Git &GitHub

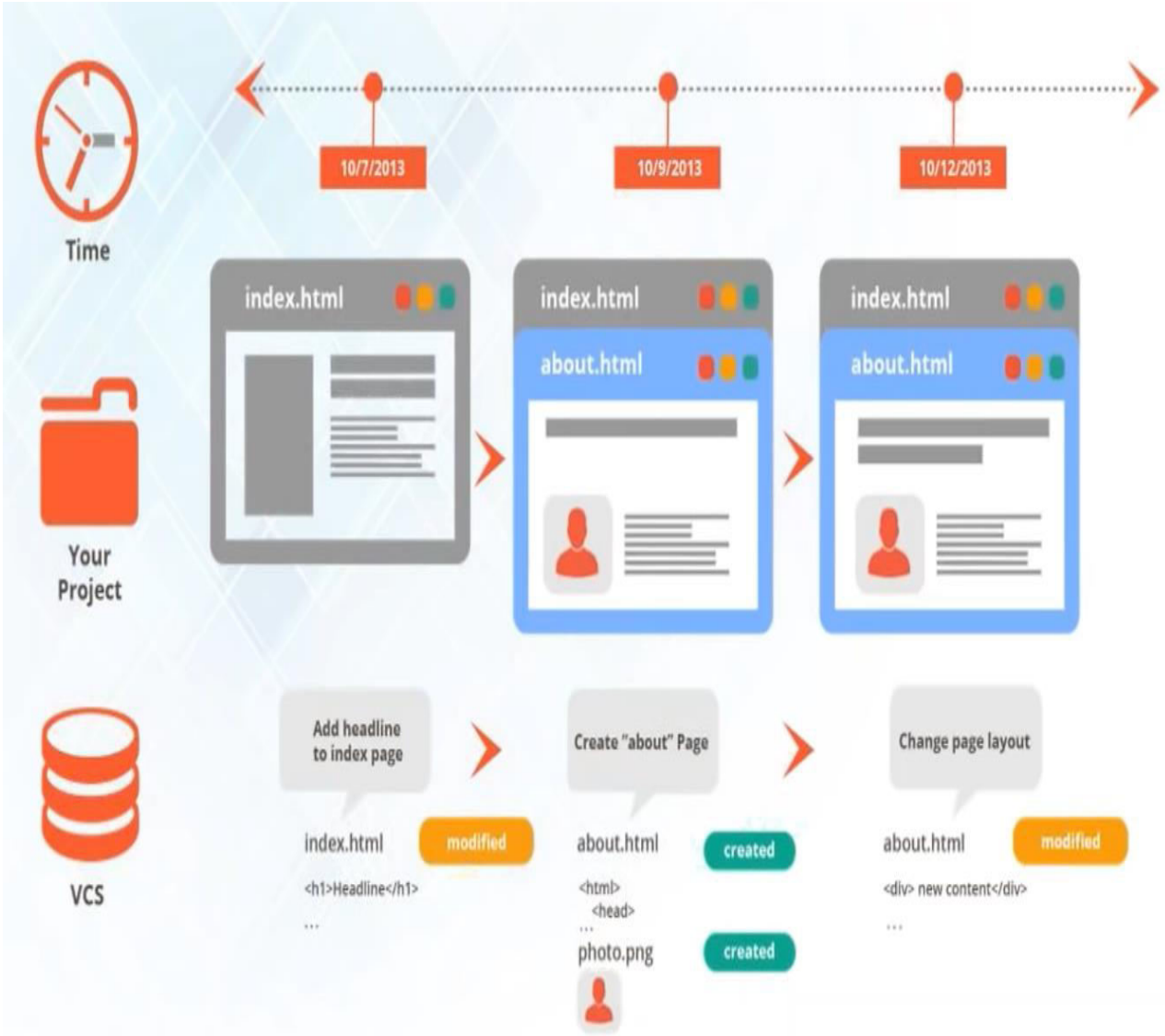
by

Girish Godbole

Git & GitHub

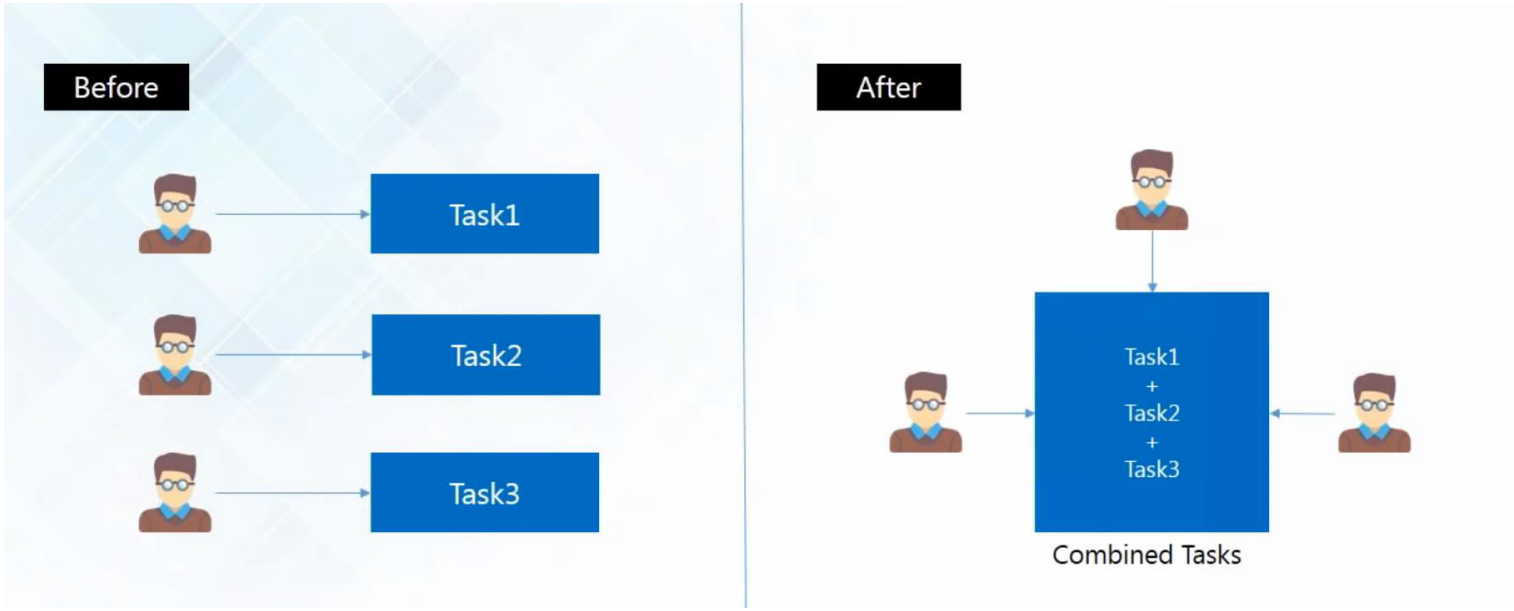
- Version Control – What & Why?
- Version Control Tools
- GitHub & Git
- Case Study: Dominion Enterprises
- Git Features
- Git Operations & Commands

