**Name:** Chetana Shivkiran Patil

**PRN:** 2020BTECS00043

**Batch:** T6

**SET Assignment 5**

**Q 1. What is version control system and why it is important?**

**Ans:** A version control system (VCS) is a software tool that helps track and manage changes to files and code in a software development project. It allows multiple developers to work on the same codebase concurrently, keeps a history of changes made to files over time, and provides mechanisms to collaborate, merge, and resolve conflicts when changes overlap.

Version control systems are important in software development for several reasons:

1. **Collaboration**: VCS enables multiple developers to work on the same codebase simultaneously, and keeps track of changes made by each individual. This facilitates collaboration among team members and allows them to work independently on different features or bug fixes without interfering with each other's work.

2**. History and Recovery**: VCS maintains a complete history of changes made to a project, including who made the changes and when. This allows developers to easily revert to a previous version of the code in case of errors or issues, and provides a safety net for recovery.

**3.** **Branching and Merging:** VCS allows developers to create branches, which are independent copies of the codebase. This allows for parallel development of different features or bug fixes, and facilitates the process of merging changes back into the main codebase.

**4. Code Review:** VCS provides mechanisms for reviewing and discussing changes made by team members before they are merged into the main codebase. This helps ensure code quality, maintain consistency, and catch errors early in the development process.

**5. Auditing and Compliance:** VCS provides a traceable record of changes made to a project, which can be useful for auditing purposes and ensuring compliance with coding standards, regulations, and policies.

**6. Collaboration across Distributed Teams:** VCS enables collaboration among team members who may be geographically dispersed, allowing them to work on the same project from different locations and time zones.

Overall, version control systems are essential tools in modern software development that help teams collaborate effectively, maintain code quality, and manage changes to codebases over time.

**Q 2. Illustrate different types of version control system with example.**

**Ans:** Version control system (VCS) is a software tool used to manage changes to source code or other files over time. There are two main types of VCS: centralized and decentralized.

1. Centralized Version Control System (CVCS):

In a CVCS, there is a single central repository that stores all versions of the code, and all users check out the code from that central repository. Changes made by one user are immediately visible to all other users.

Example: SVN (Apache Subversion)

2. Decentralized Version Control System (DVCS):

In a DVCS, each user has their own copy of the code repository, including the complete history of changes. Users can work independently and make changes to their local copy, and then push those changes to other users' copies or the central repository.

Examples: Git and Mercurial

Both CVCS and DVCS have their own advantages and disadvantages. CVCS is easier to manage and control, but it requires a network connection to access the central repository. In contrast, DVCS is more flexible and allows for offline work, but it can be more complicated to manage and can lead to merge conflicts if users are not careful.

**Q 4. Differentiate Between The Git & SVN Repository?**

**Ans:**

Git and SVN are two popular version control systems that allow developers to manage changes to their code over time. Here are some of the key differences between Git and SVN:

1. Distributed vs. Centralized:

Git is a distributed version control system, meaning that each developer has a complete copy of the repository on their local machine. This makes it easy for developers to work offline, and allows for more flexible workflows. SVN, on the other hand, is a centralized version control system, where there is a single central repository that stores all versions of the code.

2. Branching and Merging:

Git has a powerful branching and merging system that allows developers to create and merge branches easily. Branches are cheap and easy to create in Git, and merging is often automated. SVN also supports branching and merging, but it can be more difficult to manage and can lead to merge conflicts.

3. Speed and Performance:

Git is generally faster and more efficient than SVN, particularly for large projects with many files and revisions. This is because Git stores the entire history of the repository locally, whereas SVN must communicate with the central repository to access historical versions of files.

4. Learning Curve:

Git has a steeper learning curve than SVN, particularly for developers who are new to distributed version control systems. However, once developers become familiar with Git, it is often preferred due to its flexibility and power.

Overall, Git and SVN have their own strengths and weaknesses, and the choice of which to use depends on the specific needs of the project and the preferences of the development team.

**Q 6. How CVS is different from SVN?**