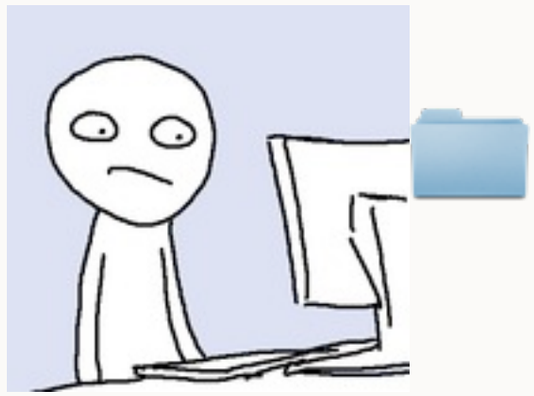
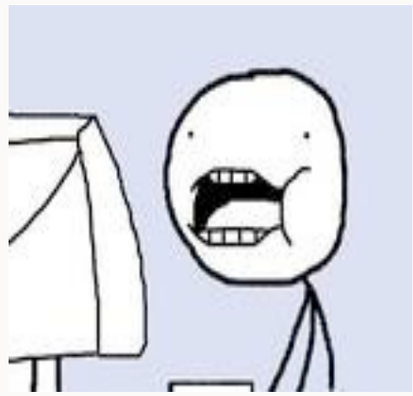
GIT

Luís Ferreira
zamith@groupbuddies.com

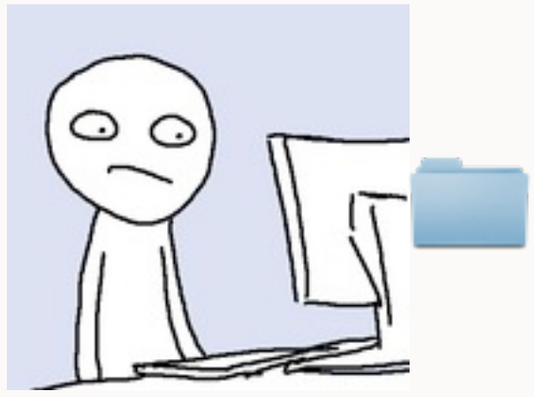
WORKING IN TEAMS

Project Sharing

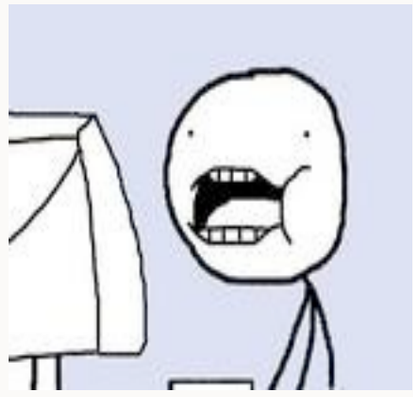




Dropboxing



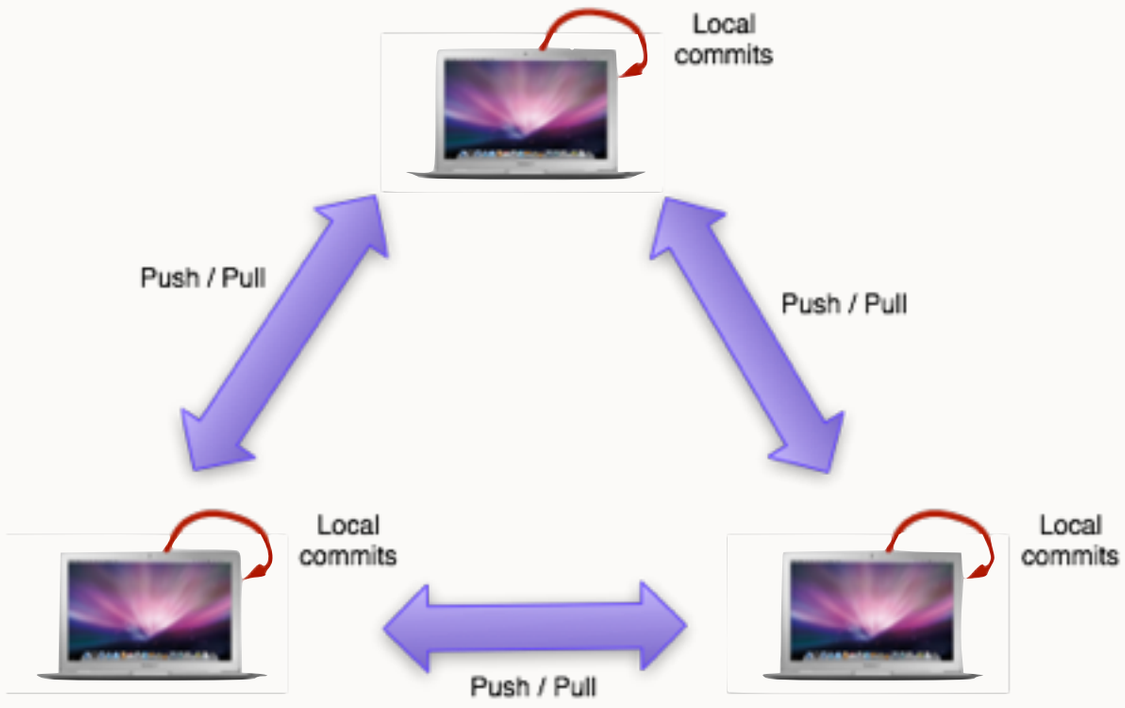
Syncing...



