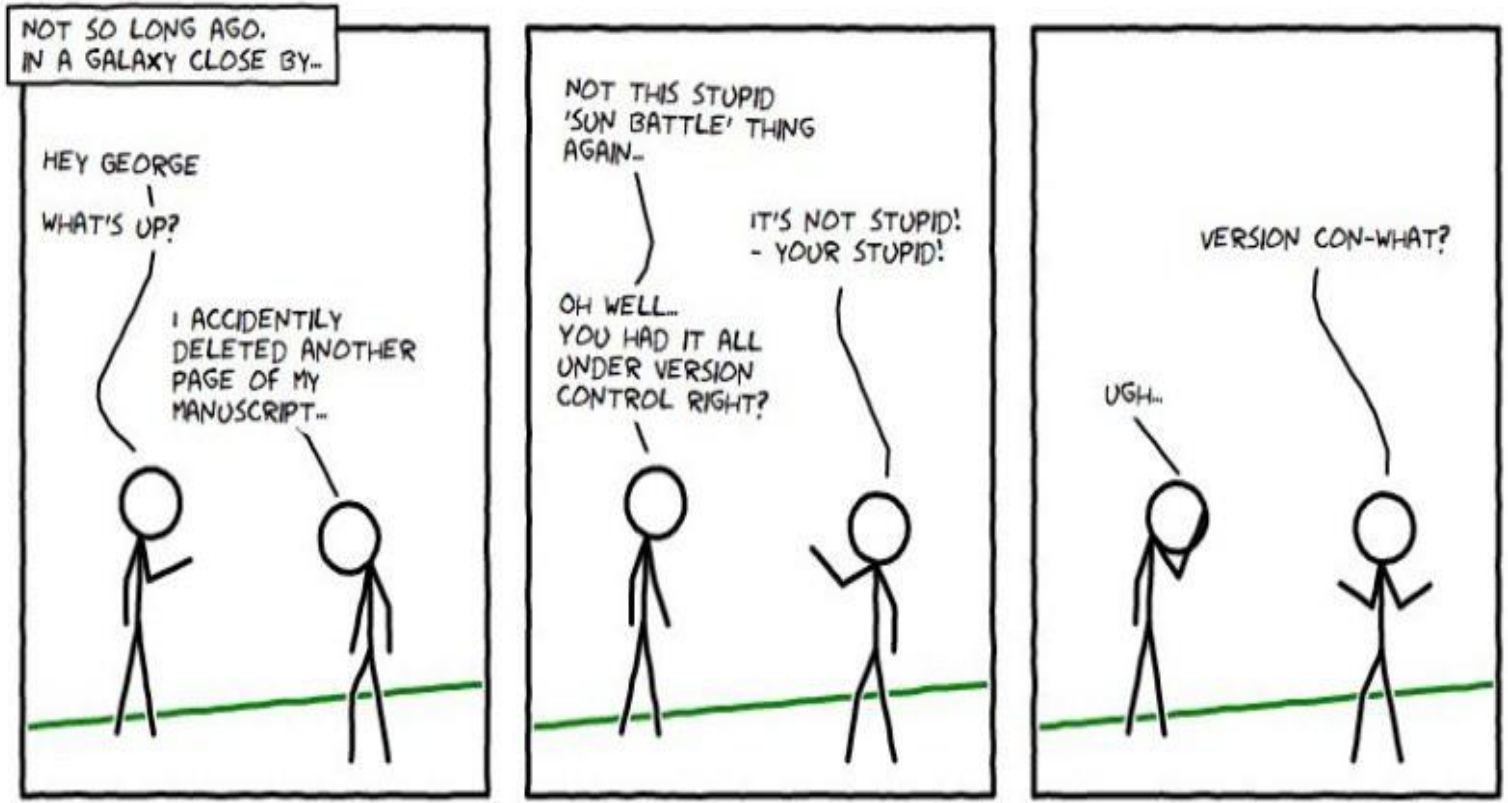

NOT SO LONG AGO.
IN A GALAXY CLOSE BY...
HEY GEORGE
WHAT'S UP?
I ACCIDENTILY DELETED ANOTHER PAGE OF MY MANUSCRIPT...
NOT THIS STUPID 'SUN BATTLE' THING AGAIN...
IT'S NOT STUPID! - YOUR STUPID!
OH WELL... YOU HAD IT ALL UNDER VERSION CONTROL RIGHT?
VERSION CON-WHAT?
UGH...

