



FAKULTÄT FÜR  
INFORMATIK

# Collaborative Software Development: Git and Github

---

Sandro Schulze, 26.04.2016

# Important Note

---

- Git is a tool for
  - YOU to enable more efficient development
  - ME to investigate contributions of team members
- You HAVE TO use git
  - commit frequently
  - use the README to spread the word (e.g., how to use your app)



# Roadmap

---

- Git
  - What makes it different?
  - Basic concepts
- Github
  - Social Coding in-the-large
  - how to use it for an SW development project
- Travis CI
  - Basic idea
  - How does it integrate with the development process





# Git

A brief introduction

“I am an egoistical bastard, and I name  
all my projects after myself. First Linux,  
now git”

–Linus Torvalds

# History

---

- originally, the Linux kernel has been developed by sending around patches or archive files (1991—2002)
- in 2002: use of proprietary DVCS called BitKeeper relationship broke down in 2005
- Hence, they (most notably, Linus Torvalds himself) decided to develop a proprietary tool  
—> Git was born

# Goals

---

- Speed
- Simple Design
- support for non-linear development
- fully distributed
- able to handle large projects such as the Linux kernel efficiently



# Git Basics

---

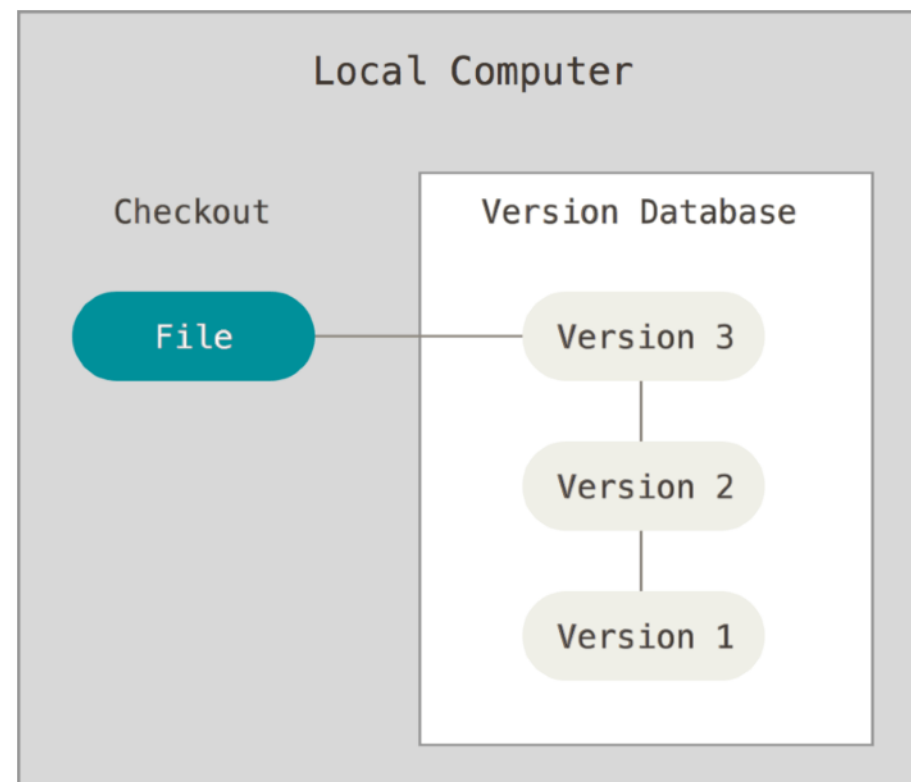




# From Local Versioning to Distributed VCS

## *Version Control Systems (VCS)*

---

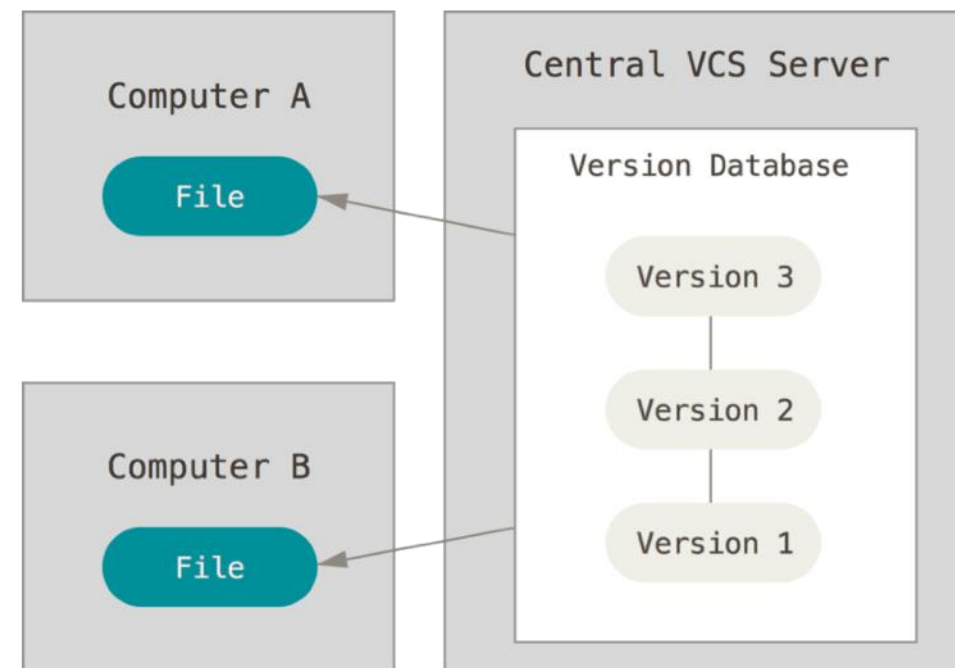


- introduced to overcome simple copy&paste of files
- database to keep changes to files
- Tool: RCS (try rcs in your shell ;))
- Technique: storing of patch sets on disk

# From Local Versioning to Distributed VCS (II)

## *Centralized Version Control Systems (CVCS)*

---

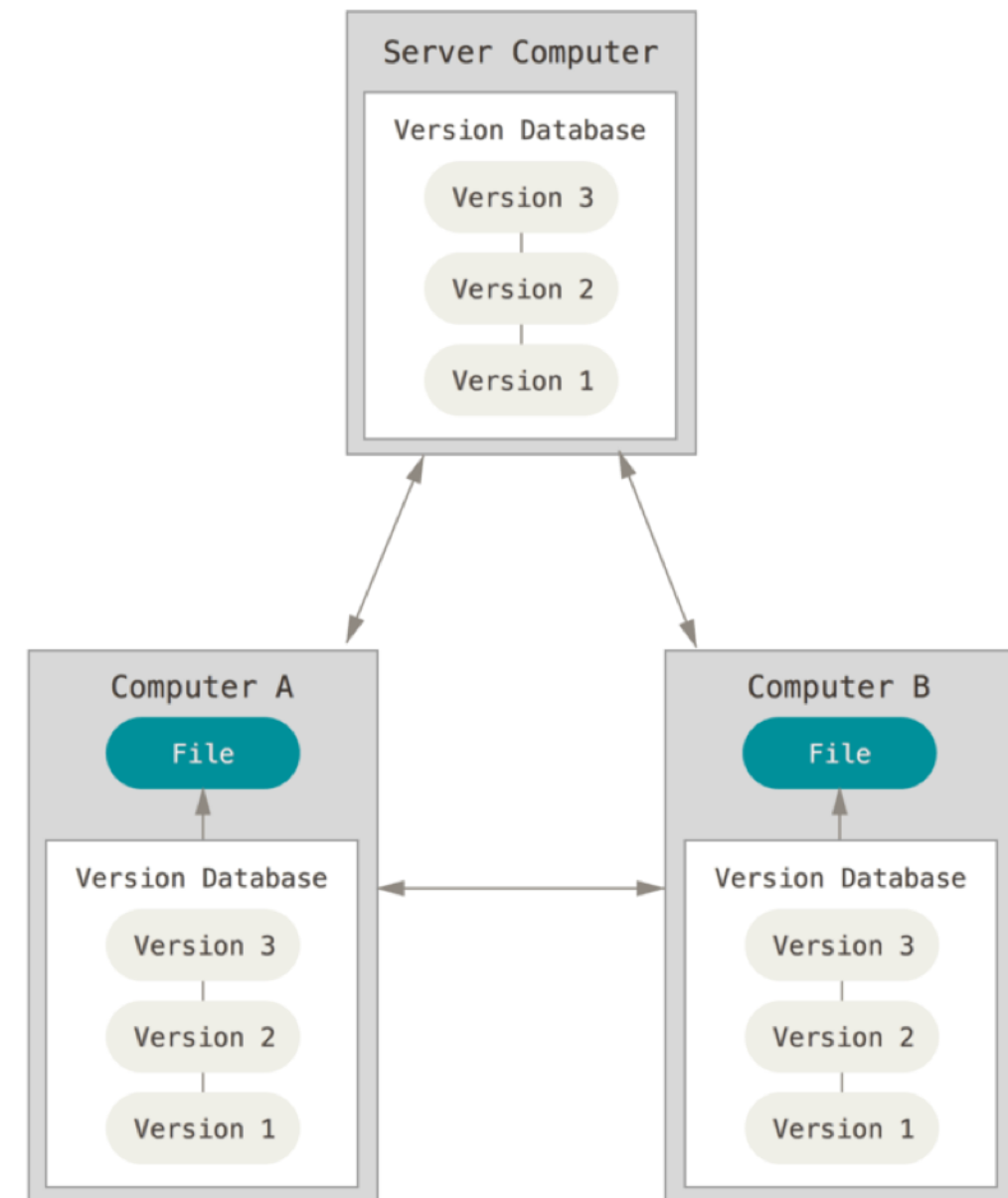


- Originated by need to collaboration between developers
- central server that contains DB with all change information (versions)
- de-facto standard for many years
- Tools: SVN, Perforce

