Basic Teaching Module

Version control basics - Introduction to GitHub

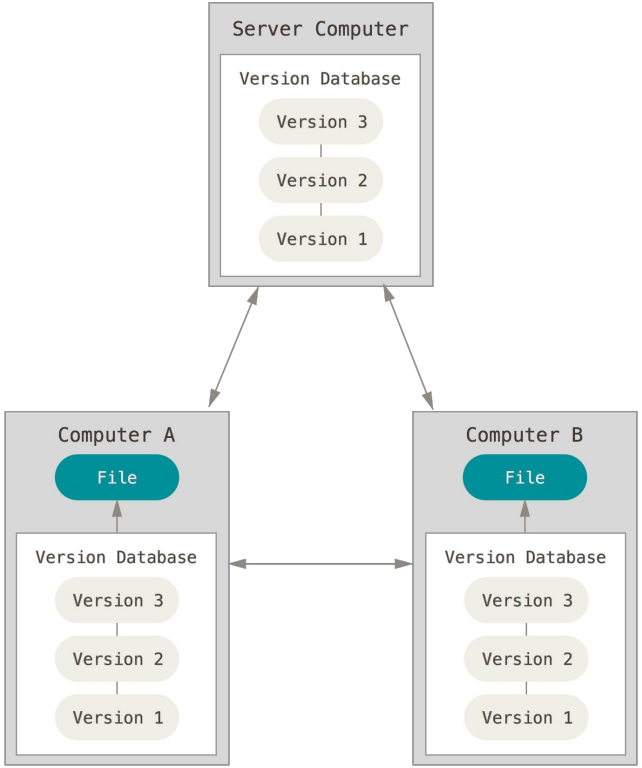
Sérgio Santos
28 October 2015

Version control

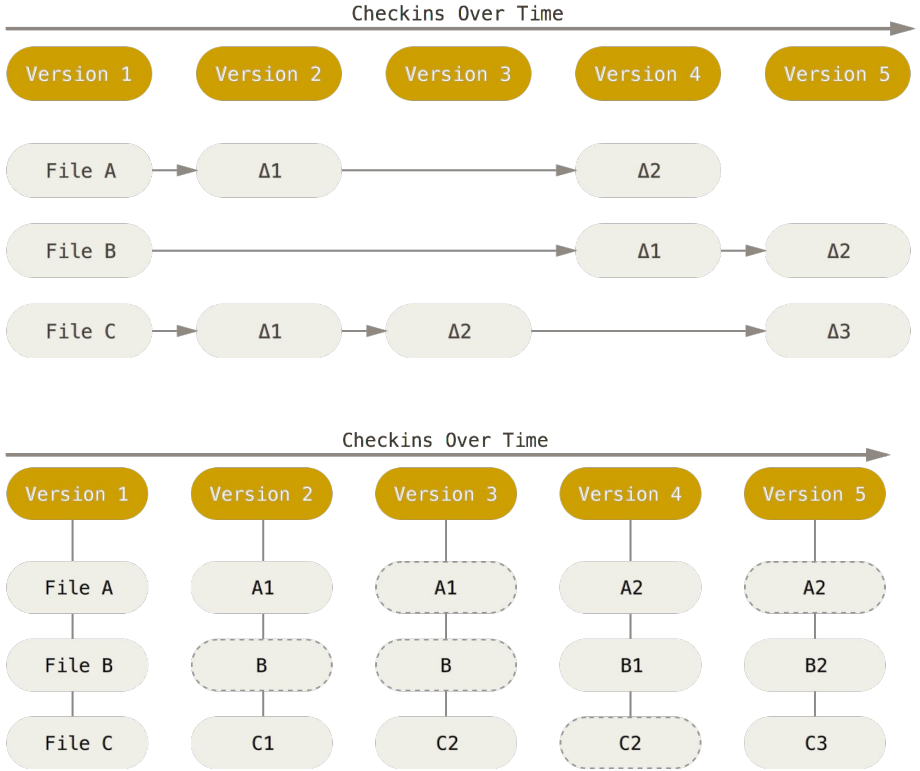
- A system that records changes to a file or set of files over time so that you can recall specific versions later

Distributed version control system

- Clients don’t just check out the latest snapshot of the files, they fully mirror the repository
- If any server dies, any of the client repositories can be copied back up to the server to restore it
- Every clone is really a full backup of all the data



