Federico Cantini
federico.cantini@lib4ri.ch

Frank Hösli
frank.hoesli@lib4ri.ch

Make your life easier with GIT

GIT basics

Motivation

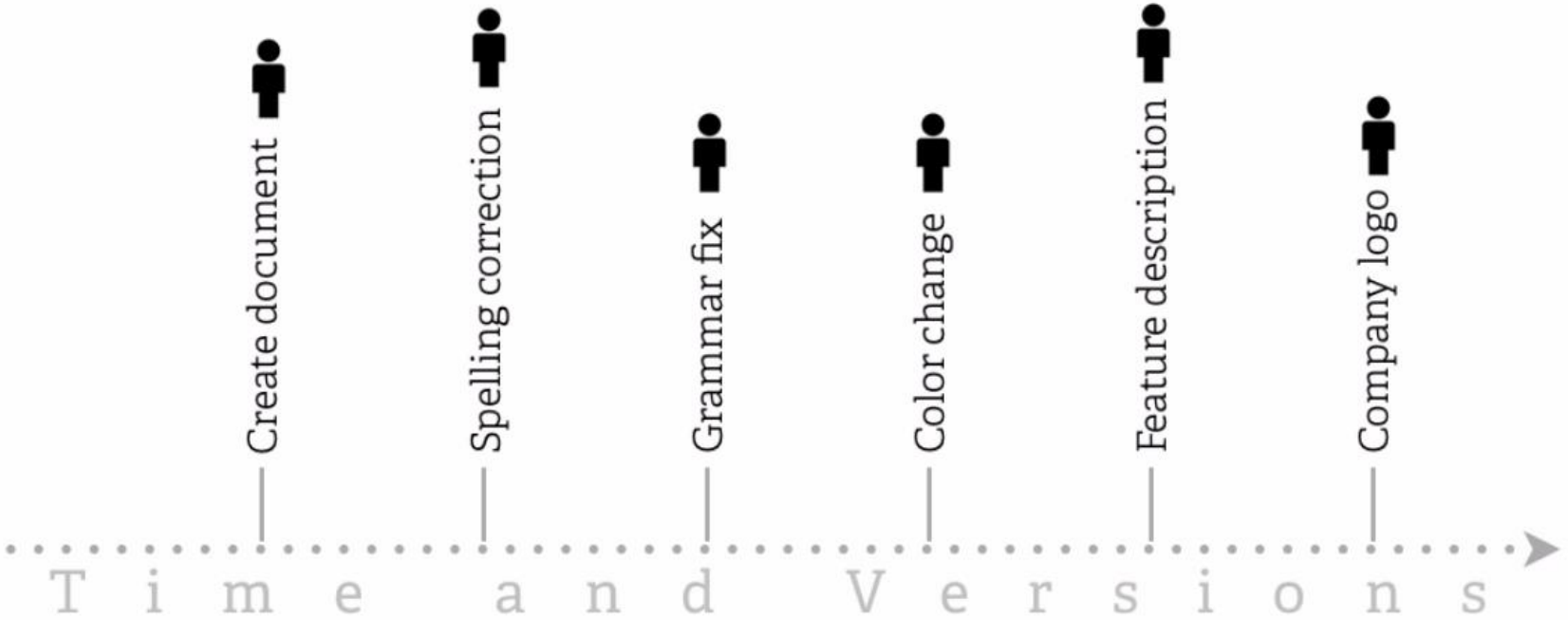
Have you ever:

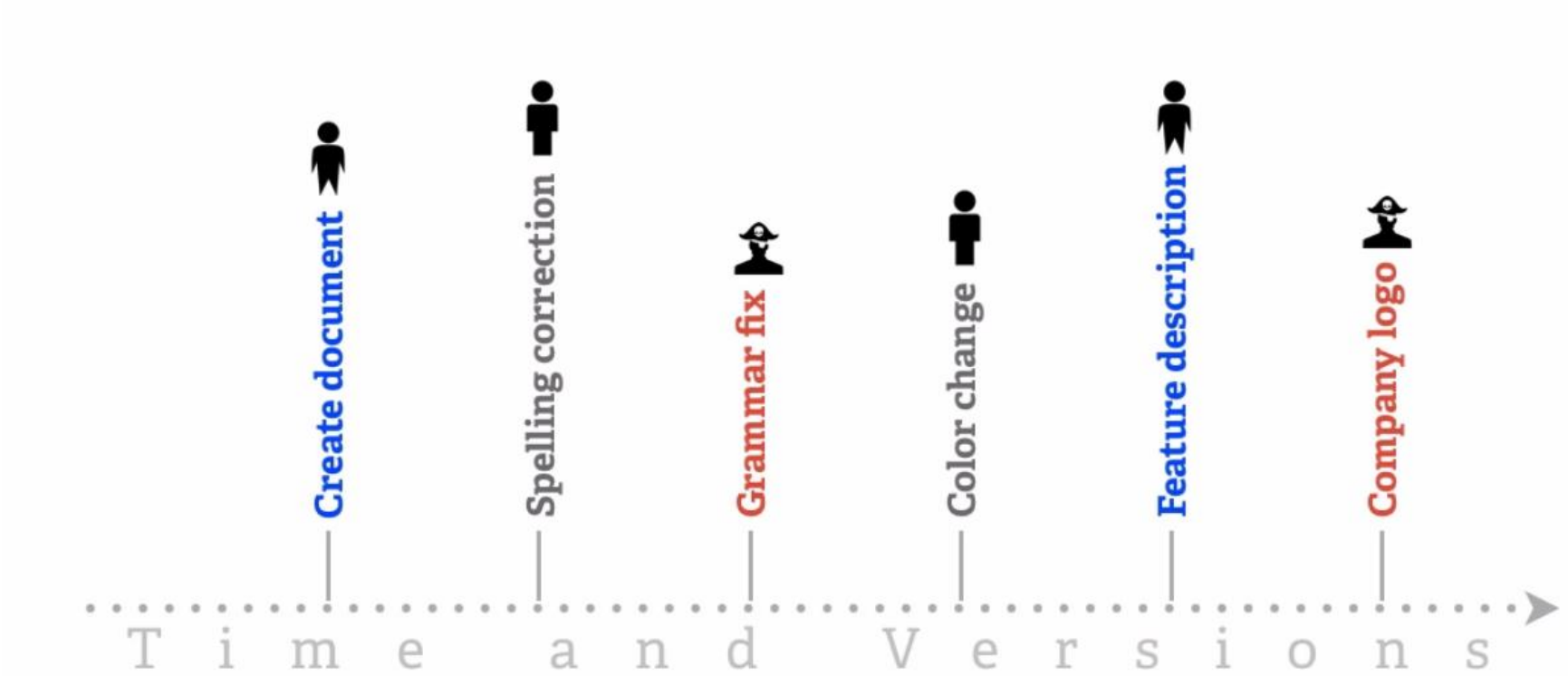
- Made changes you didn’t mean to?
- Made some changes, but can’t remember what they were?
- Wanted to try something but was worried about breaking something else?
- Went crazy trying to figure out which version of your code produced certain output?
- Tried to set up a convenient workflow for working on code together with others?