VERSION CONTROL (SCM)

GIT IS DISTRIBUTED

- Every user has a copy of the repository
- Does local commits to his copy
- Can then pull changes from others and merge them locally
- Can push his changes to others



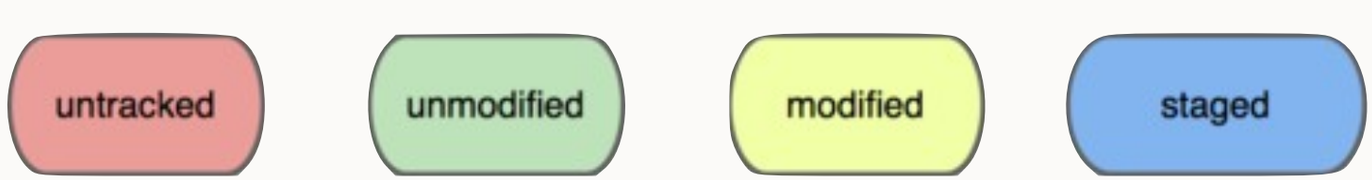
MY FIRST REPO

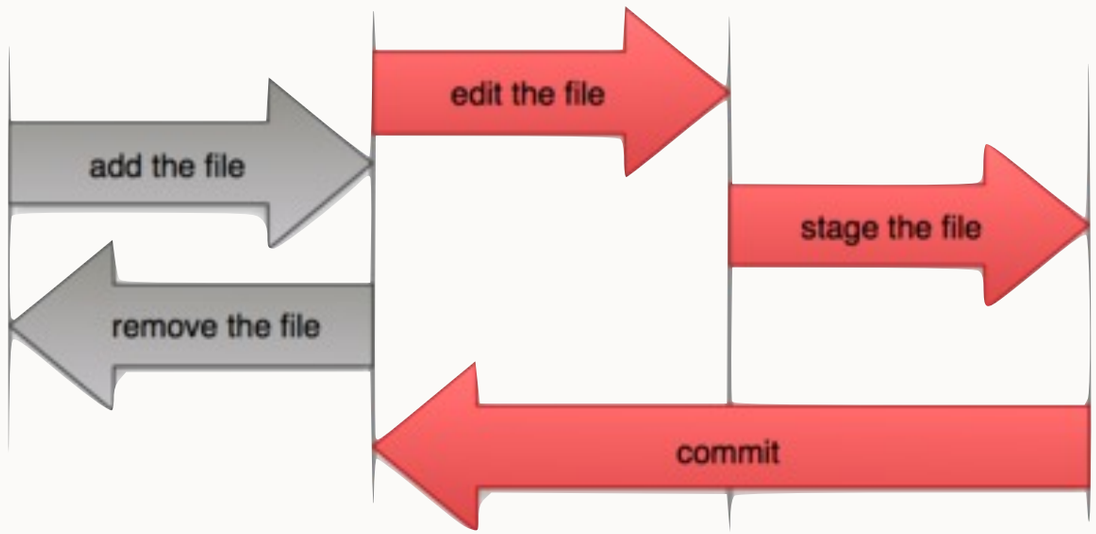
Creating a project with git is as easy as:

```
$ cd to/project/folder
$ git init

project related stuff...

$ git add .
$ git commit . -m ‘My first commit’
```

FILE STATUS LIFECYCLE

- Files in a git repo are constantly changing between these states
- You can check them with *git status*