Git & GitHub



- Version control is the management of changes to documents, computer programs, large web sites, and other collections of information.
- These changes are usually termed as "versions".

Git & GitHub

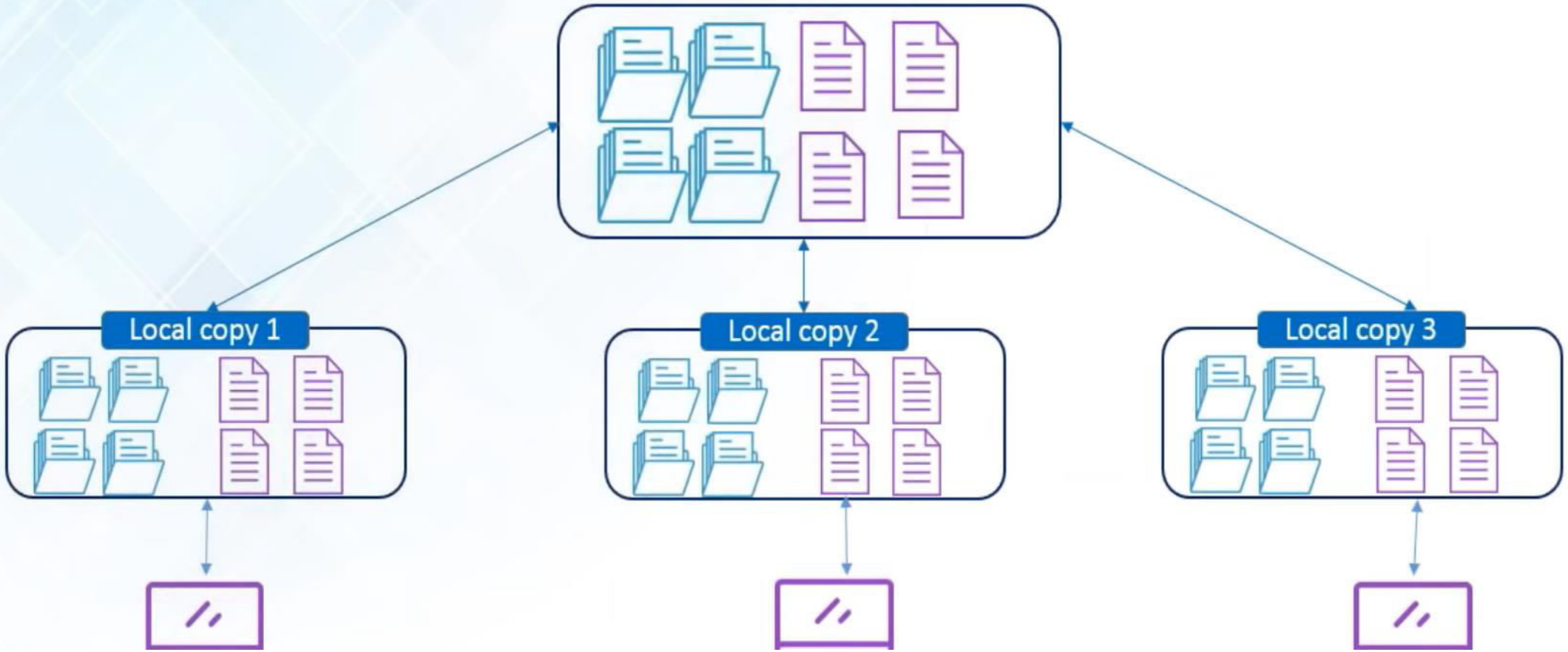

Before
Task1
Task2
Task3
After
Task1
+
Task2
+
Task3
Combined Tasks

Git & GitHub

- Snapshots of all versions are properly documented and stored.
- Versions are also named accurately.

Git & GitHub

In any case if your central server crashes, a backup is always available in your local servers.



