Week 2.1

Version Control

School of Information Technology and Electrical Engineering
The University of Queensland

Outline (svn)

- High-level
  - Concepts
  - Operations
- Subversion
  - Demo
- Subversion versus DIY

Outline (svn)

- High-level
  - Concepts
  - Operations
- Subversion
  - Demo
- Subversion versus DIY

Version Control

- **Version control (source control) –** Tools to manage changes during a project's development.
  - Many systems. Eg cvs, subversion(svn), git, bazaar, source safe, mercurial, ...
  - Lots of arguments about the best tool or method.
- The main concepts transfer between tools.
- We are only focusing on centralized VC.
- See http://svnbook.red-bean.com.

Concepts

- Repository – stores the history of the project. You do not modify this directly.
- Working copy – a copy of the files in the project where normal programming activity happens. (Could be on a different computer to the repository).
- *State** – the contents of all the files in the project.

Operations

- Single user
  - checkout – *I'd like a working copy.*
  - commit – *remember this state.*
  - add/remove/rename
  - diff/status – *what have I changed*?
  - clean copy/revert – *put it back the way it was*.
  - tag – assign a label to a state.
    - Eg: ass1 complete, release_V1

Operations

- If multiple users are committing to the same repository, there mat be commits which you don't know about.
- update – *Bring my working copy up to date with changes from the repository*.
  - What if I've made changes as well?
  - Intelligent merging rather than blind copying.
  - Will report a conflict if merging won't work.

Operations

- Blame/praise/annotate – *who changed that line last and in which revision*.
- Branching – make a separate line of development within the repository. Changes to a branch do not affect other branches or the “trunk”.
- Useful for experiments or when making large changes without disrupting people until they are done.

Subversion (svn)

- svn is a replacement for CVS.
- svn is self documenting.
  svn help
  svn help *command*
- svn checkout *URL working-dir*
  - URL – where to find the repository.
    - https://example.com/svn/project/trunk
- A working copy has hidden audit info in .svn directories.

Svn demo

- svn status
- svn diff
- svn revert
- svn help
- svn commit
  - Editor for log messages (or –m)

Version numbers

- In svn, the repository as a whole has a version number. Each time a commit is made the version number goes up.
- cvs has a more complicated system.

Svn demo

- svn add *files*
- svn move *oldname newname*
- svn mkdir *dirname*
- svn rm
  - Note: The above operations need to be committed.
- svn status

Svn demo

- svn status -u
- Dealing with conflict
  - svn resolve - “I have investigated and fixed the problem”
  - svn revert - “Forget about my changes”

Svn vs DIY

- How does svn compare with doing your own backups?
  - You can view the project in any previous committed state.
    - Backup systems might only be able to produce the latest state. Or, they thin out older backups.
  - Efficiency – (for text formats) svn stores differences between files rather than a whole new copy.

Svn vs DIY

- Times when backups/snapshots are made may not coincide with states you wish to preserve.
- How do you manage multiple developers?