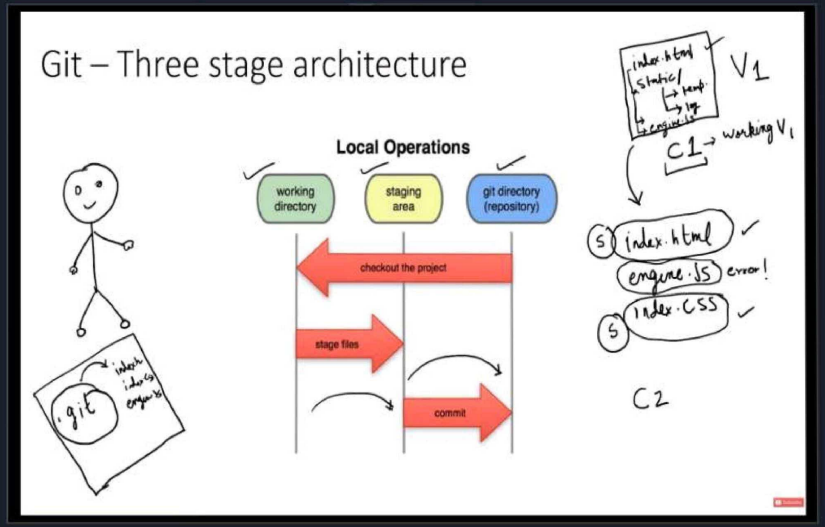
Types of Projects :-

Product	Service
• Creating a tangible item for customer	• Provides intangible services to & customers.
• Success is checked by product sold & profite made.	• Success is measured by customer satisfaction & loyality.
• Planning includes market research & product design.	• Planning includes consideration of customer needs & & service delivery
• Management includes managing scope & budget.	• Management includes service quality & customer satisfaction.
• Challenges includes product development delays & market competetion.	• Challenges includes service delivery delays & customer complaints
• Solid marketing stratcgy is must.	• Effective communication & pro-active problem Solving is must.

Git Three Stage Architecture



Working Directory:
The working directory is where you make modifications to your project files. It's simply the folder on your local machine where you edit, add, or delete files. When you create or modify a file, it exists in the working directory.

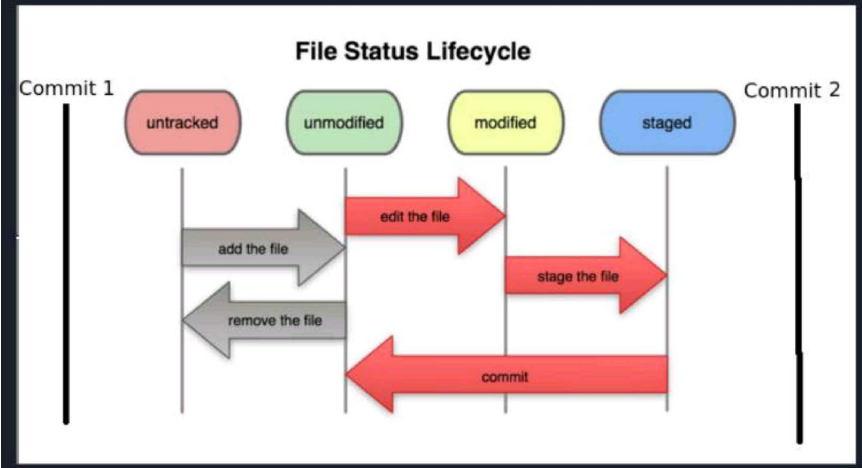
Staging Area (Index):

The staging area, also known as the index, acts as a buffer between the working directory and the repository. It is a conceptual space that holds the changes you want to include in the next commit. When you modify files in the working directory, you need to explicitly add them to the staging area before they become part of the next commit. This allows you to control which changes will be included in the commit.

Repository (Commit History):

The repository contains the complete history of your project, including all the commits you've made. Each commit represents a snapshot of the project's state at a specific point in time. It contains the changes from the staging area along with a commit message describing the changes made. Commits are permanent and immutable, ensuring that the project's history is well-documented and can be revisited at any time.

File Status Life-Cycle



Untracked:

A file is in the "Untracked" state when it is newly created in the working directory or if Git has not been instructed to track changes to that file. These files are not yet part of Git's version control system, and Git is unaware of any changes made to them.

Unmodified:

All the changes done after it has been in the tracked are checked.

If no changes done then file remains in unmodified stage.

Modified:

After you have created or modified a tracked file in the working directory, Git recognizes it as "Modified." This means that the file's content has changed since the last commit.

Staged (Changes to be committed):

To include the changes made to a modified file in the next commit, you need to add them to the staging area. The staging area is a space where you assemble changes that you want to be included in the next commit.

Software Design Approaches

Function Oriented Design

- System is designed from a functional viewpoint.
- Top-down decomposition
- Divide & Conquer approach
- DFD is used

Object Oriented Design

- System is viewed as a collection of objects (i.e., entities)
- Bottom-up approach
- UML is used

Classical waterfall	Iterative waterfall	Prototype Model	Incremental Model	Evolutionary Model	RAD Model	Spiral Model	Agile Model
Basic, Rigid, Inflexible, Not for Real Project	Basic, Problem is well understood	User Requirement Not clear, Costly, No Early lock on Requirements → High User Involvement → Reusability	Module by Module Delivery, Easy to test and debug	Large Projects	Time and Cost Constraint, User at all levels → Reusability	Risk, Not for small Projects, →No Early lock on Requirements, →Less Experience Can work	Flexible, Advanced, Parallel, Process divided into sprints

White Box vs Black Box Testing | Software Engineering

White-box testing	Black box testing
The developers can perform white box testing.	The test engineers perform the black box testing.
what the software is supposed to do, also aware of how it does it.	what the software is supposed to do but is not aware of how it does it.
To perform WBT, we should have an understanding of the programming languages.	To perform BBT, there is no need to have an understanding of the programming languages.
In this, we will look into the source code and test the logic of the code.	In this, we will verify the functionality of the application based on the requirement specification.
In this, the developer should know about the internal design of the code.	In this, there is no need to know about the internal design of the code.
Test design techniques:Control flow testing, Data flow testing, Branch testing, Statement coverage, Decision coverage, Path testing.	Test design techniques: Decision table testing, All-pairs testing, Equivalence partitioning, Boundary value analysis, Cause–effect graph
Can be applied mainly at unit testing level but now in integration, system level also.	Can be applied virtually to every level of software testing: unit, integration, system and acceptance