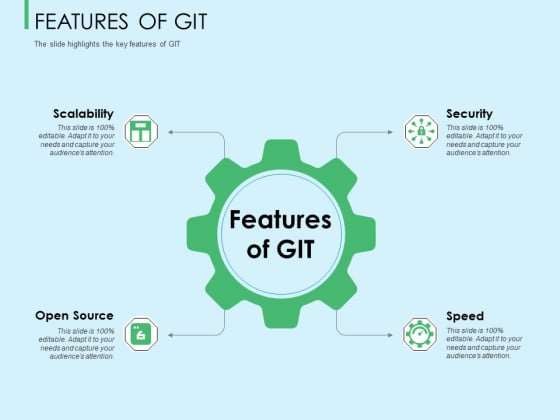
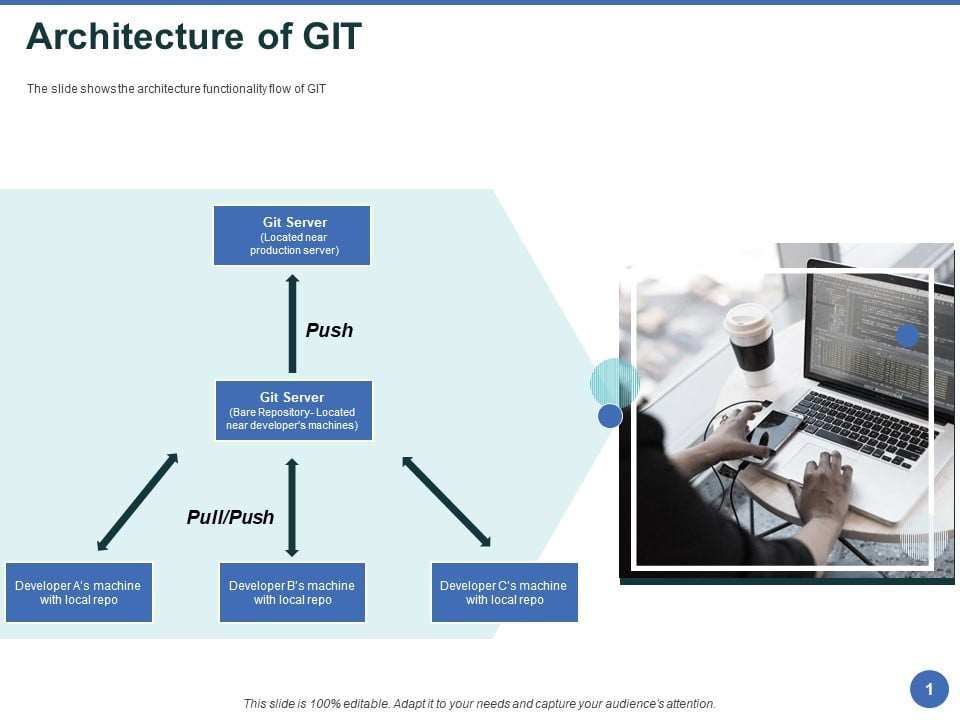


# What is Version Control System?

A Version Control System (VCS), also known as a Source Code Management (SCM) system, is a software tool or system that helps developers track and manage changes to source code, documents, and other files in a collaborative software development environment.



