Version Control with Git
Introduction

Dr Simon Katan

www.simonkatan.co.uk

What is version control ?

- Backup
- Record changes over time
- Make drafts
- Collaborate

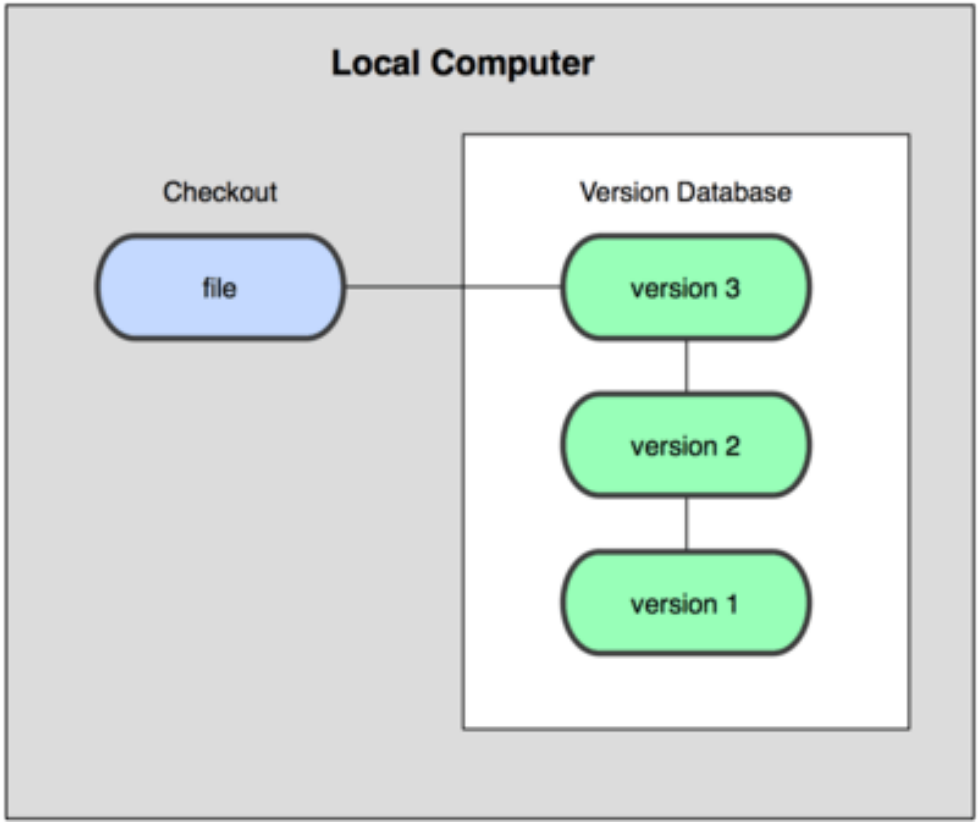
To deal with these requirements programmers developed VCSs [Version Control Systems].

