Introduction to Git and Github Repositories

Benjamin Audren

École Polytechnique Fédérale de Lausanne

29/10/2014

Version Control

survey

Survey

- Who uses it daily/weekly?

Version Control

survey

Survey

- Who uses it daily/weekly?
- svn, git, mercurial?

Version Control

Goals

Benefits

- keep track of **modifications** in case of bugs
- **clear history** , to roll back to an old but stable version
- allowing **collaboration**
- backing up a project

Two paradigms

Centralised Version Control

- history lives **only in one place**.
- **need server communication** to commit a change.
- **branching gives headache**.

Two paradigms

Centralised Version Control

- history lives **only in one place**.
- **need server communication** to commit a change.
- **branching gives headache**.

Distributed Version Control

- **virus-like way** of storing the history.
- **no internet required** to work.
- **powerful branch** system.
- **problems may appear** when editing the same file.

Git

Characteristics

- **Distributed** version control, with **remote repository**.
- Used for the development of the **Linux kernel**.
- Has a huge community, with **Github**.
- Cool **features to share the code** online.

Installation

Unix

```
sudo apt-get install git
```

Mac

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

or with Brew/Mac Ports:

```
sudo port install git-core +svn +doc
```

Windows

Install a VirtualBox with Ubuntu, please!

Basic Usage

Exercice

Example

- `mkdir example`, `cd example`

Basic Usage

Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.

Basic Usage

Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.
- `touch README.md`, then edit it.

Basic Usage
Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.
- `touch README.md`, then edit it.
- `git status` check the status of the folder.

Basic Usage
Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.
- `touch README.md`, then edit it.
- `git status` check the status of the folder.
- `git add README.md` put in the staging area.

Basic Usage
Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.
- `touch README.md`, then edit it.
- `git status` check the status of the folder.
- `git add README.md` put in the staging area.
- `git commit -m 'first commit'`

Basic Usage

Exercice

Example

- `mkdir example`, `cd example`
- `git init` This creates an empty git repo.
- `touch README.md`, then edit it.
- `git status` check the status of the folder.
- `git add README.md` put in the staging area.
- `git commit -m 'first commit'`
- `git log --graph` or use `gitk`

Staging Area

Concept

- **local modifications** to be commited to the history: **HEAD**

Staging Area

Concept

- **local modifications** to be commited to the history: **HEAD**
- It can be **several files** that constitutes more than one logical commit!

Staging Area

Concept

- **local modifications** to be commited to the history: **HEAD**
- It can be **several files** that constitutes more than one logical commit!
- `git add` selects which files to commit, and put them in the staging area.

Staging Area

Concept

- **local modifications** to be commited to the history: **HEAD**
- It can be **several files** that constitutes more than one logical commit!
- `git add` selects which files to commit, and put them in the staging area.
- `git commit -m 'first commit'`

Branching

Easy as pie

- `git branch` list all existing branches (master)
- `git branch modif` create the branch **modif**
- `git checkout modif` switch to the branch **modif**
- `touch toto.py`, edit the file
- `git add, commit`
- `git log --graph modif`
- `git checkout master`
- `git merge modif --no-ff`

Branching
Exercise

Different merging strategies

- What did the `--no-ff` do?
- What happens if you omit it?

Remote Repositories

Adding a remote from an existing repo

- `git remote add origin https://github.com/you/yourcode.git`
- `git pull origin master`
- `git push origin master`

Remote Repositories

Adding a remote from an existing repo

- `git remote add origin https://github.com/you/yourcode.git`
- `git pull origin master`
- `git push origin master`

Cloning an existing repo

- `git clone https://github.com/lesgourg/class_public.git`
- `cd class_public`
- touch, add, commit
- **Warning:** you can not push this!

Github

Most important for end-users

The `https://github.com/baudren/montepython_public/issues` page

Forking on Github

Forking

- To **clone a public repository and modify it**.
- You have the rights to commit changes to your fork.
- Send a **Pull-Request** to us via Github.
- Note that your fork will be **public**.
- You can request an academic account
  `https://github.com/blog/1840-improving-github-for-science`

