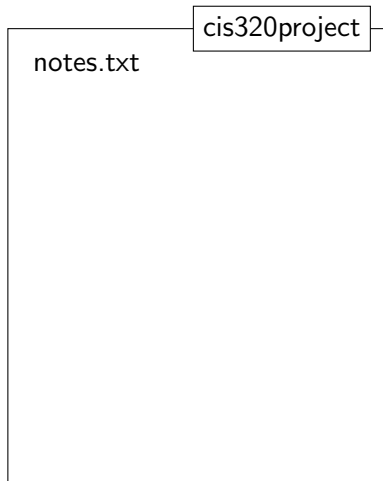


Version Control Systems

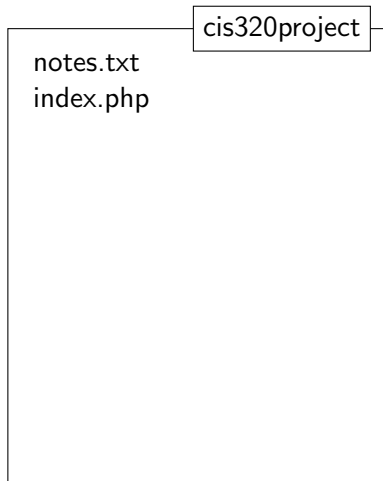
Christophe Rhodes
c.rhodes@gold.ac.uk

12th January 2013

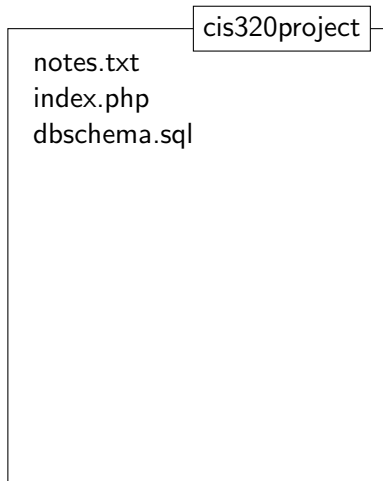
Version Control



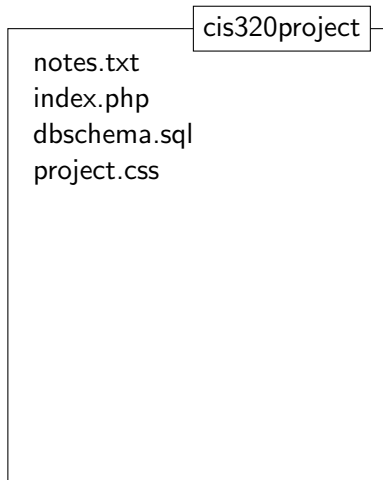
Version Control



Version Control



Version Control



Version Control

cis320project

notes.txt
index.php.old
dbschema.sql
project.css
jquery.js
project.js
index.php

Version Control

cis320project

notes.txt
index.php.old.bak
dbschema.sql
project.css
jquery.js
project.js.bak
index.php.old
project.js
index.php

Version Control

cis320project

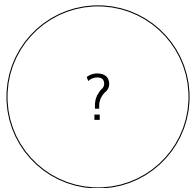
notes.txt
index.php.old.bak
dbschema.sql
project.css
jquery.js
project.js.bak
index.php.old
project.js
index.php
newdbschema.sql

Version Control

cis320project

notes.txt
index.php.old.bak
dbschema.sql
project.css
jquery.js
project.js.bak
index.php.old
project.js
index.php
newdbschema.sql
...