Characteristics of GIT

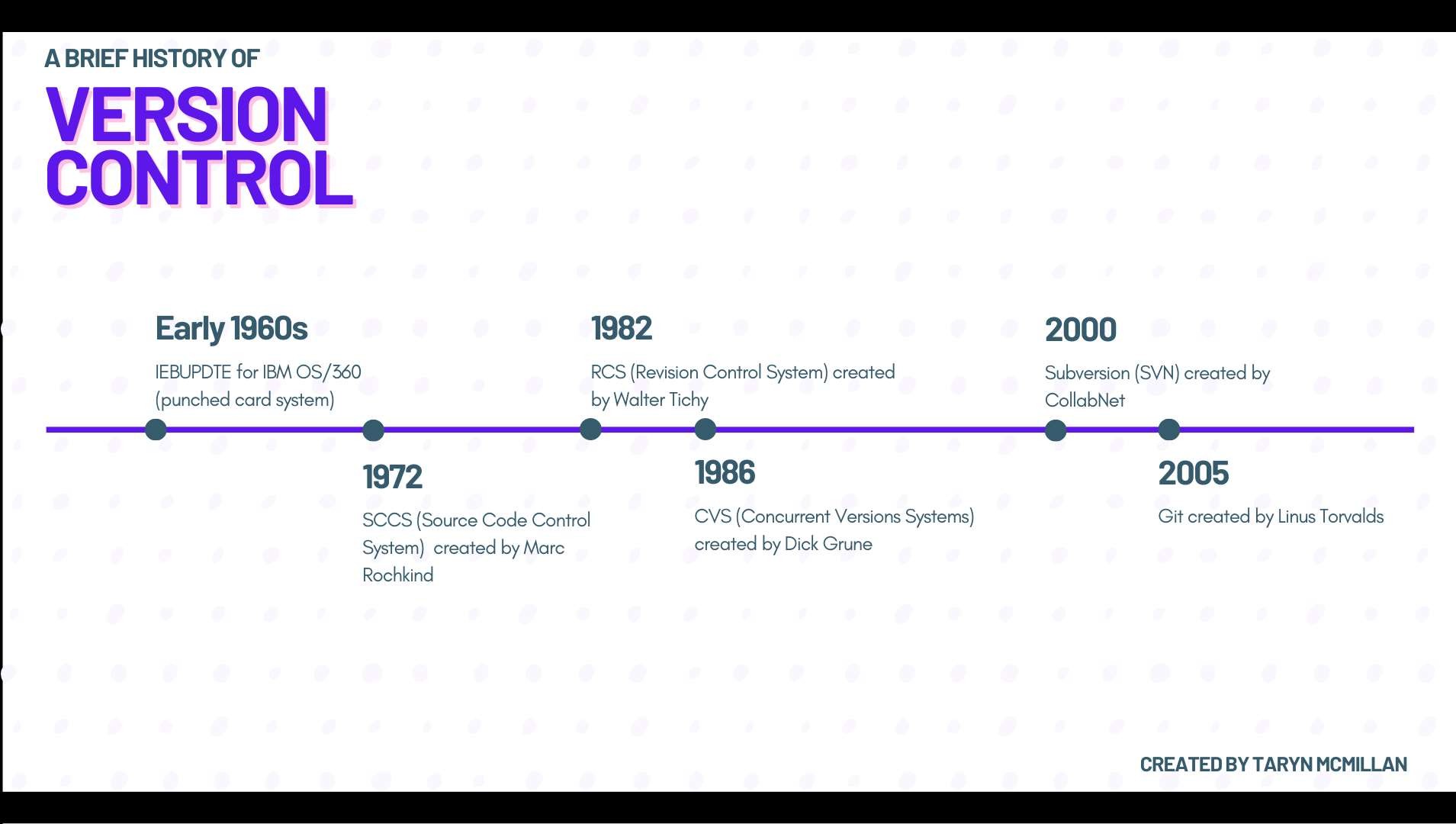
* They are free and open-source, with multiple backups and added staging areas.
* GIT provides users to work both online and offline.
* GIT has the last commit (undo operations) and revert options for unnecessary changes.
* It has a good feature called Restore delete a commit, which is essential in larger projects.
* They have good security protection against the alteration of files and maintain the authentic history of source files. In git, any revision is given a 40-bit key for all the repositories copies.
* GIT incorporates integrity (check-summed) with SHA-1 internally and enables distributed development by copying changes from one repository to another.
* They have portable control systems and are compatible with protocols like HTTP, FTP, and rsyn to publish the repositories.
* The essential characteristics include they support full package versions.
* APPLICATIONS OF GIT
* GIT in product management – they give up more frequent customer feedback and updates.
* It can be used as a graphical application.
* Government Agencies use GITHUB to share and collaborate processes (They can add their organizations and create pull requests).
* Many commercial organizations and Android applications popularly use it to build real- time web applications. They increase the team’s productivity with continuous integration and development.
* GIT can be used in operating [**systems like Fedora**](https://www.educba.com/install-fedora/), Windows, Red Hat, and other Linux systems using GIT commands.
* They are used to develop the core Linux kernel.
* They are used in open-source projects [**like Ruby on Rails**](https://www.educba.com/career-in-ruby-on-rails/), Perl, and the Linux Kernel.