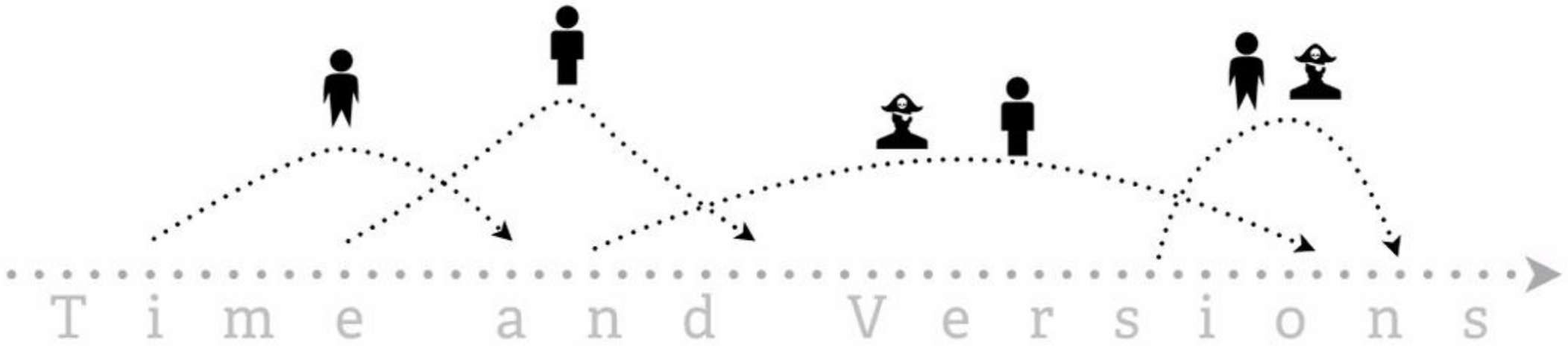
Version control

- Version control is a system that records changes to a file or set of files over time
- It allows to:
  - revert selected files back to a previous state
  - revert the entire project back to a previous state
  - compare changes over time
  - work concurrently with others
  - see who last modified something and when

- If you break things you can (usually) easily recover


Create document
Spelling correction
Grammar fix
Color change
Feature description
Company logo
Time and Versions


Create document
Spelling correction
Grammar fix
Color change
Feature description
Company logo
Time and Versions

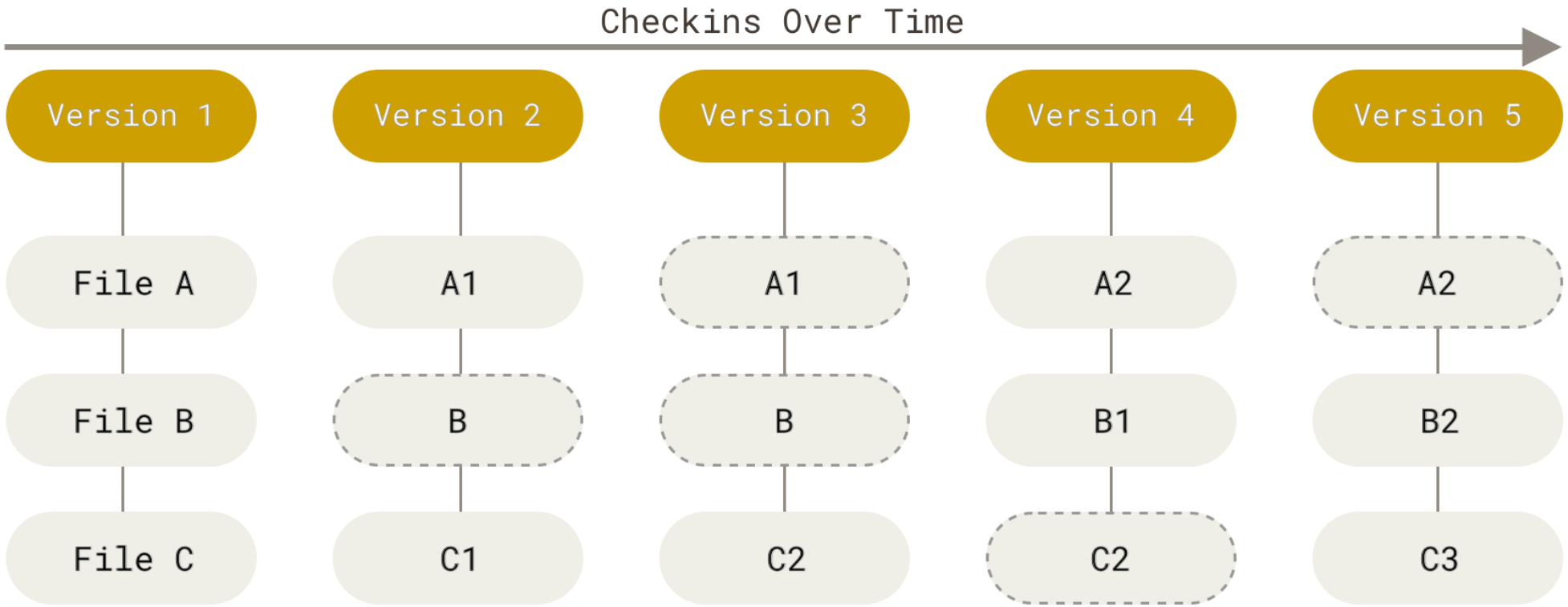

Time and Versions

https://git-scm.com

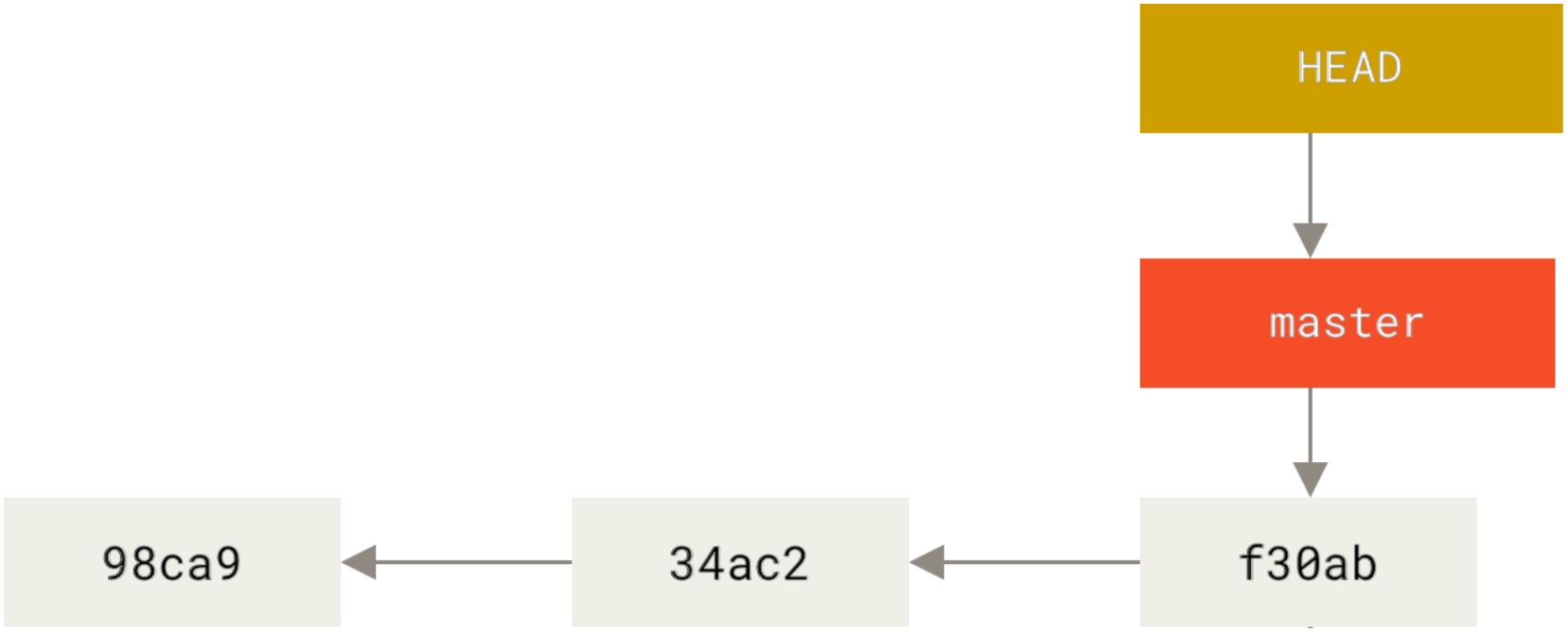
- First release April 2005
- Started by the team developing the Linux kernel
- De facto standard Version Control System
- Free/open source software (GPL v.2 license) https://git-scm.com/about/free-and-open-source
- Available for GNU/Linux, macOS and Windows https://git-scm.com/downloads
- Works with any programming language (any text file indeed and possibly other file types)
- Command line interface
- GUI clients available https://git-scm.com/downloads/guis