Version Control



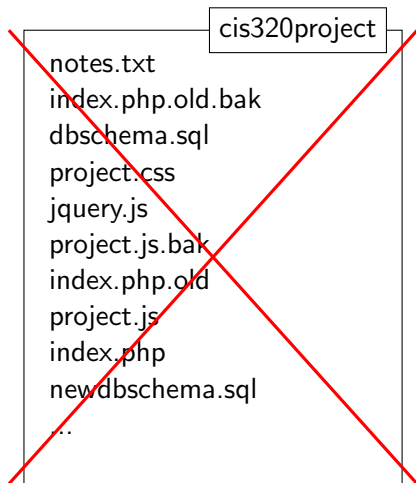
Version Control



<http://www.flickr.com/photos/purplemattfish/3188379971/>

CC BY-NC-ND 2.0

Version Control



Version Control

Why use version control?

- ▶ tracking changes made to a collection of files;
 - ▶ what was changed? by whom? when?
 - ▶ 'free' backup.

Version Control

Why use version control?

- ▶ tracking changes made to a collection of files;
 - ▶ what was changed? by whom? when?
 - ▶ 'free' backup.
- ▶ managing changes made to a collection of files;
 - ▶ revisit earlier versions;
 - ▶ 'global undo' and 'redo';
 - ▶ merging multiple changes into one consistent state.

Version Control

Why use version control?

- ▶ tracking changes made to a collection of files;
 - ▶ what was changed? by whom? when?
 - ▶ 'free' backup.
- ▶ managing changes made to a collection of files;
 - ▶ revisit earlier versions;
 - ▶ 'global undo' and 'redo';
 - ▶ merging multiple changes into one consistent state.
- ▶ dealing with collaborative work;
 - ▶ not immediately relevant to your project;
 - ▶ (likely) very relevant to your future work;

Version Control

Not just source code

Version control is not just for source code:

- ▶ text documents (e.g. \LaTeX documents);
- ▶ machine configuration files; database schemata;
- ▶ experimental data; music collection; game assets;
- ▶ ... (anything that changes).

Version Control

Not just source code

Version control is not just for source code:

- ▶ text documents (e.g. \LaTeX documents);
- ▶ machine configuration files; database schemata;
- ▶ experimental data; music collection; game assets;
- ▶ ... (anything that changes).

Embedded Version Control Systems:

- ▶ Microsoft Word, OpenOffice Writer ('Track Changes');
- ▶ Content Management Systems;
- ▶ Wikis (e.g. Wikipedia); Dropbox;
- ▶ ...

Version Control

Version Control Systems

- ▶ Centralized systems: 'master' repository with 'working copies'
 - ▶ ancient/dead: RCS, SCCS;
 - ▶ creaky: CVS;
 - ▶ active(ish): Subversion, Perforce.
- ▶ Distributed systems: peer-to-peer, working copies come with history
 - ▶ creaky: svk, arch, tla, baz, Bitkeeper;
 - ▶ active: git, darcs, mercurial, bzr, monotone.

Version Control

Version Control Systems

- ▶ Centralized systems: 'master' repository with 'working copies'
 - ▶ ancient/dead: **RCS**, **SCCS**;
 - ▶ creaky: **CVS**;
 - ▶ active(ish): **Subversion**, Perforce.
- ▶ Distributed systems: peer-to-peer, working copies come with history
 - ▶ creaky: svk, arch, tla, baz, Bitkeeper;
 - ▶ active: **git**, **darcs**, **mercurial**, bzr, monotone.

Version Control

Version Control Systems

- ▶ Centralized systems: 'master' repository with 'working copies'
 - ▶ ancient/dead: RCS, SCCS;
 - ▶ creaky: CVS;
 - ▶ active(ish): **Subversion**, Perforce.
- ▶ Distributed systems: peer-to-peer, working copies come with history
 - ▶ creaky: svk, arch, tla, baz, Bitkeeper;
 - ▶ active: **git**, darcs, mercurial, bzr, monotone.

