Version Control with Subversion

James Curran and Tara Murphy

15th April, 2009
Outline

1. Why?
2. Why me?
3. Subversion
This is a typical multi-user situation

from *Version Control with Subversion*
And here’s what goes wrong

from *Version Control with Subversion*
The lock-modify-unlock solution

from *Version Control with Subversion*
The merge solution – checking up-to-date-ness

from *Version Control with Subversion*
The merge solution – merging changes

from *Version Control with Subversion*  
Why should I use version control?

- as a constant backup (accessible from anywhere/anytime)
- allows syncing between laptop, work and home desktops, . . .
- allows collaboration on source code, papers and lecture slides
- allows students and supervisors to share code/resources
- know who made particular changes, when and why
- because version control is the e-version of a log book
What can I store in there?

- practically anything, but...
- any text-based file formats work the best
  - plain text files
  - source code
  - \LaTeX
  - HTML/XML
- finding the *difference* between binary files makes no sense
- big files ($\geq 10$MB) will be rather slow
Subversion is a better **CVS**

- (near) replacement for *Concurrent Versions System* **CVS**
- atomic commits
- renaming and copying without loss of history
- lots of clients, tools and GUIs
- an excellent free textbook [http://svnbook.red-bean.com](http://svnbook.red-bean.com)
- becoming the most popular version control system