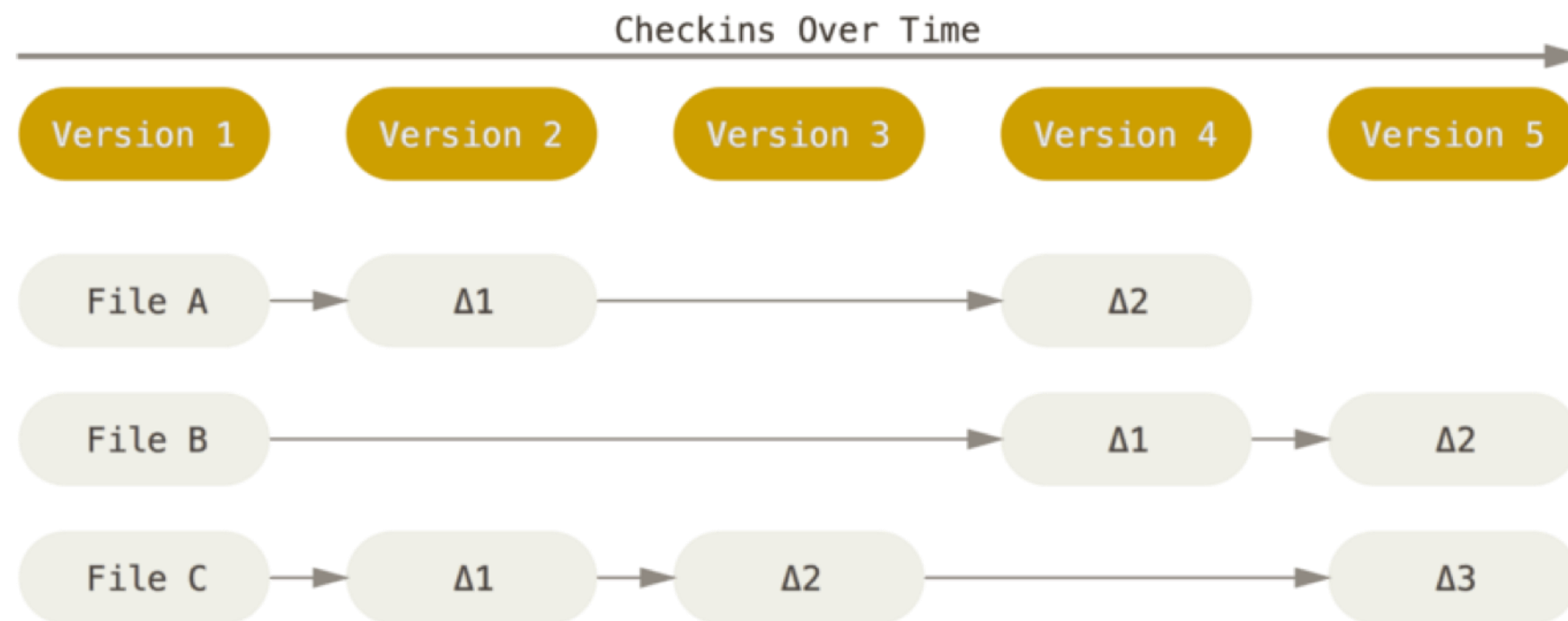
# From Local Versioning to Distributed VCS (II)

## *Distributed Version Control Systems (DVCS)*

- Clients not just check out, they fully mirror the repository (clone)
- Thus, can work even offline on the project
- support for having several repos of the same project —> enables new workflows (compared to CVCS)

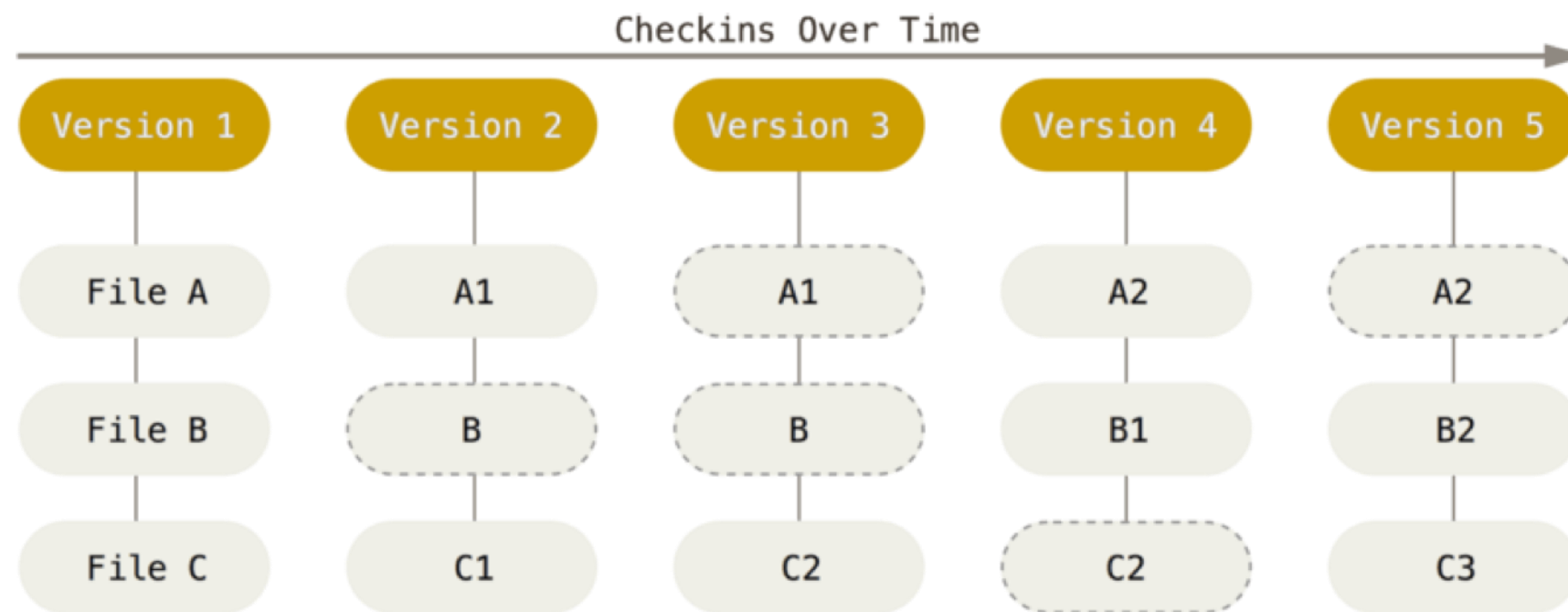


# Data Handling — Snapshots, not differences



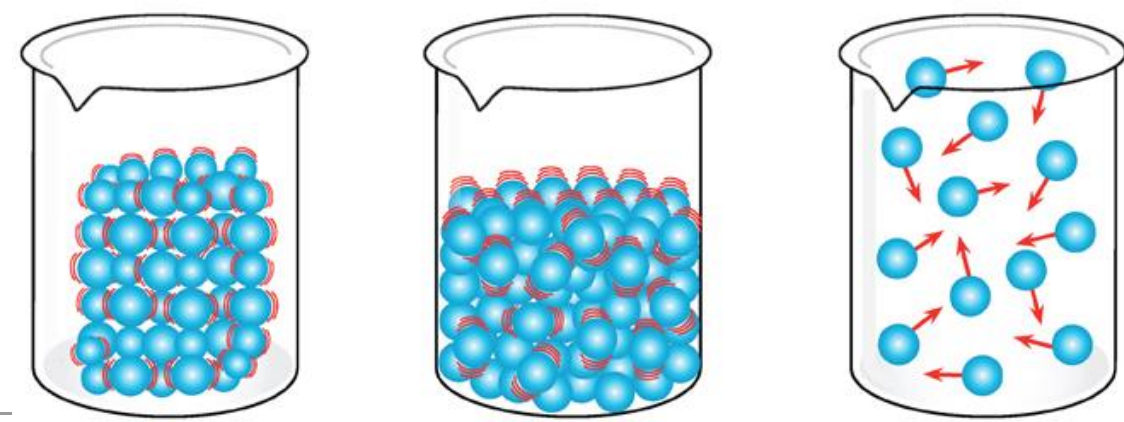
- In common version control systems (VCS) such as SVN, data is stored as changes to a base version of each file