Resources

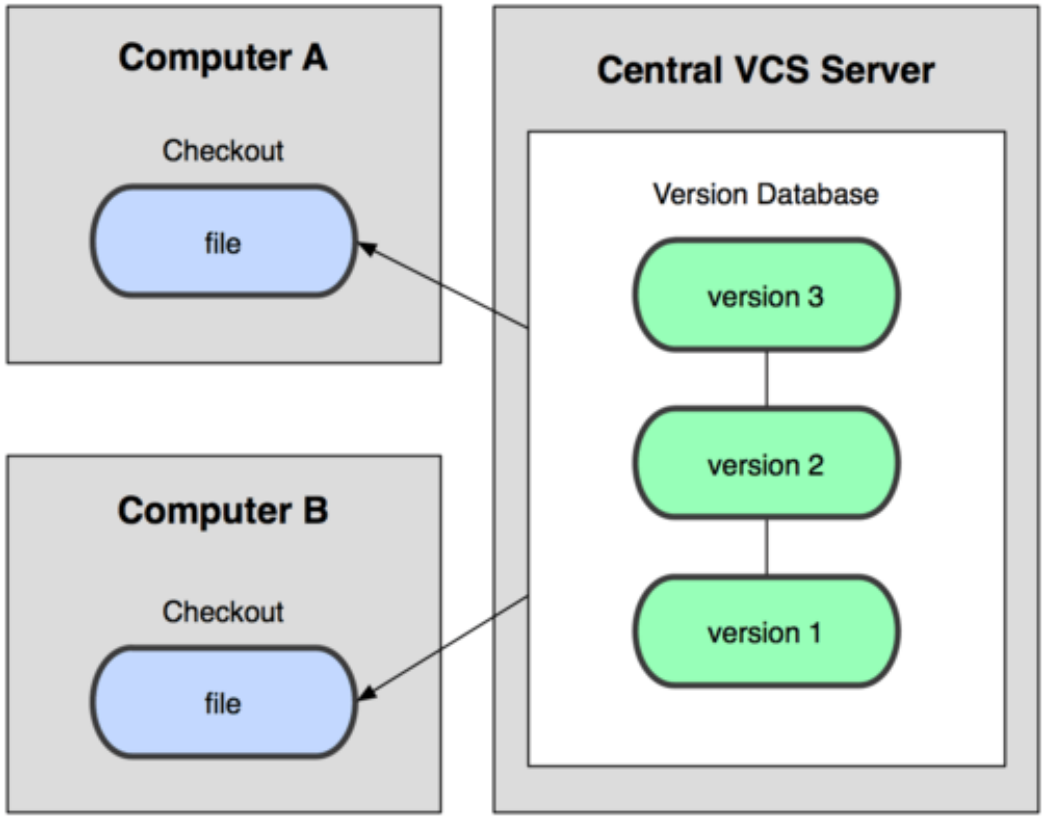
- These classes draw heavily from the resources below
  - http://git-scm.com/book
  - http://marklodato.github.io/visual-git-guide/index-en.html
  - http://cdn.bitbucket.org/themonkeymixer/cs133u/downloads/Git%20Internals%20-%20Scott%20Chacon.pdf

Local version control

- RCS
- Files only stored on hard drive.
- Changes between versions (deltas) saved as patch files.
- To get to a particular version. Start from v1 and add up all the patches to that point.



Centralised Version Control Systems CVSs



Eg. Subversion, CVS, Perforce

Advantages

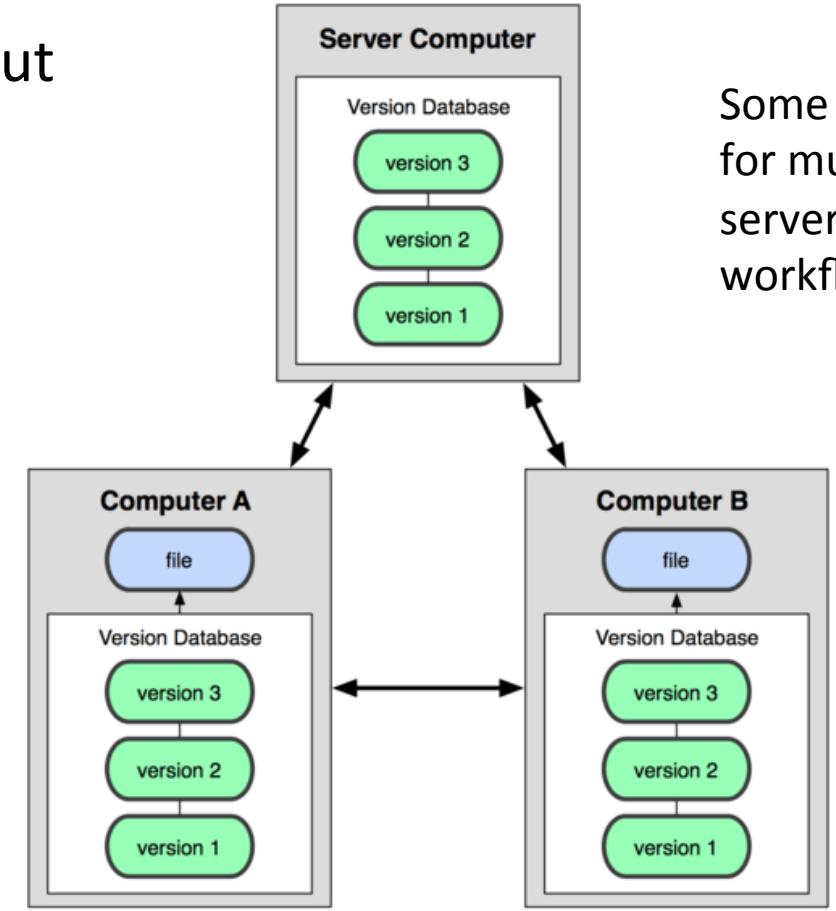
- Now collaboration is possible
- All files stored on a central server
- Everybody can see what everybody else is doing
- Administrators can control who does what
- Users checkout local versions then upload changes to the central server when ready

