

1A) Software technologies called version control systems assist software development teams in tracking changes to source code over time. Version control systems enable software teams to operate more swiftly and intelligently as development environments have increased. They are especially helpful for DevOps teams because they enable them to speed up successful deployments and cut down on development time.

Every change to the code is recorded by version control software in a particular form of database. If a mistake is made, programmers can go back in time and review prior iterations of the code to help fix it while causing the least amount of interruption to the entire team.

2A) Multiple developers, designers, and team members can collaborate on the same project thanks to version control systems. It aids in making them more productive and quick! To guarantee that everyone has access to the most recent code and that updates are recorded, a version control system is essential. The requirement to handle numerous versions and components of whole products grows as development becomes more complicated and teams expand.

3A) i) centralized version control systems: Software development teams have an option for using a central server for collaboration using a centralised version control system. A server serves as the main repository in a centralised version control system (CVCS), where each version of the code is kept on file.

ii) distributed version control system: The whole source, together with its complete version history, is replicated on each developer's workstation in a distributed version control system (DVCS). It is referred to as DVCS. File modifications are monitored across computers.

4A) Because developers clone repositories on their distributed version control workstations, making numerous backup copies, distributed version control does not have a single point of failure like a centralised version control system.

In distributed version control system everyone can act as the 'server' unlike central control system where there is only one server.

5A) Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.

6A) i) tracking history
ii) open source
iii) scalable
iv) easy branching

7A) add . — to update the added data

Commit---- to commit the changes made displays the message given by user
Clone-----clones the given repository link

8A) While GitHub is a location where copies of a Git repository can be uploaded, Git is a programme used to manage multiple versions of source code updates that are then transferred to files in a Git repository. There is no comparison between git and github.

9A) `git --version`

10A) `git add .`

11A) Committed snapshots are displayed by the `git log` command. You may list, filter, and search the project history for certain modifications. `git log` only functions on the committed history, but `git status` allows you to check the working directory and the staging area.

12A) `git init`

13A) Your files may be in one of three states in Git: edited, staged, or committed.

Modified refers to the fact that you have made changes to the file without yet committing them to your database.

Staged refers to the designation of a modified file in its current state for inclusion in your subsequent commit snapshot.

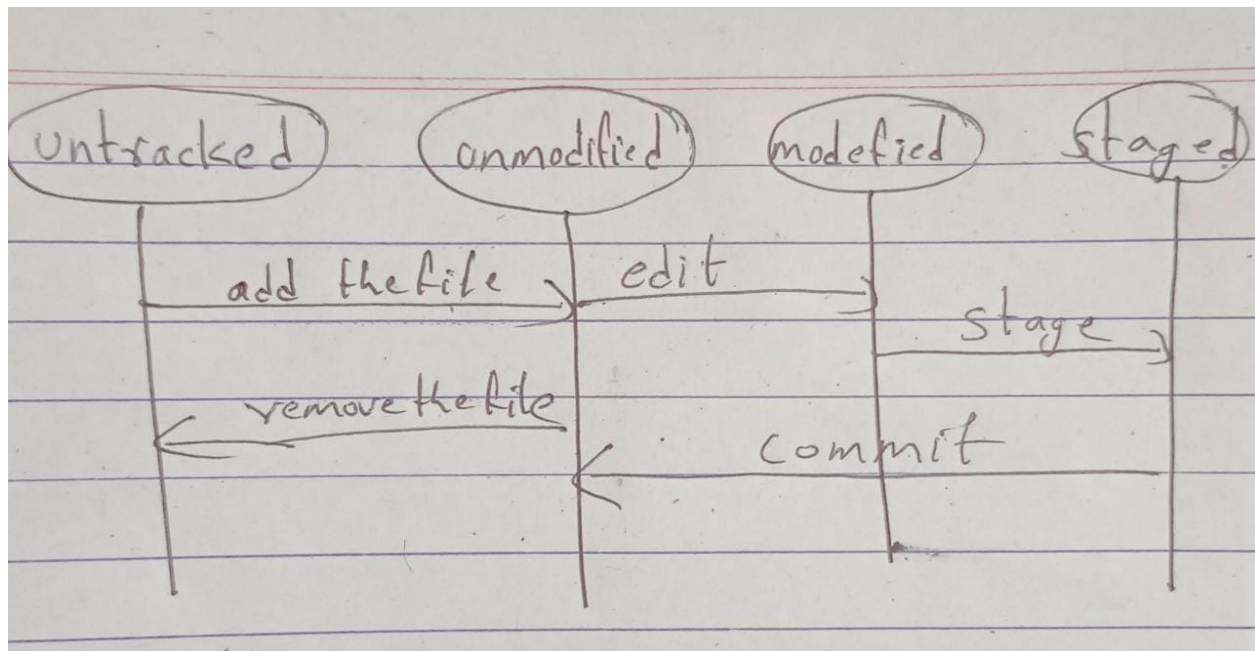
Data that has been committed to storage in your local database is kept there securely.

14A) -

15A) `git commit -m "user given statement"`

16A) `git commit -m "New email"`

17A)



18A) An independent line of development is represented by a branch. Branches represent the edit/stage/commit process abstractly. They can be viewed as a means to ask for a fresh working directory, staging area, and project history.

19A) `git branch 'new-email'`

20A) `git checkout new-email`

21A) `git switch -c "new-branch"`

22A) `git init` creates or initializes git on the current repository.

23A) forking is making a copy of a repository which can be managed by the by us i.e edit or change according to our wish. Whereas cloning makes a local copy which is not our own copy.
To clone a repo
`Git clone 'repo link'`

24A) push is to add any data or changes made in a repo which is clones using git

`Git push -u origin 'branch name'`