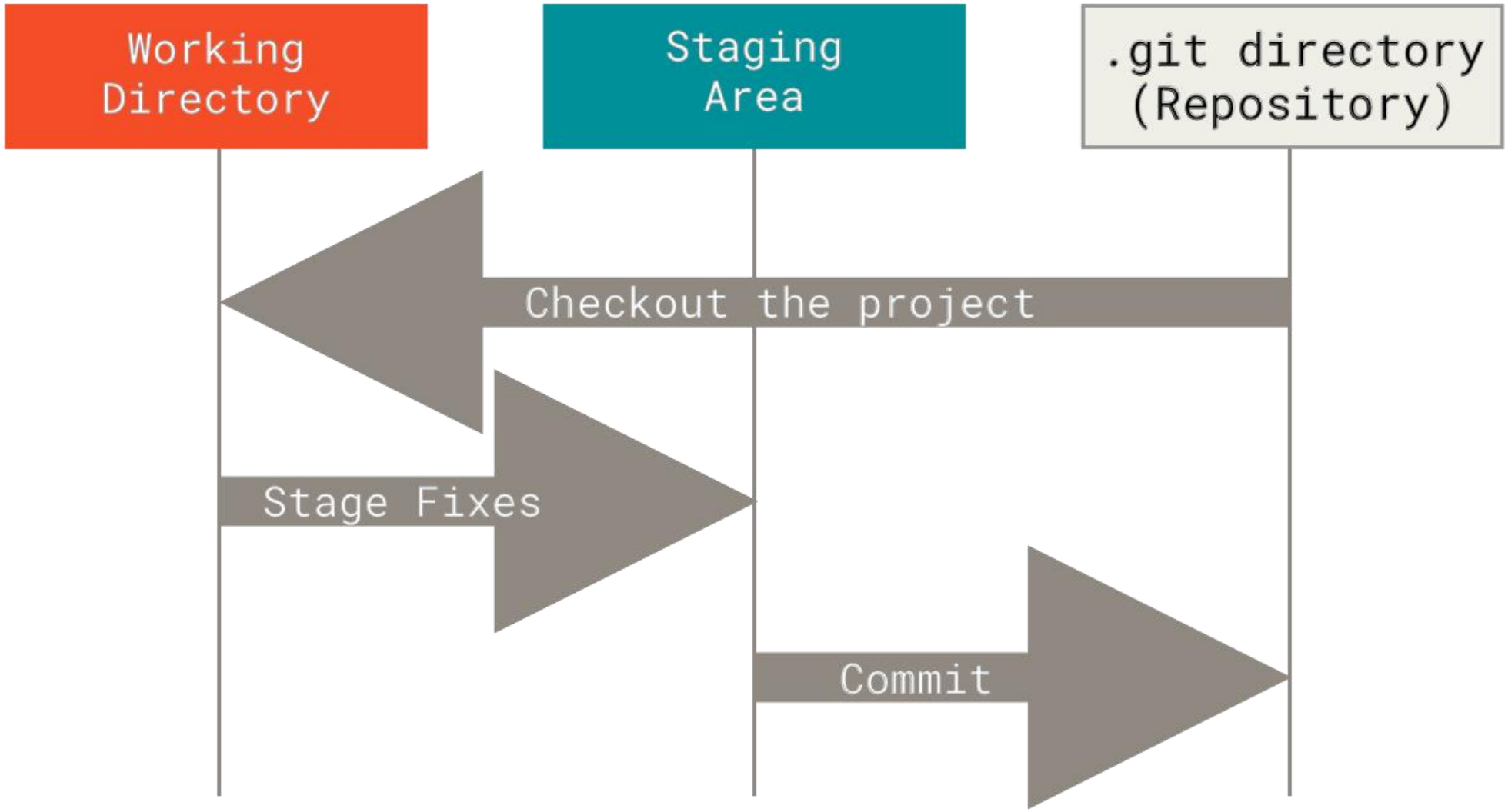
- **git ≠ GitHub ≈ GitLab ≈ Codeberg ≈ …**