Git & GitHub

When you change version -

- VCS provides you with proper description
- What exactly was changed
- When it was changed

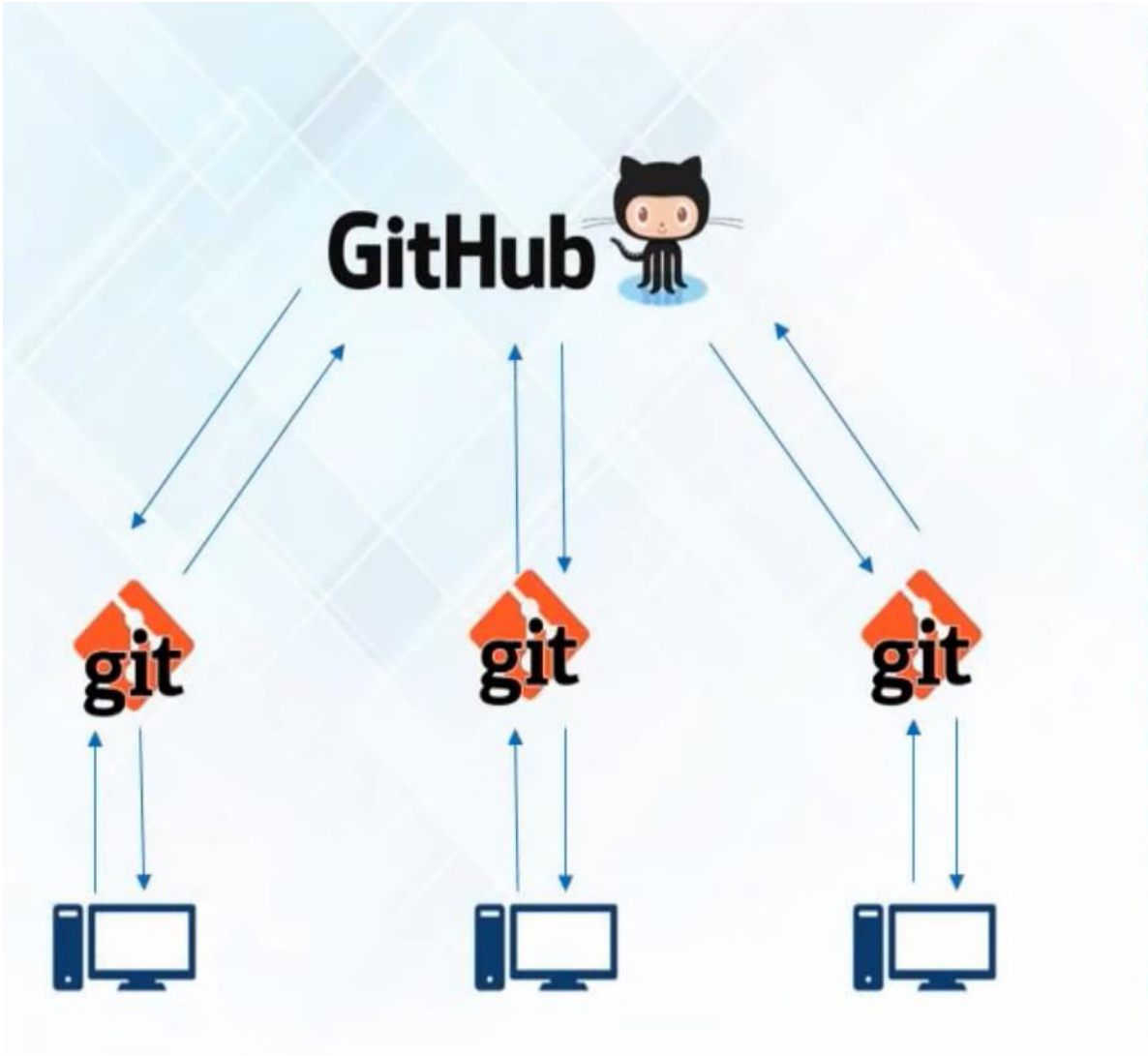
And hence, you can analyze how your project evolved between versions.

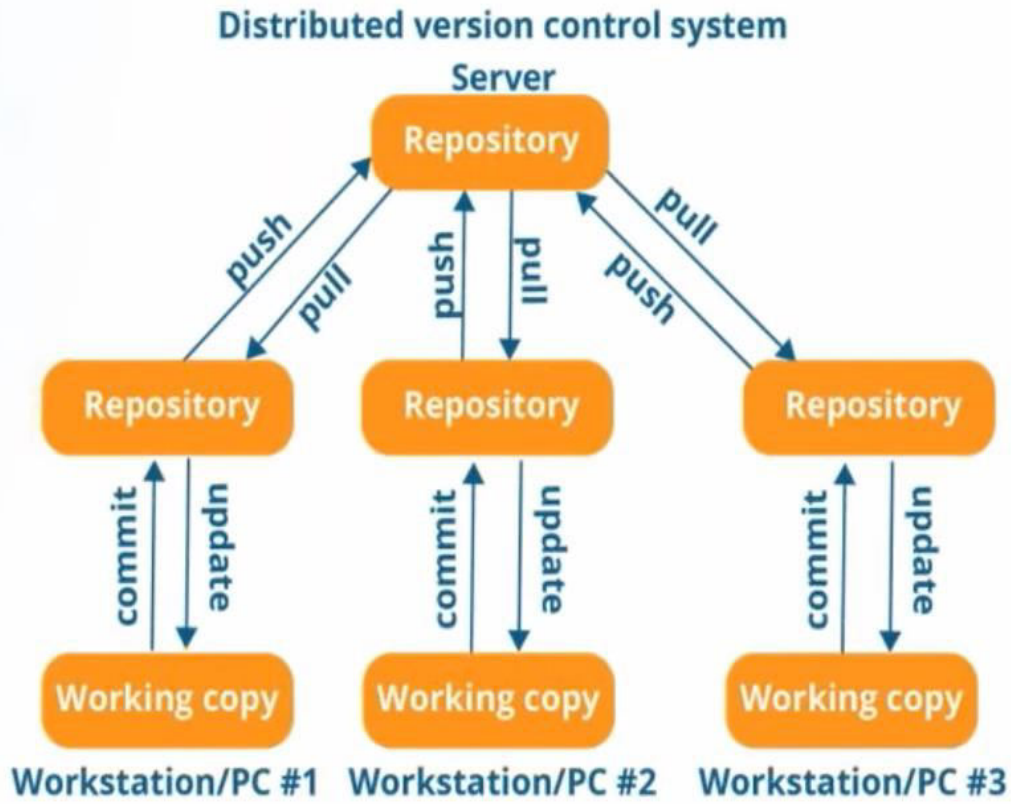
Git & GitHub

Git & GitHub

100
75
50
25
Average
Nov 6, 2011
Jul 21, 2013
Apr 5, 2015
Jan 16,2017
Git
SVN
Mercurial
CVS

