

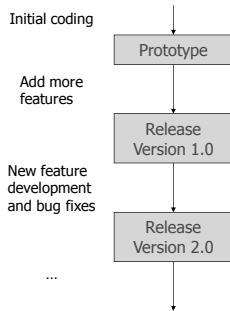
CS520 Web Programming
Version Control with Subversion

Chengyu Sun
California State University, Los Angeles

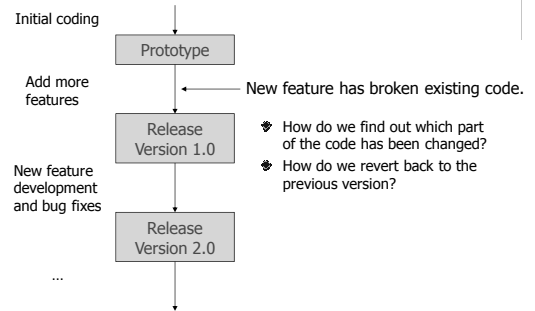
Overview

- ◆ Version control systems
- ◆ Basic concepts
 - Repository and working copies
 - Tag, branch, and merge
- ◆ Using Subversion

The Process of Application Development

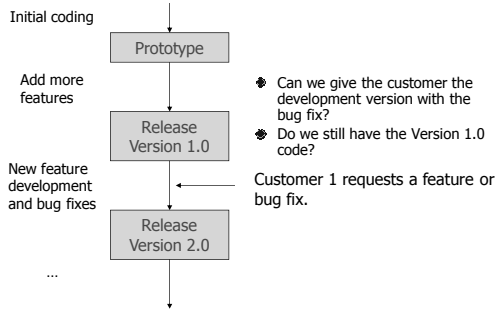


Problems During Development



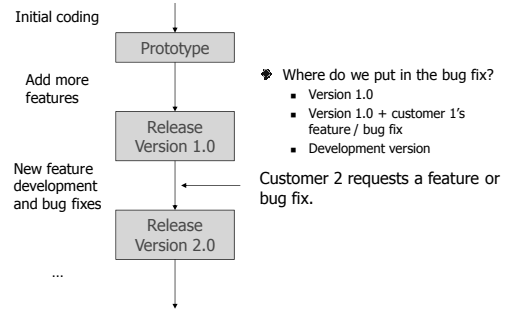
- ◆ How do we find out which part of the code has been changed?
- ◆ How do we revert back to the previous version?

Problems During Development



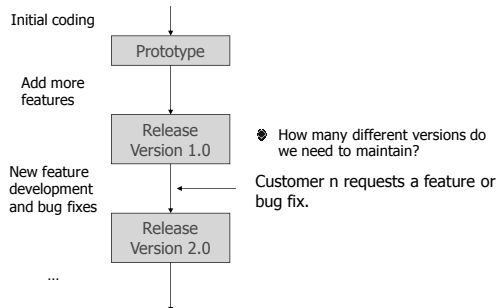
- ◆ Can we give the customer the development version with the bug fix?
- ◆ Do we still have the Version 1.0 code?

Problems During Development

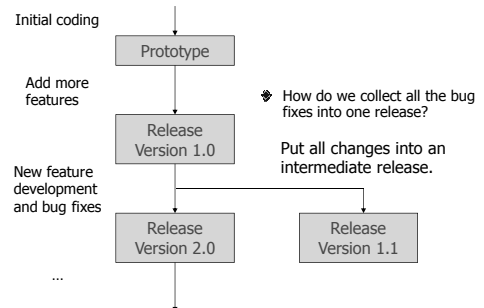


- ◆ Where do we put in the bug fix?
 - Version 1.0
 - Version 1.0 + customer 1's feature / bug fix
 - Development version

Problems During Development



Problems During Development



Version Control Systems

- ◆ **CVS**
 - *Used to be* the most popular / well known / widely used open source version control system
 - Obsolete due to some inherent system limitations
- ◆ **Subversion, git**
- ◆ **Commercial**
 - Visual SourceSafe
 - ClearCase
 - BitKeeper
 - ...