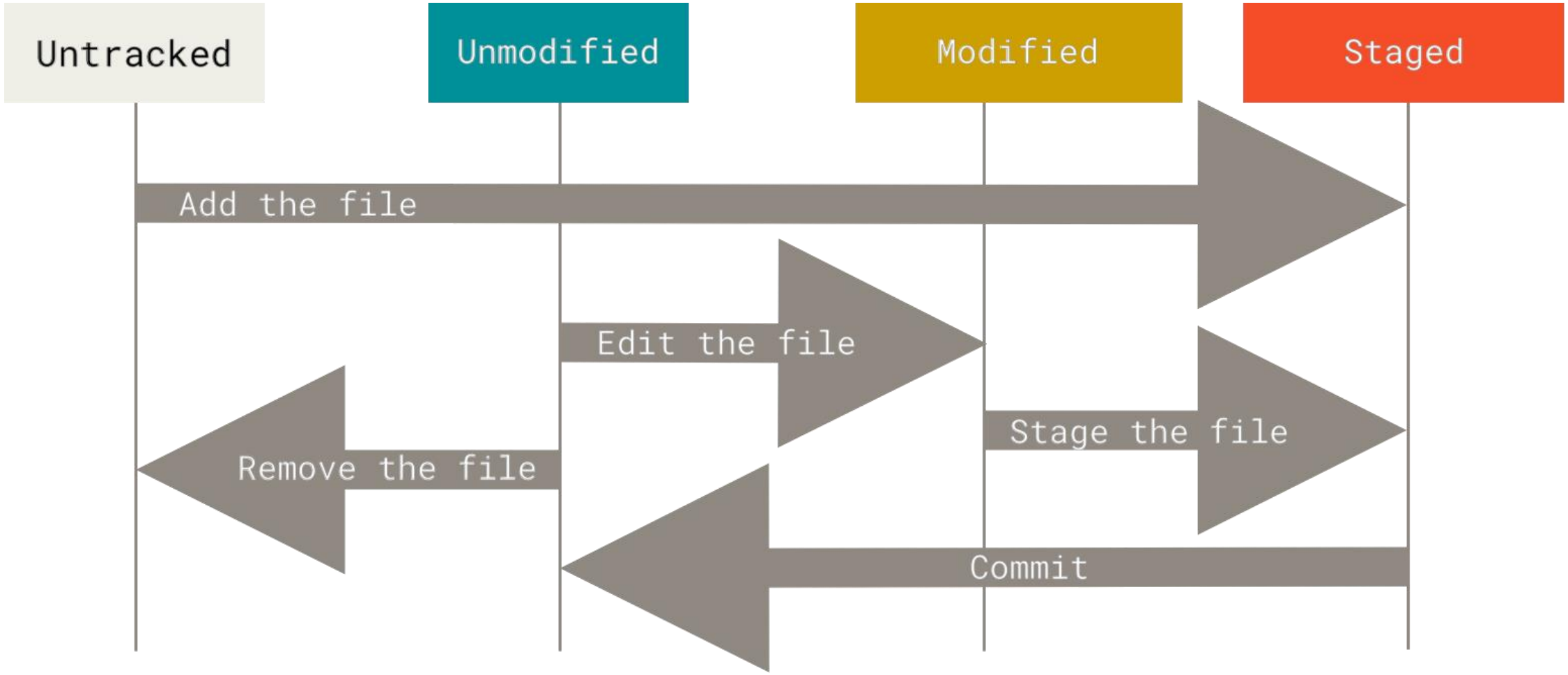
Locally track changes



Using GIT locally


HEAD
master
98ca9
34ac2
f30ab


Working Directory
Staging Area
.git directory (Repository)
Checkout the project
Stage Fixes
Commit

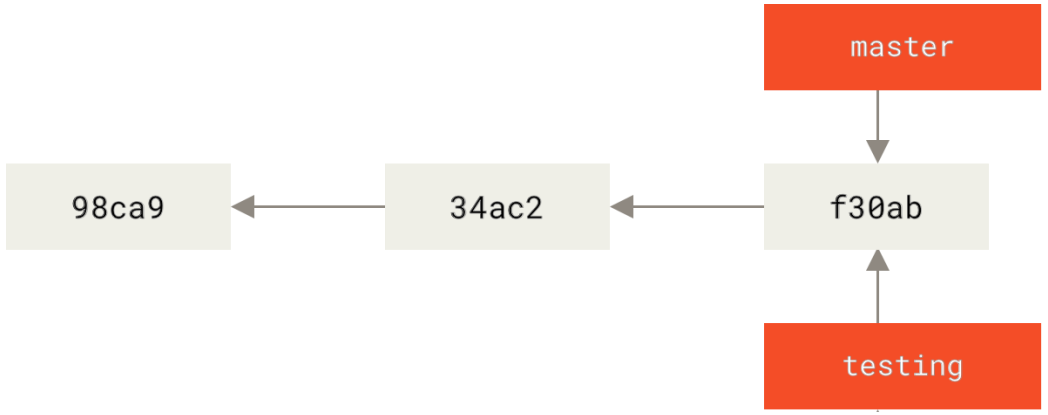

Untracked
Unmodified
Modified
Staged
Add the file
Edit the file
Stage the file
Remove the file
Commit

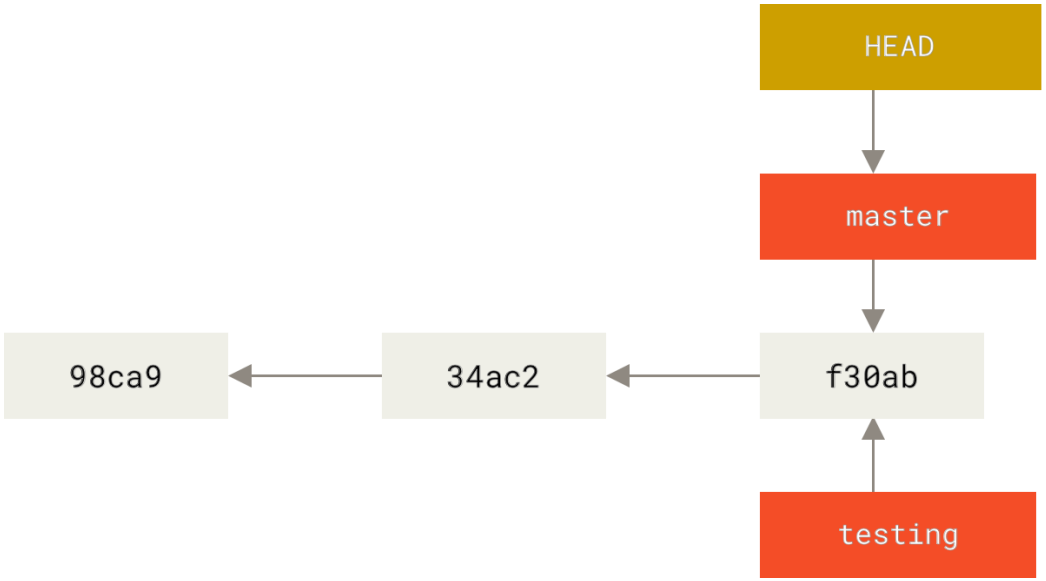
```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com

$ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"

$ git init <repository name>
$ git status
$ git add .

$ git diff

$ git log --all -graph

$ git commit -m "<message>"
$ git commit --amend

$ git reset <ID> --hard
$ git checkout <ID>

$ git tag -a v<version number> -m "<version description>"
```

