Coding Bootcamp

Git

Introduction to Git

Learning Objectives

By the end of this lesson, you will be able to:

- Apply key concepts of version control systems to explain how they manage changes in project files

- Evaluate the benefits of using version control systems in software development projects

- Construct a theoretical framework that outlines the key concepts of version control systems, including branching, merging, and commits

- Assess the impact of Git and GitHub on collaborative software development and project management
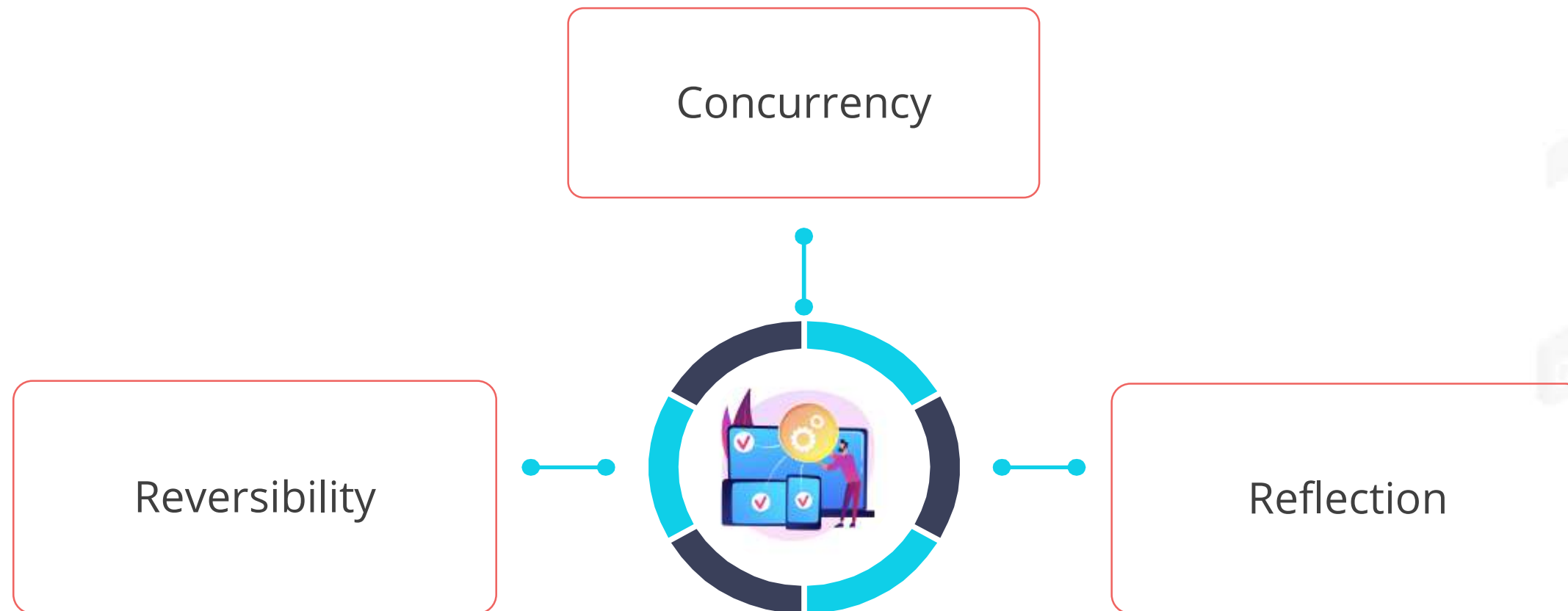
Version Control System (VCS)

It is a system that records changes to a set of files over a period to recall specific versions and can be used to store every version of an image or layout.

Example:

Version 1

Version 2

Version 3

index.htm

file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

index.htm

file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

index.htm

file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

Created

Modified

Modified

Version Control System (VCS)

A VCS offers several key functionalities including:

Concurrency

Reversibility

Reflection

By using a VCS, developers can efficiently manage changes, collaborate with others, and maintain a reliable history of their project's evolution.

Version Control System: Reversibility

Reversibility in VCS refers to the ability to undo changes and revert to previous states of the project. It is achieved through a set of built-in features and commands that allow developers to undo changes.

