Course 2 Version Control

Wednesday, 2 July 2025 12:24 PM

Module by Dhaanesh S

MODULE 1 SOFTWARE COLLABORATION

Objective : This course provides us knowledge of how modern software developers interact around the world to work on a particular project without messing up each other's code.

- How modern software teams collaborate and work on the same codebase
- List different version control systems
- Illustrate a standard software development workflow.

We will be introduced into different linux commands to interact with files present on the hard drive and create powerful workflows that automate your workflows that saves your time and energy.

- **Version control is system that keep tracks of all the changes and modifications to file for tracking purposes also called Source control.**

- Teams can know create, edit and perform deletion of files, and version control shows when and who made the changes as an important feature.

Revision History - It has the whole modification and edit history that as been done to a particular project, so that developers can roll back to what changes they made and make a useful key takeaway.

Peer Review - code inspection and code reviewed by other developers and provide feedback.

Systems of version control and tools

Different types of version controls

1. Subversion
2. Perforce
3. AWS code commit
4. Mercurial
5. Git

1. **Centralized version control system** : This has a centralized server and different users who are connected to it. The CVCS holds the copy of code which the users can pull and make changes and needs to push the code back to the server with the changes made so that other developers can see.

   Easier to learn (adv)
   Slower to establish a connection (disadv)

2. **Distributed version control system** : It is similar model as compared to CVCS but in this system when the user pulls the code from the centralized server, the user acts as the entire server who has whole history of changes made with them locally in their computer.

   No need to connect(users can work in offline state)
   DVCS is mentioned more in this course

Version control in professional software development

Version control must be complemented with other tools to maintain efficiency and integrity in delivering a quality software product. Take a situation when two engineers are editing a file and pushing the code, now it would be a conflict if they do it the same time. And other situation where a junior joins your team and you are working on a critical project in which junior cant make direct changes to the code, as its critical so a peer review system as been introduced in which other developers are necessary to review the code changes before pushing them to version control system. To resolve this type of issues we use workflows that manages changes and helps reduce mistakes from happening.

CI/CD

A history of revisions : It will help to keep track of all the changes and modification made from the start of the project till the end, so that developers can easily identify the issues solved, code changes, and helps to collaborate with fellow developers to make the process of workflow smooth.

Stagging vs Production

1. Development Environment : Every development team prior to releasing their new features or changes made needs to verify the code they do release is right or not. So teams wish to have different development environments to test and verify. In order to achieve this they normally set up QA, UAT, and other environments with stagging environments. The main purpose of this flow is to identify any potential error that might arise due to changes being made to codebase.

2. Stagging Environment : This allows QA, stakeholders to see how the new feature works or use those as pre trial. Stagging should also cover all areas of architecture of database including database and any other service that may be required.

3. Testing : Unit, Integration and performance testing.

4. Production : It is live, the application is out their for people to try out.