Branching

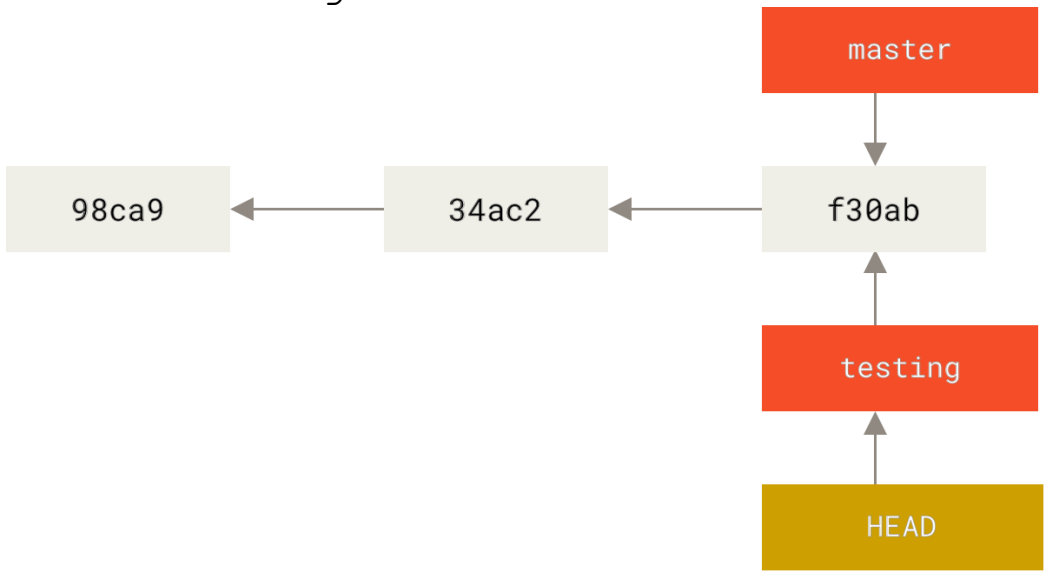

master
98ca9
34ac2
f30ab

```
$ git branch testing
```

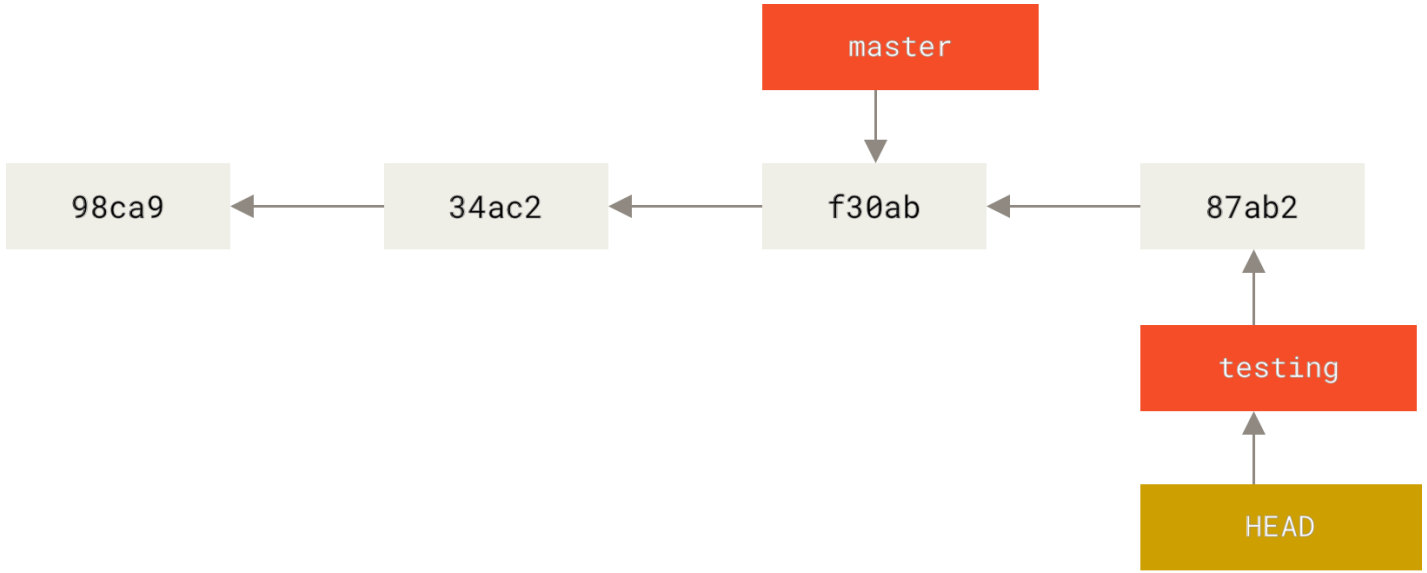


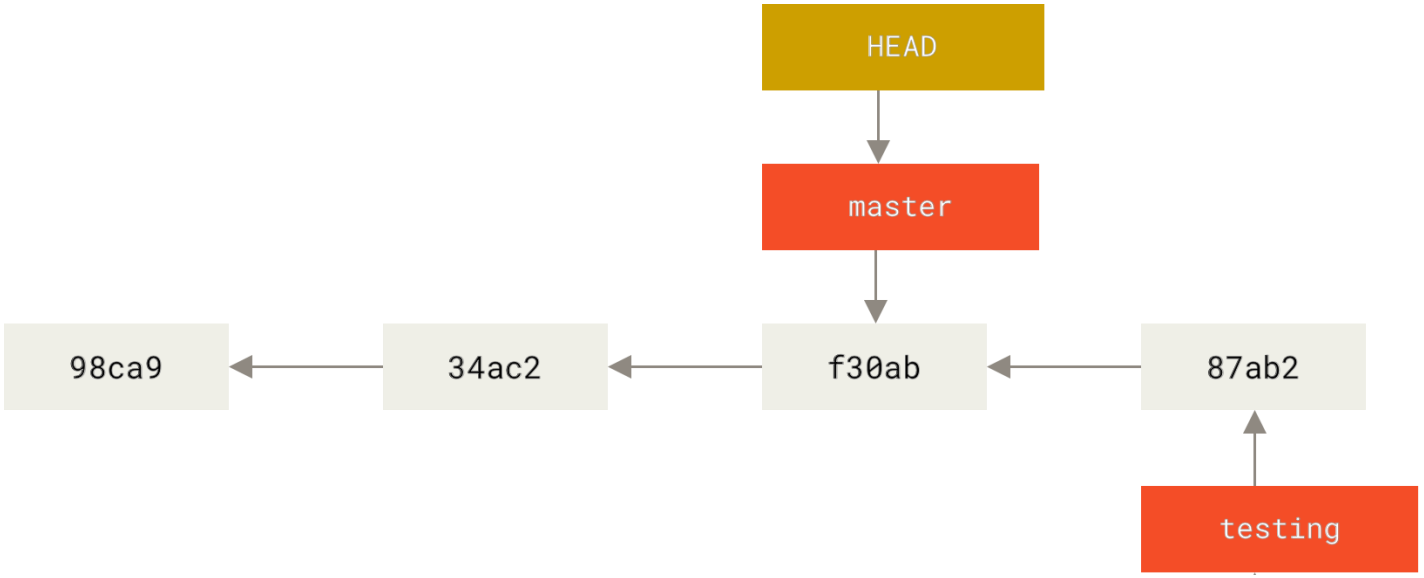

HEAD
master
98ca9
34ac2
f30ab
testing

```
$ git checkout testing
```
or
```
$ git switch testing
```



```
$ git commit -a -m 'Make a change'
```



