

CSC 255/455

Software Analysis and Improvement

Distributed Version Control

Instructor: Chen Ding

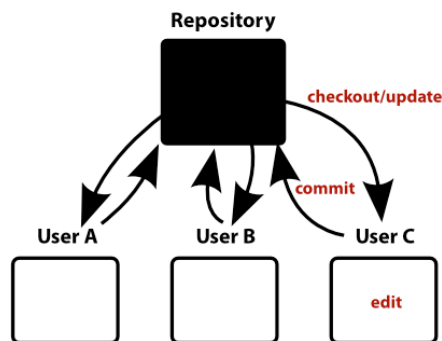
Course page: <http://www.cs.rochester.edu/drupal/u/cding/csc-255455-advanced-programming-systems-spring-2014>

Overview

- **Semantics**
 - what we want
 - sharing a class repository
 - precise properties
 - what happens at concurrent edits
- **Operations**
 - how to do it
 - use an individual branch for personal files
 - obtain updates from the main branch
 - key hg commands
 - init, add/remove, commit/update (ci/co), push/pull, merge
 - log, cat, status

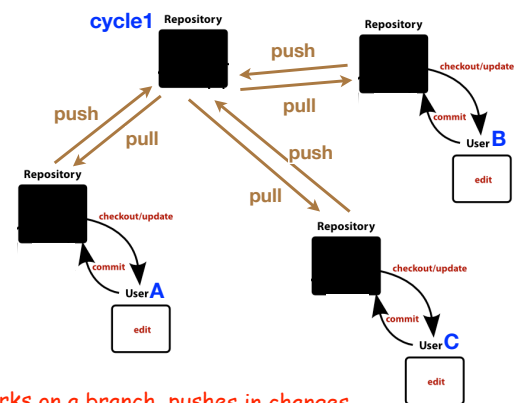
2

Centralized Version Control



3

Distributed Version Control



- Each user works on a branch, pushes in changes, fetches updates, and runs merges as needed

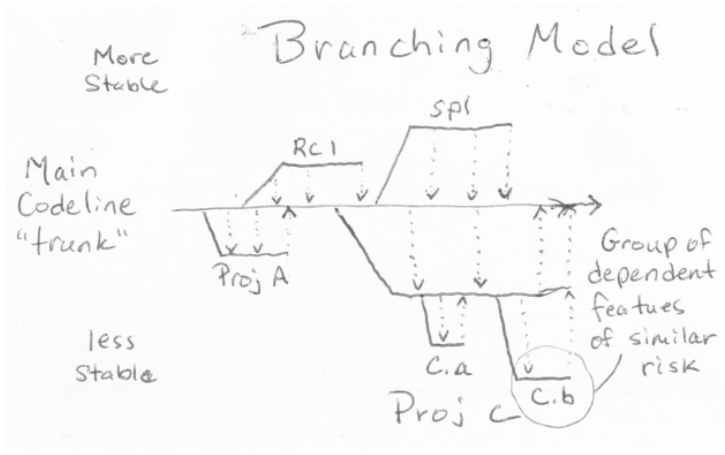
What about Concurrent Edits?

- **Branching**
 - from a common base
 - each "concurrent" commit creates a branch
 - an alternative universe from now on
- **Example uses (SVN)**
 - main trunk
 - main developments
 - feature branch
 - adding a disruptive feature but in many steps and may not work out
 - fixes branch
 - fixes for previous releases

6

Demo: Repository Creation

- hg init
- where is the repository?



http://leansoftwareengineering.com/wp-content/uploads/2008/01/branching_model.png

So... how do we decentralize?

- More formally:

Any run of the decentralized system should be indistinguishable from some run of a centralized system

- Merge: A commit made by somebody else somewhere else is indistinguishable (except for meta data) from one made by you on your local machine