Controllo delle versioni con git

Alessandro Pellegrini
pellegrini@diag.uniroma1.it

git

- Nato dalla comunità degli sviluppatori di Linux
- Linus Torvalds, 2005
- Obiettivi iniziali:
  - velocità
  - supporto per lo sviluppo non lineare (migliaia di *branch* parallele)
  - Completamente distribuito
  - In grado di gestire progetti grandi e complessi (es, Linux) in maniera efficace

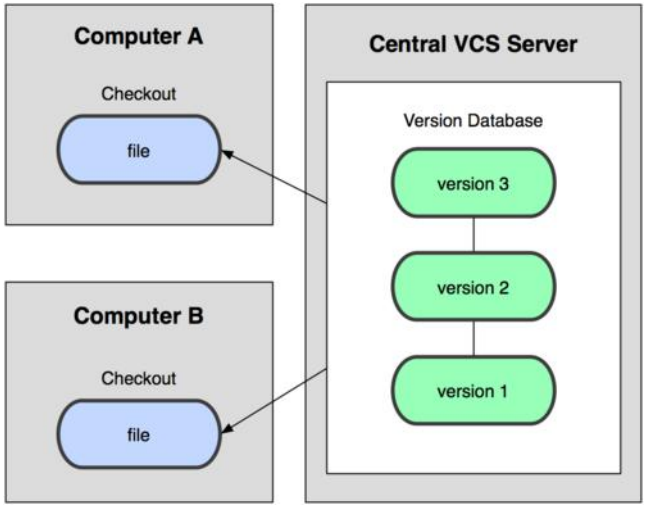
Alcune risorse

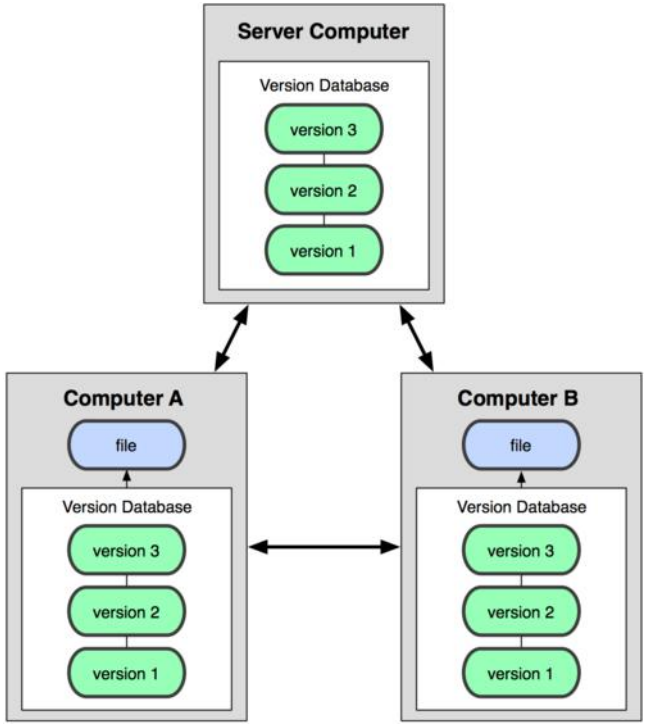
- Dalla linea di comando:
  - git help <verb>
  - git <verb> --help
  - man git-<verb>
  - <verb> è uno dei comandi di git (commit, add, config, …)

- Libro online: http://git-scm.com/book
- Tutorial: http://schacon.github.com/git/gittutorial.html
- Git for computer scientists: http://eagain.net/articles/git-for-computer-scientists/
- Dispensa caricata sul sito del corso

