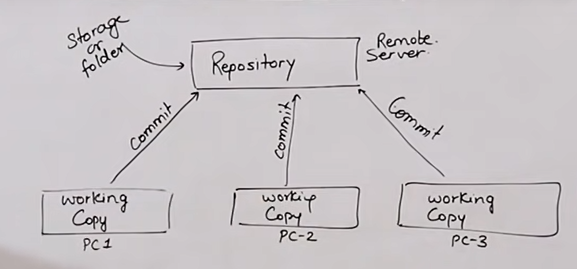
**Source Code Management or Software Configuration Management**

**---> Centralised Version Control System (CVCS)**

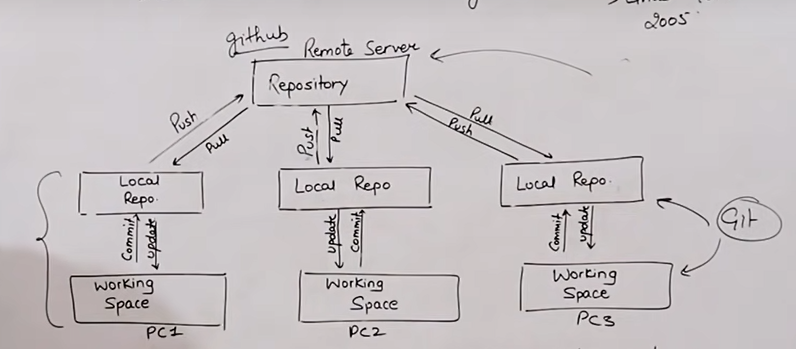
* Only one repository/folder/remote server.
* One can write its code and commit it in the repository/folder and can also update the code later.
* Can view and access other persons code/work so as to:
  + Check compatibility before writing the code.
  + Make sure that the module which is already coded is not coded again.
  + Can continue from where the other one left.
* Drawbacks
  + If the central server fails, data will be lost.
  + Accessing the server to commit or update needs internet access.
  + Downloading and Uploading used to take time/ was slow.
  + Nothing is saved locally.



**---> Distributed Version Control System Eg Git**

#Who made git : Linus Torvald (Same person who made linux) in 2005.

* Every contributor has a local copy or “clone” of the main repository.
* Local Repository contains all the files and metadata present in the main repo, and hence is not dependent on the main repo.
* For accessing local copy we don’t need internet access, so commit or updating is fast
* Every changes/work is done in the local repository only which is in the system itself.
* Main repo can be Github or Mercurial or bitbucket.

****

CVCS

1) In CVCS, a client need to get local copy of source, do the changes and commit those changes to central source on server.

2) CVCS system are easy to learn and setup.

3) Working on branches is difficult in CVCS. Developer often face merge conflict.

4) CVCS system do not provide offline access 5) CVCS is slower as every command need to communicate with server.

6) If CVCS server is down, developers cannot work

DVCS

1) In DVCS, each client can have a local repo which has complete history on it. Client need to push the changes to branch which will then be pushed to server repo.

2) DVCS systems are difficult for beginners as multiple commands needs to be remembered.

3) Working on branches is easy in DVCS. Developers face less conflicts.

4) DVCS systems work fine in offline mode as client copies the entire repo on their local machines.

5) DVCS is faster as mostly user deals with local copy without hitting server anytime.

6) If DVCS server is down, developers can work using their local copies