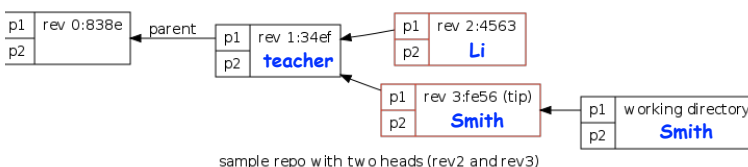
Demo: Repository Branching and Merging

- hg add/commit/push/pull/update

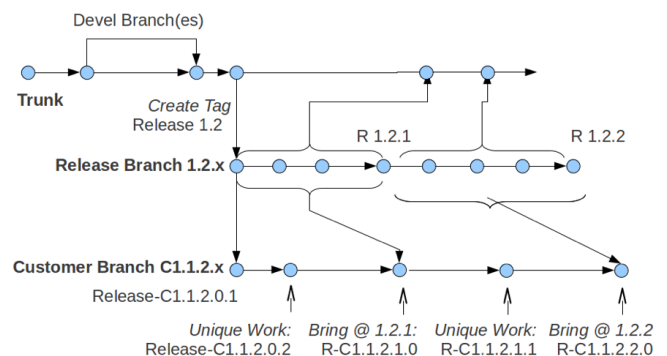
Head

hg heads

A **head** is a [changeset](#) with no child changesets. The **tip** is the most recently changed head. Other heads are recent [pulls/pushes](#) into a [repository](#) that have not yet been [merged](#).

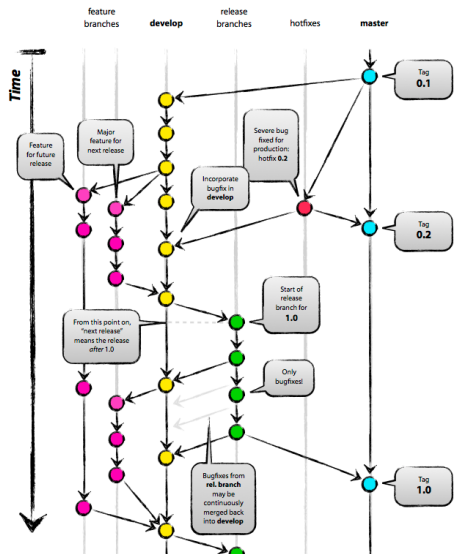


- A parent does not point to children
- A child points to its parent(s)
- A branched version has one parent
- A merged version has two parents



- How to branch at 1.2?
- How to branch at 1.2.x?
- How many merge points are there?

<http://i.stack.imgur.com/vRx5q.png>



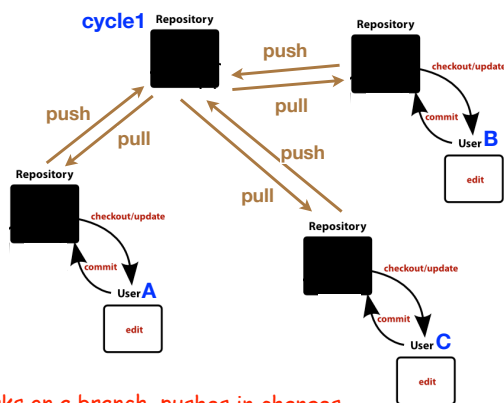
<http://i.stack.imgur.com/6rDUK.png>

More Branching

- The mainline model (Laura Wingerd's How Software Evolves)
 - active development streams
 - persistent codelines
 - task branches
 - short-lived branches for specific tasks, e.g. bug-fix
 - staging branch
 - preparing for a release
 - private branches, ad-hoc branches
 - small tasks, commit progress w/o waiting for test/review
- "Commit early, commit often"
 - "if you want to be a productive debugger, be a revision control fanatic" R. C. Metzger Debugging by Thinking

14

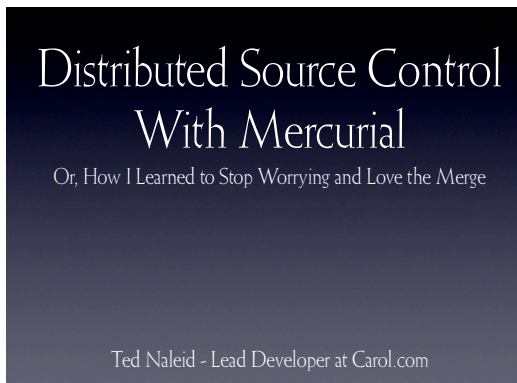
Distributed Version Control



- Each user works on a branch, pushes in changes, fetches updates, and runs merges as needed

Demo: Student Branch

- how do you name the directory of your repository?



Class Repository

See course web page -> class repository for instructions

<http://mercurial.selenic.com/wiki/Presentations>