Modello distribuito

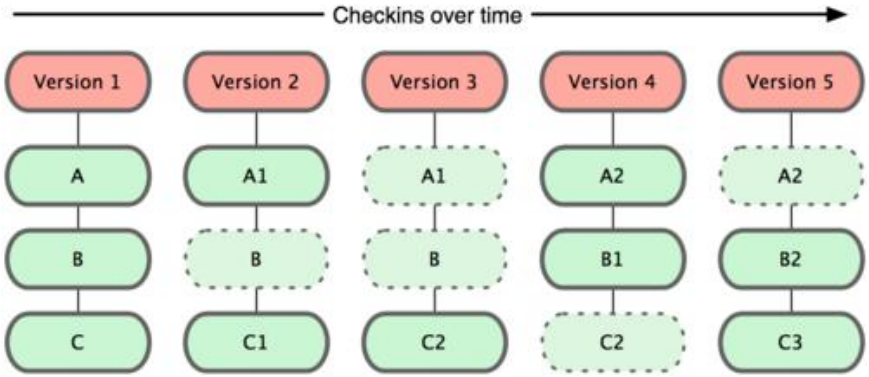
- git utilizza un modello completamente distribuito
- le versioni sono identificate da un hash SHA-1





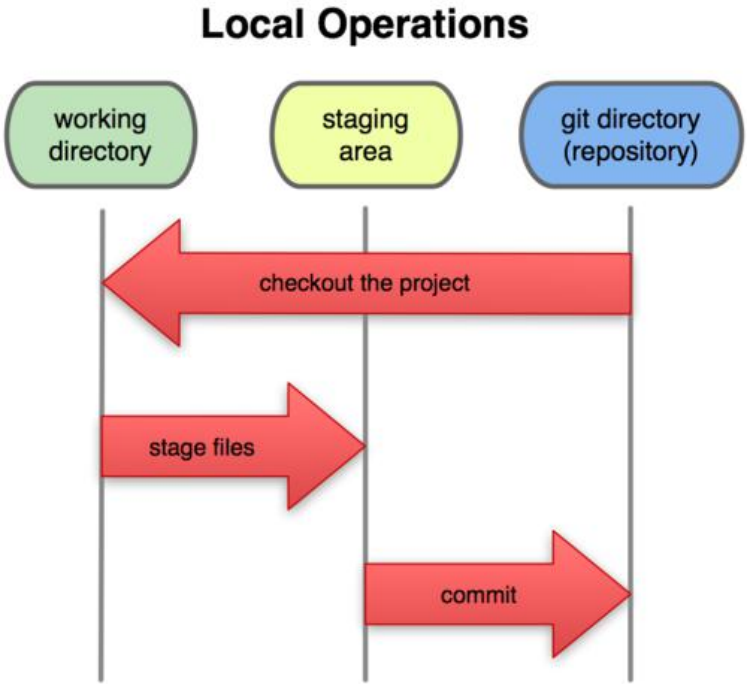
Snapshot

- Internamente, git crea degli snapshot del lavoro
- Basato sul concetto di “filesystem”
- Permette velocemente di portarsi avanti e indietro nel flusso di lavoro