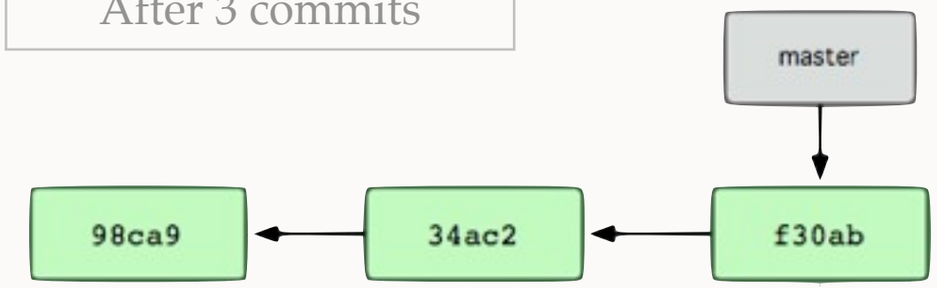
source: http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository

BRANCHES

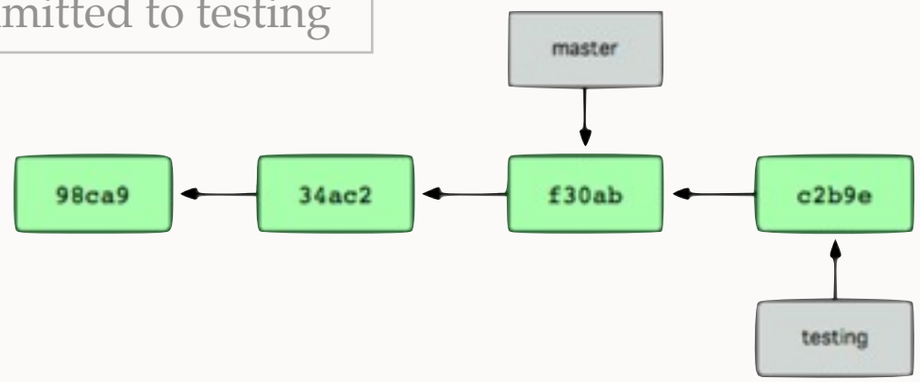
- Allow you follow different paths when developing
- Branches are cheap (41 bytes)
- By default you are in the **master** branch
- Create a branch with *git branch <name>*
- Change between branches with *git checkout <name>*