## Advantages:

* Good distributed model as each developer gets a local repository with a complete history of commits which makes git fast compared to other VCs.
* Branching capabilities and merging are easy (as they are cheap), and they have good data integrity.
* They are free and open-source. We can easily download the source code and performs changes to it. They can handle larger projects efficiently.
* The push/pull operations are faster with a simple They save time, and developers can fetch and create pull requests without switching.
* Data redundancy and replications. You can write add-ons in many languages.
* They have good and faster network performance and superior disk utilization, and they think about their data like a sequence of snapshots.

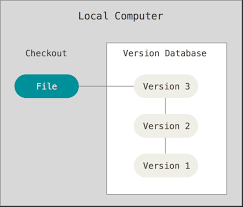
## Disadvantages:

* GIT requires technical excellence, and it is slower on Windows. They have tedious command lines to input and don’t track renames.
* They have poor GUI and usability. And also take a lot of resources, which slows down the performance.
* GIT doesn’t support checking out sub-trees. You must set up the central service for multiple package repositories for each project.
* It lacks window support and doesn’t track empty folders.
* GIT needs multiple branches to support parallel developments used by the developers.
* There is no built-in access control, and it doesn’t support binary files.
* They do not provide access control mechanisms in case of security.
* The process of Packing is entirely costly.

