Software Engineering: Theory and Practice

Configuration Management

Tom Verhoeff

Eindhoven University of Technology
Department of Mathematics & Computer Science
Software Engineering & Technology

Feedback to `T.Verhoeff@TUE.NL`

Why Configuration Management?

Important potential problems:

- Ambiguity as to what document or file is meant
- Inconsistent combinations of files
- Lost files, lost changes
- Undocumented, unapproved changes
- Not knowing the composition of released software
- Not being able to go back to a previous version

Configuration Management (CM) Terminology

- Configuration Item (CI): (document, software, hardware) entity treated as a unit for CM; atomic or composite
- Version and relations between versions:
  - revision replaces obsolete/incorrect version
  - variant exists parallel to other version
- Baseline: formally reviewed and approved CI serving as a basis for further development

Configuration Management Goals

- Identify and define the CIs of the project: all relevant artifacts
- Control the release and change of CIs throughout the project
- Record and report the status of CIs and of change requests
- Verify the completeness and correctness of CIs

Configuration Management Tools

- Tools are not a complete solution: also need to use them well.

  Establish clear and effective procedures and rules. Train users.
- CVS, Subversion, Bazaar, Mercurial, Git
- Bugzilla
- Trac: an enhanced wiki and issue tracking system for software development projects
- Also consider tools for building and releasing (not treated here).

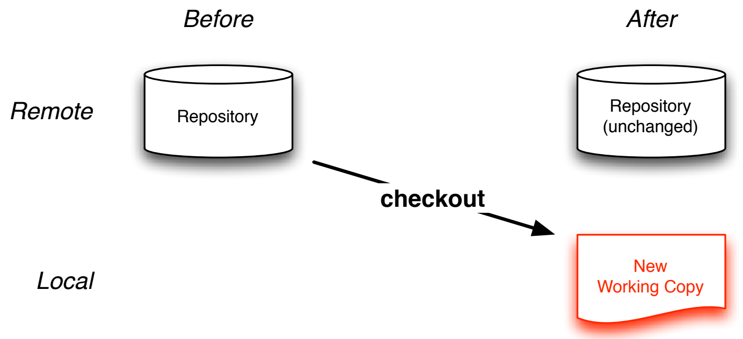
Configuration Management with Subversion

- One central repository with current and past versions
  - Trunk holds main line of development
  - Tags mark baselines, releases, . . .
  - Branches hold variants, experiments, . . .
- Multiple local working copies
  - checkout, update, commit

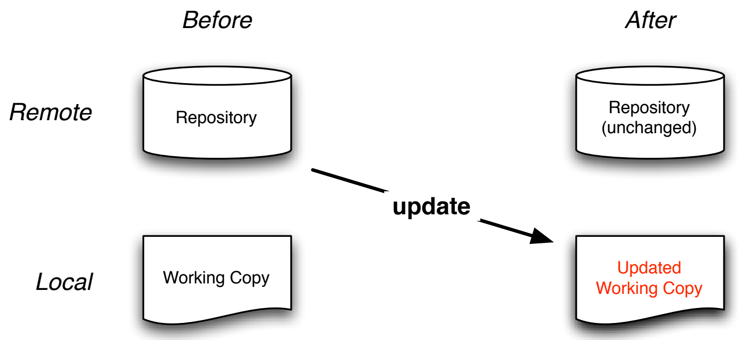
Copying in the repository is cheap, both in time and in memory usage.

Tagging and branching are done by copying. The name of the copy identifies the tag or branch.

Subversion: Create Working Copy from Repository

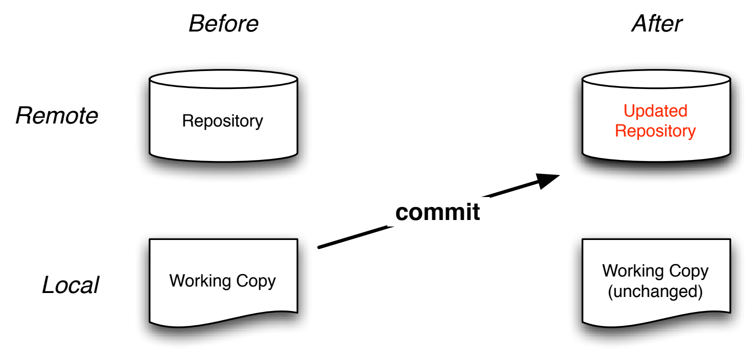


Subversion: Update Working Copy from Repository



This may result in so-called conflicts that need to be resolved.

Subversion: Commit Changes in Working Copy to Repository



Fails when repository item now differs from original working item.

Configuration Management: Good Practices

- Only commit code that compiles, runs, and passes all unit tests.
- When committing, always include a log message explaining *why* the change was made (**svn diff** can tell you *what* changed).

References

- “Configuration Management”, Ch. 17 from *Software Engineering: Planning for Change* by D. A. Lamb. Prentice-Hall, 1988.
- *Common CM Tasks in Subversion* by T. Verhoeff
- *The Subversion Book*
- Turtoise SVN: GUI client for Subversion on Windows
- svnX: GUI client for Subversion on Mac OS X