Snapshots

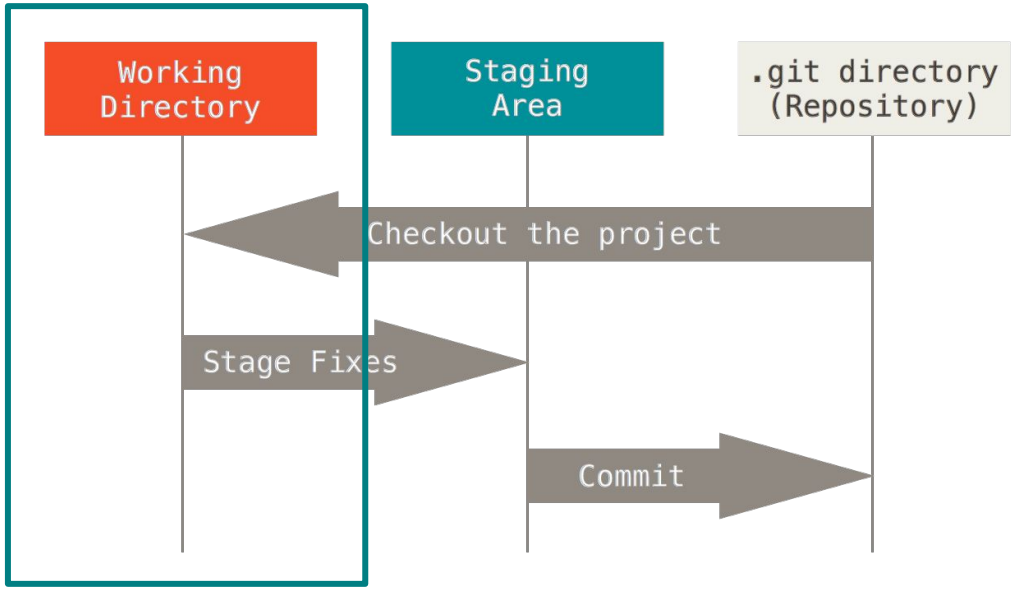

Checkins Over Time
Version 1
Version 2
Version 3
Version 4
Version 5
File A
Δ1
Δ2
File B
Δ1
Δ2
File C
Δ1
Δ2
Δ3
Checkins Over Time
Version 1
Version 2
Version 3
Version 4
Version 5
File A
A1
A1
A2
A2
File B
B
B
B1
B2
File C
C1
C2
C2
C3

Collaborate with git

- Several remote repositories
- Collaborate with different groups of people in different ways within the same project
- Set up several types of workflows that aren’t possible in centralized systems
- It’s fast, has simple design and is able to handle efficiently large projects like the Linux kernel
- Strong support for non-linear development (thousands of parallel branches)

