B N Yashaswini(16IT207)

VERSION CONTROL TOOLS

Version Control System (VCS) is a software that helps software developers to work together and maintain a complete history of their work. Version control is a way to keep a track of the changes in the code so that if something goes wrong, we can make comparisons in different code versions and revert to any previous version that we want. It is very much required where multiple developers are continuously working on or changing the source code.

CVS:

CVS is a version control system using which we can record the history of sources files, and documents. If several teams want to maintain their own version of the files, because of location or policy CVS's vender branch can import a version from another team even if they don't use CVS, and then can merge the changes from the vendor branch with the latest files . In CVS single command can manipulate the entire collection. Unreserved checkouts provide more than one developer to work on the same files at the same time. CVS does not allow atomic commits , renaming and moving, propagating changes to parent repositories.

SVS(Apache Subversion):

Subversion is a also a centralized version control tool which uses central server to store all files and enables team collaboration. It tracks changes to files, folders and directories.Simple and easy to learn .Less complex when compared with other version control tools.Better GUI than Git. It provides good remote options for http/https server and allows atomic commits,renaming and directory versioning. Branches and Tags are just directories that adhere to a convention. Subversion's use of convention rather than configuration means that we need to think about our repositories structure in advance and ensure that everyone adheres to it.Merging and mirroring were almost non-existent. It Does not work much effectively when offline.

Git:

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is mainly used for source-code management in software development, but it can also be used to keep

track of changes in any files. Git is aimed at speed, data integrity, and support for distributed, non-linear workflows. It is very flexible and user-friendly. It is a distributed model and makes it possibility of working offline.Branching and merging are easy and the workflow is flexible. Because git uses SHA1 trees, data corruption due to external reasons can be easily detected.It allows Data redundancy and replication and there is Only one git directory per repository.If our project has a non-text or binary files which are updated frequently then git becomes slow.Its very hard to remember all the commands while using Command line interface. Git does not support 'commits' across multiple branches or tags.

Mercurial:

Mercurial is a free, distributed source code version control tool.It efficiently handles projects of any size and offers an easy and intuitive interface.It is supported on Microsoft Windows and Unix-like systems, such as FreeBSD, macOS and Linux. It has advanced branching and merging capabilities. With its fully distributed collaborative development it handles both plain text and binary files robustly.Easier to learn than other version control systems.It is Good at interface ,easy to use and its history is read only.It has less functionality than other version control systems and less reliable. Mercurial also comes alongside a web-interface and various extensive documentation that can helps us to understand it better.

As I know about git a bit more than the other tools I am comfortable with it and I prefer git to the others as it is super-fast and efficient, also code changes can be very easily and clearly tracked.