

Software Engineering Assignment - 3

23 Jan 2018

Survey On Various Version Control Tools

Submitted by

A Aditya - 15IT201

INTRODUCTION - Version Control

Version control tools are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members. Version control systems (VCS) most commonly run as stand-alone applications, but revision control is also embedded in various types of software such as word processors and spreadsheets, collaborative web docs and in various content management systems, e.g., Wikipedia's page history.

1. GIT



Overview

Git was developed by Linux kernel creator Linus Torvalds. It is a distributed version control system. With a distributed version control system, there isn't one centralized code base to pull the code from. Different branches hold different parts of the code. Every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server. Git is primarily developed on Linux, although it also supports most major operating systems including BSD, Solaris, macOS, and Windows.

Pros

- Fast and efficient system.
- Distributed development.
- Compatibility with existing systems and protocols.
- Strong support for non-linear development.
- GitHub has recently helped establish Git as a great version control system, providing a beautiful front end for many large projects.

Cons

- Git is not as easy to learn as SVN or CVS is, which means that beginners may need to steer clear if they're not willing to invest time to learn the tool.
- If your project has non-text files that are updated frequently (images for websites or MS Office documents), then git becomes bloated and slow.

Basic Git Commands

- **git clone repository_url** : To clone a git repository onto the local system in the present working directory.
- **git status** : To compare your local repository copy with the git repository. Gives measure of difference in number of commits between local and global copy.
- **git add** : Add changes made on local copy to the staging area.
- **git commit** : Commit changes on the staging area with a unique name for the commit (using -m "commit_name")
- **git push origin branch_name** : Push the commits of the particular branch to the git repository.
- **git pull** : Pull the contents of git repository to local repository copy.
- **git branch branch_name** : Create a new branch for the repository.
- **git branch -d branch_name** : Delete a branch.
- **git fetch origin branch_name** : Fetch a branch from git repository to local repository copy.
- **git log** : To view the log of commits.

The version control tool I am going to use in my project is **Git**.

2. CVS (Concurrent Versions System)



Overview

CVS is the grandfather of revision control systems. It was first released in 1986, and Google Code still hosts the original Usenet post announcing CVS. CVS is the de facto standard and is installed virtually everywhere. However, the code base isn't as fully featured as SVN or other tools.

Pros

- CVS uses a client–server architecture.
- The learning curve isn't too steep for CVS, and it's a very simple system for making sure files and revisions are kept up to date.
- Excellent cross-platform support.
- CVS uses delta compression for efficient storage of different versions of the same file.

Cons

- CVS works more slowly.
- There is no integrity-checking for the source-code repository.
- CVS has very poor support for distributed source control.

3. SVN (Subversion)



Overview

SVN, or Subversion as it is sometimes called, is generally the version control system that has the widest adoption. Subversion was created by CollabNet Inc. in 2000. Most forms of open-source projects will use Subversion because many other large products such as Ruby, Python Apache, and more use it too.

Pros

- Because it is so popular, many different clients for Subversion are available.
- Commits as true atomic operations.
- Native support for binary files, with space-efficient binary-diff storage.

Cons

- Subversion does not store the modification times of files.
- Subversion uses a centralized revision control model.
- Subversion stores additional copies of data on the local machine, which can become an issue with very large projects or files.