# Data Handling — Snapshots, not differences



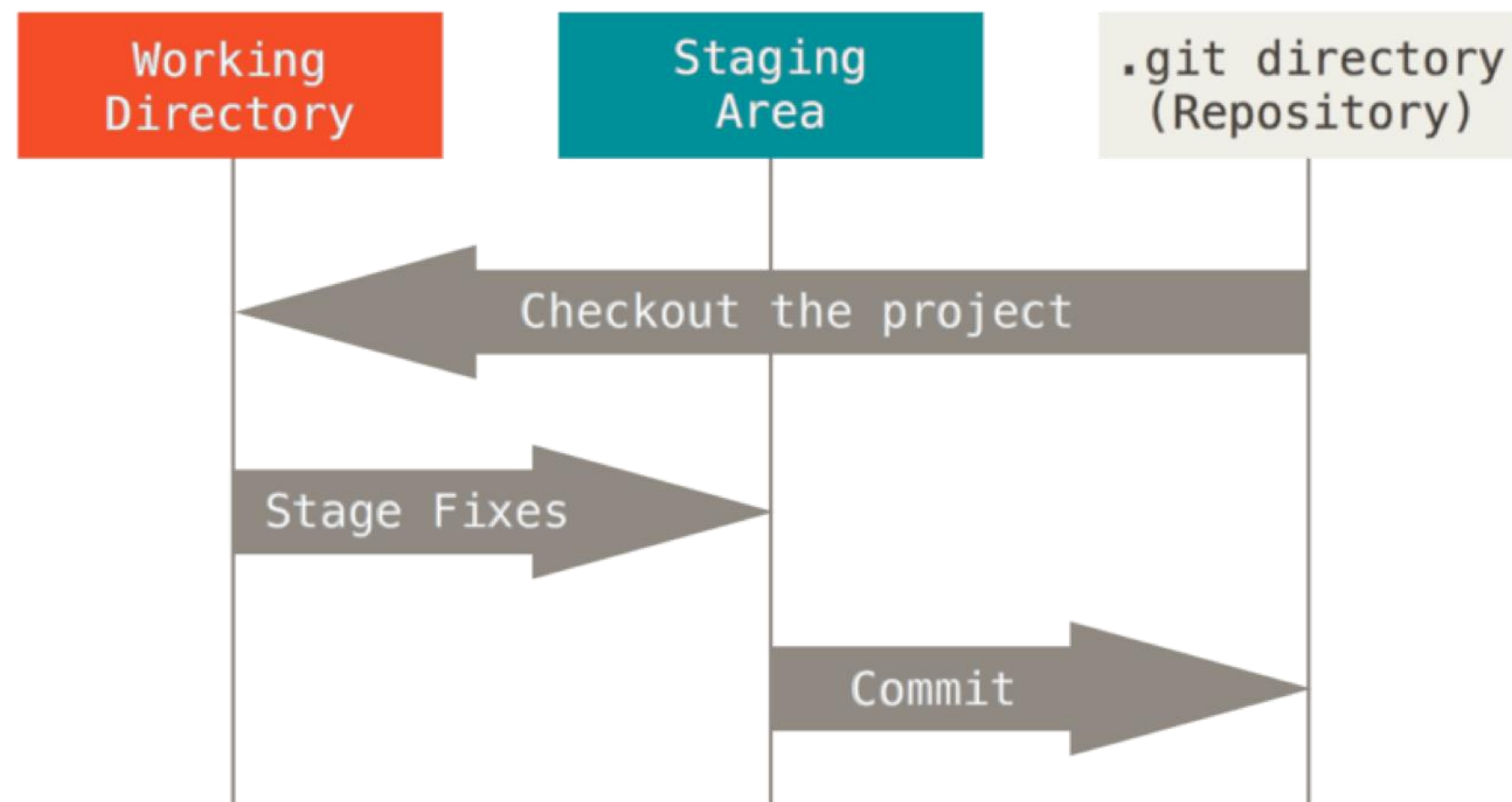
- In Git, data is handled rather as snapshots of a miniature file system
- On commit, the current “state” (i.e., snapshot) of your project is stored
- If files not change —> link to previous version of this file

# The Three States



- Git has **three** main states your files can reside in: *committed, modified, and staged*
- committed —> safely stored in (local) database
- modified —> changed file, but not committed to DB yet
- staged —> marked file (current version) to be part of next commit snapshot
- These states are reflected by the actual workflow in git

# Structure of a Git Project

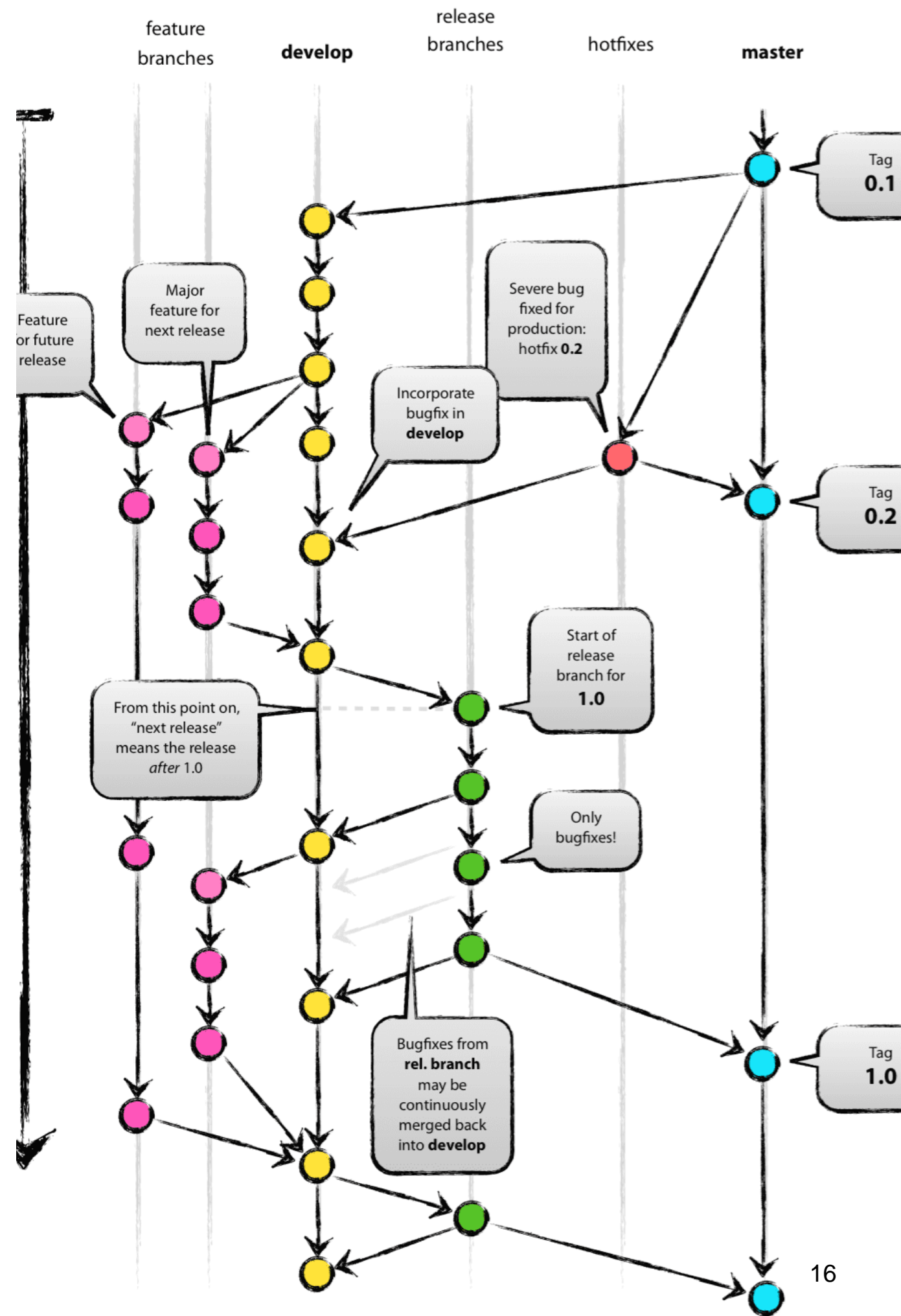


- Git directory —> stores metadata and object database
- working dir. —> single checkout of *one version* of the project
- staging areas —> file that stores information about what goes into the next commit



