

How can 3
(or 3000)
developers work
on a product
at once?

Version
Control

Slides #4

CMPT 276 © Dr. B. Fraser

Version Control

- Suppose an old version of your software has a security bug
 - How can you fix it while continuing to develop the new version?
- Suppose many team members need to change the same file
 - How can they develop without overwriting each others' work?
- Suppose you discover a bug
 - How can you find out when the bug was introduced?

15-01-16

2

Version Control

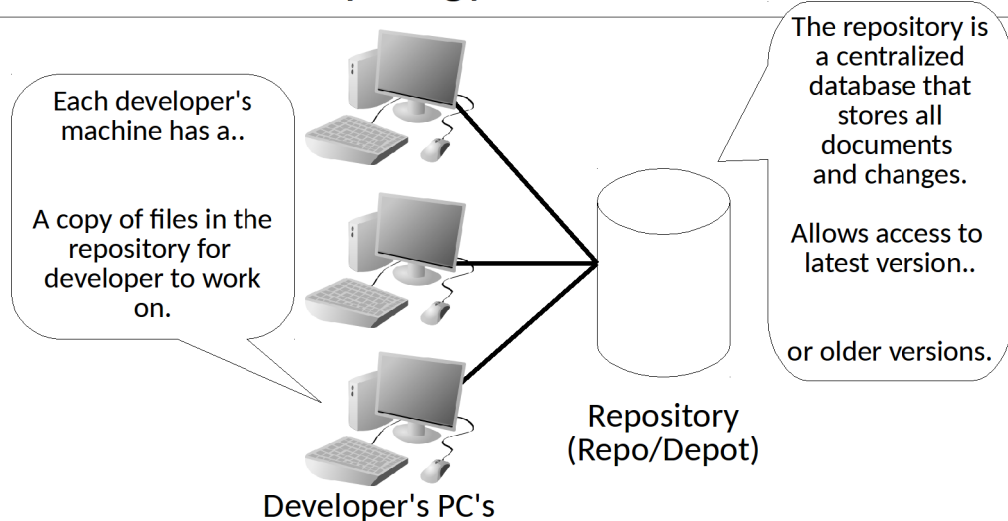
- Version Control:
 -
 - Also called revision control, source control.
- Motivation:
 - Need to manage multiple versions simultaneously.
 - Need to coordinate changes made by multiple developers.
 - Need a reliable system to ensure changes are..

My view: If a file isn't managed by version control,
it doesn't exist

15-01-16

3

Centralized Topology

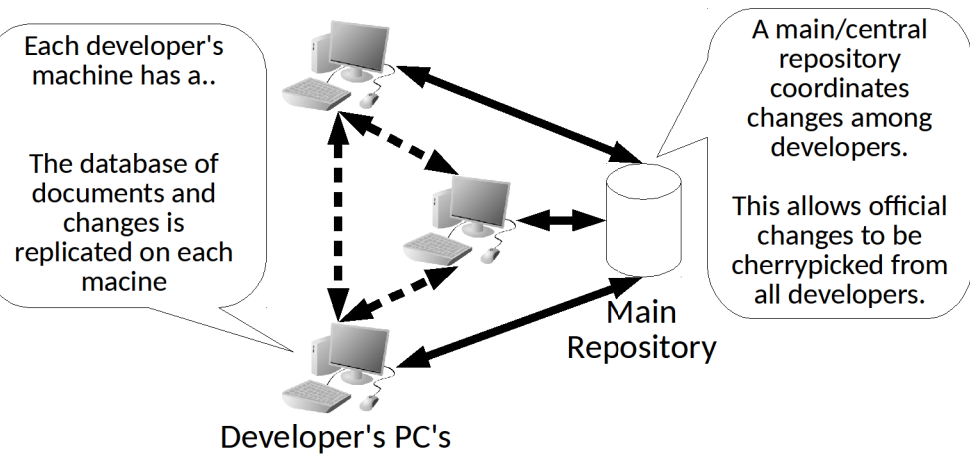


Examples: Subversion (svn), CVS, Perforce, ClearCase

15-01-16

4

Distributed Topology (DVCS)



15-01-16

5

Operations

- Init / clone:
 - Create a new repository or copy a remote one.
- Import/add:
 - Place some files under source control.
- Checkin/Commit:
 -
- Pull / Update:
- Push:
- Revert / Rollback:
 - Throw away local changes.