Disadvantages

- Server going down stops work
  - No local version control
  - No way of viewing other changes
- Vulnerability from central database corruption
  - The project could only partially be restored from piecing together the various local snapshots

Distributed Version Control Systems DVCSs

Every checkout is really a full backup of all the data



Some DVCSs allow for multiple remote servers. New types of workflow.

Eg. Git, Mercurial, Bazaar, Darcs

Advantages

- Less vulnerable to data corruption
- Individuals can work more meaningfully offline
- Allows for more fast local operations and less slow network ones
- Multiple remotes means more detailed hierarchies in workflow

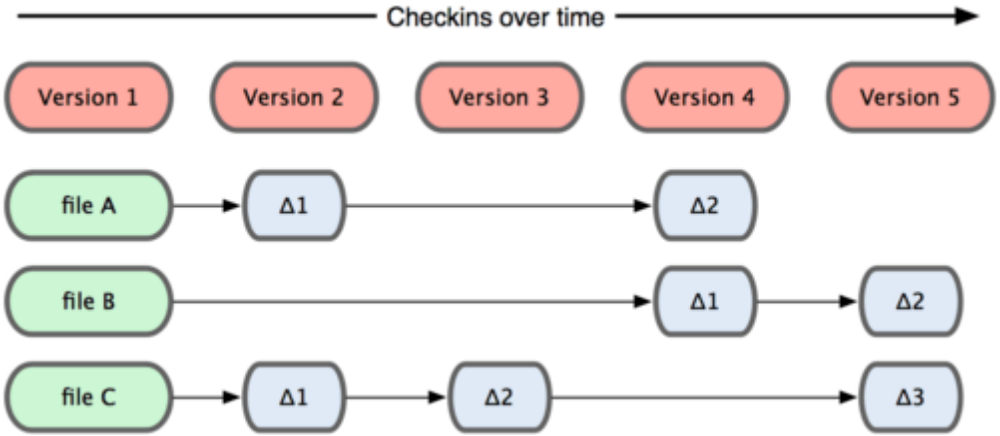
Git – A little background

"I'm an egotistical bastard, and I name all my projects after myself. First Linux, now git." – Linus

- Developed by Linus Torvalds in 2005
- Initially for Linux kernal developers
- The project very quickly spread to other Linux projects (X.org, Fedora, Wine etc.)
- Now days a very popular and mature tool
- Specialised sites to host remotes such as GitHub & BitBucket

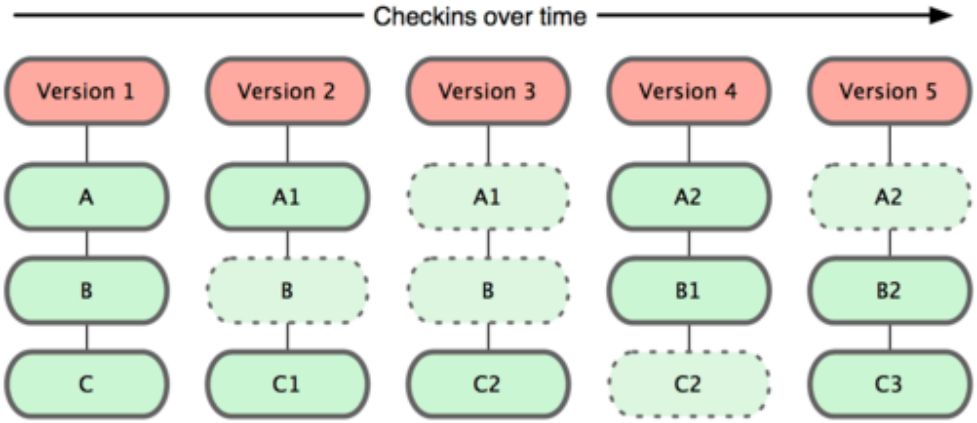
Git vs older version control systems

Older systems: record changes only (deltas)