# Branching

How data is stored in git





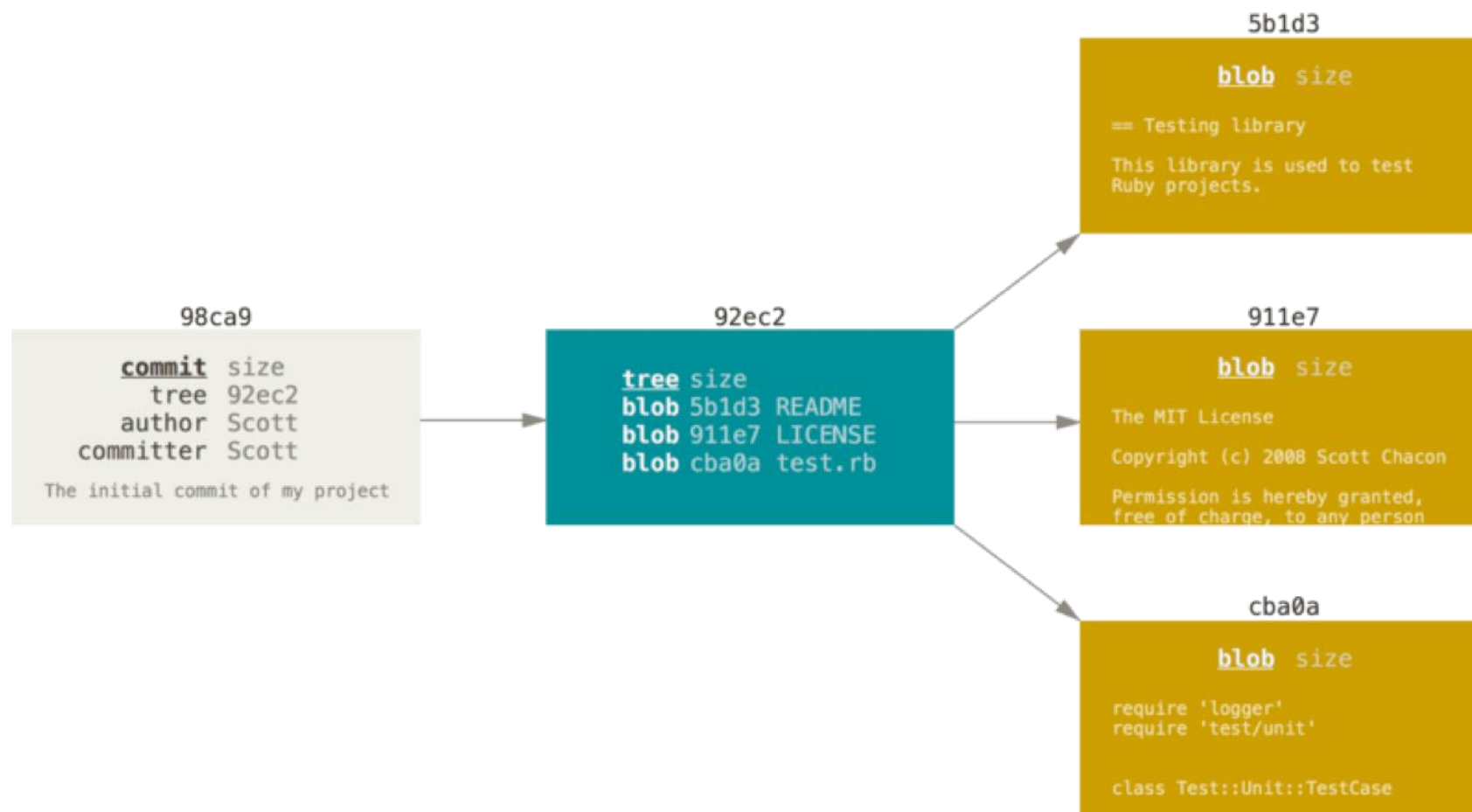
# General Process

---

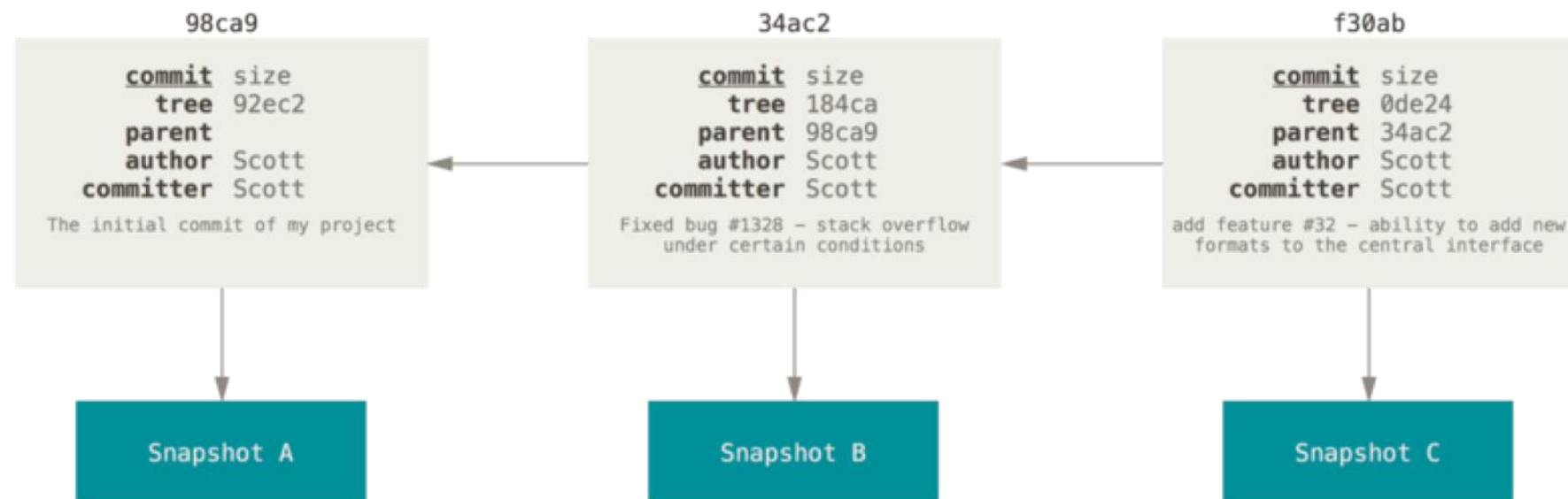
- When committing, git stores a *commit object*
  - contains pointer to the snapshot you staged
  - includes metadata (author, message, etc.)
  - contains also pointer to previous commit(s)  
—> multiple parents in case of previous merge
  - on commit, a checksum is computed —> ID of commit

