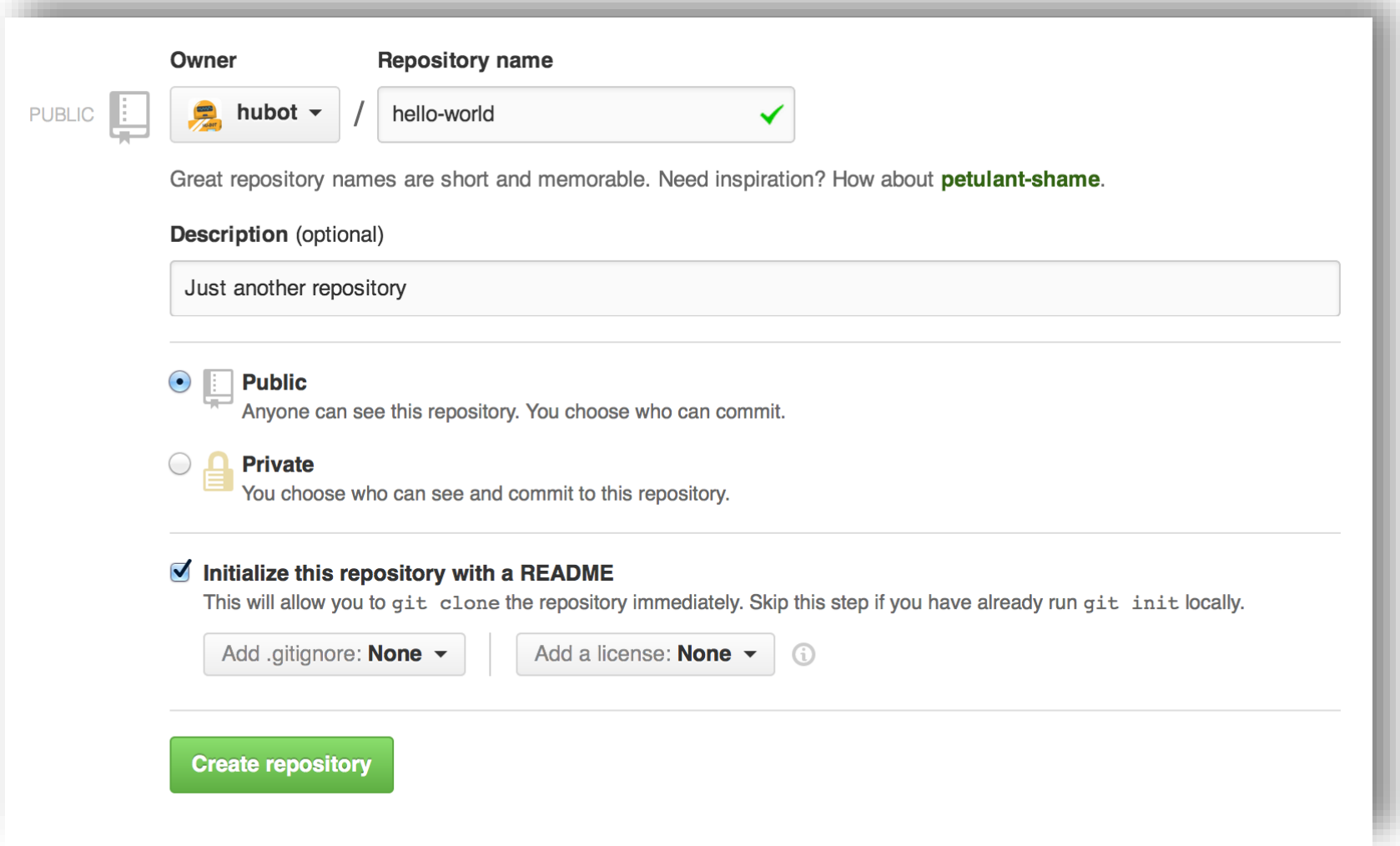
Your first repository

Upper right corner:



Choose:

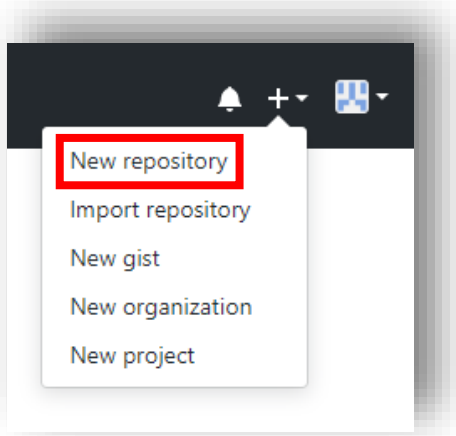
- name
- description
- status *(public/private)*
- README