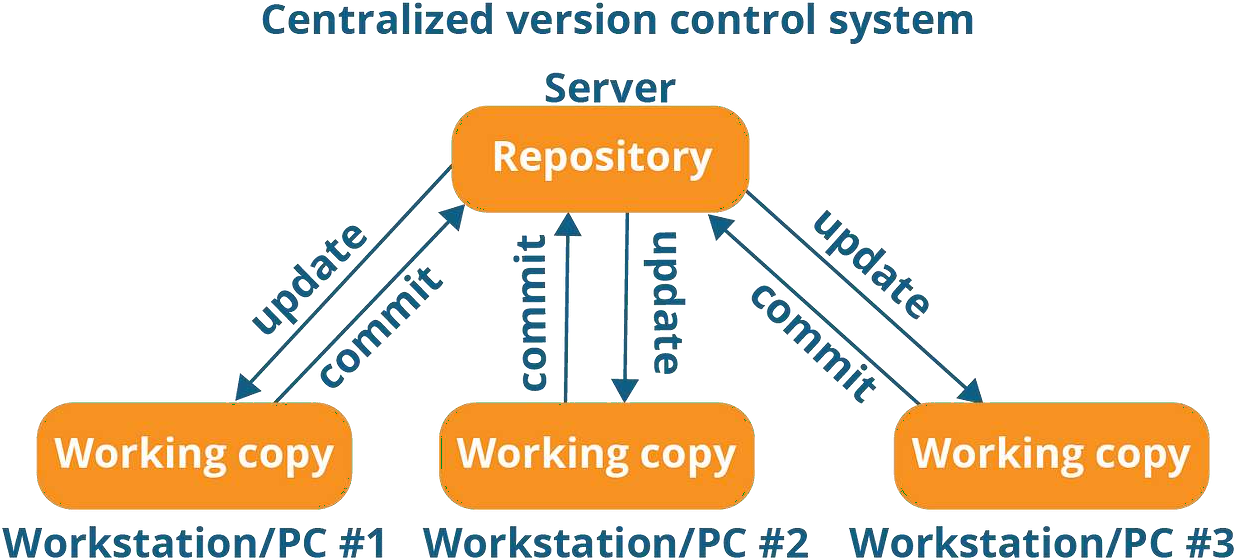


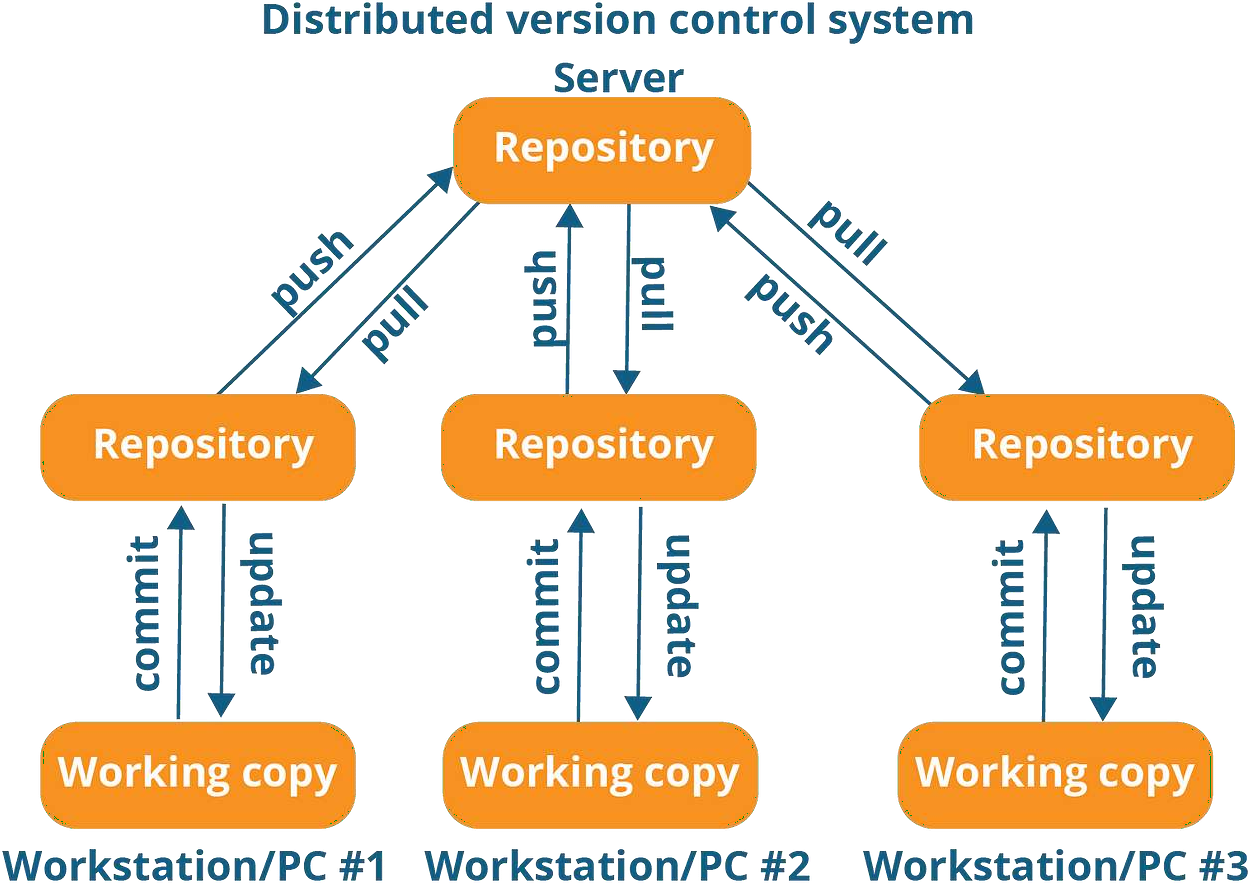


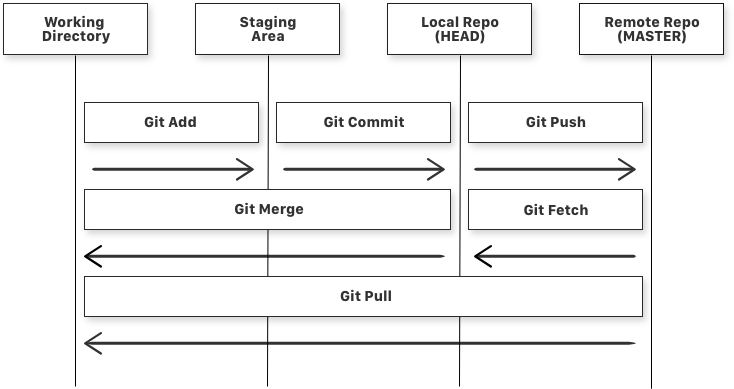
# Categories in Version Control System

* Local Version Control System
* Centralized Version Control System
* Distributed Version Control System

# Local Version Control System







git add is a command used to add a file that is in the working directory to the

staging area.