# Example: initial commit

```
$ git add README test.rb LICENSE
$ git commit -m 'The initial commit of my project'
```

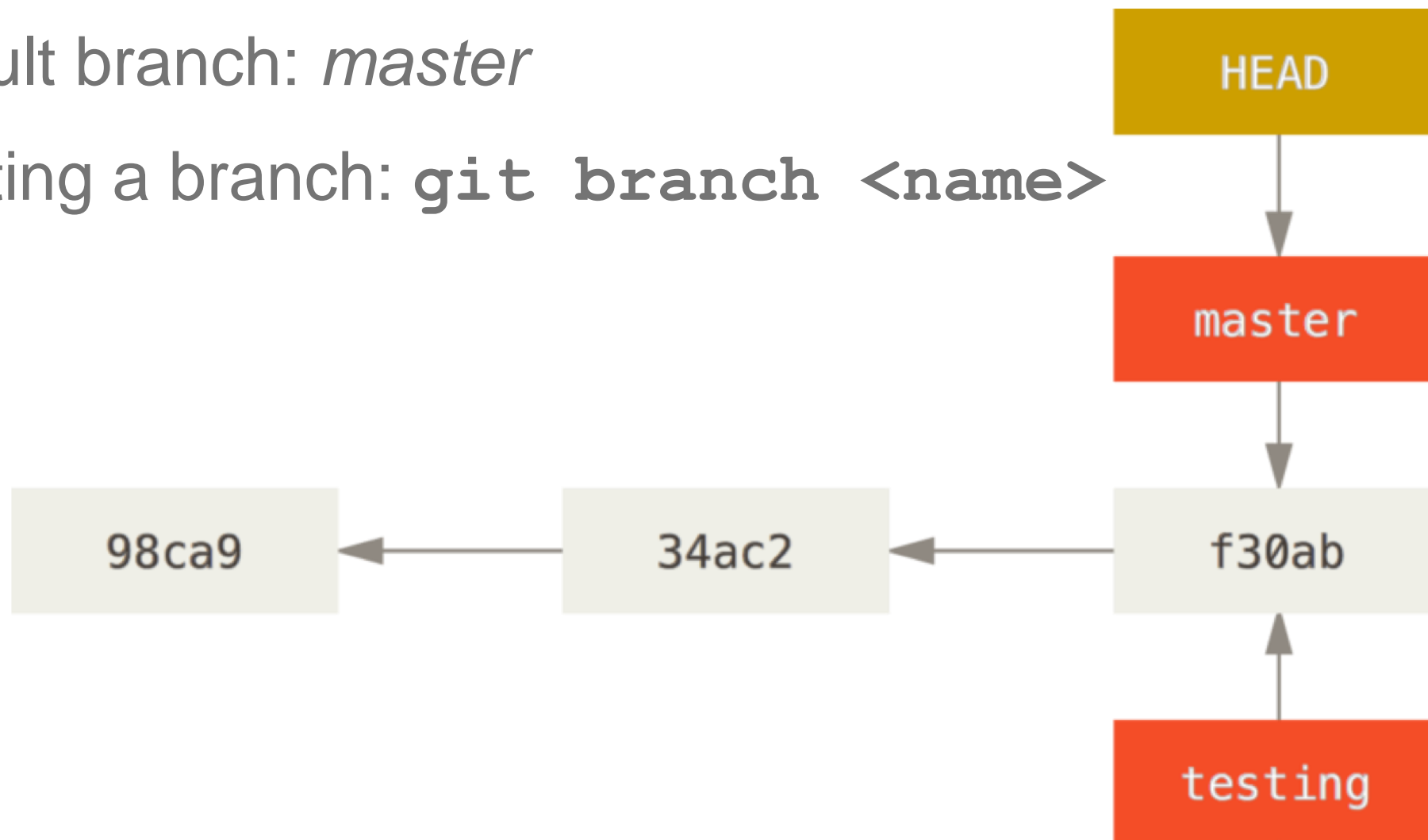


# Example: multiple commits



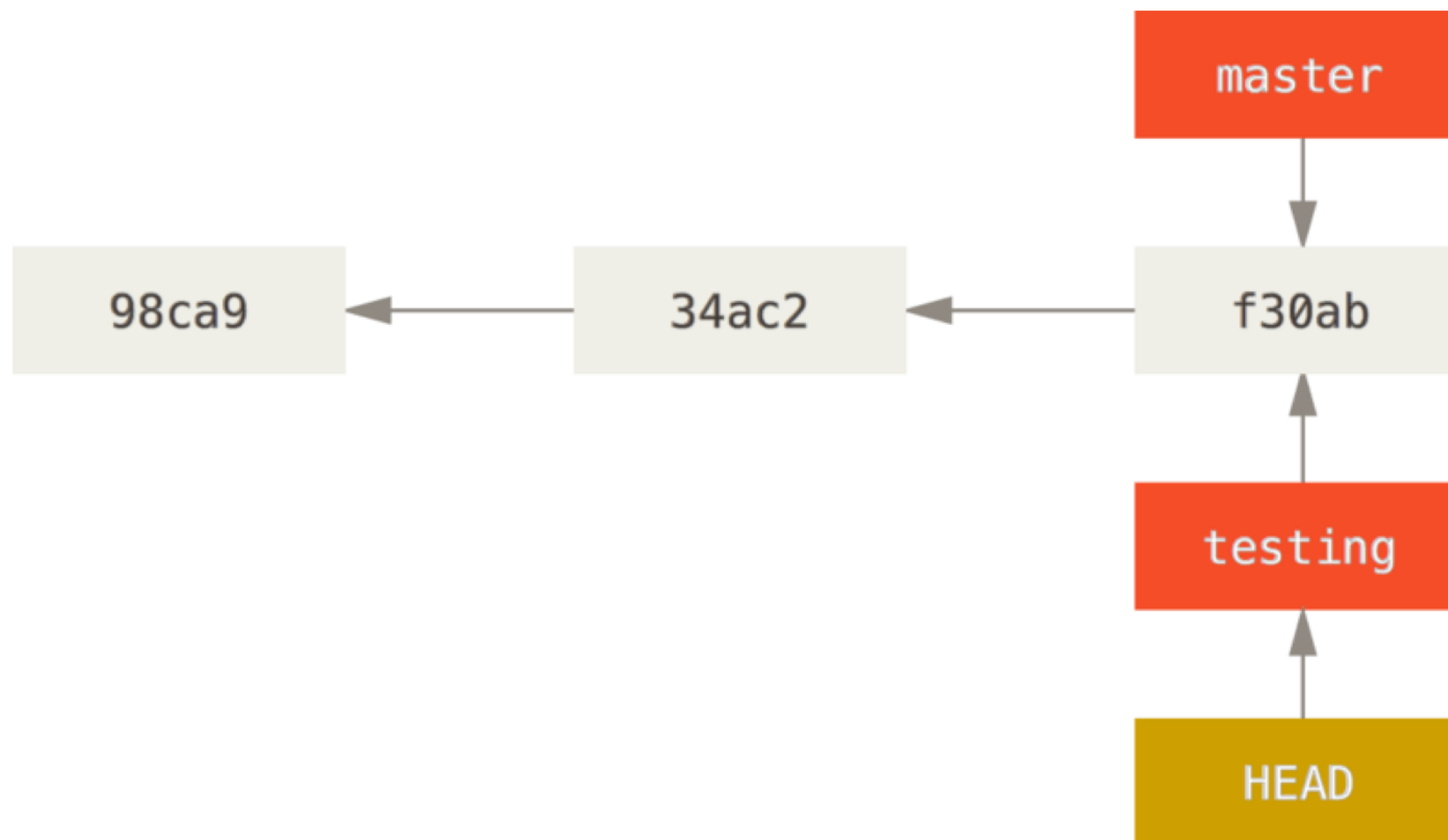
# Example: Branching

- A branch is just a movable pointer to one of the previous commits
- default branch: *master*
- creating a branch: `git branch <name>`



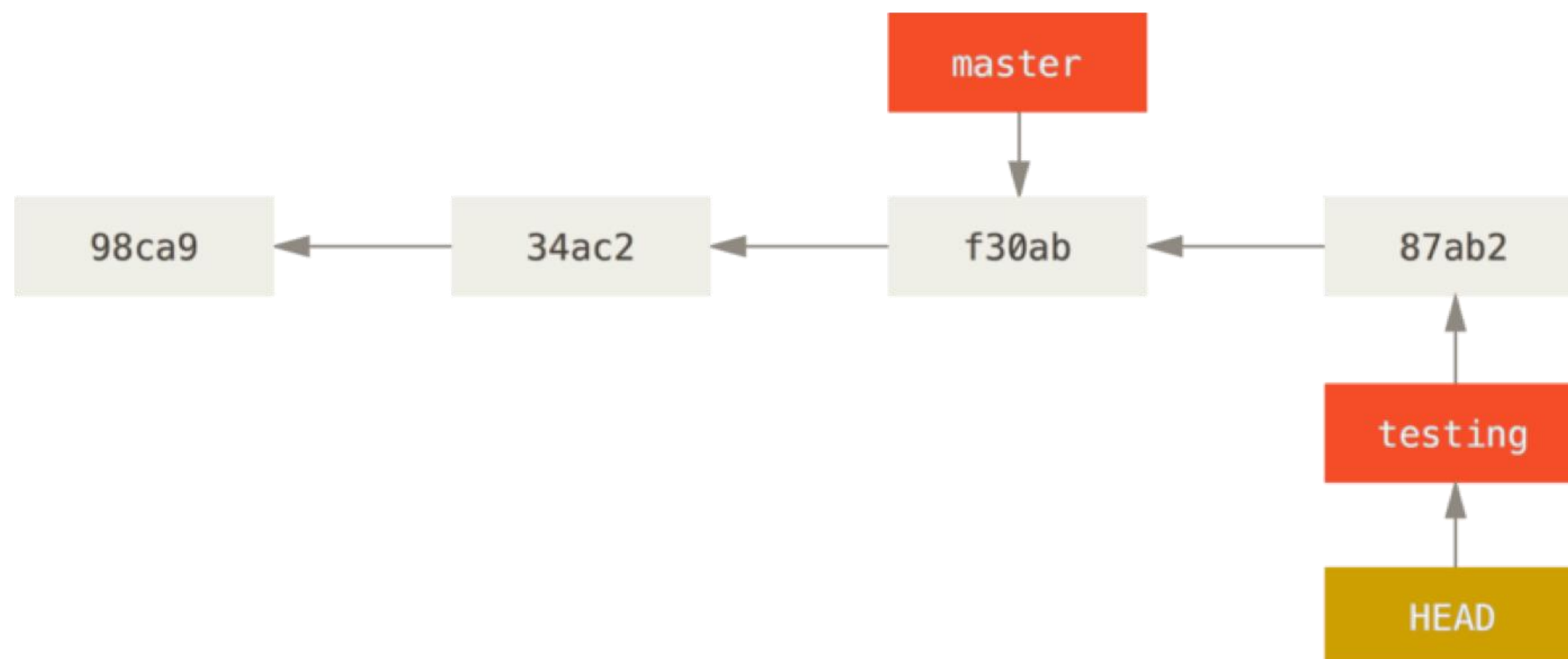
# Example: Switch Branches

- switching branches: `git checkout <name>`

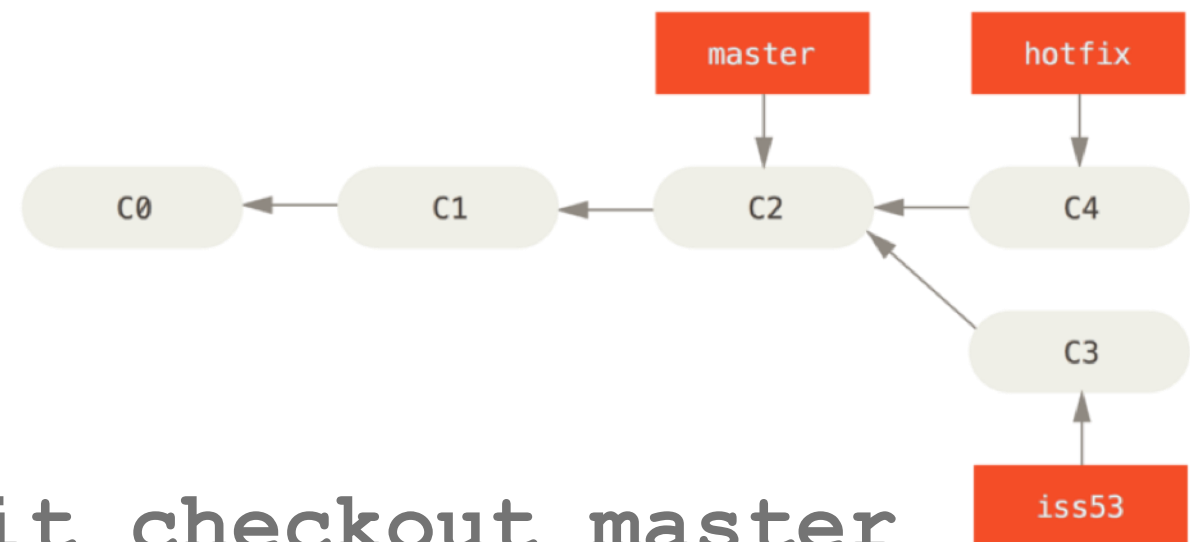


# Example: Committing to a Branch

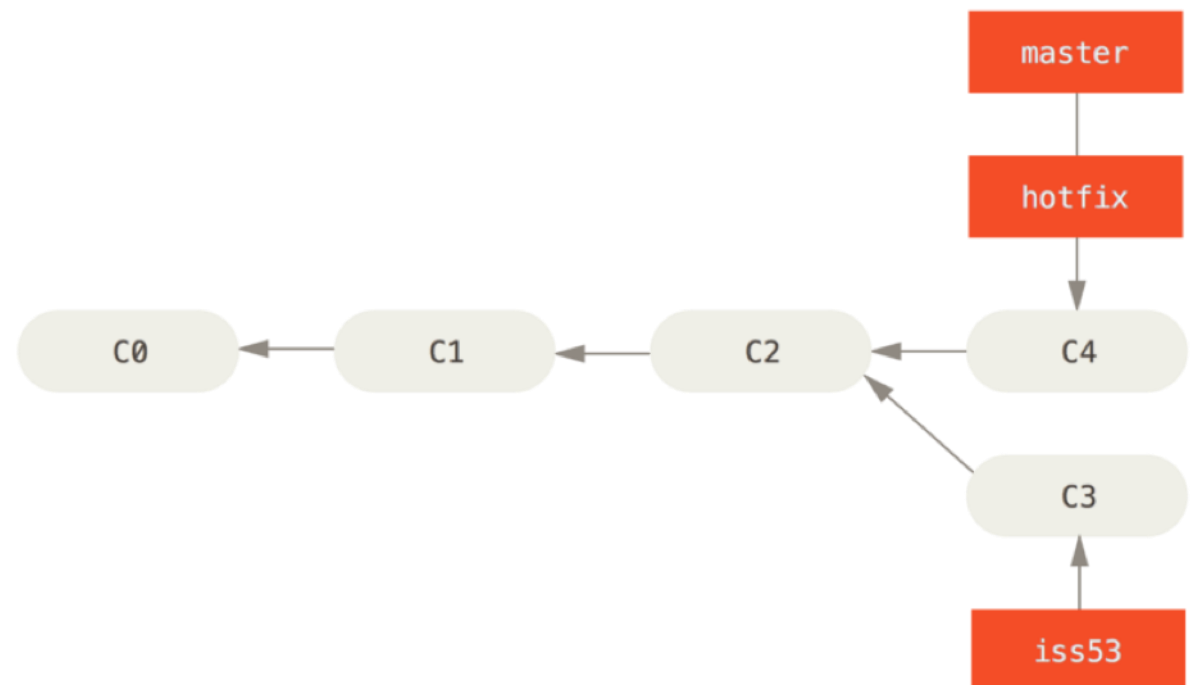
- Assume we change file `test.rb` and commit the change to the *testing* branch



# Unifying Branches — Merging



- change to master branch: `git checkout master`
- merge with hotfix branch: `git merge hotfix`





# Github

A toolkit on top of Git



1,067,856 people hosting over 3,012,331 git repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and [many more](#)

















**git** /'git/

Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.