Git: takes snapshots of whole files when they have been changed

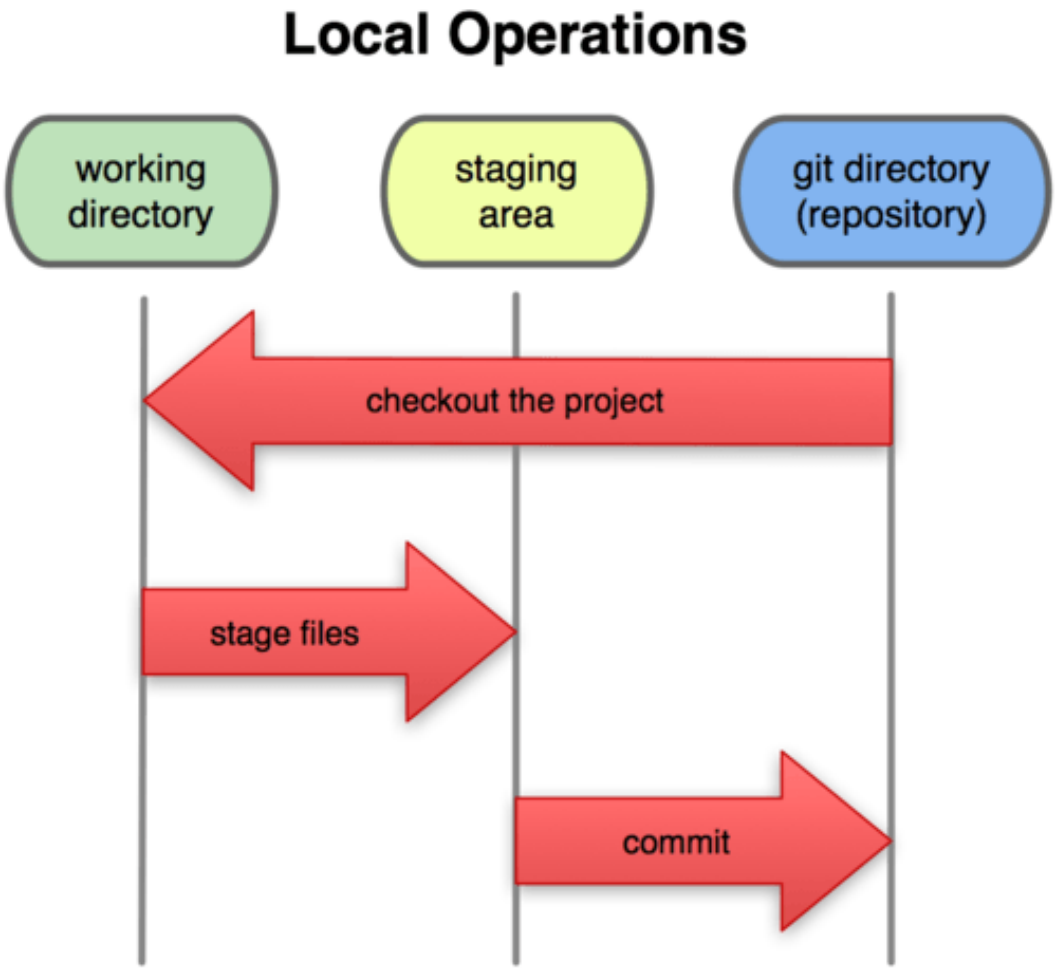
This allows for a lot of extra functionality



Git Values

- Mostly local operations = very fast
- Check-summing (Git always knows when files have been changed)
- Git generally only adds data (deleting is hard and unnecessary)
- If you have used SVN before be careful … words like add and checkout don't have the same meaning.

3 states


Local Operations
working directory
staging area
git directory (repository)
checkout the project
stage files
commit

Github for Mac

- a simple GUI for GIT

http://mac.github.com/

Concepts from GH4Mac

- Stage
- Commit
- Revert
- Branch
- Merge
- Stash

But …

- GH4Mac doesn’t give all the functionality
- Slower to keep switching to GUI
- You don’t really know what’s going on yet

Install command line git

- http://code.google.com/p/git-osx-installer