15-01-16

6

Operations

- View history/log:
 - See all changes made to a file.
- Diff:
 - Find the differences between two files / directories.
 - Often used to create a..
 - "Bundled" up set of changes
- Merge:
 - If two people change the... changes have to be merged into one correct file.
- Merge Conflict:
 - Two people changed the...
- File Locking:
 - prevent anyone else from editing the file.

15-01-16

7

Merge vs Lock

- Checkout-Edit-Merge
 - Merge support allows concurrent access to a file so multiple developers can work on same code at once
 - But can lead to...
- Lock-Edit-Unlock
 - Locking prevents merge conflicts by..
 - "I can't make any changes until Bob finish!"
 - Adds pressure to make changes quickly..
 - "I NEED THAT FILE! CHECK IN NOW!"

15-01-16

8

Features

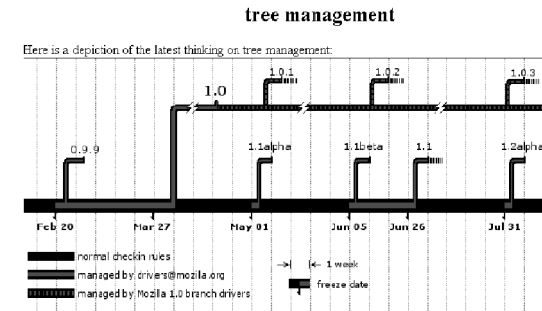
- Atomic operations:
 -
 - Change is applied all at once:
no other changes applied while you're checking in.
- Tag/Label/Baseline:
 - Mark certain versions of certain files as a group.
Ex: "Files for Version 1.0 of product".
 - Able to easily..
of the files later to fix bugs etc.
 - "Get all files exactly as they were in Version 1.0
(three years ago)".

15-01-16

9

Features

- Branch/Fork:
 - Create a second "copy" of file(s) so two versions can be...
 - Useful for applying bug fixes to released version while development continues for the next release.
 -
 - is the main-line of development.
 - other versions.



15-01-16

10

Git

- We will use Git:
 - Modern, common, free, open source version control system.
 - For this course,
 - TA and instructor will be able to view the student repos.
 - Must be used for this course.
- Client Access:
 - Will (mostly) use the Git support in Android Studio to access.
 - Other popular tools work via either the command line or the Windows shell (file-manager).

15-01-16

11

Team Work

- One rule to being a good team member:
 - When you check in, the full system must compile and run.
 - Only under exceptional circumstances should you ever check in something which breaks the build.

15-01-16

12

Coding with Source Control

- // Removed Jan 2002 for V1.01
 // cout << "Dave; I wouldn't do that, Dave.\n";
– Put meaningful comments into checkins!
- #if 0
 // Unneeded, but left 'cuz someone may want it...

 #endif
- // Written by Dr. Evil

15-01-16

13

Other Benefits

- Context
 - History records all changes & their purposes
 - Related changes occur at the same time.
(git log)
- Traceability & Searchability
 - When did a bug get inserted? Who can help?
(git blame, git bisect)
- Flexibility
 - Enables speculation & prototyping
(git branch, git stash)

15-01-16

14

From the Console

```
mkdir gitplay; cd gitplay
git init
echo first line >> README
git add README
git commit -m "Adding a line..."
git status
echo second >> README
git diff
...
git blame
```

15-01-16

15

Summary

- Version control is a critical tool for software development.
- Operations: e.g. check out, check in, update, merge.
- Features:
 - Atomic operations.
 - Branches
 - Tags/Label
- We are using Git.

15-01-16

16