Git & GitHub

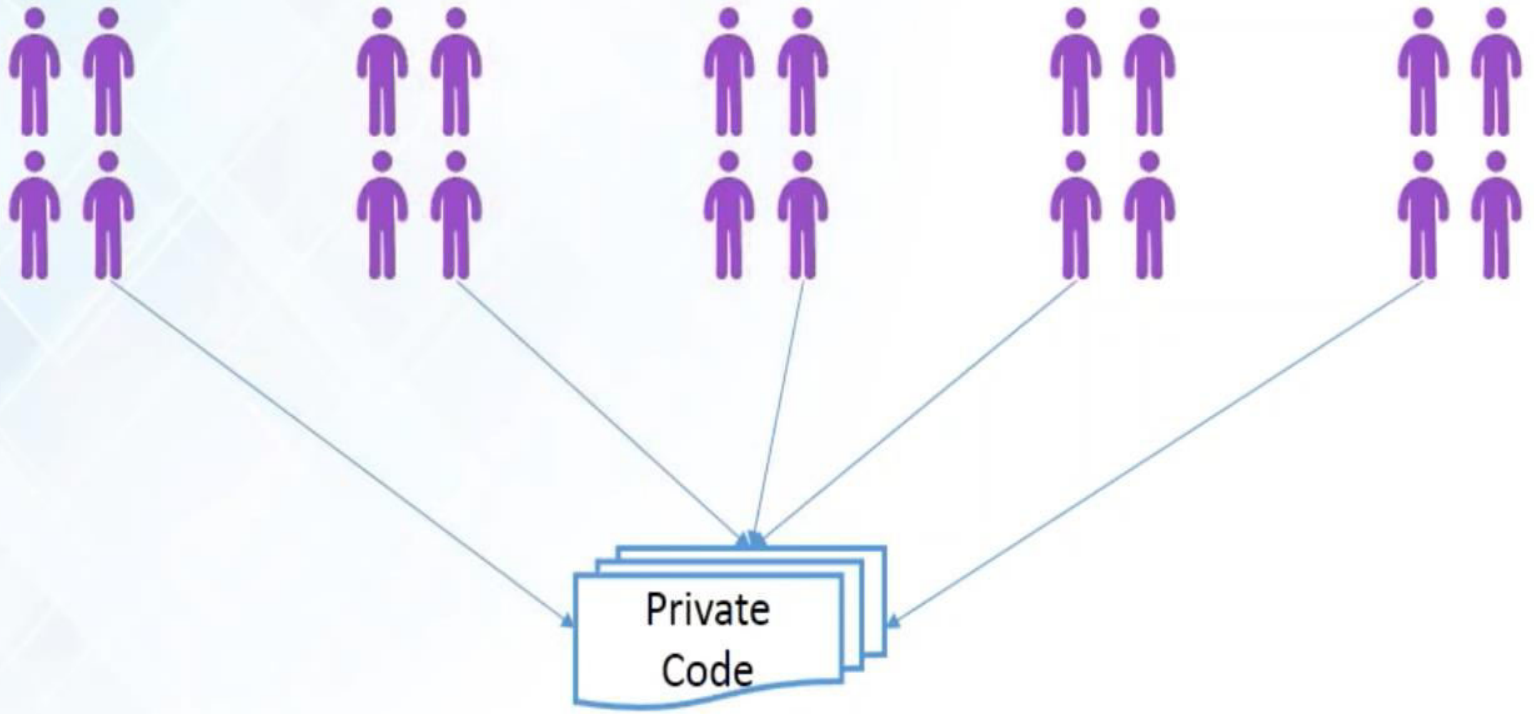

GitHub
git
git
git


Distributed version control system
Server
Repository
push
pull
push
pull
pull
push
Repository
Repository
Repository
commit
update
commit
update
commit
update
Working copy
Working copy
Working copy
Workstation/PC #1
Workstation/PC #2
Workstation/PC #3

Git & GitHub

Problem Statement:

Each team has its own goals, projects, and budgets and they also have Unique needs and workflows