The key techniques of reversibility in VCS are:

Commit: It refers to a snapshot of the project which helps to revert to any previous state.

Branches: It refers to the separate line of development that can help revert to a previous branch state.

Tags: These are references to specific commits which can be used to revert to a specific point of commit.

Version Control System: Concurrency

Concurrency in VCS refers to the ability of multiple developers to work on the same project simultaneously without interfering with each other's changes.

The key techniques of concurrency in VCS are:

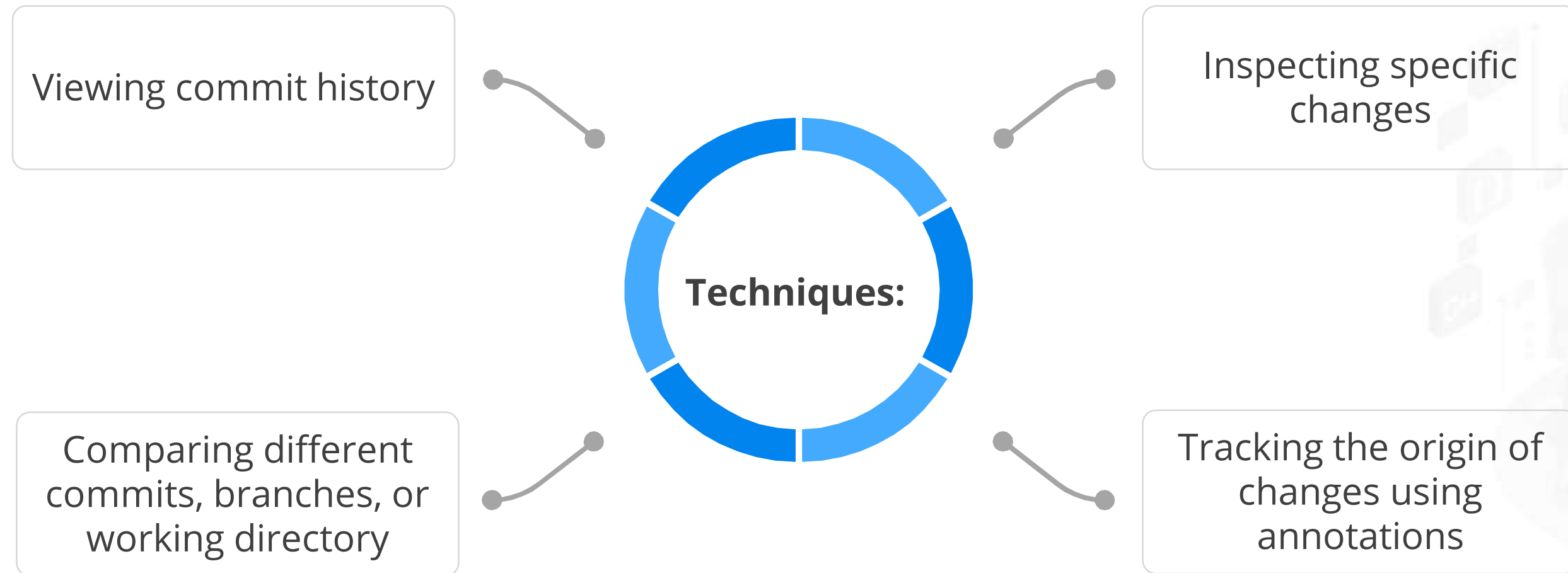**Branching:** It enables parallel work without affecting the main codebase.

Merging: It facilitates integration of parallel developments.

Conflict resolving: It refers to the manual intervention to reconcile differences while merging code branches.

Version Control System: Reflection

Reflection in VCS is the ability to review and analyze the repository's change history to understand the project's evolution, including when, why, and by whom changes were made.