Subversion

Centralised version control system:

- ▶ Open Source, Apache Foundation project
 - ▶ <http://subversion.apache.org/>
- ▶ Links
 - ▶ book: <http://svnbook.red-bean.com/>

Subversion

Centralised version control system:

- ▶ Open Source, Apache Foundation project
 - ▶ <http://subversion.apache.org/>
- ▶ Links
 - ▶ book: <http://svnbook.red-bean.com/>
 - ▶ random blog post: <http://betterexplained.com/articles/a-visual-guide-to-version-control/>
 - ▶ random video: <http://youtu.be/8wYiabh2hpM>

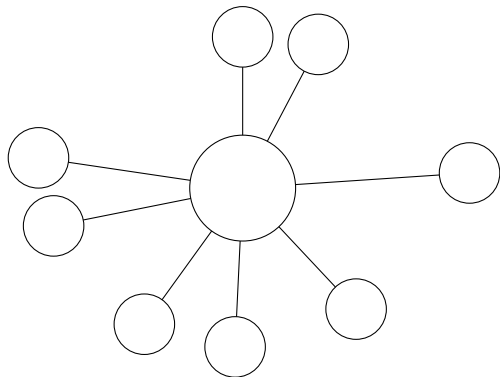
Subversion

Centralised version control

- ▶ tracks incremental versions of files, directories, ... over time;
- ▶ **repository**: privileged location, storing all the version control data;
- ▶ **working copy**: local copy of a user's version-control-managed data, on which the user can work.
 - ▶ can contain changes not yet uploaded to repository
- ▶ **client**: software able to connect to the repository, and operate on the repository's stored content and on the working copy:
 - ▶ calculate differences;
 - ▶ create new revisions;
 - ▶ merge changes;
 - ▶ generate reports.

Subversion

Centralised version control



Subversion

Working copy

- ▶ ordinary directory tree on your local system;
 - ▶ apart from `.svn` directory: do not modify!
- ▶ edit, compile, debug, add new material, move files;
- ▶ version control actions are *explicitly* invoked by client:
 - ▶ update working copy: `update`
 - ▶ make changes: `add`, `delete`, `move`;
 - ▶ examine local changes: `status`, `diff`;
 - ▶ incorporate repository changes: `update`, `resolved`;
 - ▶ submit local changes: `commit`.

Subversion

Resources

- ▶ `svn` and `svnserve` installed on igor;
 - ▶ simple repository administrator:
<http://www.doc.gold.ac.uk/svnadmin/index.php>.
- ▶ Online repository services:
 - ▶ <http://sourceforge.net/>
 - ▶ <http://code.google.com/>
- ▶ User Interfaces:
 - ▶ command-line
 - ▶ TortoiseSVN [Windows]
 - ▶ RabbitVCS [Linux]
 - ▶ svnX [Mac OS X]
 - ▶ subclipse [Eclipse]
 - ▶ vc-mode [Emacs]

Subversion

Reading

Books:

- ▶ Ben Collins-Sussman, Brian W. Fitzpatrick and C. Michael Pilato, *Version Control with Subversion*, O'Reilly (2004)
[<http://svnbook.red-bean.com/>]
- ▶ Eric Sink, *Version Control by Example*, Pyreanean (2011)
[<http://www.ericssink.com/vcbe/>]
- ▶ Lesley A. Harrison, *TortoiseSVN 1.7 Beginner's Guide*, Packt (2011)

Git

Distributed version control:

- ▶ Open Source, Software Freedom Conservancy project
 - ▶ <http://git-scm.com>
- ▶ Links:
 - ▶ book: <http://git-scm.com/book>
 - ▶ random tutorial: <http://gitimmersion.com/>
 - ▶ random video: <http://youtu.be/4XpnKHJAok8>

Git

Distributed version control

- ▶ what if there is no privileged central repository?

Git

Distributed version control

- ▶ what if there is no privileged central repository?
 - ▶ equivalently: what if every working copy were a central repository?

Git

Distributed version control

- ▶ what if there is no privileged central repository?
 - ▶ equivalently: what if every working copy were a central repository?
- ▶ each working copy carries around a full record of the history;
- ▶ clients know how to perform merges between repositories;
- ▶ never dependent on a remote system to be able to work.