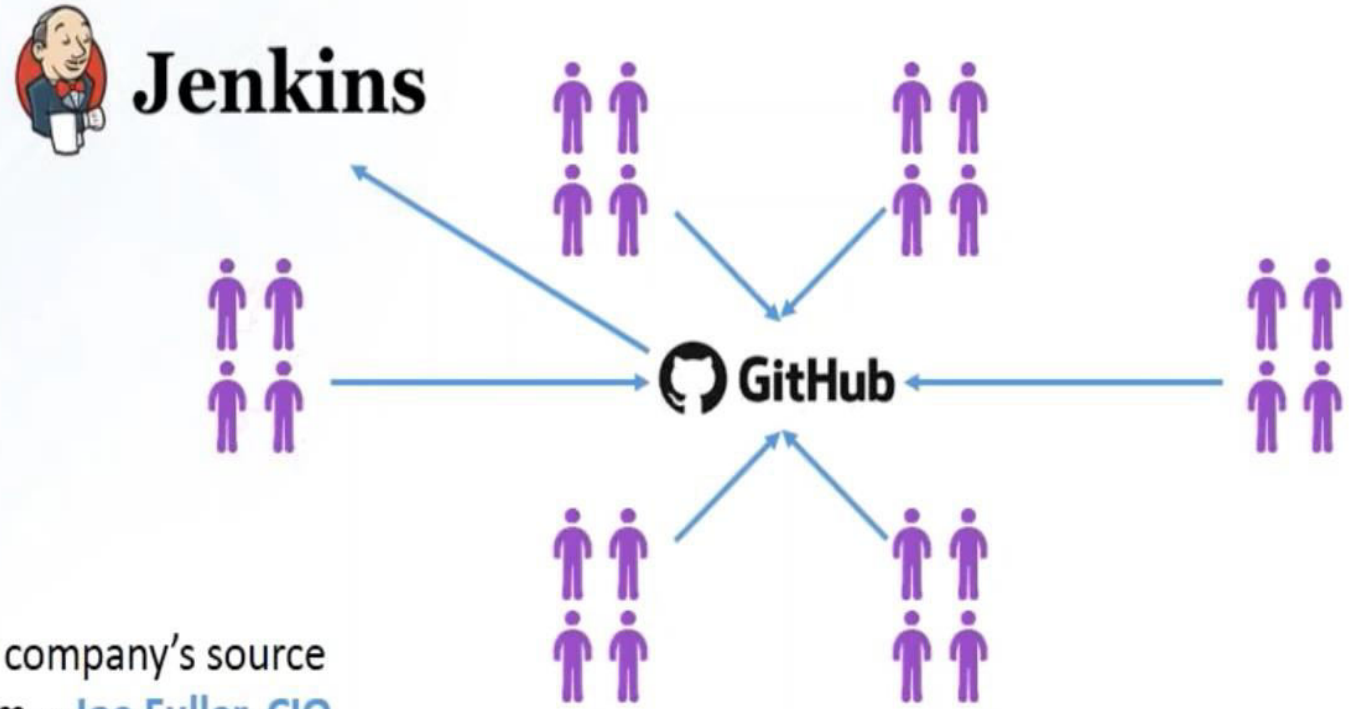


They wanted to make private code “publicly” to make their work more transparent across the company

Git & GitHub

Reason for using GitHub as the solution:

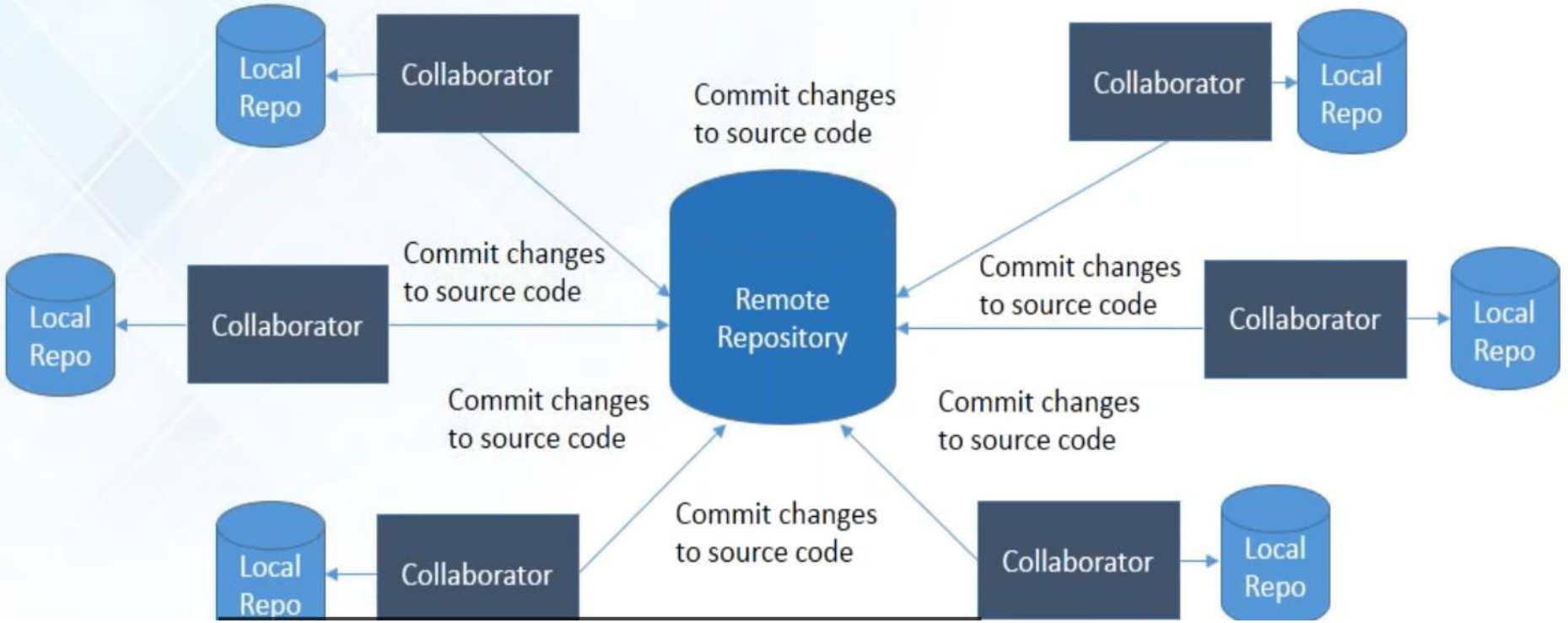
- ❑ They noticed that few of the teams were already using GitHub. Adopting a familiar platform has also made onboarding easier for new employees.
- ❑ Having all of their code in one place makes it easier for them to collaborate on projects.



