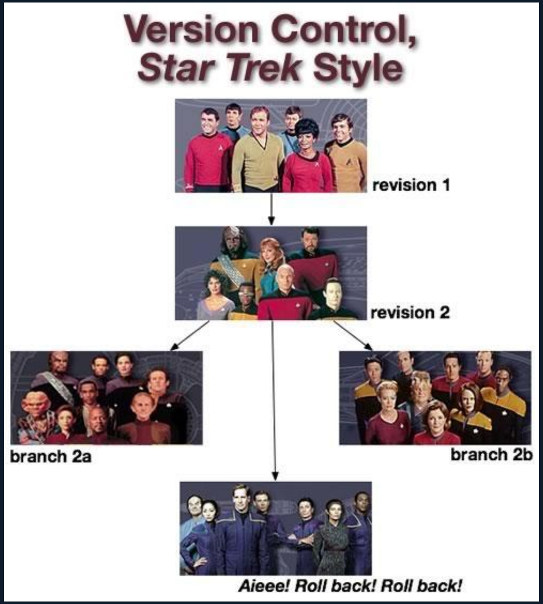
Version Control



What is Version Control? What does it do?

- A repository (database) of the source files for projects
- Backs up multiple versions of files
- Tracks changes between versions
- Merges changes in files

Why is it useful?

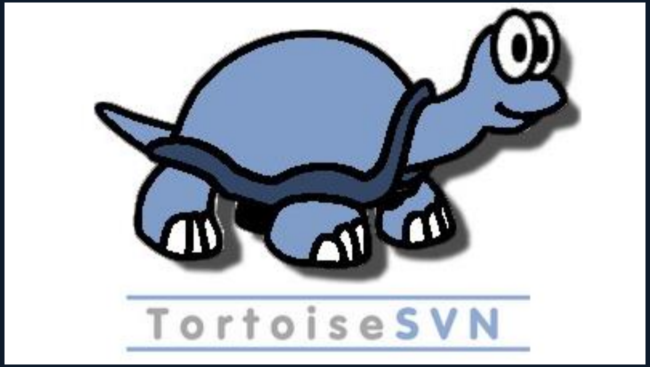
- Back-up (protection against corruption, hardware failure, general forgetfulness)
- Multiple programmers can work from the same code base, even the same file
- Ability to undo changes
- Tracking what changed, and who was responsible for those changes

Homemade Version Control

- A programmer can copy a file for backup purposes, putting the date or a version number in the file name
  - This is a very simple, manual version control

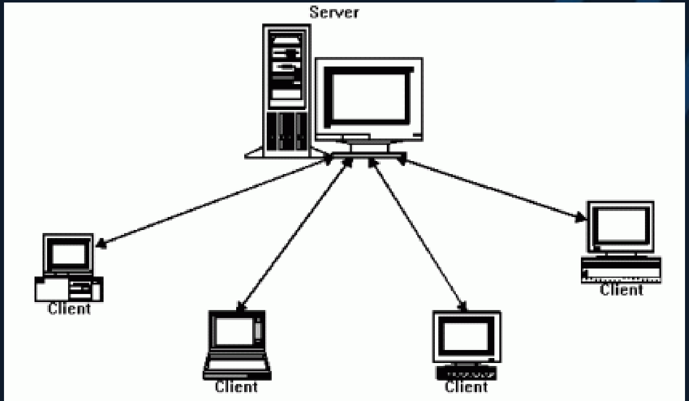
Common Control Software

- SVN (Subversion)
- Git
- Perforce



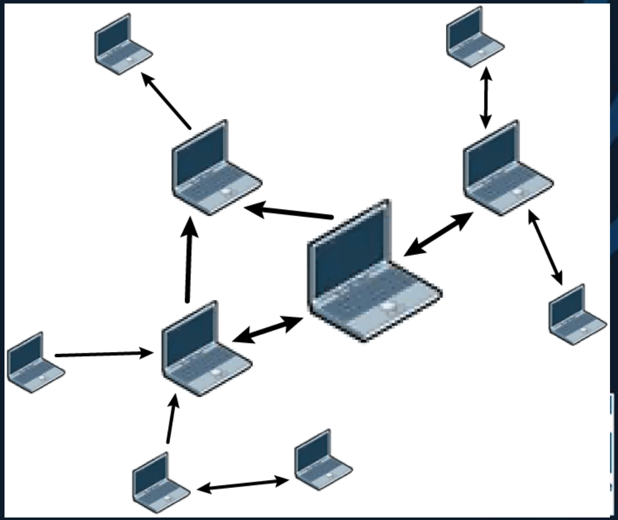
Client/Server Model

- Most version control software employ a client/server model
  - The server
    - Manages a repository (database) of files
    - Keeps track of changes
    - Synchronizes changes with clients
  - A client
    - Receives updates from the server
    - Commits new versions to the server



Distributed Model

- In a distributed model every developer has their own repository
- Multiple programmers can save changes into a common repository if desired
- Tends to share changes when desired and not force updates on others



Some definitions

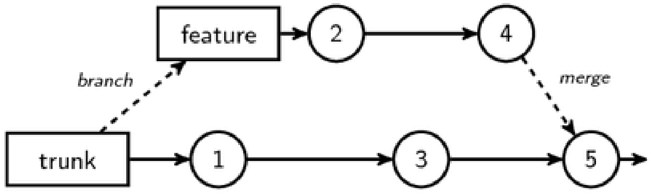
- **Repository/Depot** – the database that stores all files
- **Server** – the computer that stores the repository
- **Client** – any computer that connects to the repository
- **Trunk/Main** – the primary location for code in the repository
- **Working Copy/Local Copy** – copy of a file that exists on a client machine
- **Head** – the latest version of the repository
- **Comment** – a description of a change

Basic Actions

- **Checkout** – obtain a new local copy of the repo
- **Update/Get Revision** – update a local copy from the repository
- **Commit/Submit** – commit local changes to the repository
- **Add** – add a local file or directory to the repository
- **Delete** – delete local file or directory from the repository
- **Revert** – Throw away local changes and update from the repository
- **View Log** – view a list of changes in the repository

Other Actions

- Most software control systems permit the client additional actions
  - Rename code
  - Add log entry about changes
  - Compare different versions
  - Merge two versions
  - Create a project “branch” – a different copy of the same project



References

- Subversion Client
  - http://tortoisesvn.tigris.org/
- Git Client
  - http://www.sourcetreeapp.com
- Perforce
  - http://www.perforce.com/