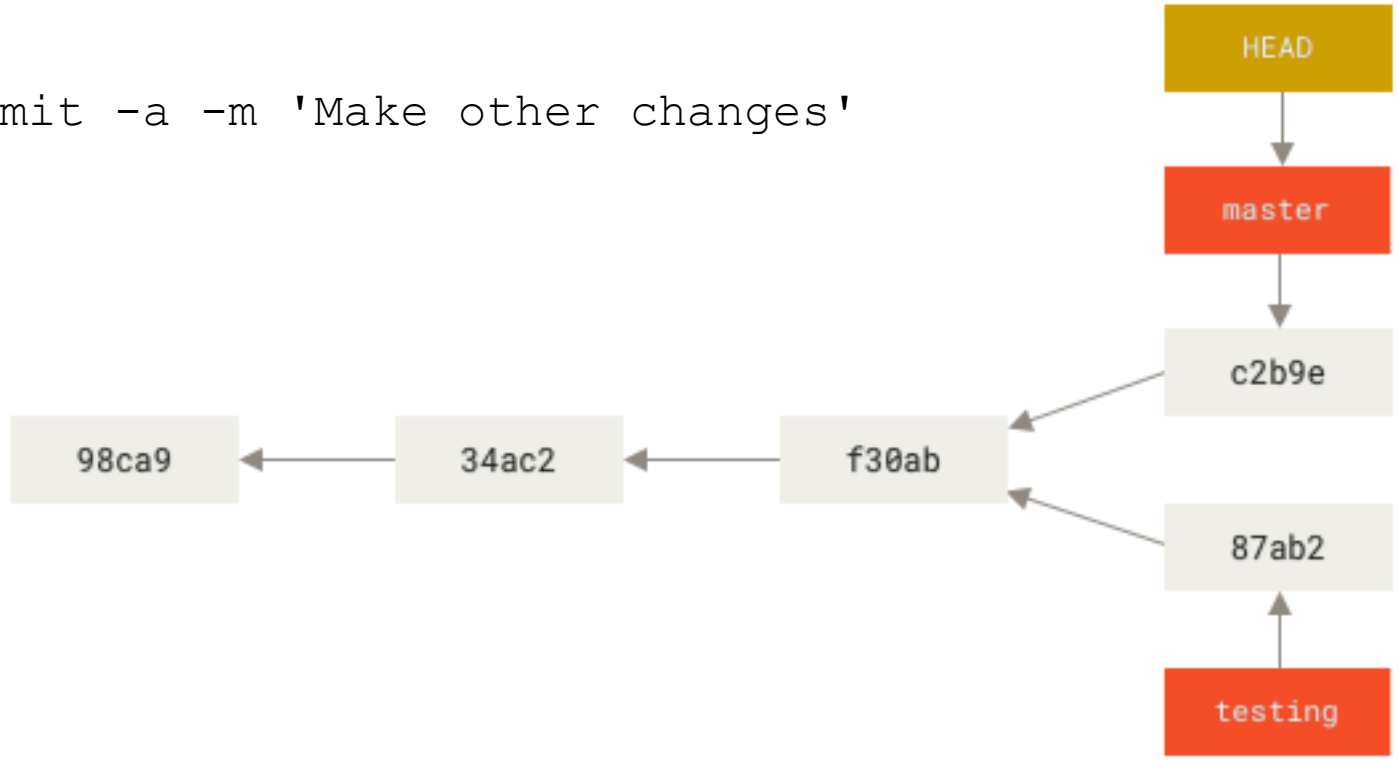


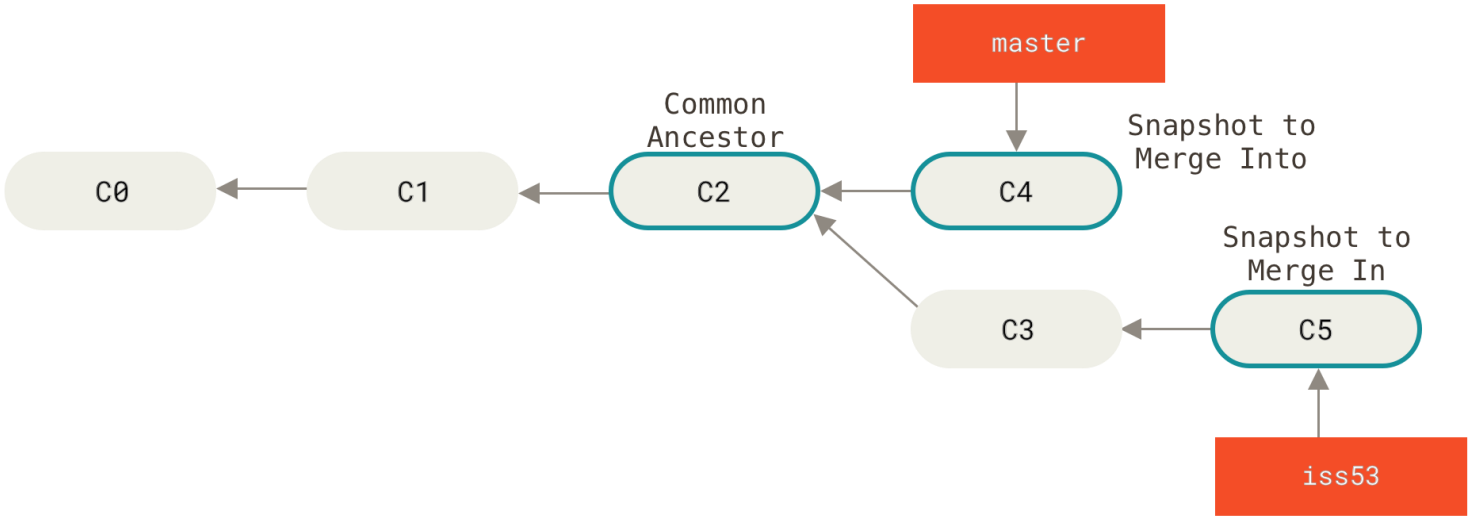

```
$ git checkout master
```

or

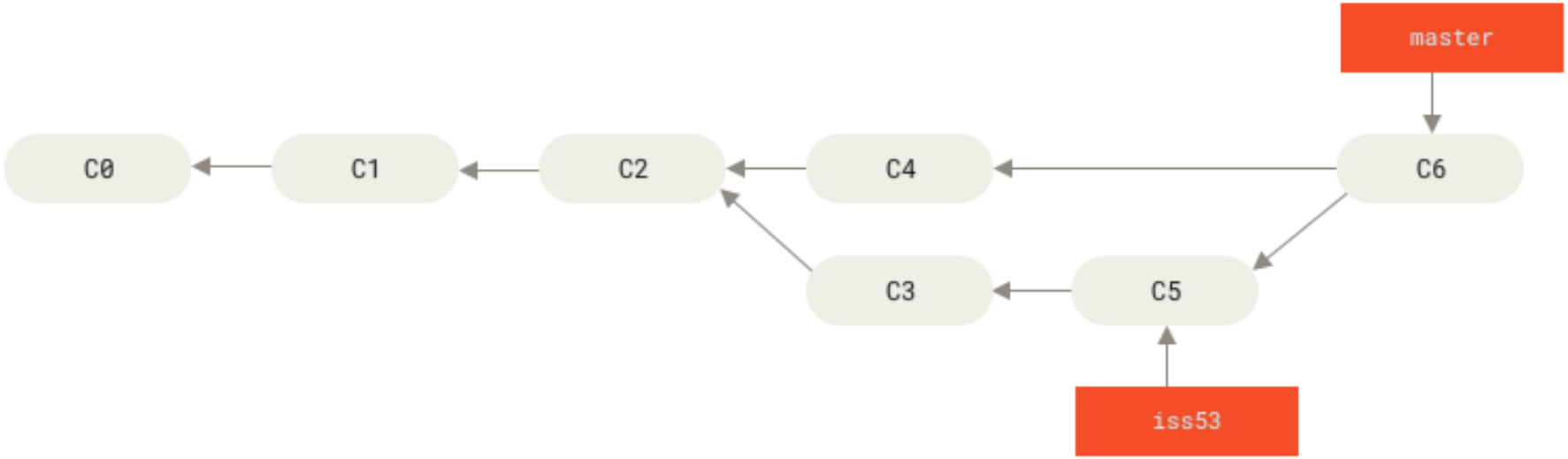
```
$ git switch master
```

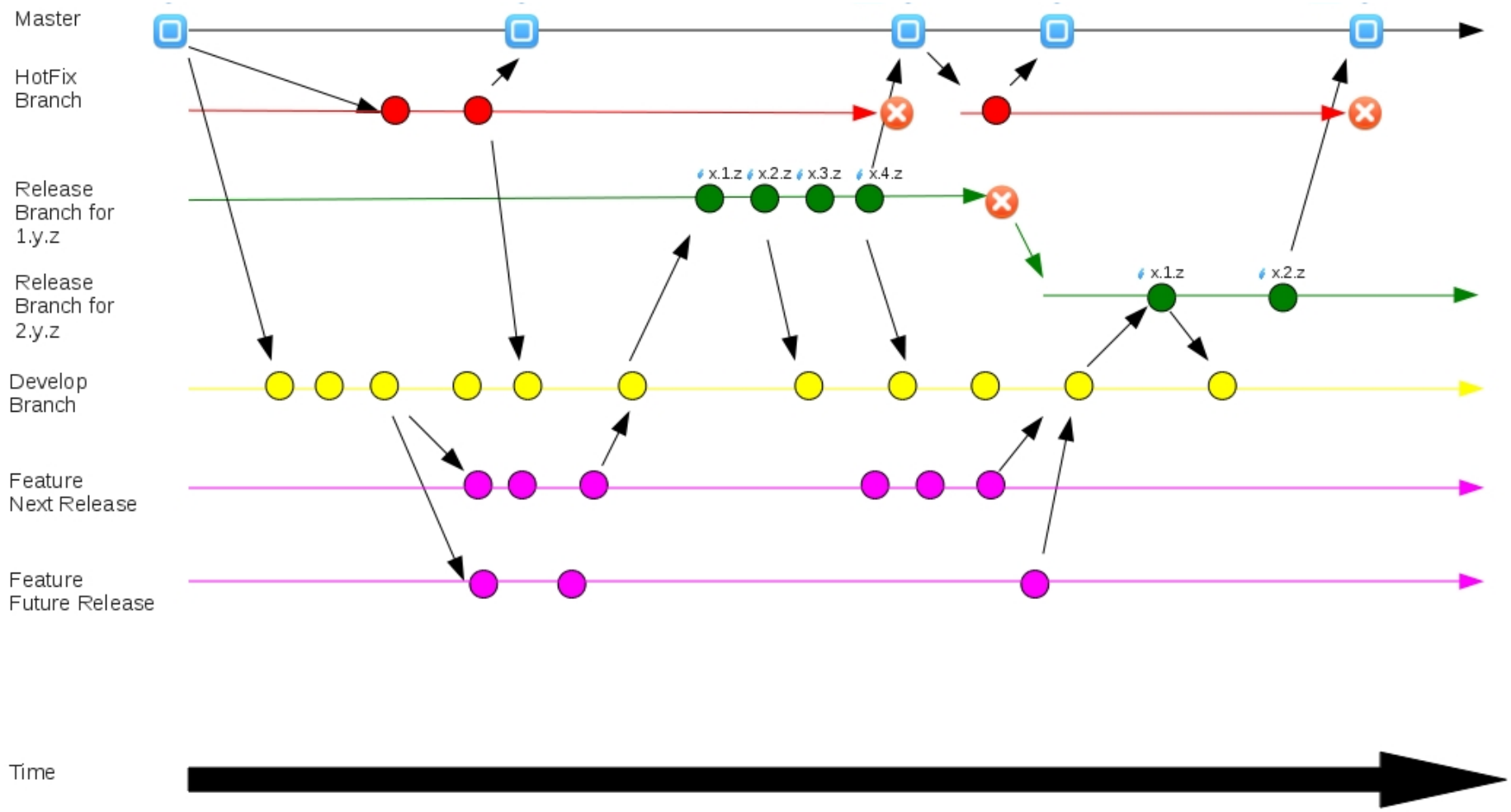
