CS 4320 / 7320
Software Engineering

Version Control Systems

Topics

- Version Control Systems
  - What are they
  - Types
- Terminologies
- Collaborative Development through VCS
- Introduction GIT

Version Control System

- Track changes and revisions to both files and file system structure of working environment
  - File Additions / Deletes
  - Folder Additions / Deletes
  - File Edits
- aka
  - Source control system
  - Source code control system
  - Revision control system

Change Tracking

- Files (and folders) exist in a temporal condition
- File A:
  - T-1 has 100 lines
  - T-101   has 1000 lines
  - How was the file changed over time from T-1 to T-101 ?
- VCS can answer this question easily
- Can also roll-back to point in time, e.g., T-99

VCS Types

- VCS come in various flavors that roughly align to three types of systems:
  - Local
  - Client-Server
  - Networked / Distributed

VCS Types : Local

- Source Code Control System (SCCS)
  - Developed at Bell Labs, 70's
  - Critical early stage use in development of UNIX
  - Part of the *Single UNIX Specification*
- File are locally version controlled
- Collaboration is limited to a single system
- Some VCS still use internals

VCS Types : Client-Server

- Client programs read and write changes to a development tree that exist on a server
  - Multiple developers can pull down to push up changes
- Concurrent Versions System (CVS)
  - Or: Concurrent Versioning System
  - 80's
- Subversion (SVN)
  - One of most popular today
  - Early 2000s

VCS Types : Distributed

- Decentralized VCS,
  - Built from the concepts of peer-to-peer trust
  - Each user has full repository in local storage
- GIT is most common
  - Developed by Linus Torvolds specifically for Linux Kernel development
  - 2005
  - GIT and other distributed VCS becoming the most popular VCS, close to SVN in usage

VCS Advantages

- Provides a control and tracking method to collaborative software development
  - Concepts can be applied to documents, e.g., non-software
- Changes (revisions)
  - Tracked by numerical or hash id
  - Timestamped
  - User stamped

VCS Advantages

- Version control is important for development groups to function effectively
- Also utilized for non-software development
  - Word-processing
  - Configuration files
  - Content management systems
  - Database records
-

VCS Terminology

- Trunk / Main / Master
  - the primary development branch
  - often the receiver of changes from other branches that are used for small development efforts, e.g., bug-fixes
- Branching
  - Duplication of a folder structure for the purpose of isolating development work from the Trunk

VCS Terminology

- Merge
  - Reconciling multiple changes to a version controlled resource
  - e.g., two versions of a file, the end result is one file with both sets of changes
- Fork
  - A branch that is not intended to be later merged
    - **NOTE, on GitHub, Forks are merged back using Pull Requests. More on this Later.
-

VCS Terminology

- Tag
  - A read-only branch that serves as the end-point of a development effort / interval, e.g., a release
  - Captures a branch at a point in time
  - Labels the point in time
- Commit
  - Saving a change to live files into the repository's set of know edits, i.e., revisions

VCS Terminology

- Baseline
  - The starting point of a branch
- Delta / Diff
  - A revision to one or more files or the file system (tree)
- Conflict
  - Two or more users have changed the same version controlled resource in a manner that cannot be automatically resolved by the VCS

VCS Terminology

- Head
  - The most recent / current / up-to-date version of a branch
- Update / Pull
  - Pulling in changes from other developers
- Working copy
  - The local working copy of files from the repository