Setting up

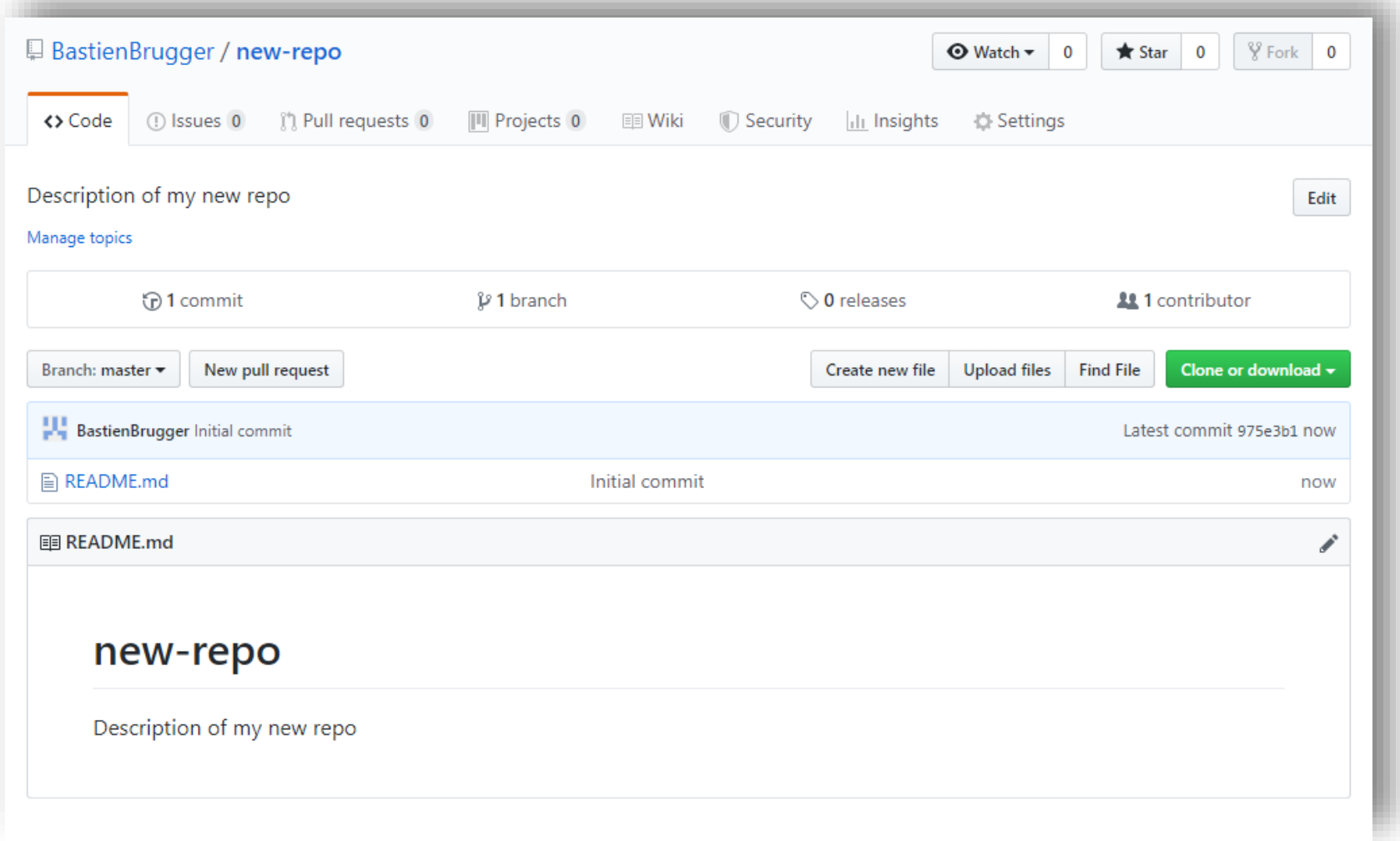
Your first repository

Upper right corner:



Choose:

- name
- description
- status *(public/private)*
- README



Managing repos

Three options:

1 Browser

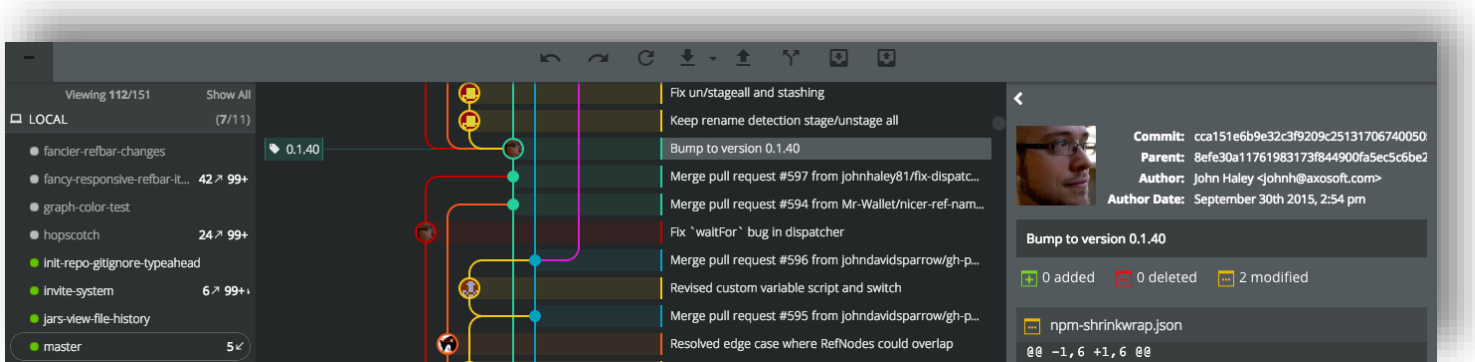
- Available everywhere, no compatibility issues
- Create/fork repos, manage files
- Be social and discover existing repos
- *Impossible to run your program online*

Managing repos

Three options:

2 Graphical User Interface (GUI)

- Multiple software options available
- Combines text editor + Git functions *(commit, pull request…)*
- User-friendly + graphic representation of collaboration



Managing repos

Three options:

3 Command Line Interface

- Control over everything you do
- Automation via custom scripts
- Most documented method online

➡ `git-scm.com/download/<linux/mac/win>`