**git·hub** /'git,hʌb/

GitHub is the best way to collaborate with others. Fork, send pull requests and manage all your **public** and **private** git repositories.

**Plans, Pricing and Signup**

Unlimited public repositories are **free!**

Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...

## Team management

**30 seconds** to give people access to code. No SSH key required. Activity feeds keep you updated on progress.

[More about collaboration](#)

## Code review

Comment on changes, track issues, compare branches, send pull requests and merge forks.

[More about code review](#)

## Reliable code hosting

We spend all day and night making sure your repositories are **secure**, **backed up** and **always available**.

[More about code hosting](#)

## Open source collaboration

Participate in the most important open source community in the world today—online or at one of our meetups.

[More about our community](#)

1,067,856 people hosting over 3,012,331 git repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and [many more](#)

twitter

facebook

rackspace  
HOSTING

digg

YAHOO!

shopify

EMI

six apart

git /'git/

Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.

git·hub /'git,hʌb/

GitHub is the best way to collaborate with others. Fork, send pull requests and manage all your **public** and **private** git repositories.

## Plans, Pricing and Signup

Unlimited public repositories are **free!**

Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...

### Team management

**30 seconds** to give people access to code. No SSH key required. Activity feeds keep you updated on progress.

[More about collaboration](#)

### Code review

Comment on changes, track issues, compare branches, send pull requests and merge forks.

[More about code review](#)

### Reliable code hosting

We spend all day and night making sure your repositories are **secure**, **backed up** and **always available**.

[More about code hosting](#)

### Open source collaboration

Participate in the most important open source community in the world today—online or at one of our meetups.

[More about our community](#)

# Overview

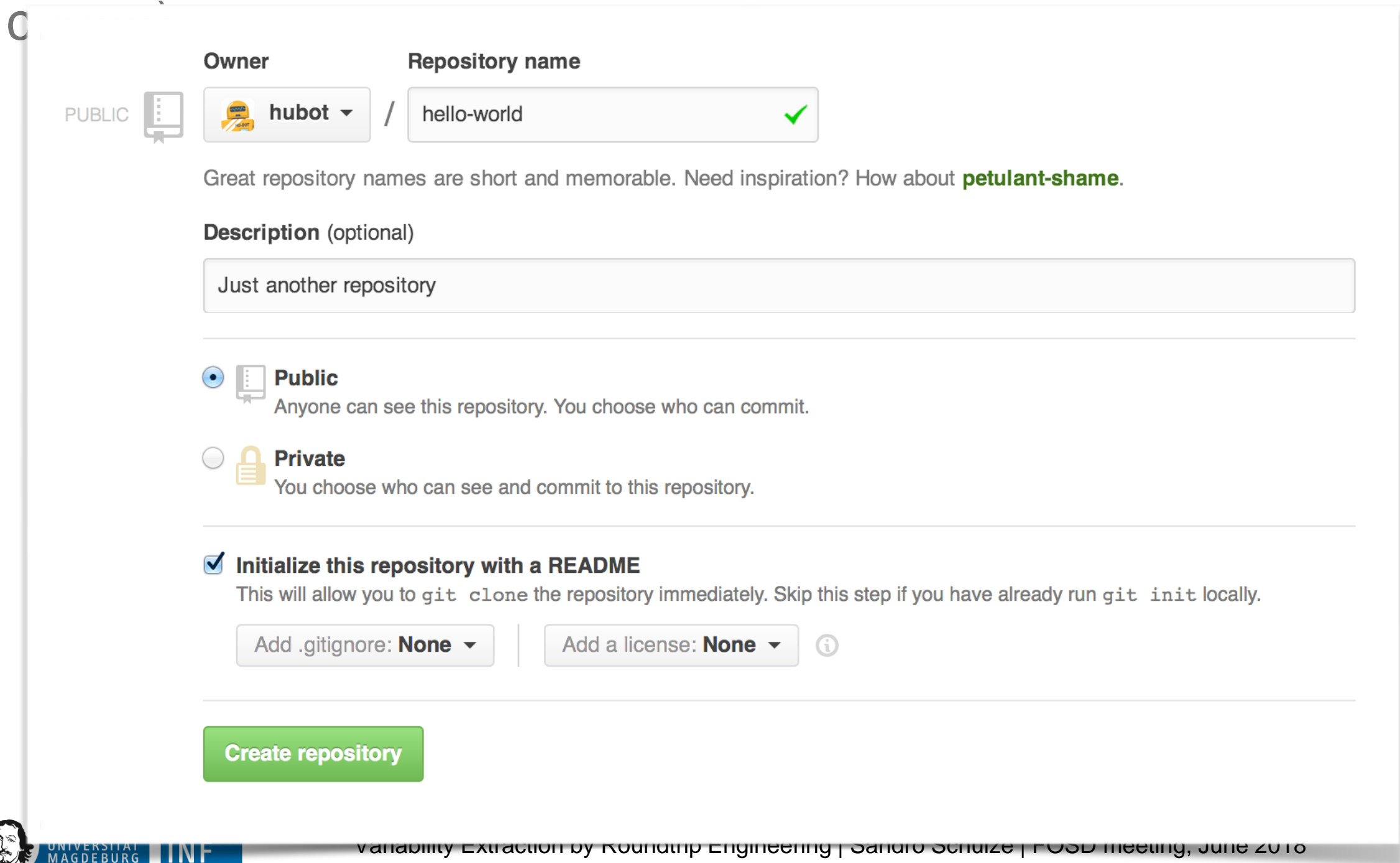
---

- Github provides remote hosting of git repositories
- Beyond that, they provide rich, web-based tool support for collaborative development
  - forking
  - issue tracking
  - pull requests
  - markdown-supported wiki, github page etc.
- provide integration with additional services, e.g., Continuous Integration



# Create a Repository

- Initiate a new repo by clicking “new repository” (upper right



The screenshot shows the GitHub 'Create a new repository' form. At the top, there's a 'PUBLIC' label with a lock icon. Below it, the 'Owner' is set to 'hubot' and the 'Repository name' is 'hello-world', which is marked with a green checkmark. A hint text says: 'Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.' The 'Description (optional)' field contains 'Just another repository'. Under the 'Visibility' section, 'Public' is selected with a radio button, and 'Private' is unselected. Below this, the 'Initialize this repository with a README' checkbox is checked. A note explains: 'This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.' At the bottom of this section, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. A large green 'Create repository' button is at the bottom.

Owner: hubot / Repository name: hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional): Just another repository

☒ **Public**  
Anyone can see this repository. You choose who can commit.