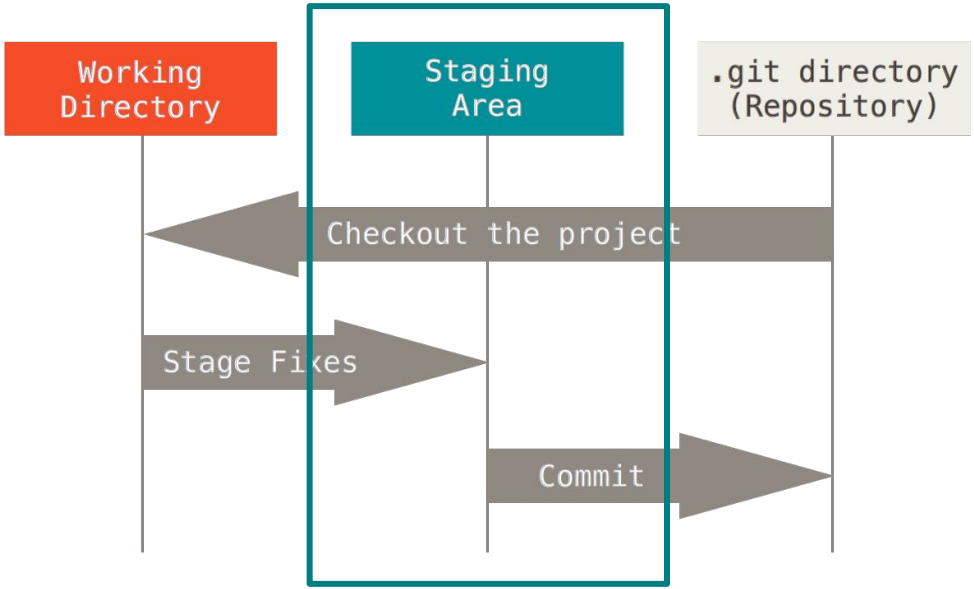
Working directory

- You modify files in your working directory



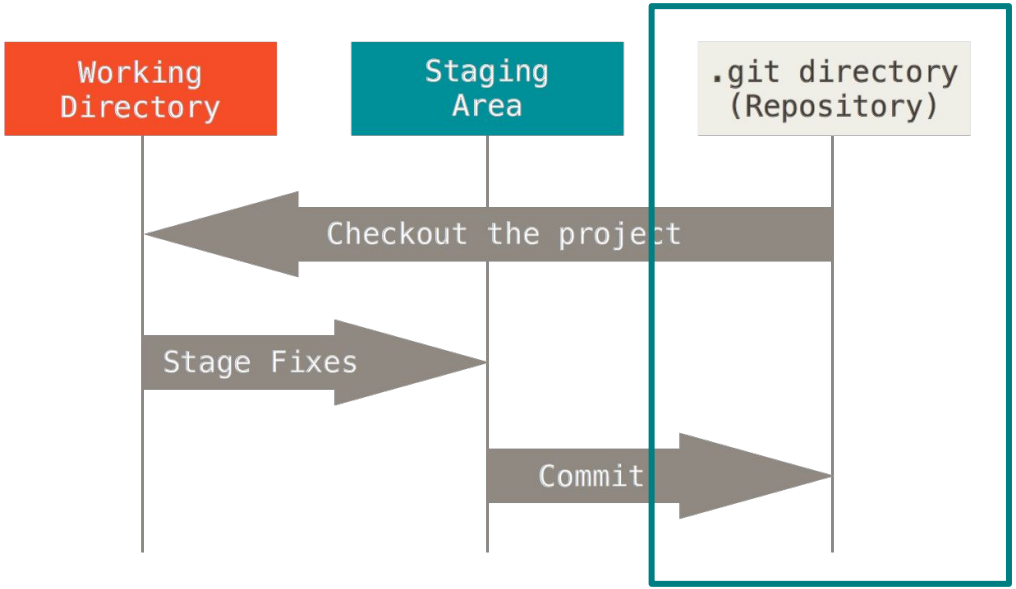
Staging area

- You stage the files, adding snapshots of them to your staging area



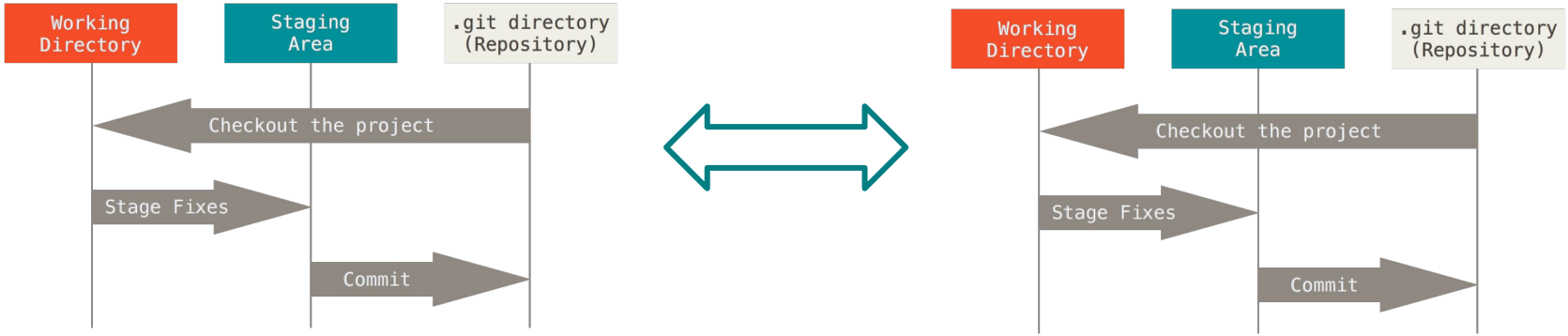
Git repository

- You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory



Add it to a remote repository

- You do a git push



Your computer

GitHub

Install git

Instructions

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Linux

$ sudo apt-get install git

Mac

http://git-scm.com/download/mac

Start using git

Create a directory and download the presentation

$ git init

$ git status

$ git pull https://github.com/sergiomcmsantos/btm

- Create a GitHub account https://github.com/
- Create a repository
- Download the repository to another directory

  $ git pull https://github.com/*repository-name*

Git config

Set your name on the configuration file

$ git config --global user.name "*My name*"

$ git config --global user.email "*email*"

