GIT

Table of contents

|  |  |  |
| --- | --- | --- |
| S No | Subject | Page No |
| 1 | What is version control | 2 |
| 2 | Types of version control | 2 |
| 3 | Source Code Management | 3 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**What is version control?**

Version control is also called source control. For developers source code is the crystal ball, it is my precious. So, to make any changes or check changes made or to experiment on code we use VCS.

Say we don’t have VCS or not using VCS then

1) You want to undo the changes made, then you need to remember it or write it down somewhere. {suppose it is a change done months ago}

2) You will also not have a history of all the changes made.

3) Say multiple developers are working on the same code then it will be troublesome.

4) Say multiple developers are working on code but one change is compatible with other change the developer making then, using VCS we can track that down.

To prevent all these problems, we are using VCS.

A few advantages are

1. Branching and merging: allows you to create a test sample code, then make changes without affecting the main source code, then do testing and everything and if it works out then merge it to the main source code. This practice is adopted by all developers.
2. A lifetime history of all the changes made to code.
3. Traceability

**Types of version control**

There are mainly two types of version control, Centralized VCS (CVCS) and Distributed VCS (DVCS).

In CVCS all developers are connected to a central server, then make changes so it needs to remain connected to the server and do the thing.

In DVCS all the history and source code are copied to developer’s system, so it will be faster because load is on individual system unlike central one and can also work offline. Example is git.

**Source Code Management**

It is synchronous to VCS. It tracks all the changes made to the source code repository. Say two developers are working on same source code but on different features, and one’s code causes the crash in other guy’s code. So, to prevent this a code changes tracking system is required then. Also, SCM blocks the changes or gives warning to prevent this.

SCM brought version control safeguards to prevent loss of work due to conflict overwriting. These safeguards work by tracking changes from each individual developer and identifying areas of conflict and preventing overwrites. SCM will then communicate these points of conflict back to the developers so that they can safely review and address.

**Git Repository**

It is like storage in PC but online, companies save their source code there and make changes to it with benefits of VCS (branching, commit etc.). Also called repo sometimes.

**How to initialize a repository or clone it**

You can go to GitHub, GitLab or any VCS and create a repository there, the process is very simple. Then to bring it to your system you can clone it using command

git clone

Another way of creating a repo only in your system is

1. Select any folder in your directory
2. Git bash there
3. Type the code git init
4. This will create a repo in your local system

All repo’s, even if they are remote or local, work the same. What I mean is when you create a local one, then add file, then commit files do operations on them all are same, but not necessarily updated with each other.

Once you clone a repo then all commit history, all files all code will be there then if you work on master branch that is the main branch as long as you don’t push those changes will not be affected. It may be a bit confusing now but you will understand it as we move on main thing is all repo works same.

**How to reach the folder where you want to do git ops?**

Just right click over the directory you want go and git bash here.

**Adding, commit and pushing files to repo**

1. Put the file in the directory cloned or other way around, but this will not add it as a part of repo.
2. You need to write a code git add filename.extension in git bash.
3. Now you have added it but in git adding a file or doing modification to a file is a two-step process.
4. First is staging, which you did just now, then commit.
5. Before commit check status using code git status.
6. To commit use code git commit –m"message”.
7. In the message section provide a comment on why this commit or if you are working in company then Jira ticket number or as per company instruction, it is considered as a good practice because if you want to look back your comment will give you the reason or anything related to it.
8. After staging and committing, we push the code to the remote repo using command git push.
9. Then it will ask you for your password to git account then done. File added to repo.
10. You can check log using git log command.