Subversion

- ◆ **Pros: A better CVS**
 - Fixed many annoying aspects of CVS
 - Recursive add, binary file handling, keyword substitution, local diff, status output etc.
 - Significant improvements
 - Atomic commit, constant time branching and tagging, better structure design etc.
 - *Feels like CVS*
- ◆ **Cons: Just a better CVS**
 - Does not scale to large, distributed development environments

Common Command Syntax

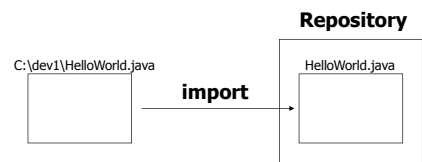
```
svn <command> [src_url] [dest_url]
```

Could be local directories or URLs.

Examples:

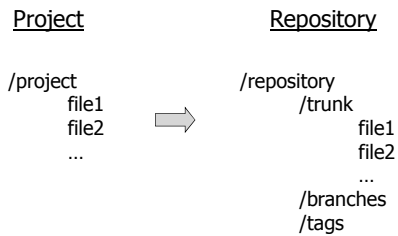
```
svn ls file:///home/cysun/subversion/cs520
svn log http://cs3.calstatela.edu/cs520
svn checkout svn://cs3.calstatela.edu/cs520/csns/trunk csns
```

Import

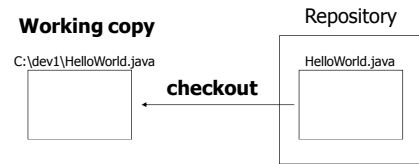


- ◆ Put a project into a repository

Directory Structure

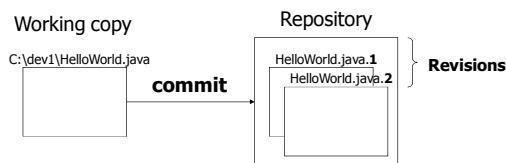


Checkout



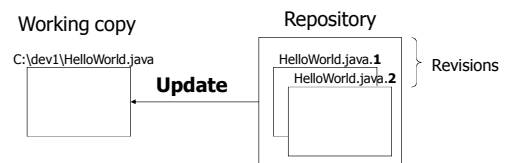
- ◆ Get a copy of the project from the repository
 - Working copy is *version controlled*

Commit (Checkin)



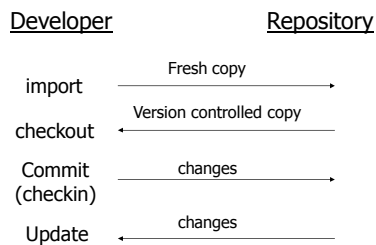
- ◆ Send changes back to the repository

Update

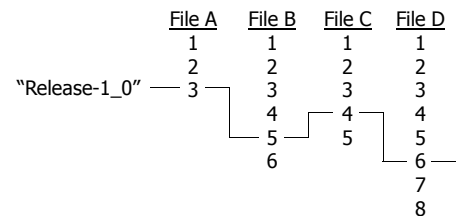


- ◆ Pull changes out of the repository and apply them to the working copy

Basic Version Control Operations

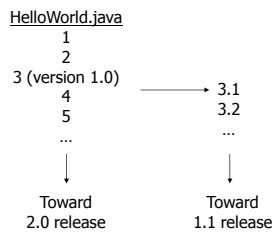


Tag – Mark A Moment in Time



`svn copy svn:///.../trunk svn:///.../tags/Release-1_0`

Branch – Work in Parallel

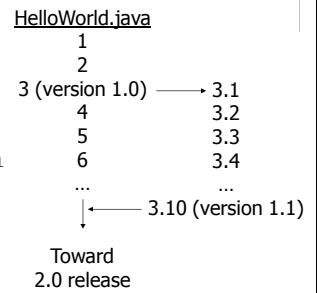


```
svn copy svn:///trunk svn:///branch/R1_1-branch
```

Merge

- ◆ Copy changes between different branches

```
svn merge src_url dest_url
```



Other Useful Commands

- ◆ Status
- ◆ Revert
- ◆ Add
- ◆ Remove
- ◆ Ls
- ◆ Log

Online Resources

- ◆ The Subversion book - <http://svnbook.red-bean.com/>
- ◆ Import and merge with Subclipse - http://csns.calstatela.edu/wiki/content/cysun/course_materials/subversion/