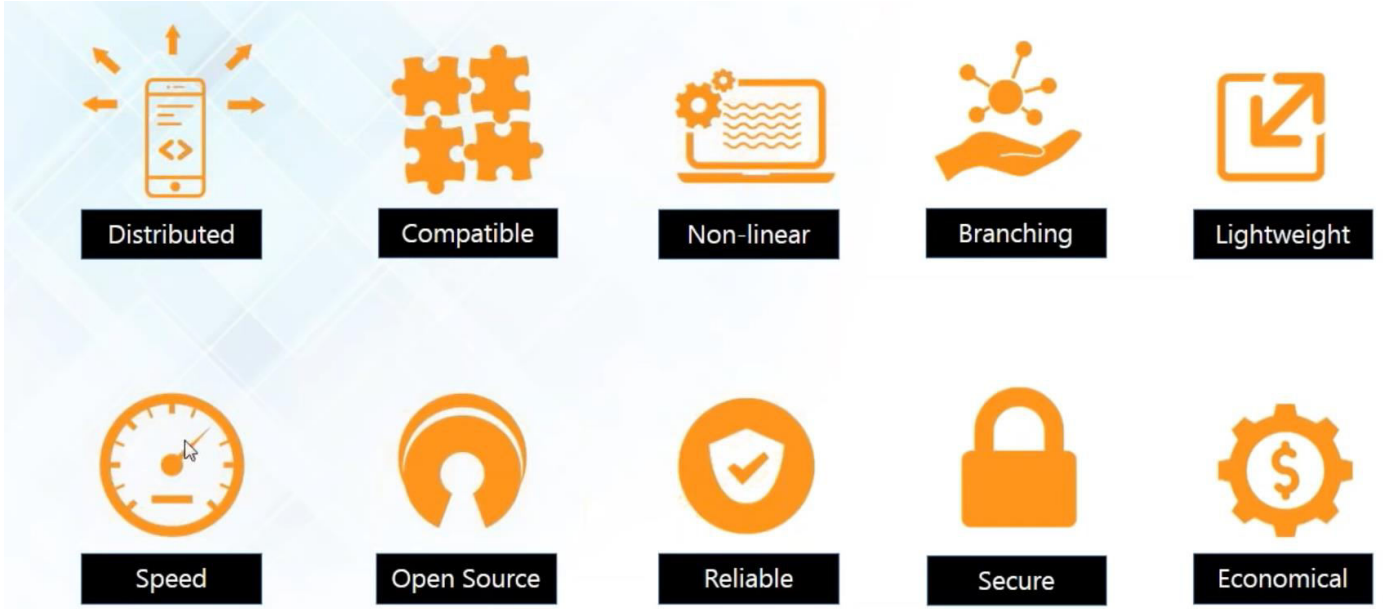
GitHub Enterprise has allowed us to store our company's source code in a central, corporately controlled system. - Joe Fuller, CIO

Git & GitHub

Git is a Distributed Version Control tool that supports distributed non-linear workflows by providing data assurance for developing quality software.



Git & GitHub


Distributed
Compatible
Non-linear
Branching
Lightweight
Speed
Open Source
Reliable
Secure
Economical