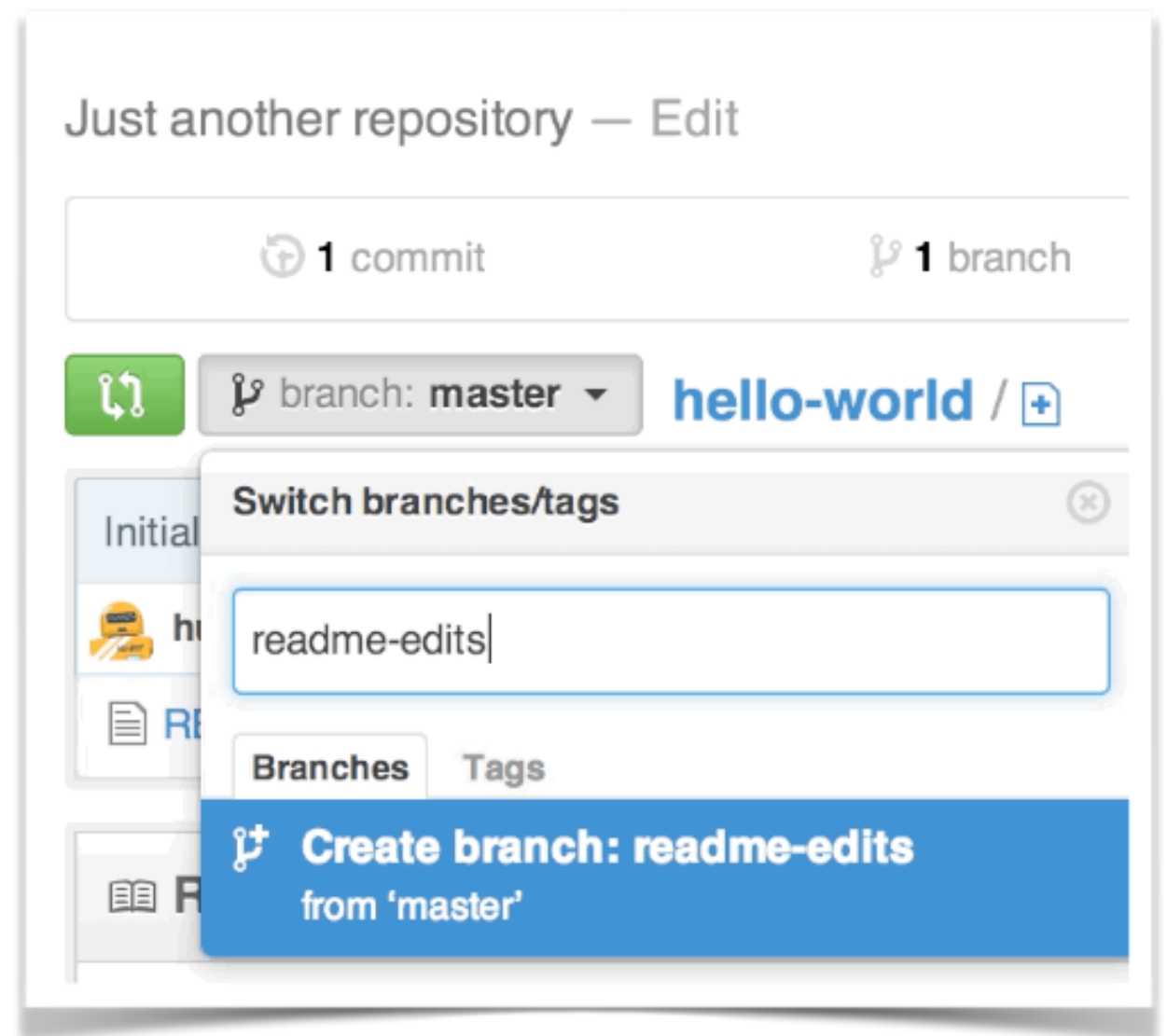
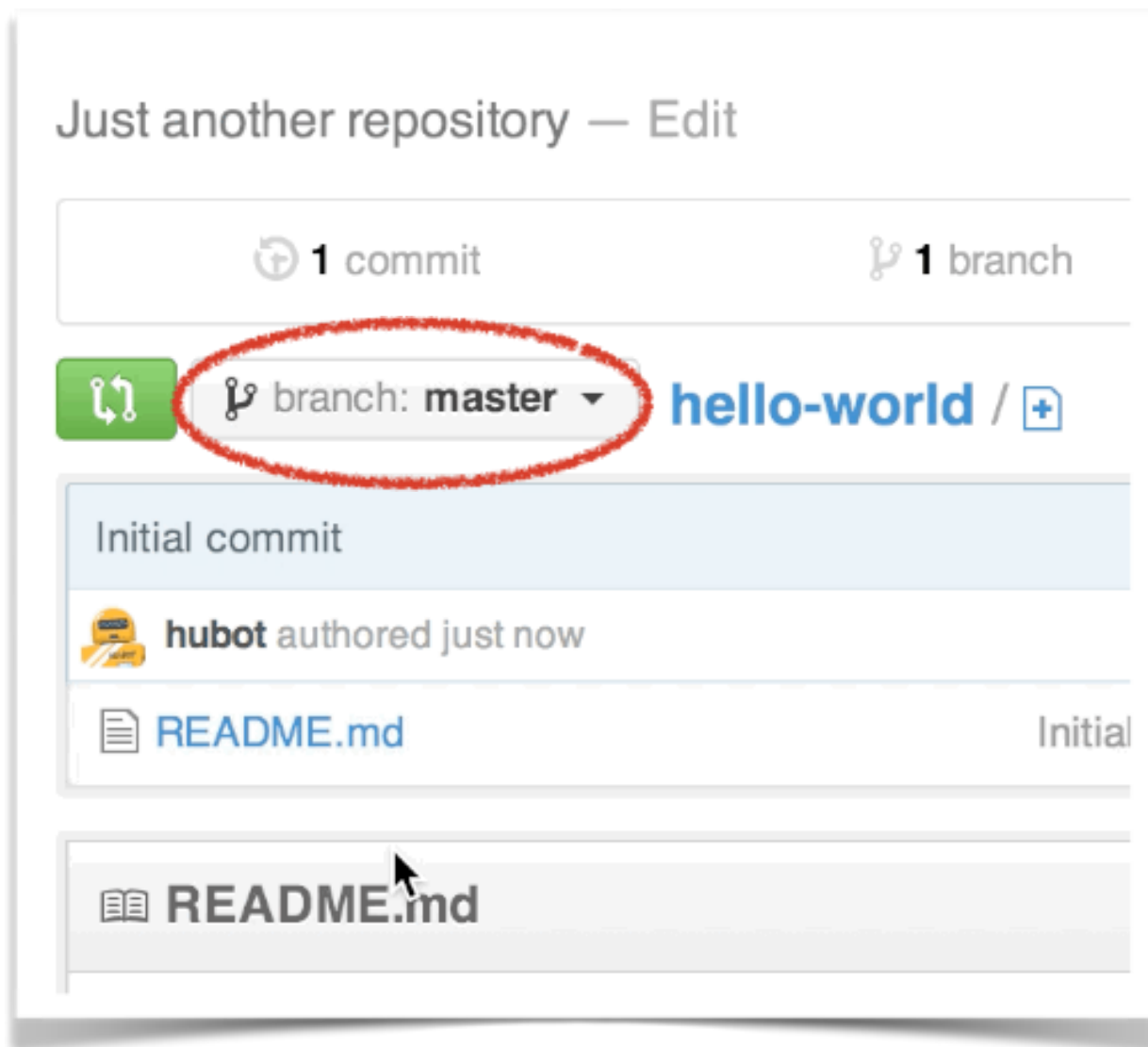
☐ **Private**  
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: None | Add a license: None ⓘ

**Create repository**

# Branching



# Forking

- Afte
- com

 **Watch** ▼ 35

 **Star** 45

 **Fork** 37

HTTPS clone URL

`https://github.com/or`



You can clone with [HTTPS](#), [SSH](#),  
or [Subversion](#). [?](#)

 **Clone in Desktop**

 **Download ZIP**

# Pull Request

---

- Getting the changes of forks in the main project is the highest benefit of distributed development/social coding
- Challenge: how to get changes from (possibly) hundreds of forks back to the main project without sacrificing stability?
- Pull Requests provide a structured workflow for doing so
- is also applicable between branches of the same fork!

# Creating a Pull Request

The screenshot shows the GitHub interface for a repository named 'Spoon-Knife'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these, the 'Pull requests' tab is selected, showing 0 pull requests. Other tabs include 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A search bar contains the text ':open'. To the right of the search bar are buttons for 'Labels' and 'Milestones'. A prominent green button labeled 'New pull request' is visible. Below the repository header, a section titled 'Your recently pushed branches:' shows a branch named 'new-file' (less than a minute ago) with a green button labeled 'Compare & pull request'. At the bottom, a status bar indicates 'This branch is 0 commits ahead and 0 commits behind master' and provides links for 'Pull Request' and 'Compare'. The main content area shows 'Adding a brand new file'.



# Creating a Pull Request

- description uses markdown: you can do formatting, adding pics etc. to make clear WHY you make this pull request
- Use @mention to get feedback from specific persons

Create CONTRIBUTING.md

Write

Preview

 Parsed as Markdown  Edit in fullscreen

Let's add a contributing file so we can work better, together!

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.



✓ **Able to merge.**

These branches can be automatically merged.

Create pull request

# Merging your PR

- This is usually done by project owner —> PRs between fork and main project need to be merged by owners of the original repo



**This branch has no conflicts with the base branch**

Merging can be performed automatically.



**Merge pull request**

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



**Pull request successfully merged and closed**

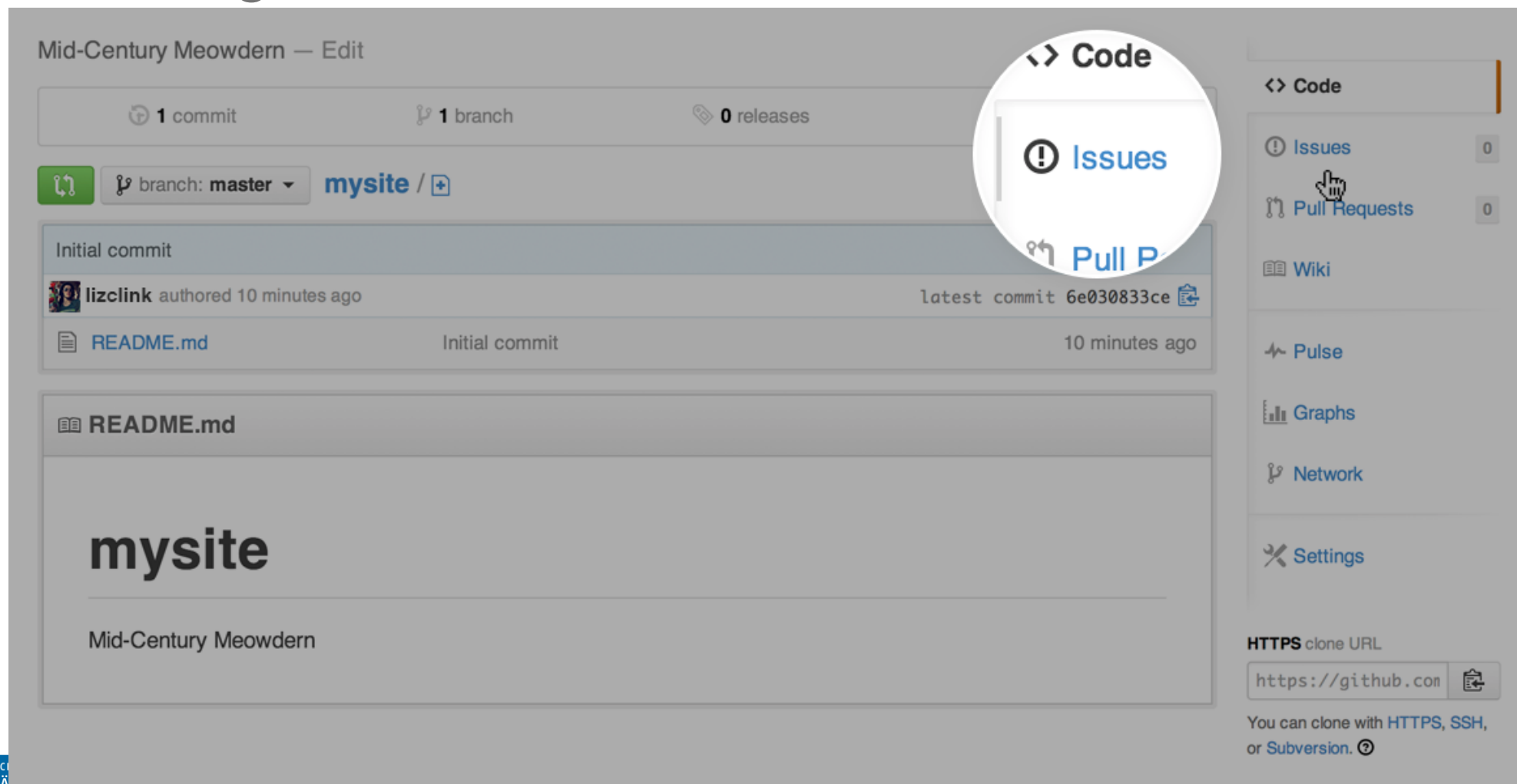
You're all set—the `readme-edits` branch can be safely deleted.



