



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS302 AGILE SOFTWARE ENGINEERING

II YEAR /III SEMESTER

Unit 2- Agile Development

Topic 8: Agile-GITHUB





Brain Storming



1. How to control and traceout the different versions of software?



What is a 'version control system?'



- A way to manage files and directories
- Track changes over time
- Recall previous versions
- 'source control' is a subset of a VCS.



Some history of source control...



(1972) Source Code Control System (SCCS)

- closed source, part of UNIX

(1982) Revision Control System (RCS)

- open source

(1986) Concurrent Versions System (CVS)

- open source

(2000) Apache Subversion (SVN)

- open source





...more history



(2000) BitKeeper SCM

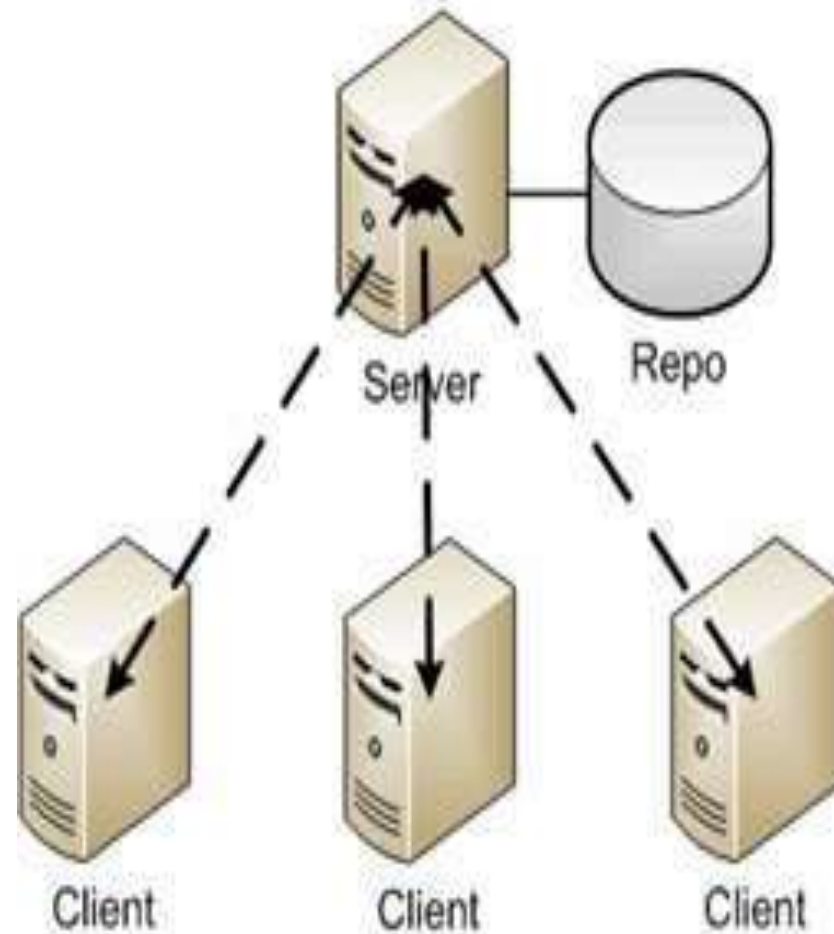


- closed source, proprietary, used with source code management of Linux kernel
- free until 2005
- distributed version control

