GIT:-It is a Source code management (SCM) software or tool.

Git is also called VCS(version control system).

Why do we require SCM or VCS?

Answer to this very simple it because it is requirement of today IT companies AND git is tool used under DEVOPS.

What is DEVOPS?

DEVOPS is not a technology ,but a culture(IT culture).

Story:-

Software Developer in order develop any application definetly a team required.

Now each a every developer , has a different way of coding.

Since all member in a team are working on the same project ,

There may be chance when one developer may face the problem due to coding of another,developer.

Hence there should be some standard for writing code.

So the standard approaches to write the code in logical manner ,we have design patterns.

**What is Design patterns?:-**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*there is research paper designed 4 people reusability of object oriented programing by gang of four.

23 patterns.

If you learn these design patterns automatically you will be in the position to developed any project / any software/in any language.

But inorder to learn, design pattern you must have basic knowledge of. Programming and advance knowledge of oops.

Like for coding and development ,design pattern are important.

Similarly ,in order to develop software as a product.

There is some standard maintains ,this

Standard(methodology/technique/process/phenomena/procedure)is called SDLC.

SDLC : software development life cycle.

This process of developing or converting user need into a software product.

When ever you are going to develop any software a software has to cover few milestone or checkpoint on the behalf of which it is ready for the market .or ready for the end user.

These checkpoint or phases are called as software design phases.

There are two design phases

1. Development phase : the phase where need are converted into product.
2. Deployement phase : where the product is tested , monitored , maintained and delivered so that customer can use it.

List of phases

1. Requirement gathering
2. Design analysis and planning
3. Desining and coding
4. Testing
5. Deliver/deployement
6. Feedback/monitoring

Android version:-

A-Astro

B-Blender

C-Cupcake

D-Donut

E-Ecliar

F-Froyo

G-Gingerbread

H-HoneyComb

I-Icream-sandwhich

J-Jellybean

K-Kitkat

L-Lollipop

M-Marshmallow

N-Nogaut

O-Oreo

P-Pie

Q-Quit beta

SDLC Models:-

\*\*\*\*\*\*\*\*\*\*\*\*\*\*SDLC models are taken a reference and we try to apply there design principle in our product or software to that a high quality product can be delivered in the market.

1. Waterfall model:-:

Requirement gathering

Design analysis and planning

Desining and coding

Testing

Deliver/deployement

Feedback/monitoring

In waterfall model each phase is unique and represented as step by step solution ,each step has its own important ,and flow of water is represented as , software flow.

The beauty of water fall model is that it is base for all working model,

The water is oldest model and considered to be foundation for any software product.

Drawback of waterfall model:-

1.Once we move out of any step we can not move backward.

1. Spiral model:-:

Is a advance version of water fall model, in this , we start with a center.

1. Evolution model
2. Prototype model
3. Agile model:-: it is independent model of SDLC , and does not own any property of , water fall model.

Agile model is modern approach for many product companies and software companies working for service .

Agile model is called scrum model .

And agile is implemented using DEVOPS. (development + operational)

Development = requirement analysis , design analysis, coding , designing, build, testing .

Operational = client interaction ,reporting ,feedback , maintainance , delivery, deployment, and monitoring.

There n of tools which work on basis of agile,

1. Youtrack (intellgie) is PMT
2. Jira (Atlassian) is PMT
3. Slack is PMT

WBS :-: Work breakdown structure

***Important Command to work with Git Base:-***

Pwd: printworkingdirectory

~: print root directory

Clear:clearscreen

Whoami:tell the current user name

Cd: to change the path of directory

**Git base or linux or unix or ubantu**

cd/d

**how to go to root folder**

cd/

***how to list of folder and file;***

ls: list all directory

ls -a : list all directory with current and parent path.

Current path ./

Current path../

Ls -attr

***How to list all directory with permission***

Ls -1

***How to create any file with extension***

Touch<file-name.txt><file-name.txt> <file-name.txt> <file-name.txt>

**Cat command:-**the cat command stands for content at,

$cat<filename>

**Most important command for terminal use**

1. Shift+insert key paste
2. Ctrl+insert key copy
3. Esc+w+Q:exit

**How to print massage on console**

$echo”massage”

**How to print massage on file using write mode**

$ echo”massage” > filename

**How to print massage on file using append mode**

$ echo “massage”>>filename

**How to find head**

1. Using git log
2. Using tree/f
3. Manual
4. Using branch

**Git log:-**command is it used to find all the information about the repo.

**$git show**: -help: command help

**$git show:** –help : browser documentation

**Difference b/w git show and git log**

1. Git is show the information about initial commit and readme file information.
2. Git log: gives you the information of the commit if commit are more you can move to difference page by using page name

$git log

…………..

