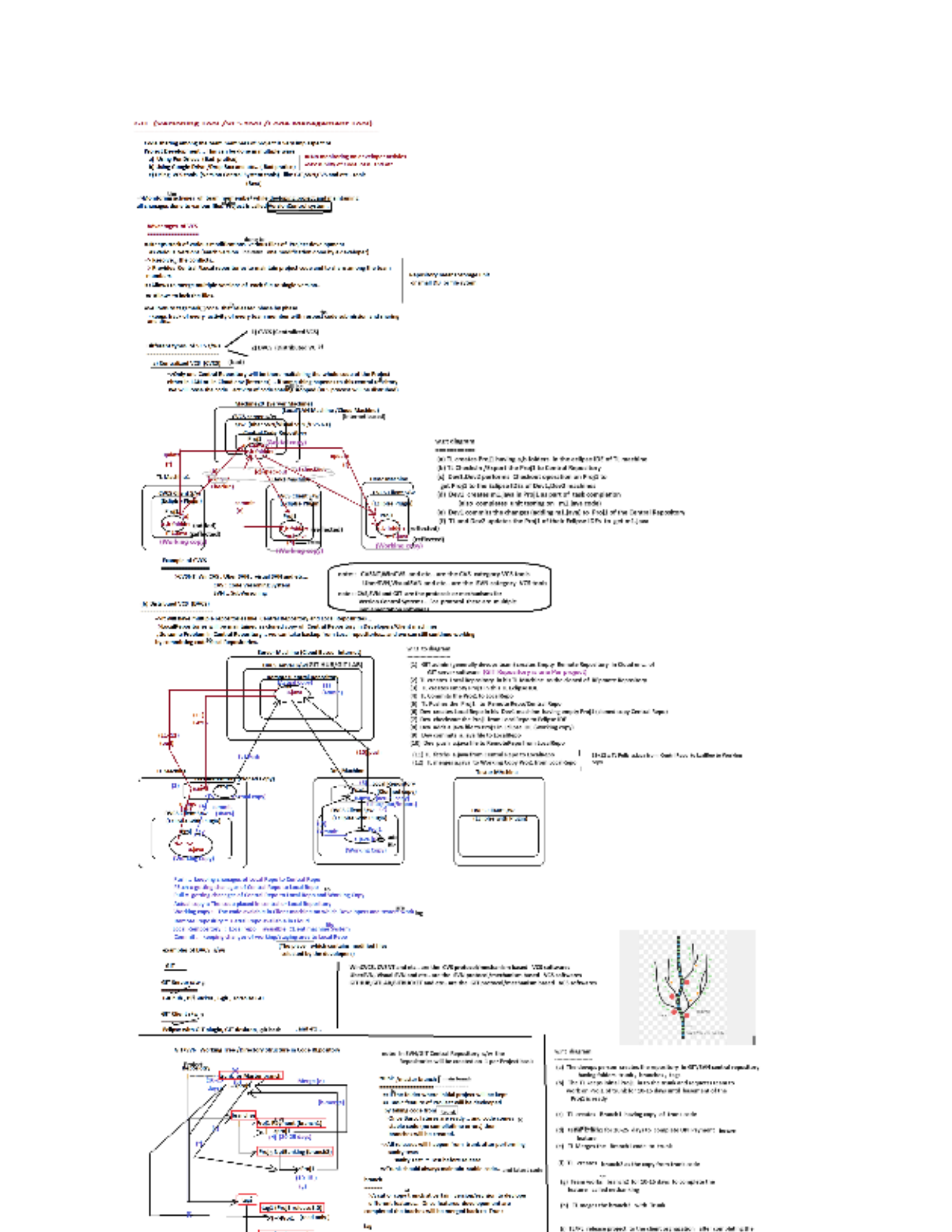

GIT (Versoning Tool /VCS tool /Code Management Tool)

Code sharing among the team members of project is very imp aspect of

Project Development .. This can be done in multiple ways

a) Using PenDrives (Bad pratice)

b) Using Google Drive /Drop Box and etc.. (Bad pratice)

=>No monitoring on developer activies =>Possibilty of Code losss and etc..

c) Using VCS tools (Version Control System tools) like GIT,SVN,CVS and etc.. tools

the

(Best)

=>Monitoring activiies of team memember while devloping project and maintaining all chanages done to various files project is called VersionControl system.

