Projecte d'Enginyeria del Software: Git and GitHub

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

Agenda

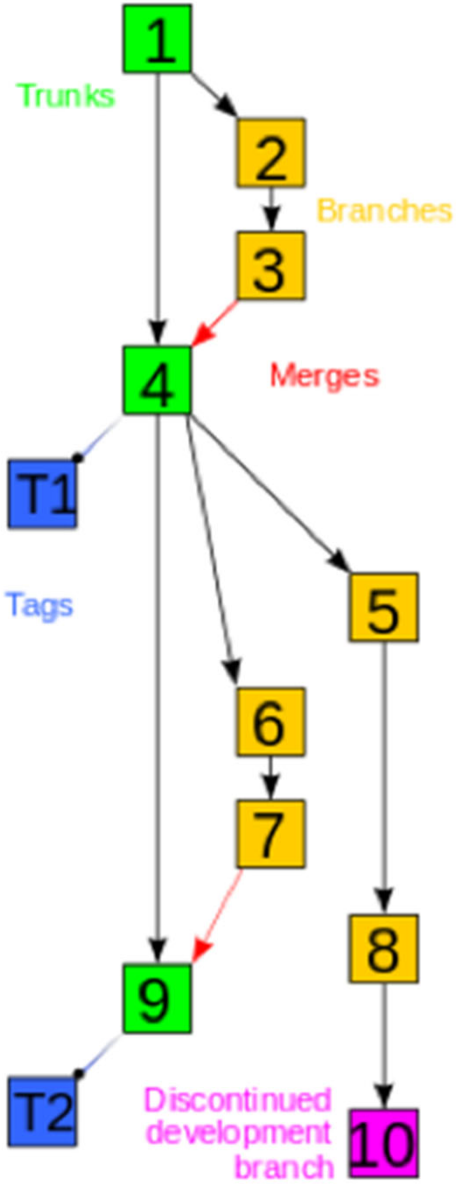
- Version control systems
- Git
- Git Flow
- GitHub
- Steps to use GitHub
- Recommendations

VERSION CONTROL SYSTEMS

VCS - Version Control System

Version Control or Source Control is the management of changes to documents, computer programs, large web sites, and other collections of information.

VCS - Graph


1
Trunks
2
Branches
3
4
Merges
T1
Tags
5
6
7
8
9
T2
Discontinued development branch
10

Tools

- CVS
- Subversion
- Visual SourceSafe
- Darcs
- Monotone
- Mercurial
- BitKeeper
- …
- and Git

GIT

What is Git

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Common vocabulary

- Repository
  - The repository is where files, current and historical data are stored, often on a server.
- Working copy
  - The working copy is the local copy of files from a repository, at a specific time or revision.
- Revision
  - Also version: A version is any change in form

Common vocabulary

- Clone
  - Cloning means creating a repository containing the revisions from another repository.
- Checkout
  - To check out is to create a local working copy from the repository.
- Commit
  - To commit is to write the changes made in the working copy back to the repository.

