SDLC:

s/w used by s/w development teams to design develop test and develop high quality s/w.

**GIT** is distributed version control system that tracks the versions of files, It is often used to control source code by programmers who are developing s/w collaboratively.

**GitHub** is s/w dev platform. It is used for storing, tracking & collaborating on s/w projects.

Version Control: it is

1. Bugfix
2. Minor
3. Major

**GitHub hierarchy:** Enterprise 🡪Organizations 🡪 Repositories

**Basic Commands**

$ git clone https://github.com/Demo-org1-2024/Demo-repo.git

$ git config –list

$ git config --global user.name "Samarth"

$ git config --global user.email "samarthm@devtools.in"

**SDLC:**

1. Requirements gatherings: JIRA, Trello, Microsoft, Asana
2. System analysis: Confluence, Track & rally, Click up
3. Design: Draw.io, Adobe XD, Sketch
4. Implementation: Coding phase, Git, GitHub, Jenkins
5. Testing: Junit, Selenium, Postman, JMeter
6. Deployment: Docker, Kubernetes, Ansible, Terraform
7. Maintenance: Nagios, Prometheus, Dynatrace

**Version control**

* DVCS (Distributed Version Control Systems)
* CVCS (Centralized Version Control Systems)

$ git clone <https://github.com/Demo-org1-2024/Demo-repo.git>

$ git config

$ git config –list

$ git config --global user.name "Samarth"

$ git config –list

$ mkdir test-repo

$ cd test-repo/

$ git init

$ ls -la

$ vi index.html

$ git add index.html

$ git status

$ git commit -m "created index.html file"

$ git log

$ git branch

$ git branch development

$ git branch

$ git checkout development

$ vi index.html

$ git status

$ git add –all

$ git status

$ git add –all

$ git status\

$ git log

$ git checkout master

$ git merge development

$ git log

Git is Command line tool used for version control systems, GitHub is a web based platform for hosting & collaboration on git repositories

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

git add 🡪 git commit 🡪 git push