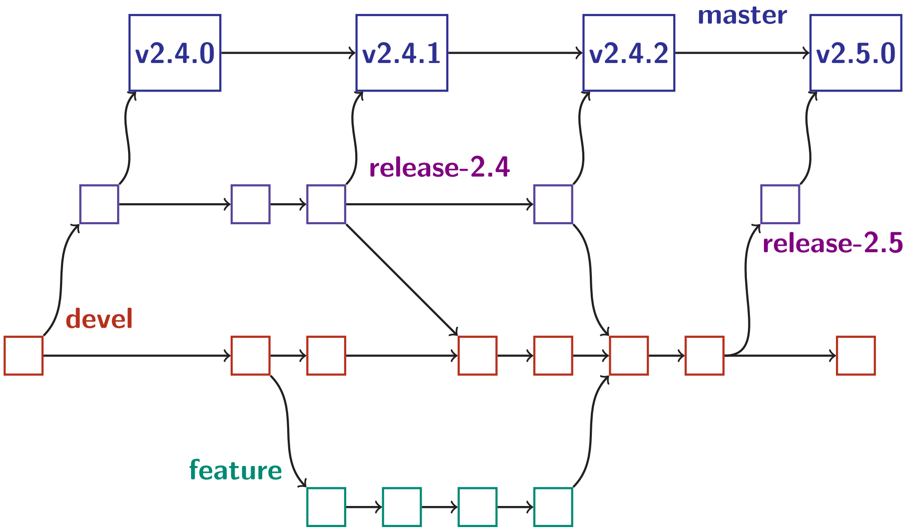
For reference

Branching model

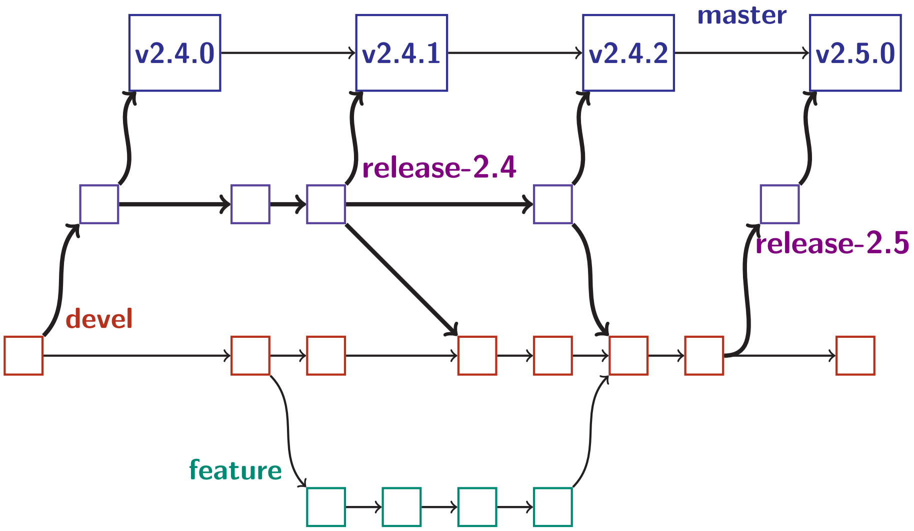
Essentially follows

http://nvie.com/posts/a-successful-git-branching-model/

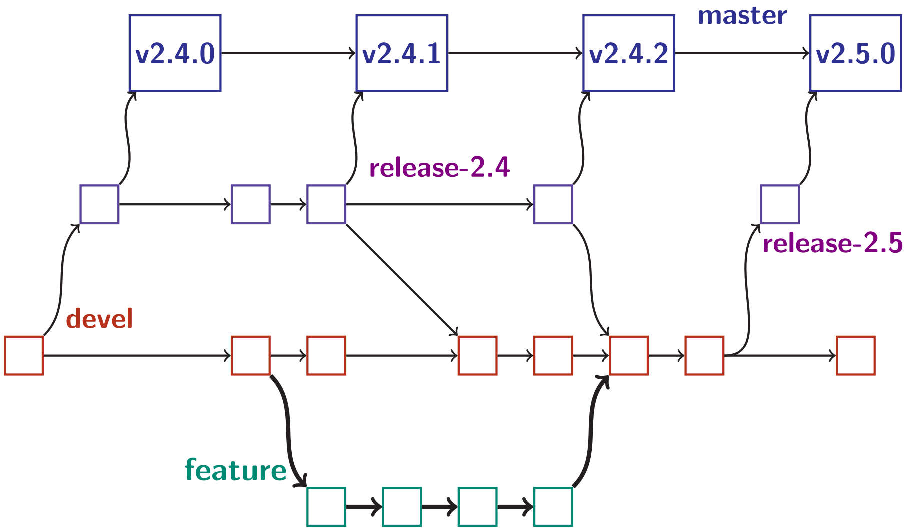
For reference



For reference



For reference



Modifying locally a public repository

Exercise I

- clone class
- copy over your modified files from (η_b or σ_8)
- `git status`
- `git add`, `commit`

Exercise II

Fork the repository to commit your own modifications
read https://github.com/lesgourg/class_public/wiki/Public-Contributing

Exercise III

Open an issue to request a missing feature
or declare a bug