

# Software Engineering - Assignment 3

IT350

## Brief Note On Different Version Control Software



Submitted to:

**Ms Raksha Nadgir**

Assistant Lecturer

Department of Information Technology

NITK Surathkal

23<sup>rd</sup> January 2018

Adwaith C D -15IT105

---

---

# Introduction

Version control systems are a category of software tools that help a software team manage changes to source code over time. They help in managing a collection of program code that provides you with three important capabilities: *reversibility*, *concurrency*, and *annotation*. The first VCSes were built as support technology for computer programmers, and that is still their primary use, but they can be used for Web content or any other sort of document or data file stored on a computer. Modern VCSes, in particular, fully support managing binary files such as images.

The most basic capability you get from a VCS is *reversibility*, the ability to back up to a saved, known-good state when you discover that some modification you did was a mistake or a bad idea. 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.

VCSes also support *concurrency*, the ability to have many people modifying the same collection of code or documents knowing that conflicting modifications can be detected and resolved.

Finally, VCSes give you the capability to *annotate* your data, attaching explanatory comments about the intention behind each change to it and a record of who was responsible for each change. Even for a programmer working solo, change histories are an important aid to memory; for a multi-person project they become a vitally important form of communication among developers. Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

---

# CVS

CVS -- or Concurrent Versioning System -- is a system for managing simultaneous development of files. It is in common use in large programming projects, and is also useful to system administrators, technical writers, and anyone who needs to manage files.

CVS stores files in a central repository, set (using standard Unix permissions) to be accessible to all users of the files. Commands are given to "check out" a copy of a file for development, and "commit" changes back to the repository. It also scans the files as they are moved to and from the repository, to prevent one person's work from overwriting another's.

This system ensures that a history of the file is retained, which is extremely useful when the boss decides he wants a feature you trashed months ago. It also ensures that backing up the repository is enough to backup a project (providing all necessary files are kept in repository).

CVS is designed for developers, either individually or in teams. For individuals, CVS provides a repository from which you can work from home, the office, or the client site without having to haul disks around. It also provides version control, allowing rollbacks without loss of data. For teams, it also keeps a record of who changed which lines of a file and prevents direct overwriting of each other's work.

## **Advantages of CVS:**

- Has been in use for many years and is considered mature technology

## **Disadvantages of CVS:**

- Moving or renaming files does not include a version update
- Security risks from symbolic links to files
- No atomic operation support, leading to source corruption
- Branch operations are expensive as it is not designed for long-term branching

---

# SVN

Subversion is an open source revision control system, similar in purpose to the well-known, widely deployed, aging and above-mentioned CVS. It is designed to provide state-of-the-art versioning, built from modern technologies. Subversion helps developers to track changes in their project source files.

Subversion has been designed and implemented as a collection of reusable libraries, written in C. This allows your programs to use the same functionality found in the command line Subversion client without having to call out to the command line client, to execute commands, or parsing output.

A *Subversion repository* acts as a filesystem that remembers sets of changes made to it. It does this by storing files in a tree structure, tracking its evolution over time. The repository increments a global revision number with every set of changes *committed* into the repository. As the whole tree is versioned, it acts as a regular filesystem. Copying and renaming files is possible; creating a project branch is as easy as copying a directory. Subversion can also produce a difference between two arbitrary revisions, or to check out some subtree at revision *N*.

## Advantages of SVN:

- Newer system based on CVS
- Includes atomic operations
- Cheaper branch operations
- Wide variety of plug-ins for IDEs
- Does not use peer-to-peer model

## Disadvantages of SVN:

- Still contains bugs relating to renaming files and directories
- Insufficient repository management commands
- Slower comparative speed

---

# GIT

Git is considered to be a newer, and faster emerging star when it comes to version control systems. First developed by the creator of Linux kernel, Linus Torvalds, Git has begun to take the community for web development and system administration by storm, offering a largely different form of control. Git offers a much different type of version control in that it's 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. Here, there is no singular centralized code base that the code can be pulled from, and different branches are responsible for hosting different areas of the code. Other version control systems, such as SVN and CVS, use centralized version control, meaning that only one master copy of the software is used. Since Git is a fast and efficient system, many system administrators and open-source projects use Git to power their repositories. However, it is worth noting that 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.

## Advantages of GIT:

- Dramatic increase in operation speed
- Cheap branch operations
- Full history tree available offline
- Distributed, peer-to-peer model

## Disadvantages of GIT:

- Learning curve for those used to SVN
- Not optimal for single developers
- Limited Windows support compared to Linux

---

# Mercurial

It was designed initially as a source for larger development programs, often outside of the scope of most system admins, independent web developers and designers. However, this doesn't mean that smaller teams and individuals can't use it. Aside from being very scalable, and incredibly fast, Mercurial is a far simpler system to use than things such as Git, which one of the reasons why certain system admins and developers use it. Mercurial "branches" are more like labels, or tags, which is why we can't delete them, or rename them; they are stored forever in posterity just like the commit message. The only way to achieve this is to learn and enable the patch-queuing system and use the hg strip command, or install the local-branches extension. But again they are put to .hg/strip-backup directory but are never deleted by Mercurial.

## Advantages of Mercurial:

- Safer history with mercurial
- Several third party GUIs are available for those who prefer graphical interfaces
- Great windows support
- Easier to learn than git

## Disadvantages of Mercurial:

- Add-ons must be written in python
- No partial checkout- which is a big limitation to large projects
- Combining features and functionality is problematic when using different extensions