```
$ git commit -a -m 'Make other changes'
```



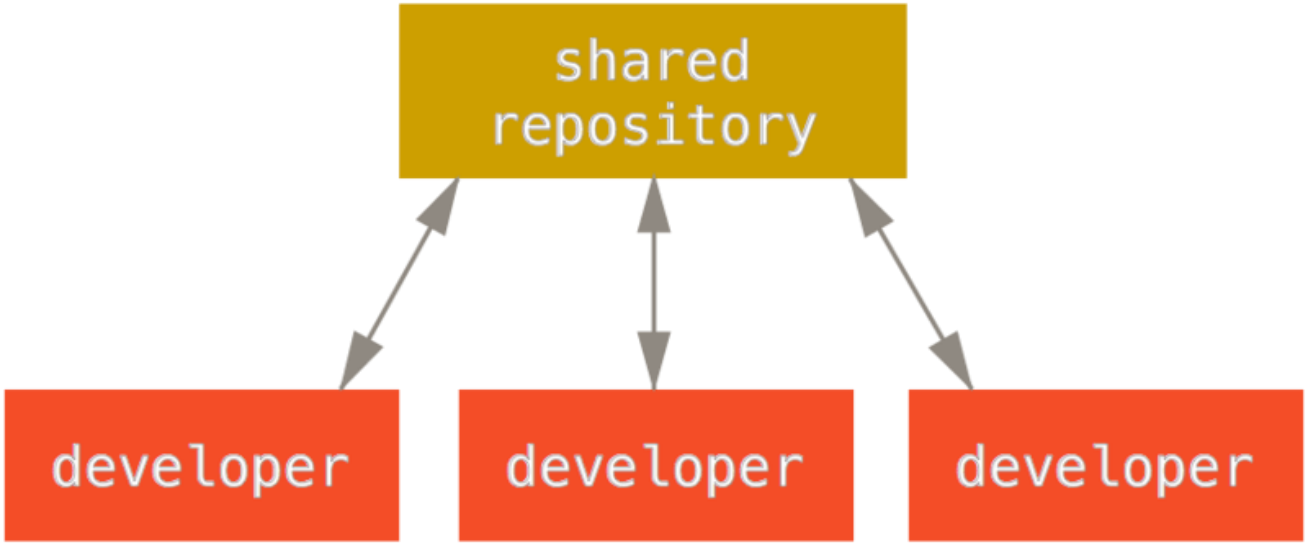

master
Common
Ancestor
Snapshot to
Merge Into
C0
C1
C2
C4
Snapshot to
Merge In
C3
C5
iss53

```
$ git merge iss53
```