Advantages of VCS

done to

=>Keeps track of various modifications various files of Project development

as various versions (earch version indicates one modification done by a developer) => Resolve the conflicts..

=> Provides Central /Local repositories to maintain project code and to share among the team members

=>Allows to merge multiple versions of each file to single version..

=> Allows to lock the files.

=>Allows to tag/mark code that released phase by phase

to

=> keeps track of every activity of every team member with respect code submission and sharing and etc...

Repository means Storage unit

or small DB or file sytem

1) CVCS (Centralized VCS)

Different types of VCS s/ws

a) Centralized VCS (CVCS) (bad)

2) DVCS (Distributed VC S)

=>Only one Central Repository will be there maitaining the whole code of the Project either in LAN or in Cloud env (internet).. if some thing hapends to this central reository we will loose the code.. activity of code sharing stopped (VCS process will be distrubed)

Machine20 (Server Machine)

(LocalLAN Machine /Cloud Machine) (Internet based)

CVCS server s/w s/w (uber SVN/Visual SVN /CVS NT)

Central Code Repository

Proj1cutal copy)

mr.java

Ha,b folders

update (f)

update (f)

(c) checkout

(c) checkout

TL Machine1

Lexport!

Dev1 Machine

Dev2 Machine

Checkin)

CVCS Client S/

CVCS Client S/w

commit

VCS Client S/W Eclipse Plugin

(e)

Proj1

mijava (reflected)

,b folders reflected) m1jáva

(Eclipse Plugin)

Proja)

,b folders added)

(Working copy

Example of CVCS

====

==========

(d) adding (Working-copy)

=>CVSNT, WinCVS, Uber SVN, visual SVN and etc...

(b) Distribued VCS (DVCS)

=========

=======

CVS ::Code Versoning System SVN :: SubVersoning

w.r.t diagram

(Eclipse Plugin)|

Proj1

,b folders (reflected) m1.java (Working copy)

======

(a) TL creates Proj1 having a,b folders in the eclipse IDE of TL machine

(b) TL Checksin/Export the Proj1 to Central Repository

(c) Dev1,Dev2 performs Checkout operation on Proj1 to

get Proj1 to the Eclipse IDEs of Dev1, Dev2 machines

(d) Dev1 creates m1.java in Proj1 as part of task completion

(also completes unit testing on m1.java code)

(e) Dev1 commits the changes (adding m1.java) to Proj1 of the Central Repository (f) TL and Dev2 updates the Proj1 of their Eclipse IDEs to get m1.java

(reflected)

note:: CVSNT,WinCVS and etc.. are the CVS category VCS tools UberSVN,VisualSVN and etc.. are the SVN category VCS tools

note :: CVS,SVN and GIT are the protocols or mechanisms for Version Control Systems .. For protocol there are multiple implementation softwares

=>It will have multiple repositories like Central Repository and Local Reposirities .. =>LocalRepositories will be maintained as cloned copy of Central Repository in Developers/Client machines

So some Problem in Central Repository.. we can take backup from Local repositories.. and we can still continue working by committing code Local Repositories.