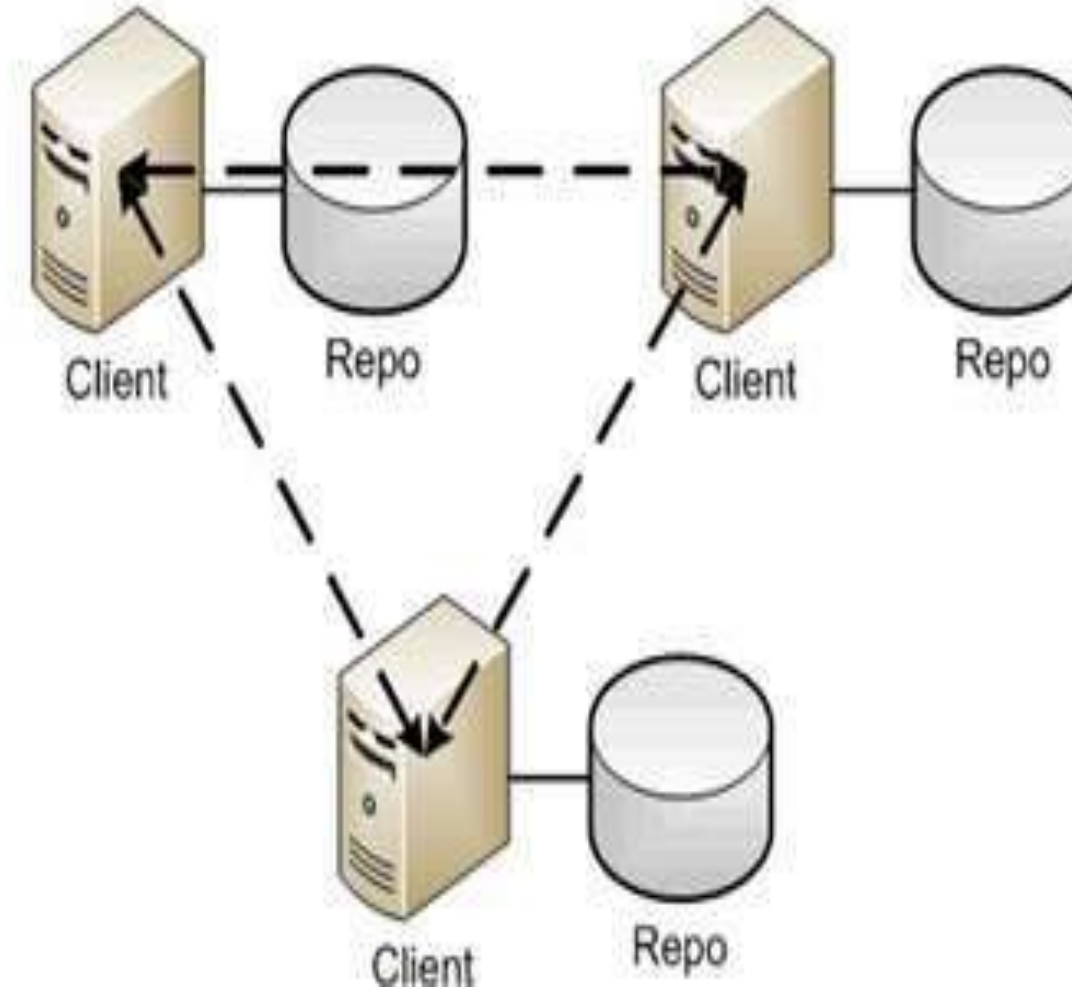
Distributed version control

- No central server
- Every developer is a client, the server and the repository

Traditional



Distributed





What is git?



- created by Linus Torvalds, April 2005
- replacement for BitKeeper to manage Linux kernel changes
- a command line version control program
- uses checksums to ensure data integrity
- distributed version control (like BitKeeper)
- cross-platform (including Windows!)
- open source, free



What is Git?



- Git is a Version Control System (VCS) designed to make it easier to have multiple versions of a code base, sometimes across multiple developers or teams
- It allows you to see changes you make to your code and easily revert them.
- It is NOT GITHUB!



Is Git for me?



- People primarily working with source code
- Anyone wanting to track edits (especially changes to text files)
 - review history of changes
 - anyone wanting to share, merge changes
- Anyone not afraid of command line tools



Most popular languages used with Git



- HTML
- CSS
- Javascript
- Python
- ASP
- Scala
- Shell scripts
- PHP
- Ruby
- Ruby on Rails
- Perl
- Java
- C
- C++
- C#
- Objective C
- Haskell
- CoffeeScript
- ActionScript

Not as useful for image, movies, music...and files that must be interpreted (.pdf, .psd, etc.)



Ok, then what is Github?



- **Github.com** is a website that hosts git repositories on a remote server
- Hosting repositories on Github facilitates the sharing of codebases among teams by providing a GUI to easily fork or clone repos to a local machine
- By pushing your repositories to Github, you will pretty much automatically create your own developer portfolio as well!