Git

Distributed version control

- ▶ tracks incremental versions of whole repository over time.
 - ▶ tracks *content* (carefully defined).

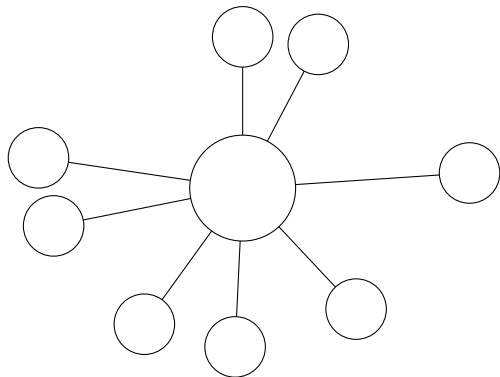
Git

Distributed version control

- ▶ tracks incremental versions of whole repository over time.
 - ▶ tracks *content* (carefully defined).
- ▶ **repository**: a location storing version control data;
- ▶ **remote**: a different repository known to a particular repository;
- ▶ **branch**: a particular revision, usually named;
- ▶ **client**: software able to operate on local and remote repositories to:
 - ▶ calculate differences;
 - ▶ create new revisions;
 - ▶ merge changes;
 - ▶ select branches;
 - ▶ generate reports.

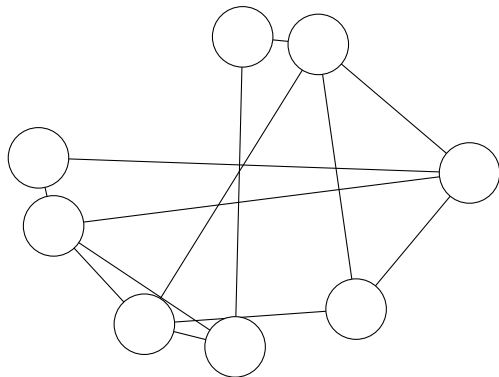
Git

Distributed version control



Git

Distributed version control



Git

Repository

- ▶ ordinary directory tree on your local system;
 - ▶ apart from `.git` directory: do not modify!
- ▶ edit, compile, debug, add new material, move files;
- ▶ version control actions are *explicitly* invoked by client:
 - ▶ make changes: `add`, `commit`;
 - ▶ examine local changes: `status`, `diff`;
 - ▶ work with branches: `merge`, `branch`
 - ▶ work with remote repositories: `pull`, `remote`;

Git

Interoperability with Subversion

- ▶ **initialization:**
 - ▶ construct new svn remote: `git svn init`
 - ▶ fetch revisions: `git svn fetch`
 - ▶ both together: `git svn clone`
- ▶ working with svn remote:
 - ▶ update repository with changes from svn: `git svn rebase`
 - ▶ commit local changes to svn remote: `git svn dcommit`

Git

Resources

- ▶ Online repository services:
 - ▶ <http://github.com/>
 - ▶ <http://gitorious.org/>
 - ▶ <http://bitbucket.org/>
- ▶ User Interfaces
 - ▶ command-line, built-in guis (gitk, git-gui);
 - ▶ TortoiseGit [Windows]
 - ▶ GitX [Mac OS X]
 - ▶ EGit [Eclipse]
 - ▶ vc-mode, magit [Emacs]

Git

Reading

Tutorials:

- ▶ Git for Computer Scientists: <http://eagain.net/articles/git-for-computer-scientists/>
- ▶ Git for Designers: http://hoth.entp.com/output/git_for_designers.html

Books:

- ▶ Scott Chacon, *Pro Git*, Apress (2009) [<http://git-scm.com/book>]
- ▶ Ben Lynn, *Git Magic*, CreateSpace (2010) [<http://www-cs-students.stanford.edu/~blynn/gitmagic/>]
- ▶ Jon Loeliger, *Version Control with Git*, O'Reilly (2009)