(11) Fetch

(11+12) (pul)

Server Machine (Cloud Based -Internet)

DVCS server s/w (GIT HUB/GIT LAB)

Remote/Central Repository Actual Copy) a.java

(1)

(admin)

w.r.t to diagram

(1) GIT admin (generally devops team) creates Empty Remote Repository in Cloud env.. of GIT server software (GIT Repository is one Per project)

(2) TL creates Local Repository in his TL MAchine as the cloned of REpmote Repository

(3) TL creates empty Proj1 in this TL Eclipse IDE

(4) TL Commits the Proj1 to LocalRepo

(5) TL Pushes the Proj1 to Remote Repo/Central Repo

(6) Dev creates Local Repo in his Dev1 machine having empty Proj1 (cloned copy Central Repo)

(7) Dev checksout the Proj1 from LocalRepo to Eclipse IDE

(8) Dev adds a.java file to Proj1 in Eclipse IDE (working copy)

(9) Dev commits a.java file to LocalRepo

(10) Dev push a.java file to RemoteRepo from LocalRepo

(5) Push

(10) push

TL Machine

Dex Machine

(11) TL fetchs a.java from Central Repo to LocalRepo (12) TL merges a.java to Working Copy Proj1 from

LocalRepo Tester MAchine

11+12 :: TL Pulls a.java from Centrl Repo to LcalReo to Working copy

(2)

Locati pository (cloned Copy) a.java Proj

(actual copy)

a.java

merge (1/2)

(4) commit

DVCS Client S//w

(Eclipse with Plugin)

Pro13)

(share)

(6) Local Repository Proj1 (Clonned copy) (actual copy) (check out/import)

DVCS Client S/w (7) (Eclipse with Plugin)

DVCS Client S/W

(Eclipse with Plugin)

a.java

Commit

Proj1 a.java (8) (Working Copy)

add file

(Working Copy)-

Push :: keeping chanages of Local Repo to Central Repo

FEtch :: getting chanages of Central Repo to Local Repo to

Pull :: getting chanages of Central Repo to Local Repo and Working Copy

Actual copy :: The code placed in central or Local Repository

are

Working copy :: The code avaiable in Client machine on which Developers and testers work ing Remote Repositiry :: Cetral Repo available in Cloud

file

Local Rempository:: Local repo avaialble Client machine System Commit :: keeping changes of working/staging area to Local Repo

examples of DVCS s/ws

=====

GIT

==========

GIT Server s/w s

(The place which contains modified files selected by the developers)

WinCVCS, CVSNT and etc.. are the CVS protocol/mechanism based VCS softwares UberSVN, Visual SVN and

etc.. are the SVN protocol/mechanism based VCS softwares GITHUB,GITLAB,BITBUCKET and etc.. are the GIT protocol/mechanism based VCS softwares

Glt hub, Bi Bucket, Ggit, Tortoise GIT

GIT Client s/w s

Eclipse with GIT plugin, GIT desktop, git bash

, and etc..

commit

branch

trunk/master branch.

GIT/SVN Working Tree /Directory Structure in Code Repository

======

Project Reposiotry

(£)

(10-15

tex

trunk or Master branch

days 1-pro

Merge (e)

(h-merge)

branches

-------> Proj1-Payment (branch1)

tags

->proj1

(d) (20-25 days)

Proj 1-NetBanking (branch2)

7 Proj1 (10-15) (g)

branch

======

→tag1 (Proj1-release1.0) -->Proj1 (read only)

tag2 (Proj1-Demo)

|--->Proj1 (read only)

Admin activities in GITHIB (GIT Server s/w) (DevOps Team work)

==========

step1)

==========

Create Account in www.github.com by submitting email id

step2) Create Remote Repository in github.com and collect url

note: In SVN/GIT Central Repository s/w the Repositories will be created on 1 per Project basis

trunk /master branch main branch

==========

=>It the folder where initial project will be kept => Basic feature of Projject will be devleoped by taking code from

=>Once Bascc fatures are ready.. and code comes to stable code (no compiletime errors) then

branches will be created.

=>All releases will happen from trunk after performing sanity tests

sanity Test Test before release..