GIT on the server


shared
repository
developer
developer
developer

Git based platforms

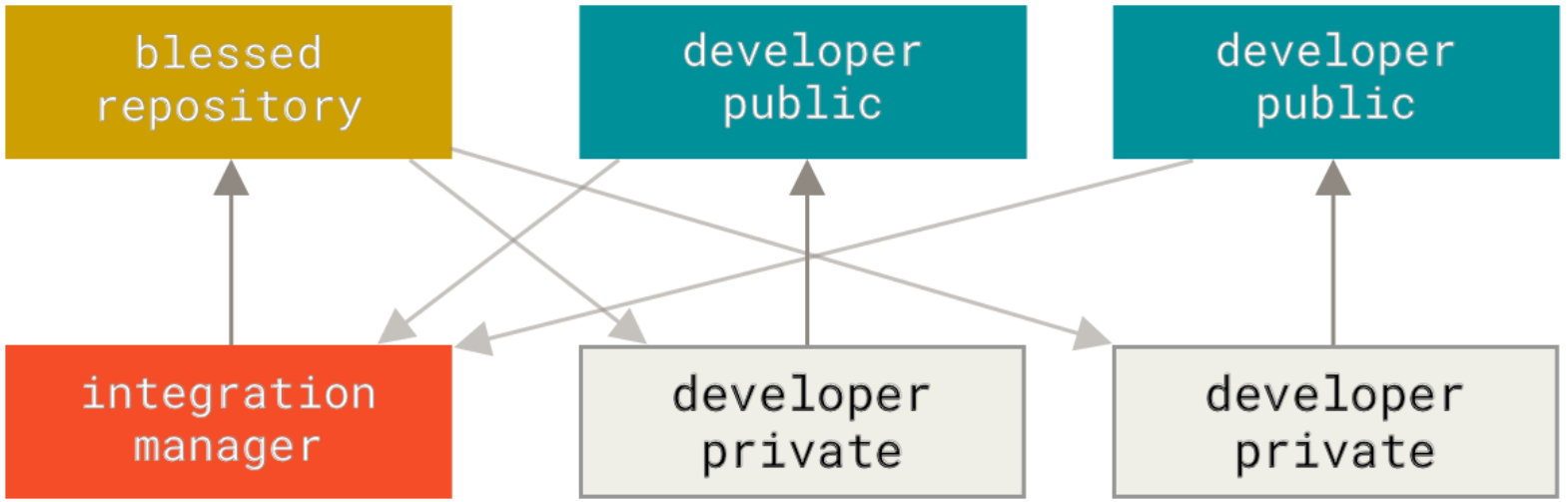
On the internet

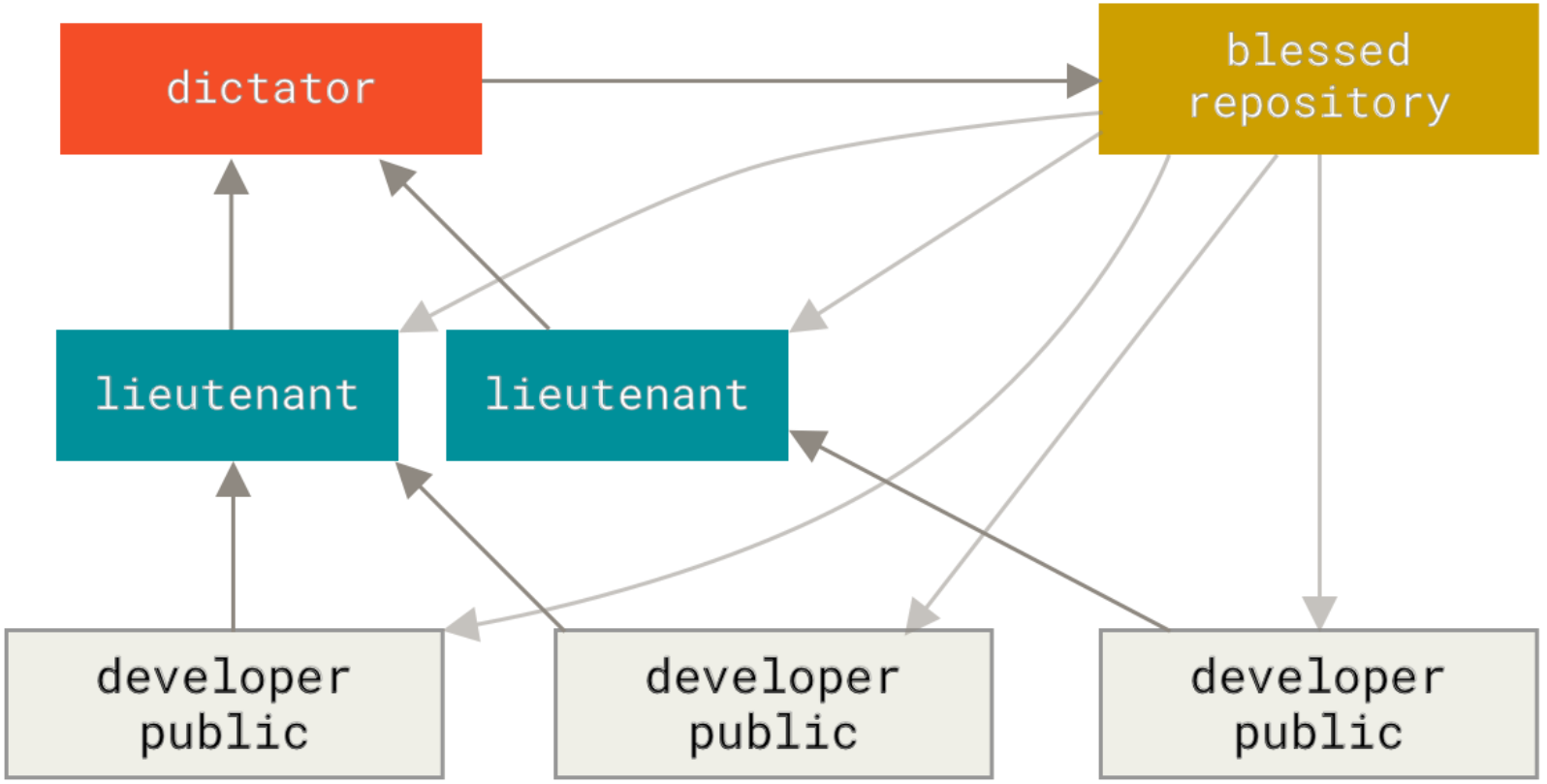
- Codeberg *https://codeberg.org/*

- GitLab *https://gitlab.com*
- GitHub *https://github.com/*
- Others…

PSI provides a GitLab instance

```
$ git branch –m master main

git remote add origin <remote URL>
$ git remote set-url origin <remote URL>

$ git remote show origin

$ git push origin <branch name>

$ git push origin <tag name>
```


blessed repository
developer public
developer public
integration manager
developer private
developer private


dictator
blessed repository
lieutenant
lieutenant
developer public
developer public
developer public

GIT resources

- Git website download page (Git, GUI clients) *https://git-scm.com/downloads*
- Git website documentation page (Reference manual, “Pro Git” book, videos) *https://git-scm.com/doc*
- GitHub Git cheat sheets *https://training.github.com/*