**Software Engineering Tools Lab Assignment No-6**

(Module 4- Version control) **Due date-04/04/2022**

Q 1. What is Microsoft’s VSS? Provide the information of VSS tool with respect to below points.

* Owner/developer: Microsoft Corporation
* Brief information/introduction: Microsoft's VSS (Visual SourceSafe) is a version control system designed for software developers who work on Windows-based platforms. It is a centralized version control system that allows developers to manage and track changes to code over time, and provides a secure, centralized repository for code storage.
* Basic operations involved: Some of the basic operations involved in VSS include:
* Adding files to the repository
* Checking out files for editing
* Checking in changes to the repository
* Branching and merging code changes
* Resolving conflicts between code changes
* Rollback of changes to previous versions
* Advantages:
* VSS is designed to be easy to use and integrates well with the Microsoft Visual Studio IDE.
* It provides a secure, centralized repository for code storage and version control, making it easy for developers to collaborate and work on code together.
* VSS provides features such as branching and merging that allow developers to work on different versions of code simultaneously and merge changes together.
* It allows for rollback of changes to previous versions of code, which can be useful in case of bugs or errors in the code.
* Disadvantages:
* VSS is a centralized version control system, which means that it requires a central server to store code and manage version control. This can make it less flexible than distributed version control systems like Git.
* VSS has been known to have issues with stability and data corruption, which can cause problems for developers using the system.
* The software is no longer actively maintained by Microsoft and has been replaced by newer version control systems like Git and Azure DevOps.

Q 2. Create a SVN repository and perform below operations on that repository using SVN. Also explain below operations.

1. **Revert**

svn revert /path/to/file

1. **Import**

svn import /path/to/local/folder file:///path/to/repository/trunk -m "Importing new files"

1. **Checkout**

svn checkout svn://server/path/to/repository

1. **Commit**

svn commit /path/to/file -m "Commit message"

1. **Update**

svn update /path/to/local/folder

1. **Copy**

svn copy /path/to/source/file /path/to/destination/file -m "Copy message"

Q 3. Perform below operations using CVS

1. **cvs checkout**

cvs checkout project-name

cvs checkout myproject

1. **cvs update**

cd myproject

cvs update

1. **cvs add**

cd myproject

cvs add newfile.txt

1. **cvs remove**

cd myproject

cvs remove oldfile.txt

1. **cvs commit**

cd myproject

cvs commit -m "Added newfile.txt and removed oldfile.txt"

Q 4. Differentiate Between The Git & SVN Repository?

1. Distributed vs Centralized Repository: Git is a distributed version control system, meaning that each developer has a complete copy of the repository on their own machine, and changes can be made and committed locally. SVN, on the other hand, is a centralized version control system, where there is a single central repository that stores the entire version history of a project.
2. Branching and Merging: Git makes branching and merging changes between branches a simple process, allowing developers to work on multiple features or bug fixes simultaneously and merge changes back into the main branch when they are ready. SVN also supports branching and merging, but it can be more complex and requires more coordination among developers.
3. Speed: Git is generally faster than SVN, especially when it comes to large repositories or frequent branching and merging.
4. File Renaming and Moving: Git tracks files based on content, which means that renaming or moving a file within a repository is a simple operation. In SVN, however, renaming or moving a file can require copying and deleting, which can be more time-consuming and error-prone.
5. Conflict Resolution: Git has more sophisticated conflict resolution capabilities than SVN, which can make it easier for developers to resolve conflicts that arise when multiple developers make changes to the same code.

Q 5. What is “branch”, “tag” And “trunk” In SVN?

1. Branch: A branch is a copy of the codebase that is created to allow for parallel development. Branches can be used to work on new features, bug fixes, or other changes without affecting the main codebase. Once the changes on a branch are complete and tested, they can be merged back into the main codebase.
2. Tag: A tag is a way of marking a specific point in the codebase, typically a release or a significant milestone. Tags are useful for creating snapshots of the code at a particular point in time, and they can be used to revert to a specific version of the code if necessary.
3. Trunk: The trunk is the main line of development in the codebase, also known as the "master" branch. All development work is typically done on branches, and changes are merged back into the trunk when they are complete and tested.

* The trunk is the main line of development, where the most up-to-date and stable version of the code is stored.
* Branches are created off the trunk to work on specific features or changes. Developers can work on the branch without affecting the trunk, and changes can be merged back into the trunk when they are complete.
* Tags are created to mark specific points in the codebase, such as a release or a milestone.

Q 6. How CVS is different from SVN?

* Centralized vs. Distributed: CVS is a centralized version control system, which means that there is a single central repository that stores the entire version history of the project. SVN, on the other hand, supports both centralized and distributed workflows, allowing users to work with a local copy of the repository.
* Branching and Merging: SVN has better support for branching and merging than CVS. In SVN, branches are cheap and easy to create, and merging changes between branches is generally straightforward. In contrast, CVS branches can be more complex to manage and merge, which can make it more difficult for developers to work on multiple features simultaneously.
* File Renaming and Moving: SVN tracks file moves and renames more efficiently than CVS, making it easier to track changes to files and directories.
* Speed: SVN is generally faster than CVS, particularly for larger repositories.
* Repository Backup: CVS requires additional software to perform backups, while SVN has built-in backup functionality.