=>Trunk should always maintain stable code.. and latest code

of

=>A cut or copy trunck at certain version/revision to devlope different features.. Once features developement are completed the braches will be merged back to Trunk

tag

====

=>A cut or copy of trunk that indicates what is being released to Client or what is being deomisstrated for client is called tag. (It is always read-only copy)

github.com ---> create repository --->name :: FullStackRepo (Remote Repo)

山

(our choice name)

create

https://github.com/natarazworld/FullStackRepo.git

=> Take Two eclipse workspaces--- 1 for TL and 1 for to Dev1 (Pointing to different worrkspace folders)

(feel Two eclipse windows as two devleopers (TL,Dev1)

Aactivities in TL Machine (TL workspace)

========

=========

a) create Local repository as Cloned repo of Remote Repository (above created)

i) get two tabs

window --->show view-->GIT ---> GIT staging, GIT repositories

ii) Go to Git Repositories tab

w.r.t diagram

(a) The devops person creates the repository in GIT/SVN central repository having folders trunk, branches, tags

(b) The TL keeps initial Proj1 in to the trunk and requetss team to work on Proj1 of trunk for 10-15 days until basement of the Proj1 is ready

(c) TL creates Branch1 having copy of trunk code

(d) tech1 for 20-25 days to complete UPI Payment feature

feature

(e) TL Merges the Branch1 code to trunk

(f) TL creates branch2 as the copy from trunk code

on

(g) Team works branch2 for 10-15 days to complete the feature called netbanking

(h) TL meges the branch2 with Trunk

(i) TL/PL release project to the client orgnazation after completing the sanity test.. also keeps the released code to as tag for future

reference

Creating Personal AccessToken in the account of GITHUB.com

=> Goto GIT HUB Account image ---> click on it ---> settings -----> developer settings -----> personal Access Token -----> note:: tkn1 --->select all main checkboxes ---> generate token and save the token in a separate file

GIT Token::

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

(This will be used as the password while connecting to GIHBUB Repository from the Eclipse IDE)

(GIT Client)

(note:: In the place of password submit the

above access token)

-->clone repository --> URI: https://github.com/natarazworld/FullStackRepo.git submit username, password (GITHUB account details) --->next---> next--> Choose LocalRepo Name and Location (c:/users/Nareshit/GIT/FullStackRepo-TLLocal)

-->finish

b) create Project (Java Project)

(Proj1) note: u may add intial code

c) Share Project to Local repo ----> Push to Remote Repo

Right click on the Project ===> team --->share Project ---> press ++ ---> write comment ---> commit and Push

Activities in Dev1 machine

==========

a) create Local repository as Cloned repo of Remote Repository (above created)

i) get two tabs

window --->show view-->GIT ---> GIT staging, GIT repositories

ii) Go to Git Repositories tab

note: if Dev1 clones the Empty GITHUB repository with out having Project .. then to get Project to the Dev1 Local Repo from Central/Remote Repo once it is been placed by TL we need to perform following operations in the GIT Repository tab of Dev1 Eclipse IDE

-->clone repository --> URI: https://github.com/natarazworld/FullStackRepo.git submit username, password (GITHUB account details) --->next---> next--> Choose LocalRepo Name and Location

a) Right on cloned Repo and perforn Fetch from Origin

b) expand remote tracking -->right click on orgin/master --> checkout

c) perform import operation as shown as discussed here

note:: Instead of password, u need to give personal access token

-->finish

(c:/users/Nareshit/GIT/FullStackRepo-TLLocal)

b) Import the Project to Eclipse from GIT Local Repository

File ---> import ---> GIT ---> import projects from GIT ---> existingLocal Repo ---> import as existing Project-->select Dev1LocalRepo ---> .....

c) Add B.java file having some code and commit it

right click on Project---> team-->commit ---> ++ ----> comment ---> ....

d) make TL pulling the changes from Remote Repo- Local Repo to Working copy

e) Make TL to add /modify /delete code/files and commit--->push

f) Make Dev1 to pull changes..

Understading confict creation and resolving the conficts..