…………..

…………..

<page on>

To quit git log use **q**

**How to find path and head**

.git🡪 head open with notepad

**Or**

$cd.git/

$cat head

**How to create a branch**

Git branch<branchname>

**How to show list of branches**

$git branch

**Note:-**\* this symbol means current working branch.

this will be head.

For example:-

\*master 🡪head🡪current branch(master)

Dec-1

Dev-2

Dev-3

**How to change or switch the branch**

$git checkout<branch-name>

For example:-

Git checkout dev-1

Switching to branch ‘dev-1’

$ 🡪(dev-1)

**What is branch referencing:-**

Whenever we create a new branch the new create requires a reference . for example dev-1 require master branch since ,dev-1 branch uses ,reference of master .the version of master .the version until a first commit.

**How to create new branch and automatically switch:-**

$git checkout -b <new branch name>

Example:

$git branch

Dev-1

\*dev-2

Master

$Git checkout -b dev-3

Switched to a new branch ‘dev-3’

$git branch

Dev-1

Dev-2

\*dev-3

Master

**Why do we create branches?**

1. Whenever you have multiple developer or you have multiple environments(developement, testing ,production ,staging) in the case keep the code , separate for each developer we create a branch.

Why? So that if anything wrong happen ,we can recloan the code from parent branch.

**What is convention:-**

1. Master 🡪live server (production)
2. Testing 🡪(testing)
3. Staging 🡪(testing+ development+ client+ quality analyst)
4. Dev 🡪 development

**Git tracking:-**

**\*\*\*\*\*\*\*\*\*\*\*\*** git tracking is related to stage phase.

It track each file and tells the status of file by using 3 status.

1. Untracked :git will show, all the file as untracked.
2. Unmodified : now file is added in stage , and it is ready to tracked but file is not modified.(no code or editing has been done ,or changes have done )
3. Modified :file is under stage , some change has been done on the file.
4. Deleted(optional) : file is under stage , and user has deleted the file.

**How to check tracking?**

$git status

* **git config**
* **git init**
* **git clone**
* **git add**
* **git commit**
* **git diff**
* **git reset**
* **git status**
* **git rm**
* **git log**
* **git show**
* **git tag**
* **git branch**
* **git checkout**
* **git merge**
* **git remote**
* **git push**
* **git pull**
* **git stash**