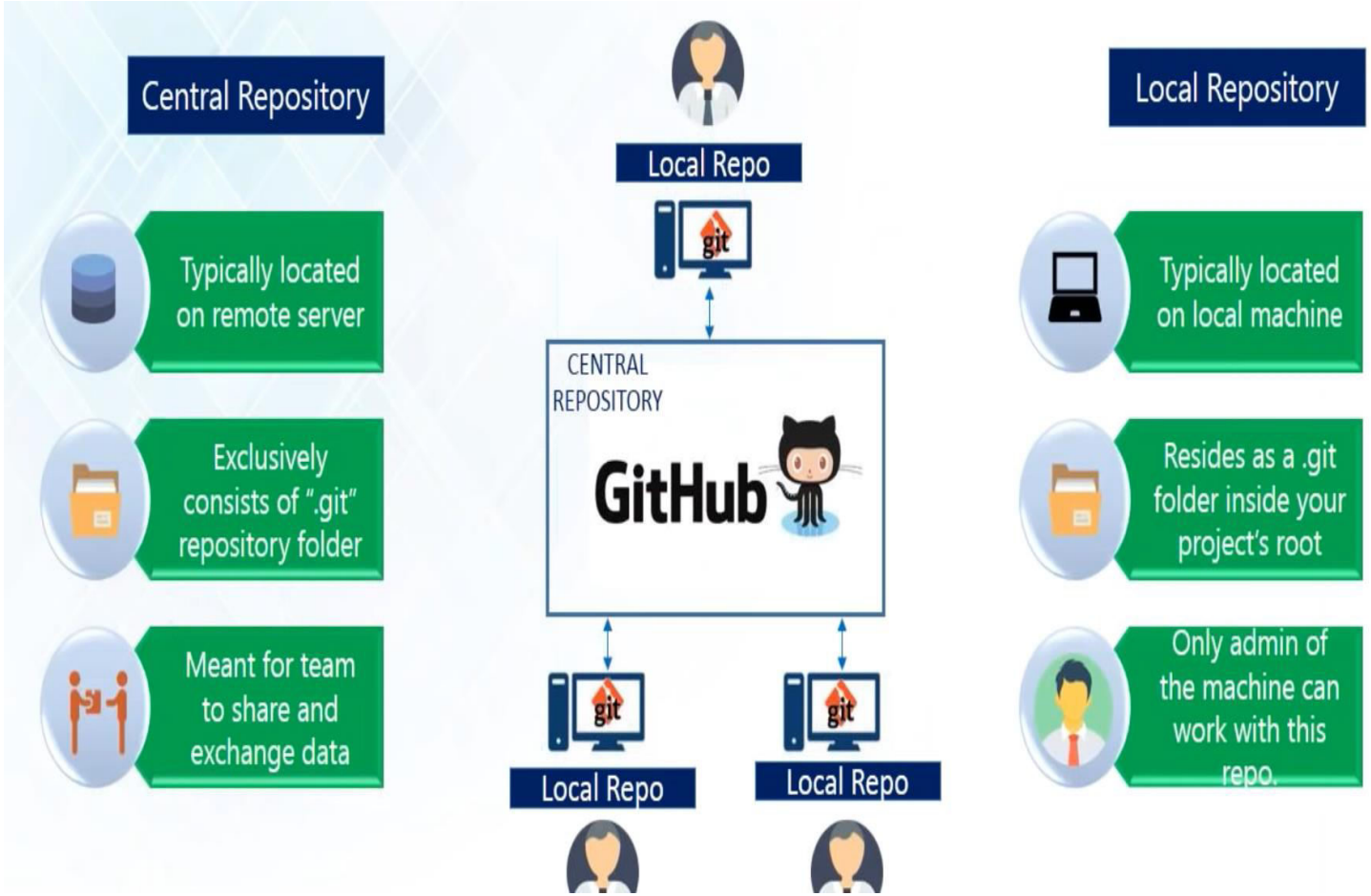
Git & GitHub

A directory or storage space where your projects can live. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, image files, you name it, inside a repository.

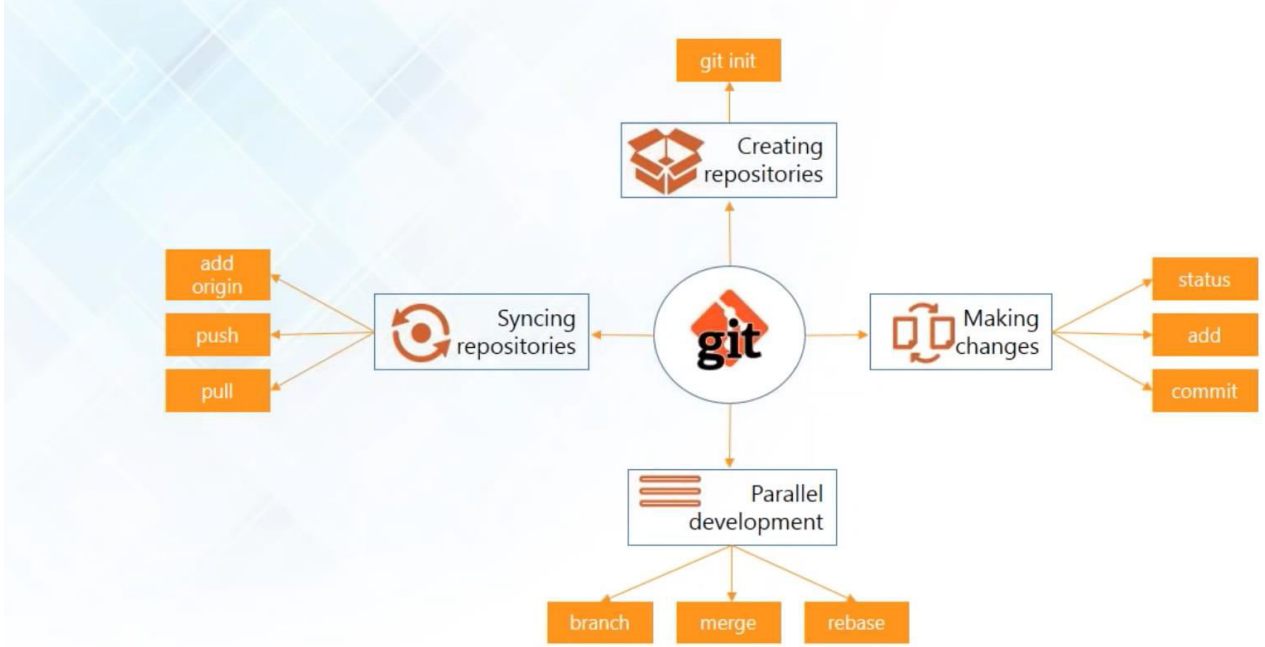
There are two types of repositories:

1. Central Repository
2. Local Repository

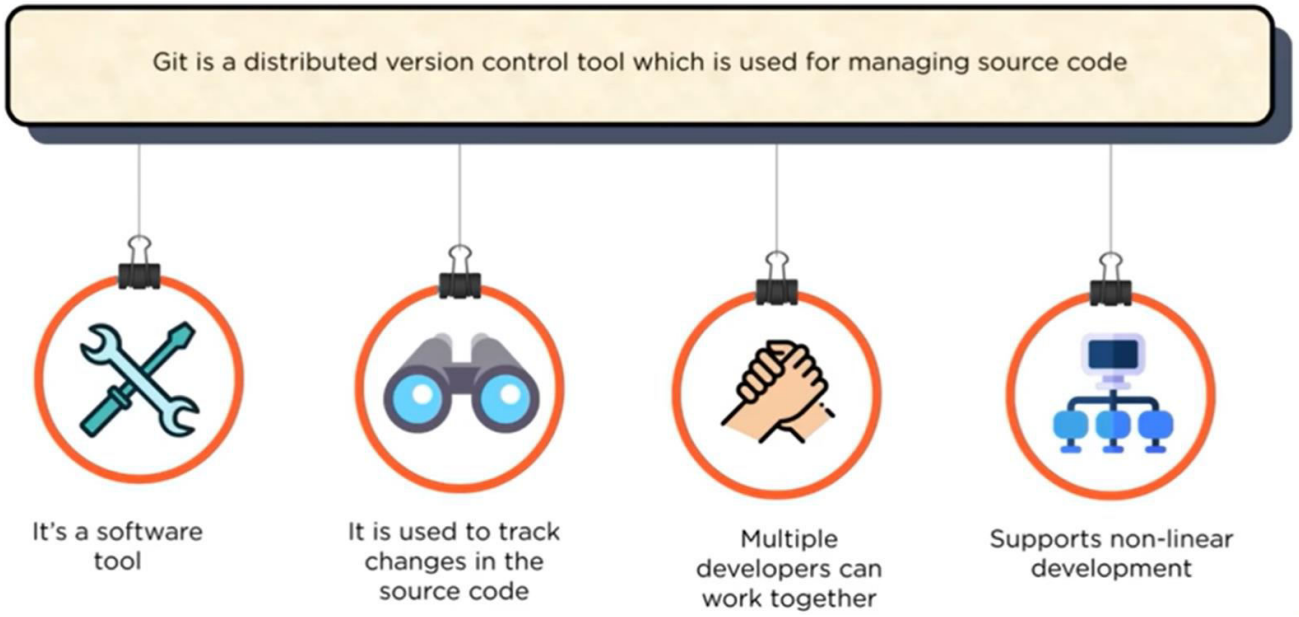
Git & GitHub


Central Repository
Typically located on remote server
Exclusively consists of ".git" repository folder
Meant for team to share and exchange data
Local Repo
git
CENTRAL REPOSITORY
GitHub
Local Repo
Local Repo
Local Repository
Typically located on local machine
Resides as a .git folder inside your project's root
Only admin of the machine can work with this repo.

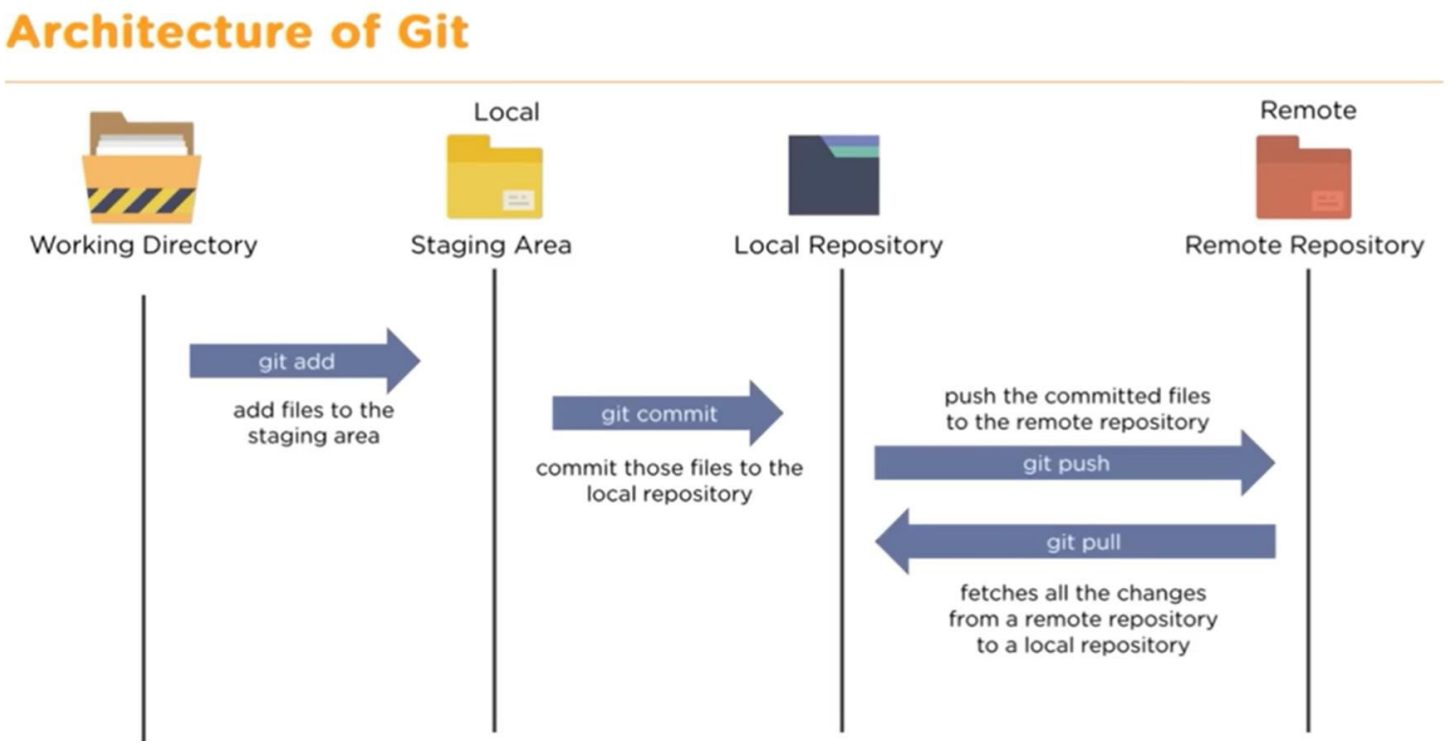
Git & GitHub


git init
Creating repositories
add origin
push
pull
Syncing repositories
git
Making changes
status
add
commit
Parallel development
branch
merge
rebase

Git & GitHub


Git is a distributed version control tool which is used for managing source code
It's a software tool
It is used to track changes in the source code
Multiple developers can work together
Supports non-linear development

Git & GitHub


Architecture of Git
Local
Remote
Working Directory
Staging Area
Local Repository
Remote Repository
git add
add files to the staging area
git commit
commit those files to the local repository
push the committed files to the remote repository
git push
git pull
fetches all the changes from a remote repository to a local repository