BRANCHING OUT

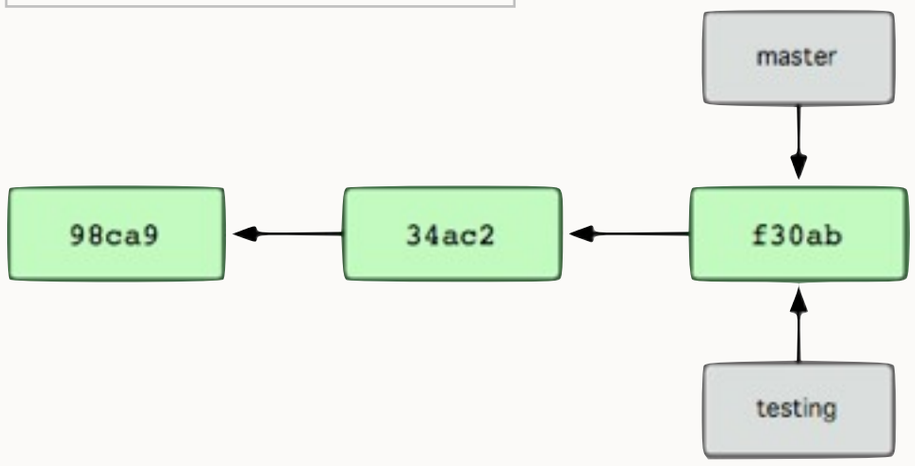
After 3 commits



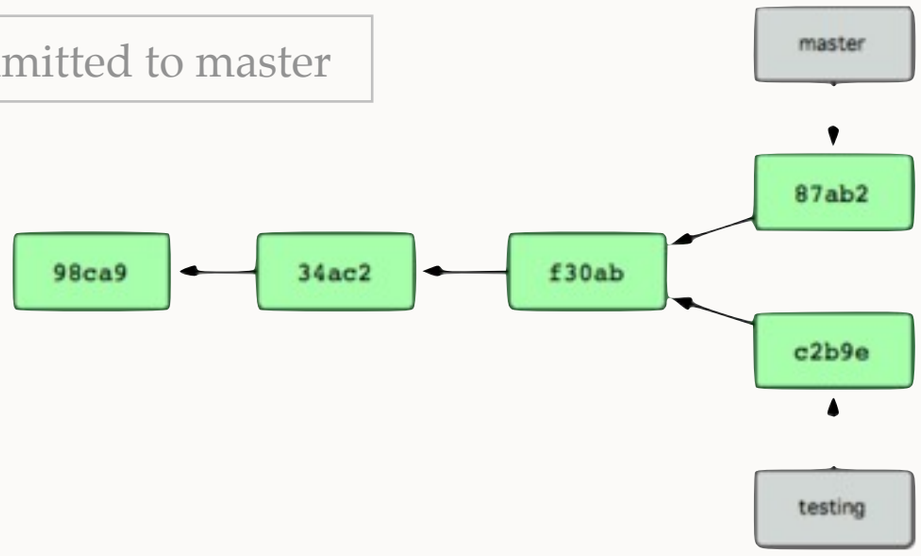
Committed to testing



Created branch testing



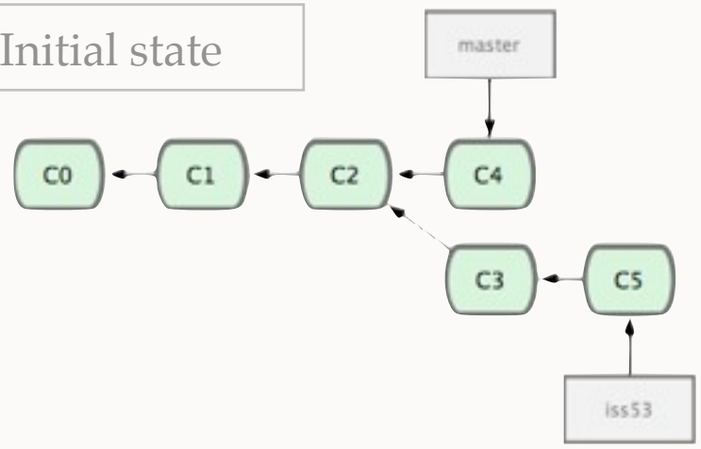
Committed to master



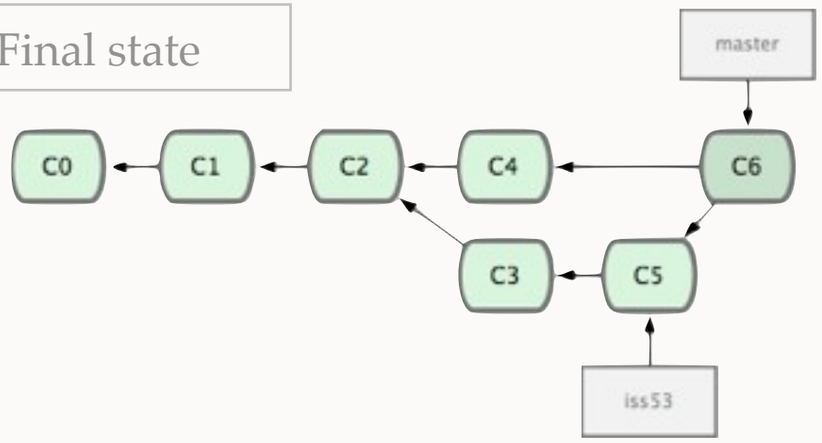
source: http://git-scm.com/book/en/Git-Branching-What-a-Branch-Is

MERGING

Initial state



Final state



- Merge done by *git merge iss53*
- After merge you can simply *git branch -d iss53*
- Conflicts may occur and they look like this:

```
<<<<<<< HEAD:index.html
<div id="footer">contact : contact@groupbuddies.com</div>
=======
<div id="footer">
  please contact us at contact@groupbuddies.com
</div>
>>>>>>> iss53:index.html
```

A BIT MORE USEFUL THINGS

- Use *git diff* to see what has changes between commits
- Add a *.gitignore* file where each line represents one or more files or directories for git not to track
- Untrack files with *git rm*
- View the commit log with a visual tool such as **gitk**, **GitX** or **GitHub for Mac** (alternatively use *git log*)

GITHUB

- Social network for your code
- Free account for as many public repos as you want
- Nice interface for many git related tasks
- Great graphs to analyze what is happening in the project
- Great exposure for the your projects

COOPERATION

- Sharing your commits is really easy
- Pull before you push!
- Resolve conflicts locally