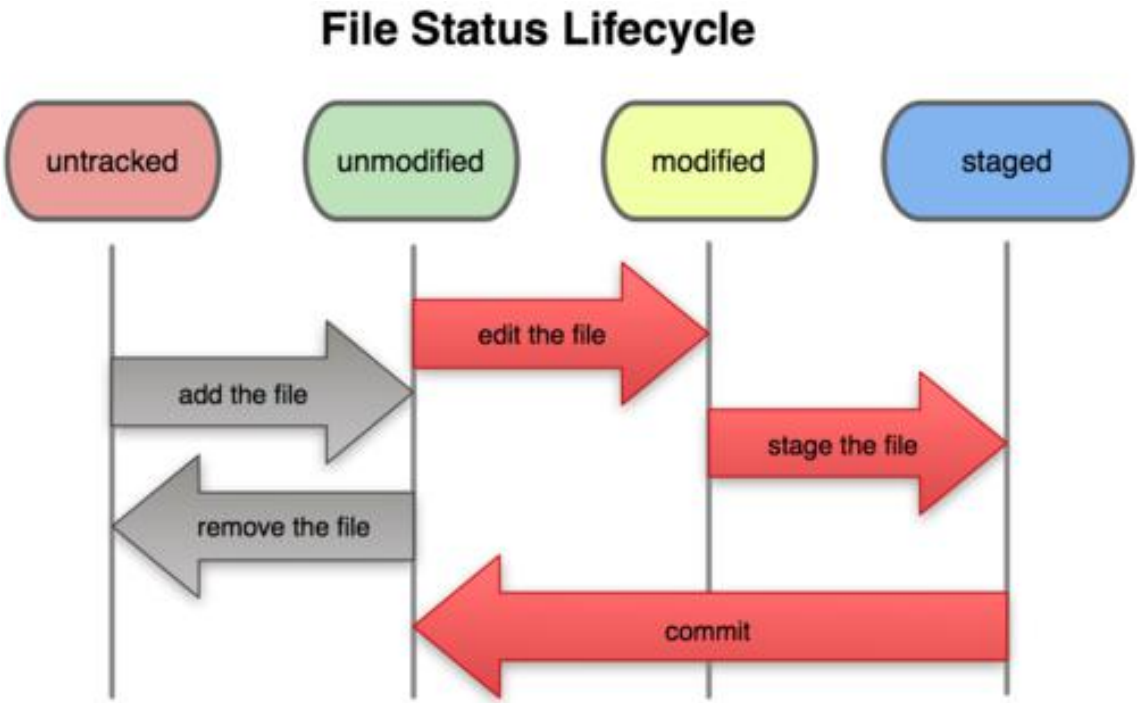


Aree di un progetto locale

- Un progetto locale ha tre aree



Ciclo di vita di un file



GitHub

- github.com è un sito che ospita copie di reposiroty git
- Utilizzato da molti progetti open source (anche dal Kernel Linux)
- Offre spazio gratuito per progetti open source

- Devo utilizzare github per usare git? No!
  - git è un sistema di controllo delle versioni distribuito

Preparazione all'uso di git

- Imposta il nome utente e l'email con cui verranno “firmati” i commit

```
git config --global user.name “Bugs Bunny”
git config --global user.email bugs@gmail.com
```

- Si può utilizzare `git config -list` per verificare la loro corretta impostazione
- Omettengo il flag `--global` si possono impostare variabili di configurazione locali per ciascun repository

Creare un repository locale

- Ci sono due scenari principali:
  - Voglio *clonare* un repository già esistente da un server remoto:
  - **$ git clone <url> [local dir name]**

  - Voglio creare *da zero* un nuovo repository:
  - **$ git init**

    $ git add file1.py

    $ git commit –m “initial project version”

Comandi di git

command	description
`git clone` ***`url [dir]`***	copy a git repository so you can add to it
`git add` ***`files`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	

Commit di file

- La prima volta che aggiungiamo un file al repository, *ed ogni volta che ne effettuiamo una modifica che vogliamo salvare*, dobbiamo aggiungerlo all'area di staging:
  - **`$ git add README.txt hello.java`**
- Questo comando crea uno *snapshot* della versione corrente
- Per spostare il file dall'area di staging all'interno del repository, dobbiamo effettuare l'operazione di *commit*:
  - **`git commit –m “Fixing bug #22”`**

- Attenzione: questi comandi operano unicamente sulla *versione locale* del repository!

Status e diff

- È possibile vedere lo stato attuale del repository locale:
  - **`$ git status`**

- Si può chiedere di mostrare cosa è stato modificato ma ancora non inserito nell'area di staging:
  - **`$ git diff`**

- Per mostrare le differenze nell'area di staging:
  - **`$ git diff --cached`**

Logs

- Si può far mostrare l'elenco delle modifiche che sono state effettuate nelle varie versioni:
  - `$ git log`

- Se si è interessati solo alle ultime 5 modifiche:
  - `$ git log -5`

Sincronizzare repository remoti

- Per recuperare da un repository remoto le ultime modifiche pubblicate:
  - **`$ git pull origin master`**