Viewing commit history

Inspecting specific changes

Techniques:

Comparing different commits, branches, or working directory

Tracking the origin of changes using annotations

Advantages of VCS

Some benefits of version control system are:

Restoring previous versions
Revert to earlier versions of files or code

Storing versions
Track the changes and maintain a history of revisions

Collaborating
Facilitate teamwork by allowing multiple users to work on the same files

Code evolution
Gain insights into the evolution of code or files through version history

Backing up
Ensure data safety by having backups of all versions stored in the system

simplilearn

Key Concepts of a VCS

Repository: Maintain a central location that stores and displays all project versions and history

Branching: Enable parallel development for multiple versions of a project to coexist

Tagging: Create named markers for specific points in the project history

Merging: Combine changes from one branch into another

Commit: Save changes into the version history with a description

Cloning: Create a local copy of a remote repository

Key Concepts of a VCS

Forking: Create an independent copy of a repository

Pull request: Propose changes to merge from one branch or fork into another

Annotate: Identify who made specific changes in the code

Rollback or Revert: Return to a previous state of the codebase

Stash: Save changes temporarily that are not ready for committing

Changelog: Document all the changes made to a project

VCS Tools

Version control system tools are essential for managing and tracking changes in software development projects. Some popular tools widely used in various development scenarios are:

Types of Version Control Systems

Types of Version Control Systems

Version control systems
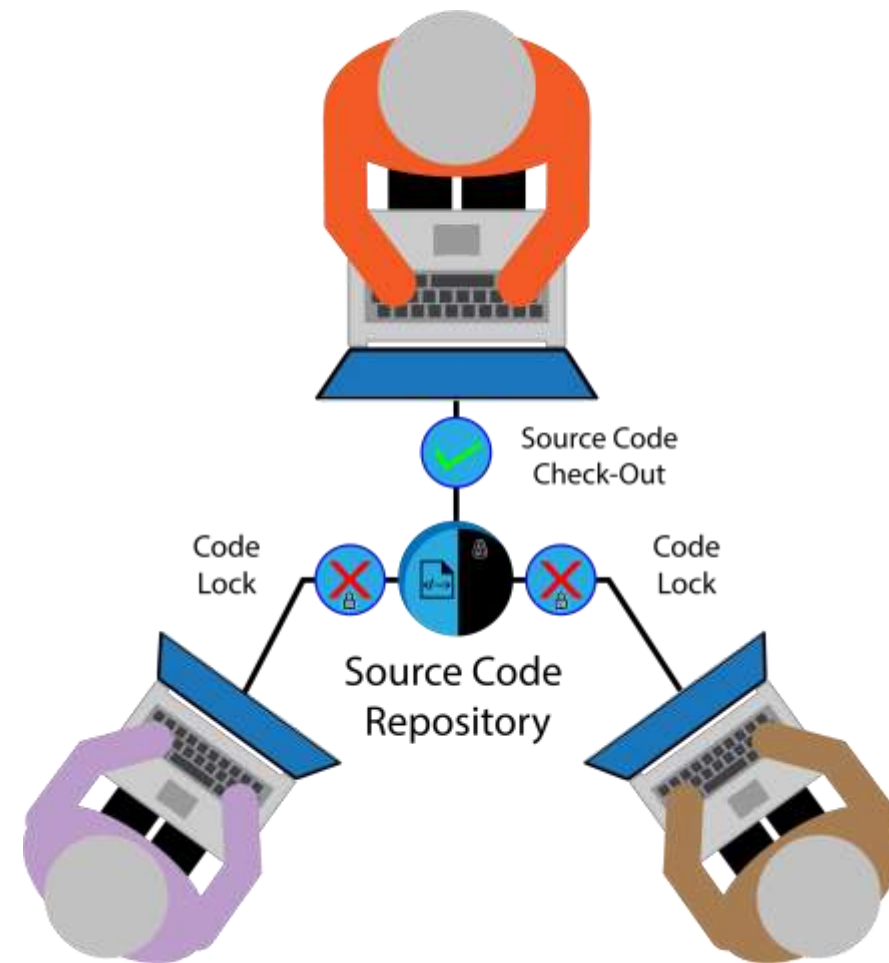
01

Centralized version control system (CVCS)

02

Distributed or Decentralized version control system (DVCS)

Centralized Version Control System

A **CVCS** is a type of version control system where a central server stores the entire history of a project, including all the files and changes in it.