Common vocabulary

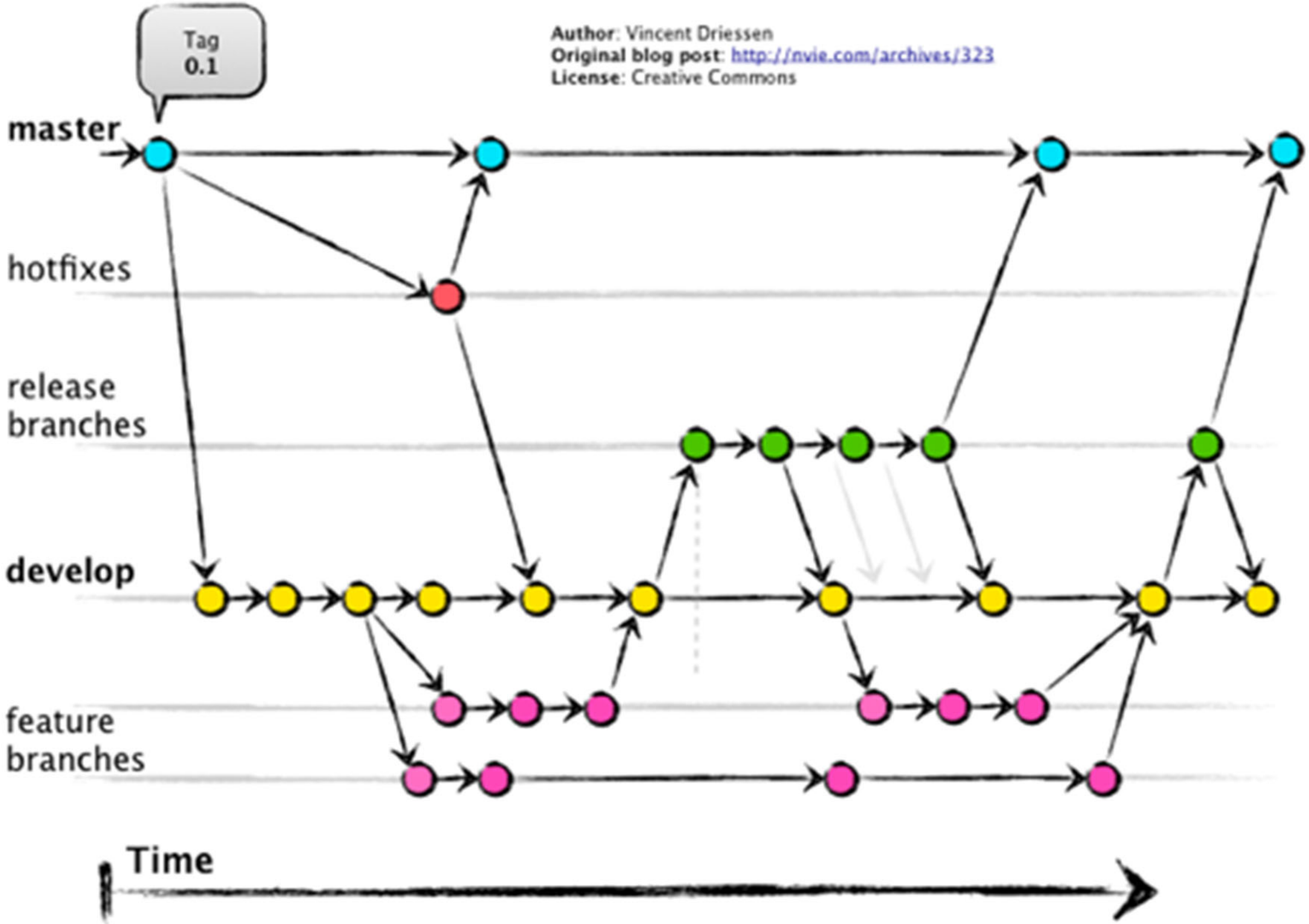
- Pull, push
  - Copy revisions from one repository into another.
- Tag
  - A tag or label refers to an important snapshot in time, consistent across many files.
- Head
  - Also sometimes called tip, this refers to the most recent commit, either to the trunk or to a branch.

Common vocabulary

- Branch
  - A set of files under version control may be branched or forked at a point in time so that, from that time forward, two copies of those files may develop at different speeds or in different ways independently of each other.
- Merge
  - A merge or integration is an operation in which two sets of changes are applied to a file or set of files

GIT FLOW

GitFlow



Author: Vincent Driessen
Original blog post: http://nvie.com/archives/323
License: Creative Commons