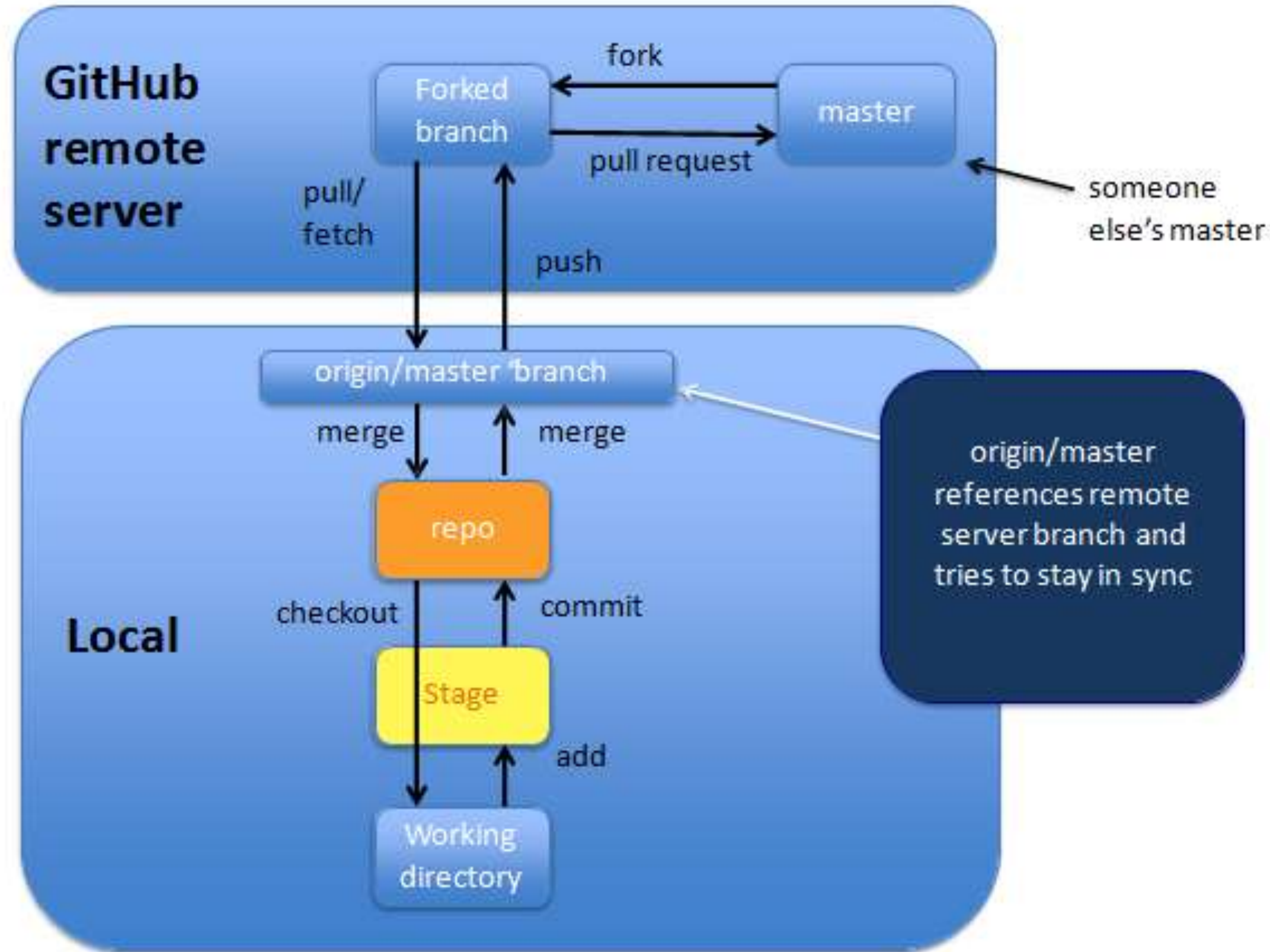
Ok, then what is Github?



- a platform to host git code repositories
- <http://github.com>
- launched in 2008
- most popular Git host allows users to collaborate on projects from anywhere
- GitHub makes git social!
- Free to start