Sync with remote repo

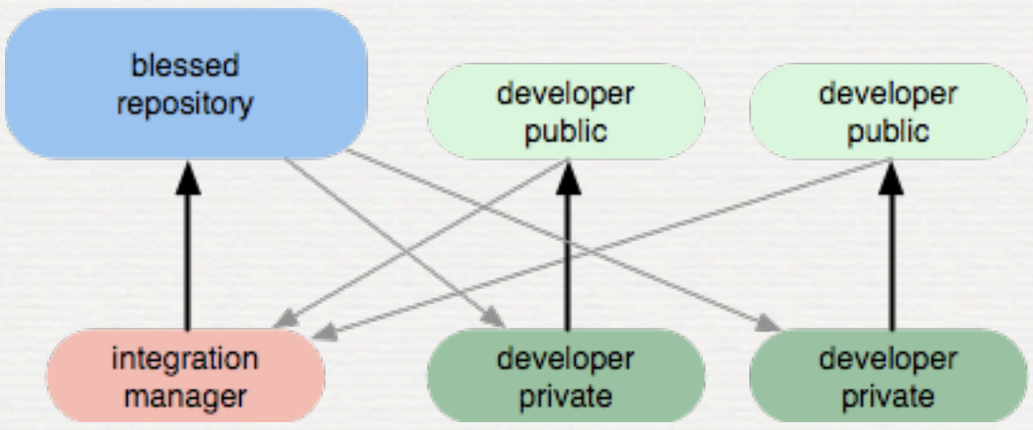
```
$ git add remote origin git@github.com:zamith/backbone-rails.git
$ git pull [origin master]
$ git push [origin master]
```

Resolve conflicts

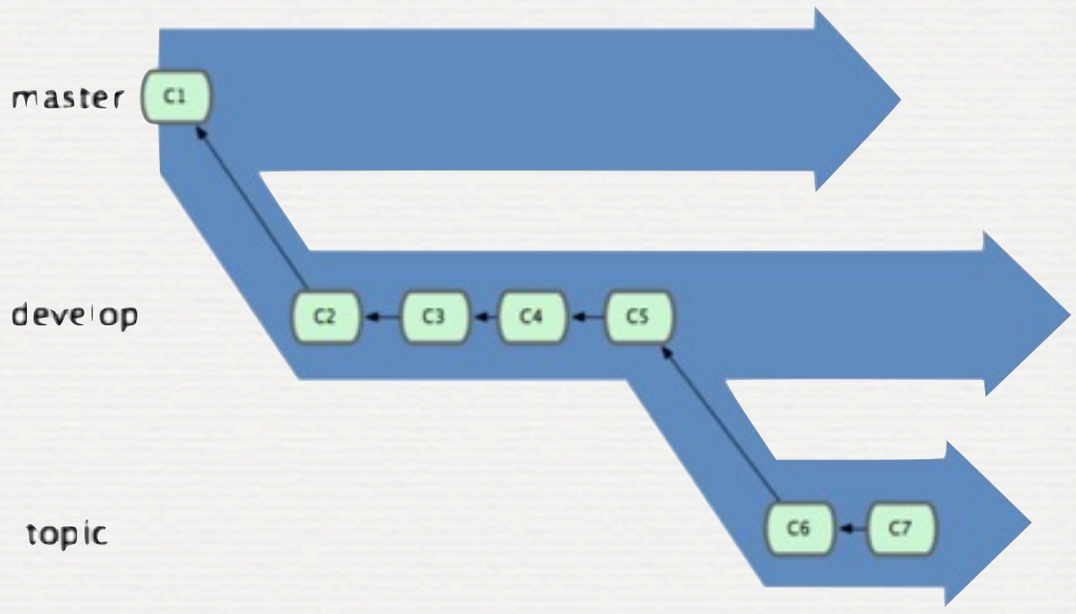
```
$ git pull [origin master]
... conflicts in some files ...
... resolve them ...
$ git add .
$ git commit
$ git push [origin master]
```

GETTING A WORKFLOW

- Github introduces an easy way to do an **Integration Manager Workflow**
- Don't break stuff, use branches



source: http://git-scm.com/about/distributed



source: http://git-scm.com/book/en/Git-Branching-Branching-Workflows

HELPING THE COMMUNITY

- Fork a project
- Get it to your workstation through *git clone*
- Commit locally
- Pull request

TL; DR

```
$ mkdir Project
$ cd Project
$ git init
$ touch README
$ git add README
$ git commit -m ‘first commit’
$ git remote add origin git@github.com:zamith/Project.git
$ git push -u origin master
```

And you’re done!

http://help.github.com/

QUESTIONS?

Visit *git-scm.com* for more info