GITHUB

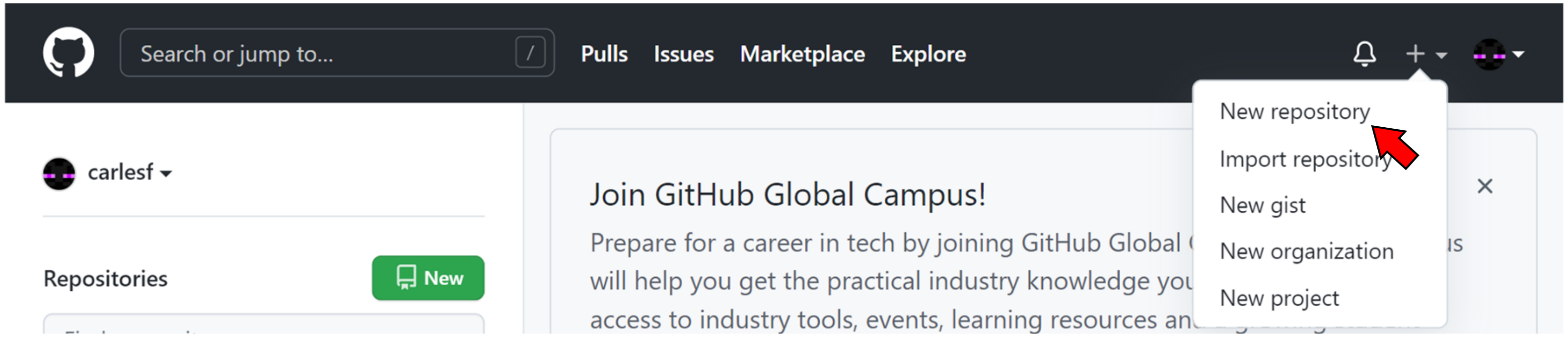
What is GitHub

- GitHub is a **distributed version control and source code management tool** for Git repositories.
- Repositories on GitHub can be accessed and managed using the standard Git command-line interface.
- GitHub also provides other collaboration features such as bug tracking, feature requests, task management, continuous integration, and wikis.
- GitHub was acquired by Microsoft in 2018.

https://en.wikipedia.org/wiki/GitHub

STEPS TO USE GITHUB

Create a new repository (I)

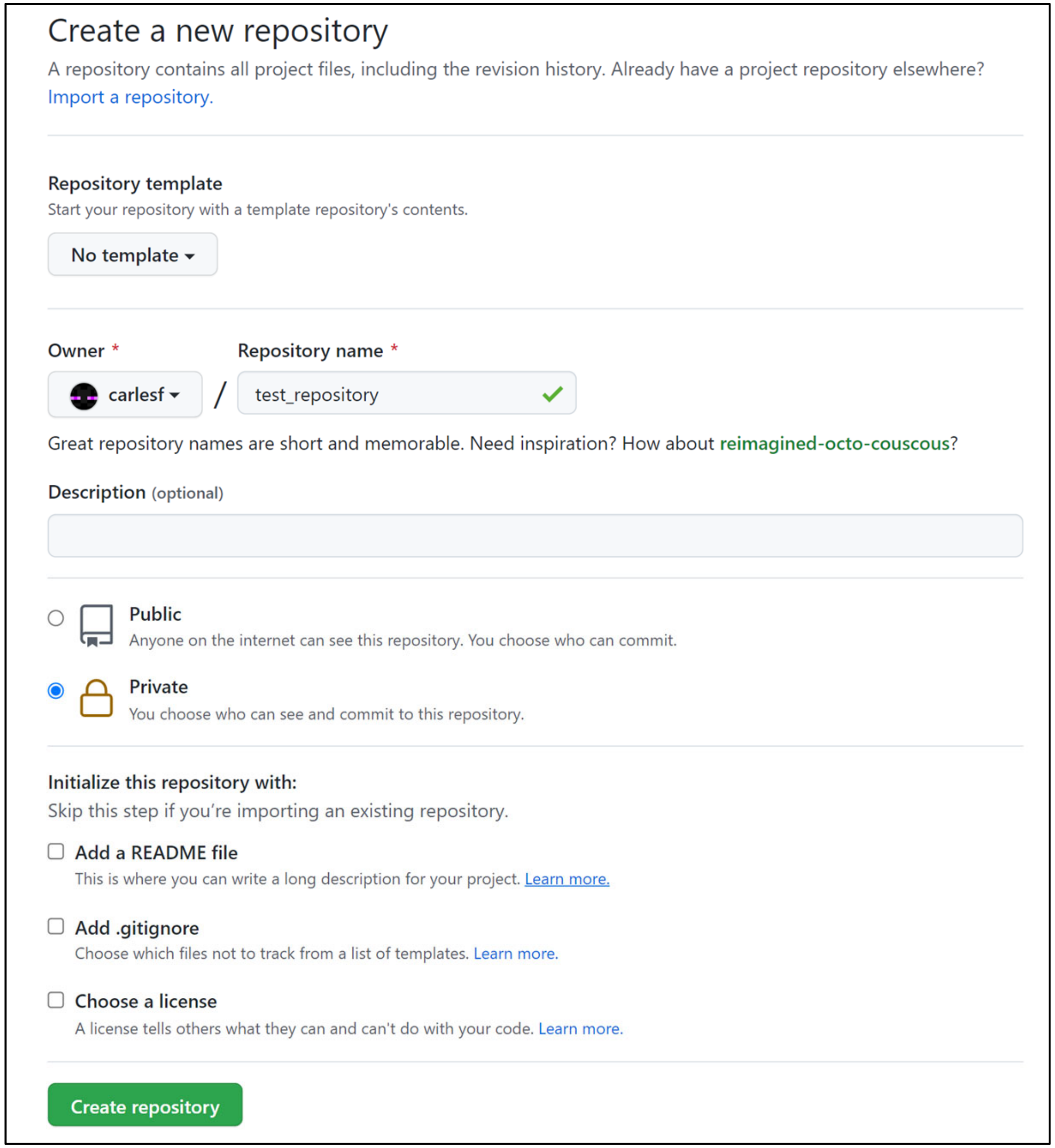


Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global (... us will help you get the practical industry knowledge you ... access to industry tools, events, learning resources an...