Verify the setting

$ git config user.name

Change to your favorite text editor

$ git config --global core.editor *vim*

Create and add file to repository

Create a file

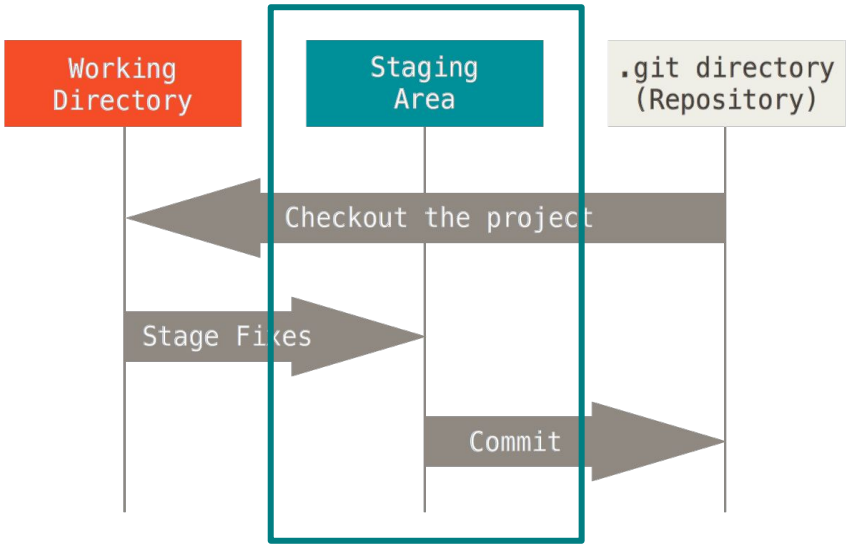
```
$ touch file.txt
```

```
$ git status
```

Add the file to the repository

```
$ git add file.txt
```

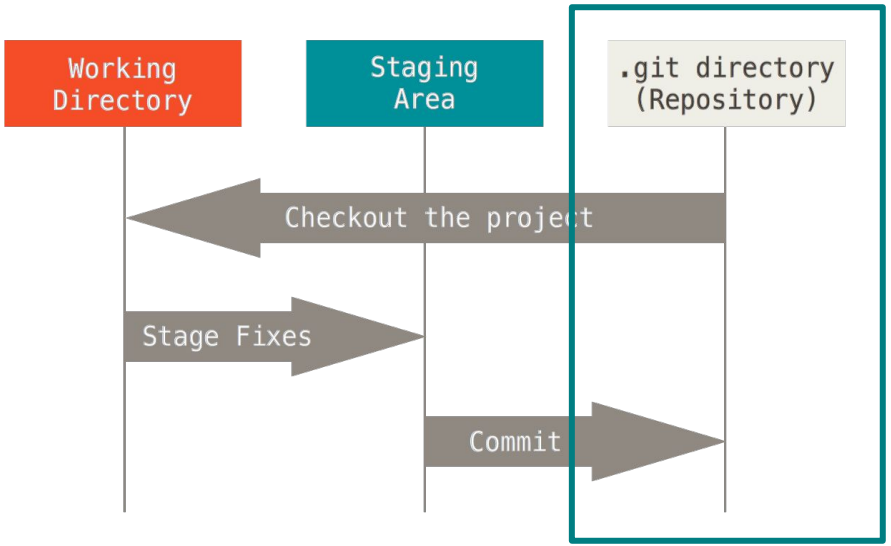
```
$ git status
```



Commit changes

Commit changes

$ git commit -m "My first file”



Change file and add it to the repository

Change file, add and commit

```
$ echo first change > file.txt

$ git status

$ git add file.txt

$ git status

$ git commit -m "First change”

$ git log
```

Push it to GitHub

Push it to GitHub

```
$ git push
```

On GitHub:

- Find and navigate the history of the file
- Change file using the web interface

Pull the changes made to your computer

```
$ git pull
```

Why you might need branches

On GitHub:

- Make a change to the file using the web interface

On your computer, change the file and try to push it

```
$ echo a_different_change >> file.txt
$ git add file.txt
$ git commit
$ git push    # What happened?
```

Create a branch

Check the branches you have

```
$ git branch
```

Create a new branch

```
$ git branch new-branch
```

```
$ git branch
```

Change to the new branch

```
$ git checkout new-branch
```

Change file in the new branch

Change file, add, commit and push it

```
$ echo new_branch_content > file.txt
```

Push changes in all branches

```
$ git push --all -u
```

- Change between branches and look at the content of the file
- Create a different file on the branch and merge the branch using “git merge”

Still have time?

Navigate the history using the terminal

$ git log

$ git checkout *commit*

Explore git

https://git-scm.com/book/en/v2