**Delete branch**

# Issue tracking with Github

- best way to keep track of any progress  
—> tasks, features, enhancements, bugs
- you may think of using it to structure and document your project management



# Example: A project using Github's Issues

Issues

Pull requests

Labels

Milestones

Filters

New issue

104 Open ✓ 9,660 Closed

Author ▾ Labels ▾ Milestones ▾ Assignee ▾ Sort ▾

!

**.form-group-sm .form-group-lg shrink textarea**

confirmed

css

4

#13989 opened 11 hours ago by limitstudios ☐ v3.2.1

!

**Tooltip unnecessarily breaks into multiple lines when positioned to the right**

confirmed

js

0

#13987 opened 15 hours ago by hnrch02 ☐ v3.2.1

!

**Tooltip Arrows in Modal example facing wrong way**

css

6

#13981 opened a day ago by SDCore

!

**Table improvement**

css

0

#13978 opened a day ago by Tjoosten

!

**docs/dist files**

docs

7

#13977 opened 2 days ago by XhmikosR ☐ v3.2.1

!

**Potential solution to #4647**

js

4

#13976 opened 2 days ago by julioarmandof

!

**Bootstrap site: right-hand navigation text becomes rasterized after scrolling**

css

docs

4

#13974 opened 2 days ago by mg1075 ☐ v3.2.1

!

**Dropdown toggle requires two clicks**

js

1

#13972 opened 2 days ago by Kizmar

# A Typical Issue — Overview

## ◀ The no-conflict mode should be the default behaviour #12395

[Edit](#)[New Issue](#)

**Open** thewebdreamer opened this issue 3 days ago · 10 comments



thewebdreamer commented 3 days ago

The no-conflict mode should be the default behaviour. Why would a Bootstrap client need to implement this?



cvrebert commented 3 days ago

I believe no-conflict-is-not-the-default is the norm for jQuery plugins?



thewebdreamer commented 3 days ago

It is true that it is the norm for jQuery plugins.

Couldn't there be a clash with other jQuery plugins with the current implementation of Bootstrap though?

Labels

js

Milestone

No milestone

Assignee

No one assigned

Notifications

Subscribe

3 participants



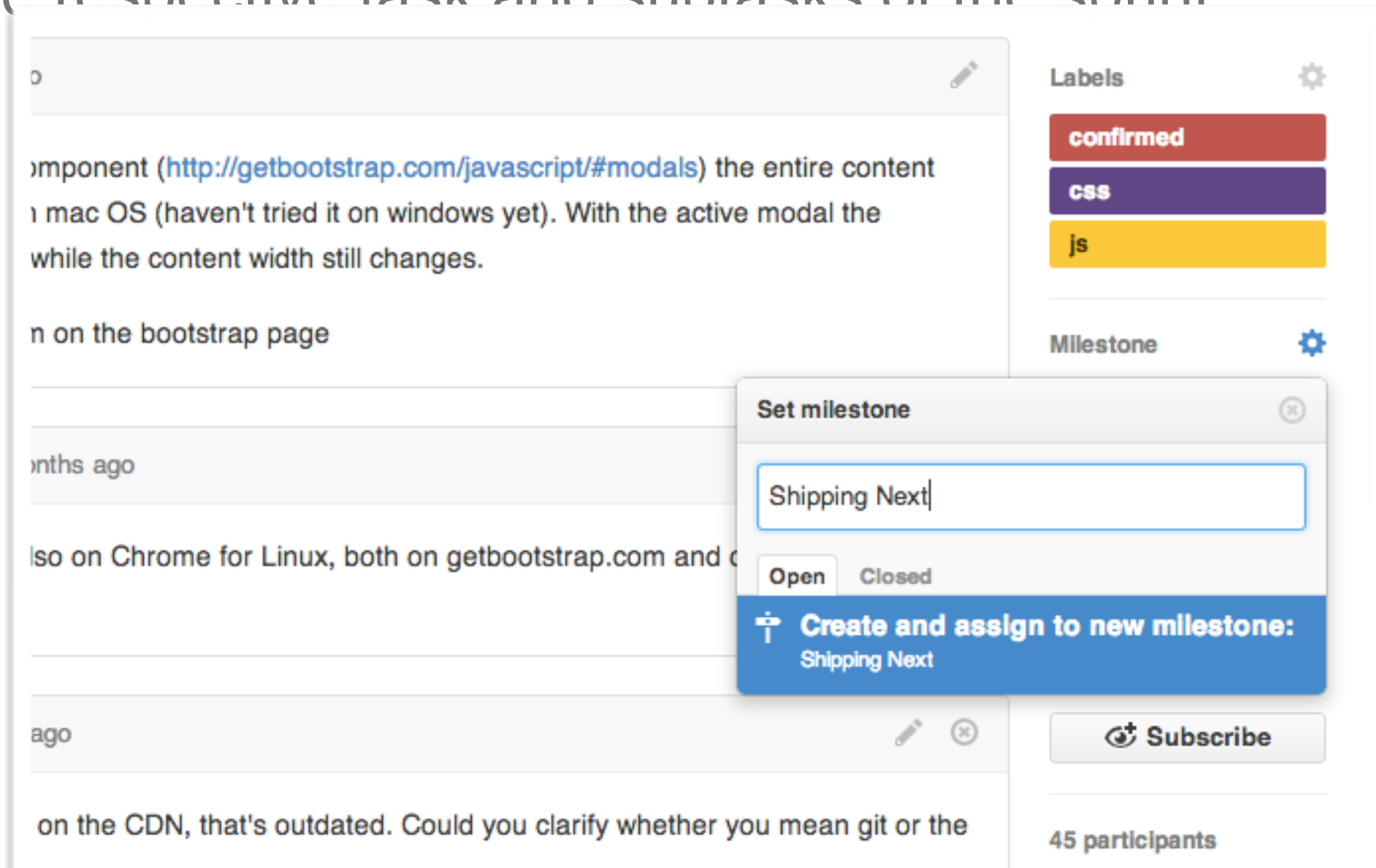
# Milestones, Labels, and Assignees

---

- after a while, you may get lost in all your issues
- Use milestones and labels to manage and structure them
- Moreover, assignees are useful to directly assign some work to be done to a team member (e.g., to track who is working on what)
- add or change all of them on the right side of your issue

# Milestones























- Milestones could be used to manage your sprints  
—> one milestone per sprint, afterwards you can open issues for the respective task and subtasks of the sprint





# Labels

- Use labels to organise different types of issues


	<b>Open modal is shifting body content to the left</b>   	#9855
Opened by mat0r 3 months ago  62 comments		
	<b>Navbar issues</b>   	#11243
Opened by Nugrata a month ago  36 comments		
	<b>Add support of extra styling class on collapse event</b>   	#11350
Opened by zioaschr 22 days ago  24 comments		
	<b>Redundant responsive utility styles</b> 	#11214
Opened by AlexYursha a month ago  24 comments		
	<b>Select tag not properly styled on stock android browser</b>  	#11055
Opened by ADmad a month ago  19 comments		




# Make use of Markdown facilities

- @mention, references, but also text formatting

Please review the [guidelines for contributing](#) to this repository.

 This is an example of a new issue

**Write** Preview MD Parsed as Markdown  Edit in fullscreen

Text fields on GitHub are parsed with Markdown. This means we can @mention people like @githubstudent or @octocat to reference them into the issue. You can also cross-reference other issues and pull requests with the number of those issues and pull requests like #14020

You can also create tasks list to monitor the progress of smaller tasks as well:

- [ ] #23
- [ ] #142
- [ ] #140

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

**Submit new Issue**

# Never miss a change to Issues

---

## How you receive notifications

### Participating

When you participate in a discussion or someone brings you in with an **@mention**.

☒ Email ☒ Web

### Watching

Updates to any repositories or threads you're [watching](#).

☐ Email ☒ Web

# The Big picture

- Make use of Issue Dashboard
- Check the Pulse feature on Github

August 01 2013 - August 08 2013

Period: 1 week ▼

## Overview



65 active pull requests



257 active issues

🔗 49

merged pull requests

🔗 16

proposed pull requests

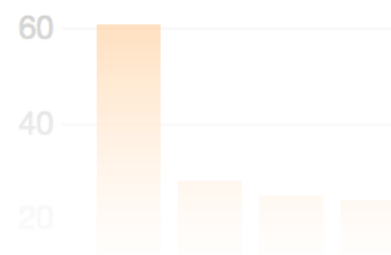
🔒 230

closed issues

🚨 27

new issues

54 authors have pushed 253 commits to all branches, excluding merges. On 3.0.0-wip, 64 files have changed and



# More on this

---

<https://guides.github.com/>



Greetings from