**What is VCS/SCM/RCS?**

Version Control System (VCS). Also known as Source Code Management (SCM) or Revision Control System (RCS).

**Define:** VCS is basically a "software configuration management" that manages all the changes done in codebase or any document or file.

**Types of VCS:**

**1) CENTRALIZED VCS**: Centralized repository is checked out by every developer in their local system known as working copy.

Drawback: As the central server stores all the files, so if the disk of the central server gets corrupted, the entire history of the project is lost.

**2) DISTRIBUTED VCS**: In this, the developer clones the main repository on their local system and then works on it. The developer can also create the another branch or can directly work on local repository. This branch in which developer makes the changes is the working copy.(refer diagram module 1)

. The developer makes/update the changes in the WORKING COPY and the COMMITS the code to the LOCAL REPOSITORY.

. Then the developer can push the code from LOCAL REPOSITORY to the REMOTE REPOSITORY/CENTRAL SERVER.

. Now after the changes that are made is updated in the central server/ remote repository, from there the developers can pull these changes to update their personal repositories.

Advantage: Even if the central server gets corrupted, the data or files are not lost. Due to cloning in the local system and branching, the copy of the files exists in the other repositories that the developers have.

. Also enables the offline COMMIT, BRANCHES CREATION and so on.

**What is GIT?**

. Moat Popular VCS tool.

. Distributed VCS.

. Centralised VCS.

. Open source project – Developed in 2005.

. Supports many OS and IDE’s.

. Emphasis on Speed & Data Integrity. Also, customized for performance.

. Focuses on file content instead of name.\

. As most operations performed on the GIT are LOCAL, which states that: it does not require network connection for most of the commands, making it very fast.

. It’s free.

Why we need GIT**:** Because large software projects require a VCS to keep track of all the changes made to the codebase in order to keep the track of the changes. Understand the need ogf GIT based on WHO, WHAT and HOW.

. Who edited a certain file?

. What they changed?

. How to get back to the original code if necessary?

What GIT can do?

. Helps recovering the code, reverting the changes, sharing of code with other developers through central server, keeping the track or history of changes made to the codebase and allows to keep the backup of your code.

Benefits of Git: Fully Distributed, Simple Design, Speed as it is faster because of its distributed nature, thus no load on single server, Non-Linear development support as team can work on multiple branches parallel, Efficiency as it is able to handle large projects.

**GIT BRANCHING CONCEPTS:**

Branch Terminolgy:

. Commit: Recorded changes in history.

. Branch: A branch in GIT is simply a lightweight movable pointer to one of the commits.

. Checkout: It is a command to create a new branch or switch to existing branch.

. Master: Default branch name when new project is added to GIT repository.

. Push: This command is used to push changes from Local Repository to Remote Repository/Central Server.

. Pull: This command is used to pull changes from Remote Repository to Local Repositiries.

. Conflicts: When 2 or more people make the changes in the same file simultaneously, conflicts occur.

. Reset: This command is used to set branch to any Commit Point.

**COMPARING GIT WITH OTHER TOOLS:**

Other VCS Tools are:

. CVS(Concurrent Version): Free-client server VCS.

. Mercurial: It is a Distributed VCS/RCS.

. SVN(Subversion): It is an Open-Source VCS under Apache License.

Overall, GIT is better in every aspect as compared to SVN/CVS.

**WEB SCALE ARCHITECTURE:**

**GIT VIRTUAL FILE SYSTEM (GVFS):**