Github-Workflow

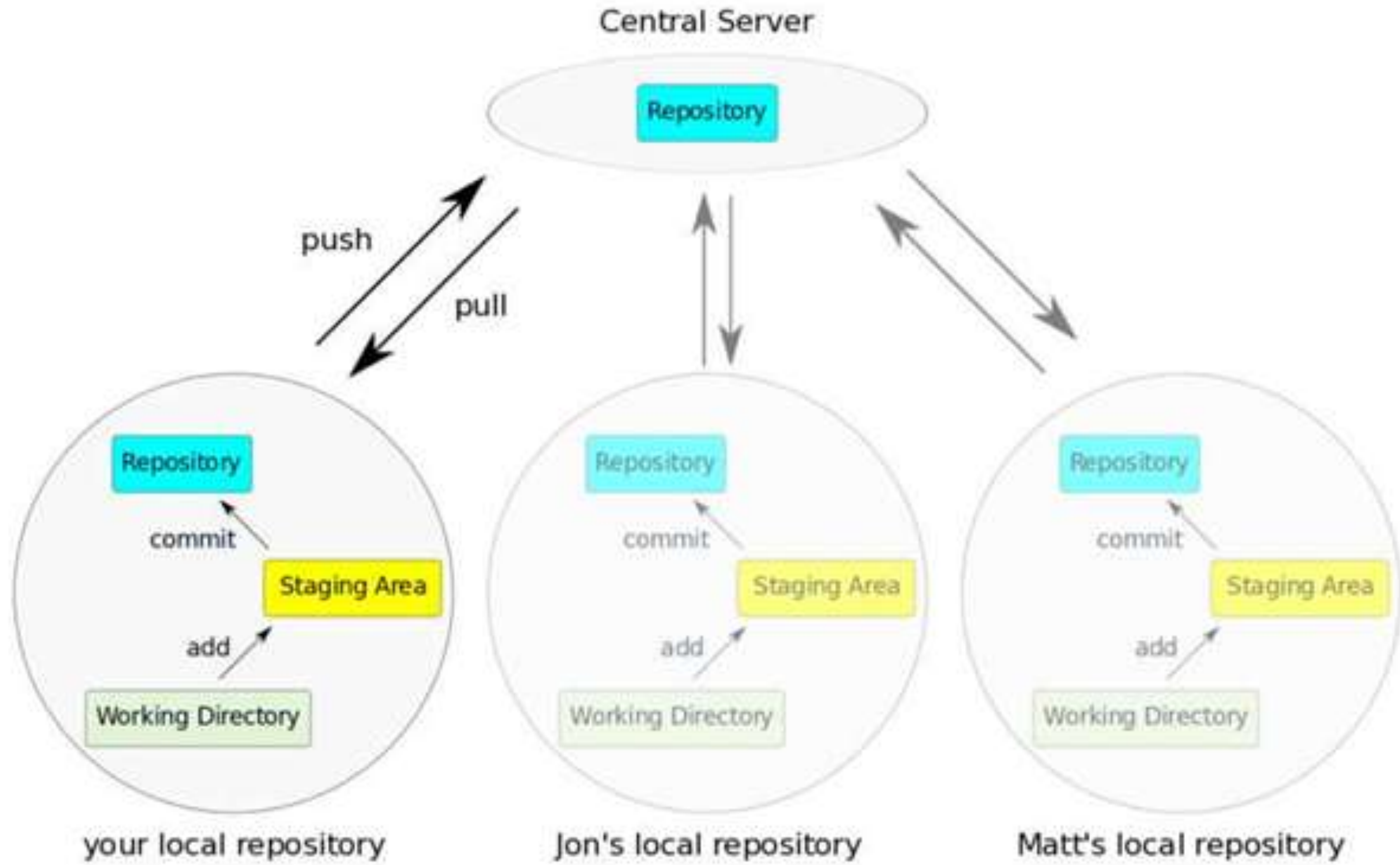




Important to remember



- Sometimes developers choose to place repo on GitHub as a centralized place where everyone commits changes, but it doesn't have to be on GitHub.





How does git work?



- Can be complicated at first, but there are a few key concepts
- Important git terminology in following slides are **blue**

Key Concepts: **Snapshots**

- The way git keeps track of your code history
- Essentially records what all your files look like at a given point in time
- You decide when to take a snapshot, and of what files
- Have the ability to go back to visit any snapshot
- Your snapshots from later on will stay around, too



Key Concepts: Commit



- The act of creating a snapshot
- Can be a noun or verb
 - “I committed code”
 - “I just made a new commit”
- Essentially, a project is made up of a bunch of commits



Key Concepts: Commit



- Commits contain three pieces of information:
 1. Information about how the files changed from previously
 2. A reference to the commit that came before it
 - Called the “parent commit”
 3. A hash code name
 - Will look something like:
fb2d2ec5069fc6776c80b3ad6b7cbde3cade4e



Key Concepts: **Repositories**



- Often shortened to 'repo'
- A collection of all the files and the history of those files
 - Consists of all your commits
 - Place where all your hard work is stored



Key Concepts: Repositories



- Can live on a local machine or on a remote server (GitHub!)
- The act of copying a repository from a remote server is called **cloning**
- Cloning from a remote server allows teams to work together



Key Concepts: Repositories



- The process of downloading commits that don't exist on your machine from a remote repository is called **pulling** changes
- The process of adding your local changes to the remote repository is called **pushing** changes



Key Concepts: **Branches**



- All commits in git live on some branch
- But there can be many, many branches
- The main branch in a project is called the master branch



Key Concepts: Branching off of the **master** branch



- The start of a branch points to a specific commit
- When you want to make any changes to your project you make a new branch based on a commit



Key Concepts: How do you make a commit anyway?

- There are a lot of 'states' and 'places' a file can be
- Local on your computer: the '[working directory](#)'
- When a file is ready to be put in a commit you add it onto the '[index](#)' or '[staging](#)'
- Staging is the new preferred term – but you can see both 'index' and 'staging' being used



Key Concepts: How do you make a commit anyway?



- The process:
 - Make some changes to a file
 - Use the 'git add' command to put the file onto the **staging environment**
 - Use the 'git commit' command to create a new commit'



Assessment 1



1. List out the Agile Testing methods?

Ans : _____

2. List out the Phases in Agile SDLC?

Ans : _____





References



1. Roger S. Pressman, Software engineering- A practitioner's Approach, 10th Edition, McGraw-Hill, 2017.
2. Ken Schawber, Mike "Agile Software Development with Scrum" Pearson Education, 2nd Edition, 2015.

Thank You