**what is GIT?**

**\*\*\*\*\*\*\*\*\*\*\*\***it is linux or unix based terminal tool used for VCM(version control managment)

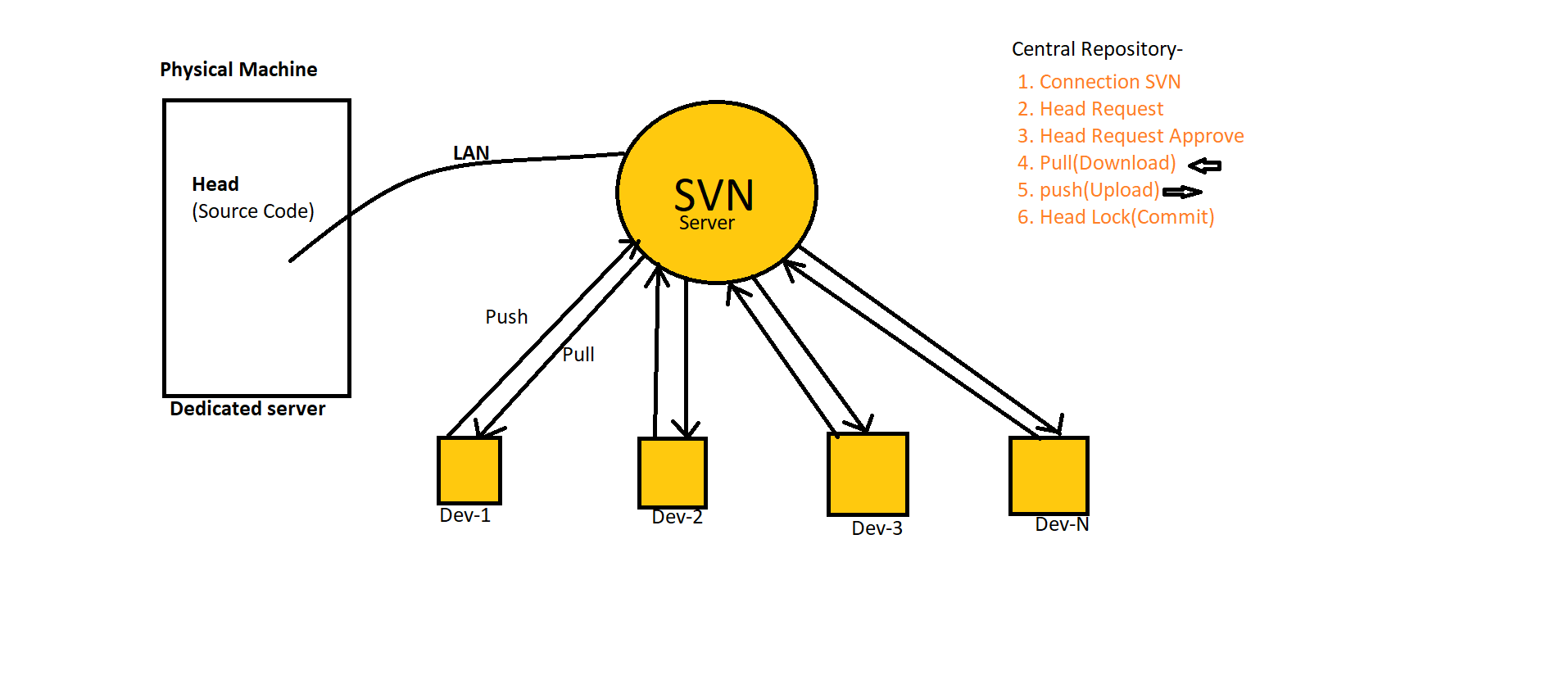
**What is VCM?**

**\*\*\*\*\*\*\*\*\*\*\*\***version control machanism.

Earlier ,when we use to developer any code we have to maintain to backup of every code daily and organise then inform folder.

**Other tools of VCM**

1. svn: subversioning : server Centralised :water fall



1. git : git tools: server distributed :agile

**what is GITHUB?**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Git hub is cloud base server where each and every developer is going to maintain there source code.

Source code: collection of all version code.

Version code: code by each developer is called version code.

**Note :-** VCM tools-GIT, SCM tools-GITHUB,GITLAB,BITBUCKET,BITKEEPAR

**Repository:-** it is very similar to folder in your local system.

1. This repo path has unique URL

2. This URL can be accessed by any one if repo is public.

3. If it is private repo only the owner can access it.

**Type of repository:-**

1.public::-anyone can access

2.private::-only authenticated Owner can access it.

**Difference between public repo/private repo==**

1. Public URL: Google search engine

2. Private URL:it is private and Google can also not search it.

3. Public repo can be accessed any general or guest user

4. Private repo only authenticated user,with valid read/write permission can access it.

5.public repo: https or SSh authentication both can be used , directly download (.zip)

6.private repo: SSh authentication access.

**How GitHub URL for repo looks likes?**

Https://GitHub.com /username/reponame

**What is Readme.md file?**

1. It is very first file added in any repo .

2. It is use to write the documentation with lot, features like heading, paragraph, comments, code, image, license.

3. Every package with readme allows user to use it.

**What is .gitignore file?**

1. . gitignore file is use to ignore which file not be added will pushing the code.

**There are two types of . gitignore**

1. Created at GitHub

2. Created at local

**Note:-** when local system and GitHub are not connected, then in that case, local . gitignore and GitHub . gitignore can be different but as soon as you connect your local system to GitHub system, only one . gitignore file will be allowed.

**What is license file?**

This file contains declaration , code of conduct of the legal license.

License are required when your public.

Head ,branch(master,main) to the server.

**Note:-** branches can divided on the basis

1. Environment

2. Developer role

**First command**

**Git clone:** first time download any repo.

**Git pull:** second time it is used pull.

**Difference between git clone and git pull**

1. Git clone is the first command for download any project git pull is used to download the code with latest changes from any specific branch.

2. In case of git clone all the project with all the branches will be downloaded.

3. Git pull is used to download the code from any specific branch.

**How to clone?**

Git clone<https: URL>

**Git flow:-**

Git flow tells how, your project flow from local to server.

It define the life story or life cycle of repo from local to server.

**Since git is a distributed VCM**

There are four stages (flow channel) of git

1. Local system

2. Stage

3. .git repo

4. Cloud repo(GitHub, gitlab, bitbucket, bitkeeper)