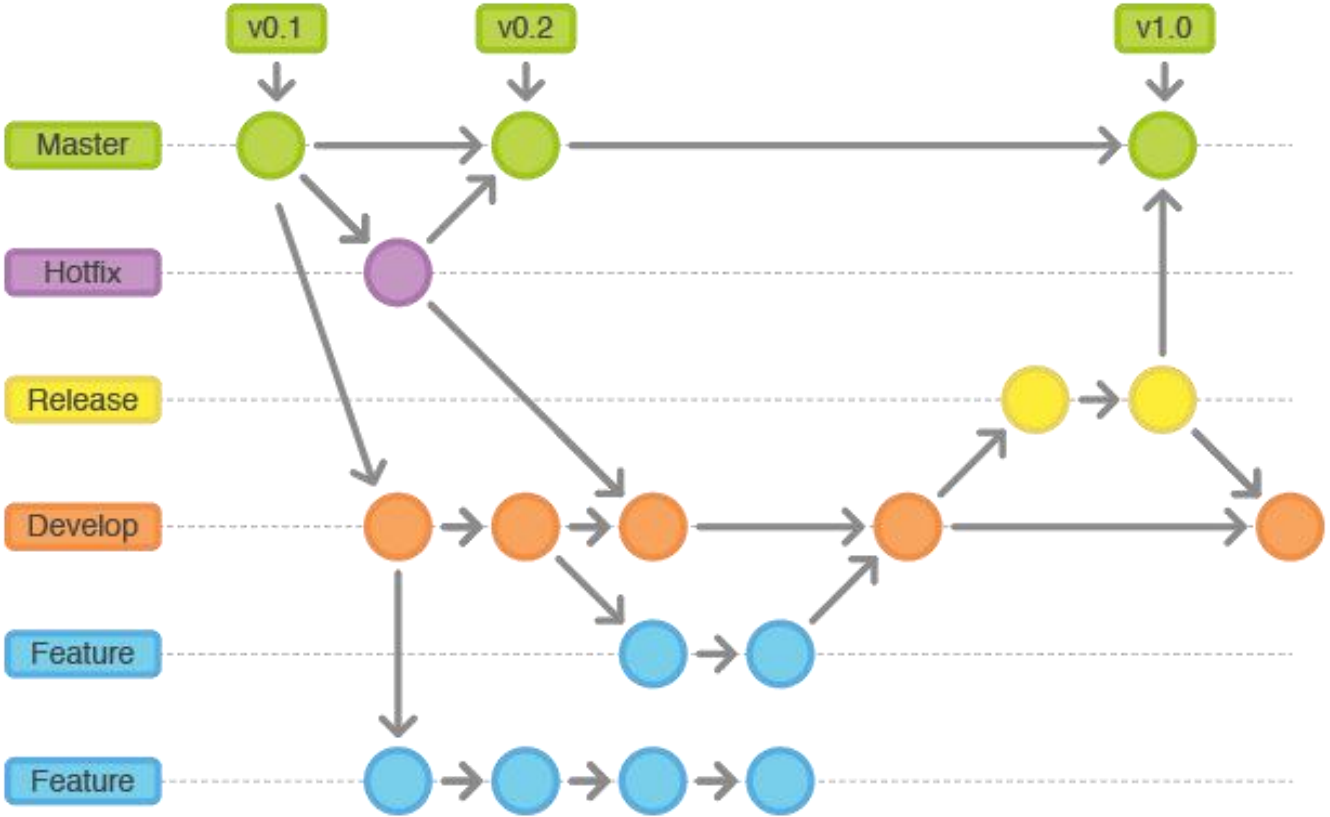
- Per pubblicare sul repository remoto i commit dal repository locale:
  - **`$ git push origin master`**

- Informazioni sul/sui repository remoti:
  - **`$ git remote -v`**

Branching

- Si possono creare versioni “multiple” di lavoro, chiamate branch
- Per creare una branch:
  - **$ git branch <nome>**
- Per mostrare tutte le branch locali:
  - **$ git branch**
- Per passare da una branch all'altra:
  - **$ git checkout <nome>**
- Per “fondere” insieme le branch:
  - **$ git checkout master**
  - **$ git merge <name>**

Branching model



Alcune risorse

- Dalla linea di comando [<verb> è uno dei comandi di git (commit, add, config, ...)]:
  - git help <verb>
  - git <verb> --help
  - man git-<verb>

- Libro online: http://git-scm.com/book
- Tutorial: http://schacon.github.com/git/gittutorial.html
- Git for computer scientists: http://eagain.net/articles/git-for-computer-scientists/
- Dispensa caricata sul sito del corso