SOURCE CODE MANAGEMENT

INTRODUCTION

- Source code management is used to track modifications/revisions to a source code repository.
- Each revision is given a timestamp and includes the name of the person who is responsible for the change.
- In a distributed model, developers work directly with a local repository, where their individual source code revisions are collected.
- The developers then update the central repository with their local revisions so that revisions can be shared among developers.

FEATURES OF SCM

- Authenticated access for commits
- Revision history on files
- Atomic commits of multiple files
- Versioning/Tagging

WHY SCM?

- Big projects need a version control system to track the changes and avoid misunderstanding.

Functions of a good SCM:

- **Backup and Restore** – Files can be saved at any moment and can be restored from the last saved.
- **Synchronization** – Programmers can get the latest code and fetch the up-to-date codes from the repository.
- **Short-Term Undo** – We can do a short-term undo to the last known version.
- **Long-Term Undo** – It helps when we have to make a release version rollback.

- **Track Changes** – We can track the changes as when anyone is making any change, he can leave a commit message as for why the change was done.
- **Ownership** – With every commit made to the master branch, it will ask the owner permission to merge it.
- **Branching and Merging** – You can create a branch of your source code and create the changes. Once the changes are approved, you can merge it with the master branch.

TYPES OF VERSION CONTROL SYSTEMS

CENTRALIZED VERSION CONTROL

- The main concept of Centralized Version Control is that it works in a client and server relationship.
- The repository is located in one place and allows access to multiple clients.

Benefits:

- More powerful and easy change tracking.
- No need of a centralized server. Most of the functionalities work in offline mode also apart from sharing the repositories.
- Branching and Merging strategies are more easy and reliable.
- It's faster than the other one.

Drawbacks:

- It is harder to understand.
- It's new, so less GUI clients.
- It is easier to make mistakes until you are familiar with the model.

DISTRIBUTED VERSION CONTROL

- In Distributed Version Control, each user has their own copy of the entire repository as well as the files and history.

Benefits:

- More powerful and easy change tracking.
- No need of a centralized server. Most of the functionalities work in offline mode also apart from sharing the repositories.
- Branching and Merging strategies are easier and more reliable.
- It's faster than the other one.

Drawbacks:

- It is harder to understand.
- It's new, so less GUI clients.
- It is easier to make mistakes until you are familiar with the mod

SCM TOOLS

- Github
- GitLab
- BitBucket
- SourceForge
- Beanstalk
- Apache Allura
- AWS CodeCommit
- Launchpad
- Phabricator
- GitBucket