Create a new repository (II)

private! →



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * / **Repository name** *

carlesf ▾ / test_repository ✓

Great repository names are short and memorable. Need inspiration? How about reimagined-octo-couscous?

Description (optional)

○ **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:

Skip this step if you're importing an existing repository.

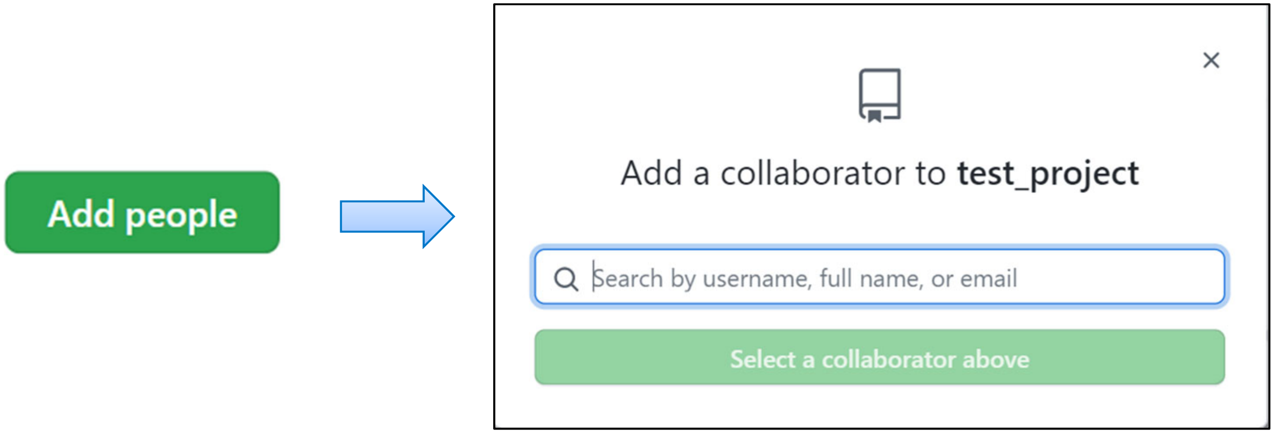
□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Add Collaborators

- On your repository page, go to ⚙ Settings and then to Collaborators
- Start adding your team members:



Add a collaborator to **test_project**

Add people
Search by username, full name, or email
Select a collaborator above

- Don't forget to add your teacher! (They will give you their username/email to do it)

Repository setup (I)

- One team member should create a local (i.e., in their computer) git repository through the new project command of your framework/tool of choice (rails, django, angular, react, …)
- Go inside the new directory created by that command:

```
cd my_new_project
```

Repository setup (II)

- **If git has been already initialized** (check if you have a .git subdirectory), then just execute

```
git remote add origin https://github.com/carlesf/test_repository.git
git branch -M main
git push -u origin main
```

The URL of your GitHub project

Syncs the local and the GitHub repository

Repository setup (III)

- **If git has not been initialized** (you don't have a .git subdirectory), then execute

```
echo "# test_repository" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/carlesf/test_repository.git
git push -u origin main
```

Creates the repo's README.md file

The URL of your GitHub project

Syncs the local and the GitHub repository

Clone the GitHub repository

- The other team members should clone the GitHub repository to have their own local copy:

```
git clone https://github.com/carlesf/test_repository.git
```

The URL of your GitHub project

README.md

- Edit the README.md so that it contains the following info (markdown):

```
## Team Members

| Name | GitHub username | Taiga username |
| --- | --- | --- |
| John Doe | johndoe34 | doejohn24 |
```

A row for each team member (teacher excluded)

Commits

- Add the related Taiga's task id in each commit command:

```
git commit –m “task #34: Login form updated”
```

- Even if you write code in pairs, make sure that all team members commit often and evenly.

RECOMMENDATIONS

Recommendations

- If your project involves the use of different technologies/frameworks, use more than one repository. Typically:
  - 1 repository for the front-end
  - 1 repository for the back-end

Projecte d’Enginyeria del Software: GitHub

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d’Informàtica de Barcelona