=======

CreatingConflict

a) Make DEV1 modifying A.java at line no: 5 (// 456)

and make him not to commit the code

b)) Make TL modifying A.java at line no: 5 (// 123)

and make him to commit the code

c) when Dev1 tries to commit the code we will get conflit error (Fail-Fast foward rejected)

Resovle Conflict

======

d) Make Dev1 Pull the Project ... then A.java show both

Repo code and our code --> keep repo code, delete our code --> comnit and push

Creating Branch

==========

(keep both codes or only one side code)

note:: To avoid conflicts (a) Before commit and push,perform pull

a) make TL creating branch

(b) Pull the code at regualar intervals

Right click on Project -->team -->switch to -->new branch--> branch name:: Proj1-B1-Payment

b) make TL Commiting and pusing the branching

c) Make Dev1 getting Proj1-B1-Paymen branch

note: if Project is having empty folder like folder with out files or sub folders they will not reflect in Local/Centrl Repository when we commit the code

To see the history of Project /folder/file

Right click on the Project/file/folder=>

team ----> showhistory ---> displays all the versions/revisions

--> click on any revision to get the code..

=>Go to GIT Repositories tab --->right click Dev1Local ---> Fetch from orgin

=> right click on Remote tracking ---> checkout -->check as new Local branch

e) Make Dev1 adding new code to Proj1-B1-Payment branch to complete feature and commit -->push code

f) Make TL to pull the chnages to Proj1-B1 -Payment

g)make TL merging Proj1-B1-Payment branch code Trunk

Go Repositories Tab ---> Local ---> check out master --->right click --->merge -->select Proj1-B1-Payment ---->....

(from Remote Trakcking)

--->go commit and push the code..

Creating Tag

==========

stashing

a) TL Machine -->check out master ---> right click on Project --->team ---> advanced ---> tag --->

tag name :: Proj1-release1.0 ---> create Tag push code..

=>After commiting some changes to any Brach1.. if have any incomplete logics .. we save them in temporary

place to complete them later.. this is calle stashing..

=>modify code ... having incomplete works (AS TL/Dev1)

=>create stash :: right click proj--> team->stashes --> create stash--> name: st1 (stash changes wll disapper from branch code) =>complete u r merge activity of brach with master (develop other priority logics in the branch and commit it)

=> To get stashes and apply changes

right click project --->team ---> stashes -->choose stash name -->apply stash changes ---> contine u r work..

(corner button)

What is .gitignore file and how to use it?

(or)

How to make certain files and folders of the project not moving to Local/Remote Repository as part of GIT Operations?

Ans) By adding .gitignore file to the Project and specifying the list of folders and files that want to stop going to Local/Central Repository.

step1) add .gitignore file to the root folder of the Project as shown below

.gitignore

HELP.md

/target

step2) share project to Local and Remote Repositories

/src/test/java

!.mvn/wrapper/maven-wrapper.jar

STS-Eclipse###

.apt_generated

.classpath

.factorypath

.project

step2) observe directory structure in GITHUB.com

.settings

.springBeans .sts4-cache

TestProj03

.mvn

src/main/java/nit/TestProj03

App.java

FAQs

====

.gitignore

1) What Code repository?

2) What is VCS (Version Control System)

pom.xml

3) Types of VCS softwares (CVCS and DVCS)

4) Examples for CVCS softwares

.gitignore

5) Examples for DVCS

6) Limitations of CCVS and avantages of DVCS?

7) GIT fall under which VCS software ?

8) What is main advantage of GIT over SVN /CVS category VCS softwares

9) Explain the following terminologies?

a) Check out

b) Export/Share

c) Fetch

d) Pull

e) Commit

f) Push

g) Merge

10) Explain

Code Repository Directory structure?

a) Trunk/master branch/main branch

b) Branches

c) tags

11) What is Sanity Test?

12) What is conflict in Code Repositories and how to resolve them?

13) what is the neccessarity of creating branches ?

14) what is the neccessarity of creating Tags ?

15) What is Stashing and how it can be useful in day to day actitivies?

16) what is the use of .gitignore file?

17) How to make certain folders and file the Project not participating in GIT oepations