•git commit is a command used to add all files that are staged to the local repository.

•git push is a command used to add all committed files in the local repository to the

remote repository. So in the remote repository, all files and changes will be visible to

anyone with access to the remote repository.

•git fetch is a command used to get files from the remote repository to the local

repository but not into the working directory.

•git merge is a command used to get the files from the local repository into the

working directory.

•git pull is command used to get files from the remote repository directly into the

working directory. It is equivalent to a git fetch and a git merge

**Create Git Repository**

**Use the bellow command to create Git Repository:**

$ git init

How to Configure Git?

• To configure the Git Use following Commands:

**Create the User name and add Email**

$ git config –global user.name

$ git config – global user.email

• To check the configuration:

$ git config

**Logs in Git**

Git log – The git log command is used to view the commit history of a Git Repository

Command:

$ git log : This command shows the entire commit history

$ git log –oneline : Displays a compact one-line summary of each

commit

$ git –author=<author>: Filters commits by a specific author.

$ git –log –graph : Shows an ASCII representation of the commit

history.

$ git log –since and –until : Filters commits by date.

**Status in Git**

Git Status – The git status command is use to show the current state of

your working directory concerning your Git repository. It provides the

information about which files have been modified , which files are

staged for next commit, and which files are untracked.

• To check the status use below command:

$ git status

Creating directory, adding file in working directory,

• Creating directory:

**Create a folder for your Project**

• Adding file in Working directory:

create a project file in working directory

Staging in Git.

In Git, "staging" refers to the process of preparing changes to be

included in the next commit

• Staging file from working directory to Staging area:

**Use below command to add single file:**

$ git add file name

Use below command to add single file:

$ git add .

How to Commit the stage file?

In Git, "committing" refers to the process of creating a snapshot of

your project's current state, including all the changes that you've staged in the

"staging area."

• To commit the stage file Use the below command:

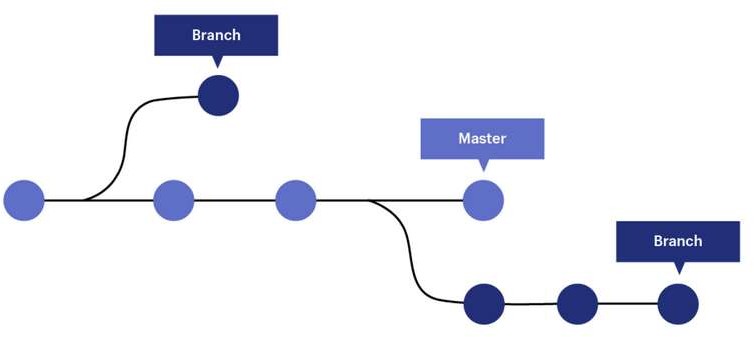
$ git commit –m “Commit message”

• To commit the file without adding to stage area:

$ git commit –a –m “Commit message

# **Branching in Git**

Branching is a fundamental concept in Git that allows you to work on multiple parallel lines of development within the same Git repository. Each branch represents an independent line of development, and you can create, switch between, and merge branches to work on different features, bug fixes, or experiments simultaneously.



Create a Branch:

$ git branch new-feature Switch to a Branch:

$ git checkout new-feature

$ git switch new-feature Create and Switch to a new Branch:

$ git checkout –b new-feature View Branches:

$ git branch

Merge Branches:

$ git checkout master

$ git merge new-feature Delete Branch:

$ git branch –d new-feature



# GitHub

GitHub is a web-based platform and service that provides a wide range of features for software developers and teams to collaborate on and manage their software projects.

# Push Operation

In Git, the "push" operation is used to upload or send the changes and commits from your local repository to a remote repository

Command:

$ git remote add origin <url>

$ git remote –v

$ git branch

$ git branch –M main

$ git push –u origin main

# Clone Operation

In Git, the "clone" operation is used to create a copy of a remote Git repository on your local machine.

Command:

$ git clone <URL>

# Pull Command

In Git, the "pull" operation is used to fetch and merge changes from a remote repository into your local repository.

Command: git pull [<options>] [<repository> [<refspec>…​]]