Developers working on the project access this central repository to check out files, make changes, and commit their revisions.
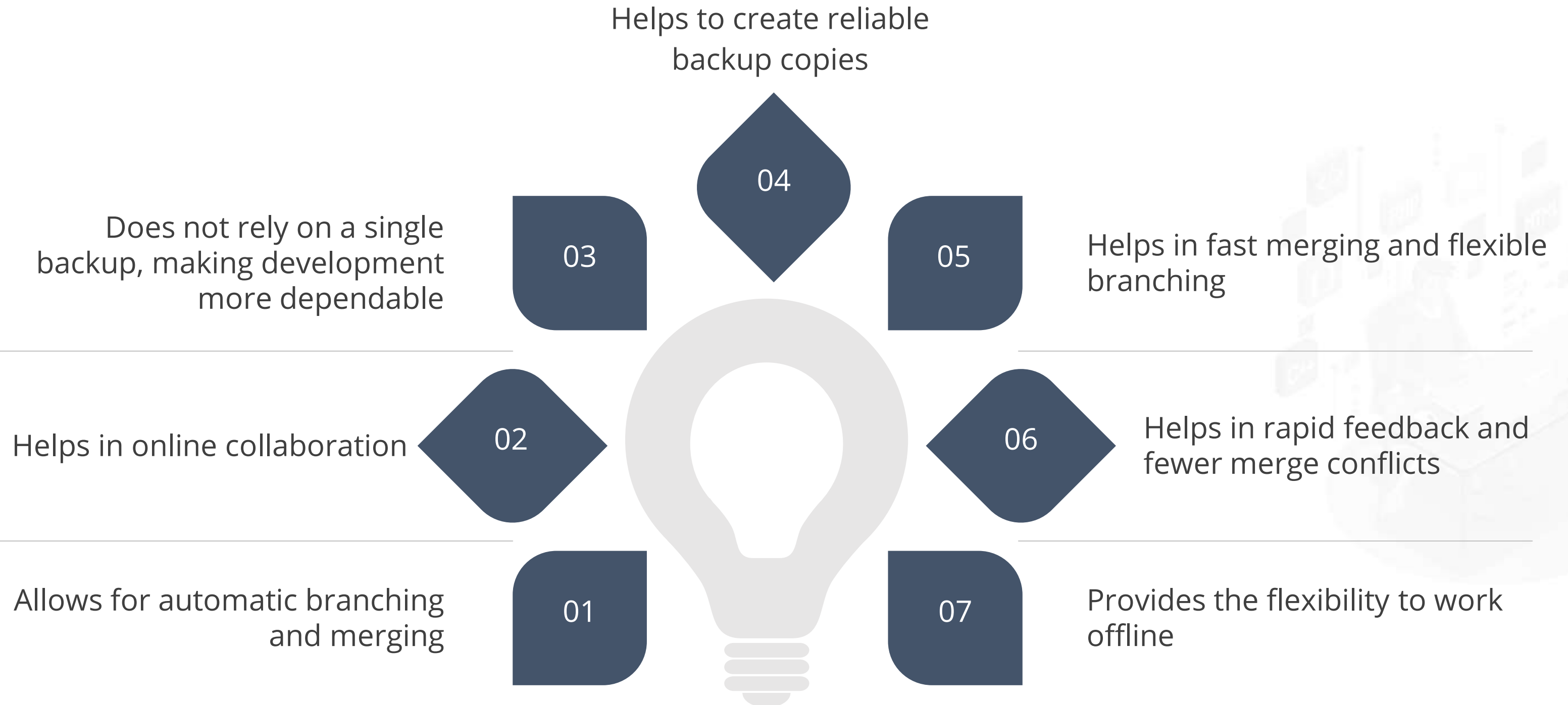
Centralized Version Control System: Features

Stores all project versions and histories in one central location

04

03

Provides selected access control to the codebase

05

Simplifies the process of back up and restoring

Ensures that all developers are working with the same version of the project

02

06

Maintains a detailed log of all changes made to the project

Ensures that either all changes in a commit are applied or none are applied

01

07

Supports creating branches and tags within the central repository

Distributed Version Control System

A **DVCS** is a type of version control system where every developer has a complete copy of the project's history and can work independently on their local machine.
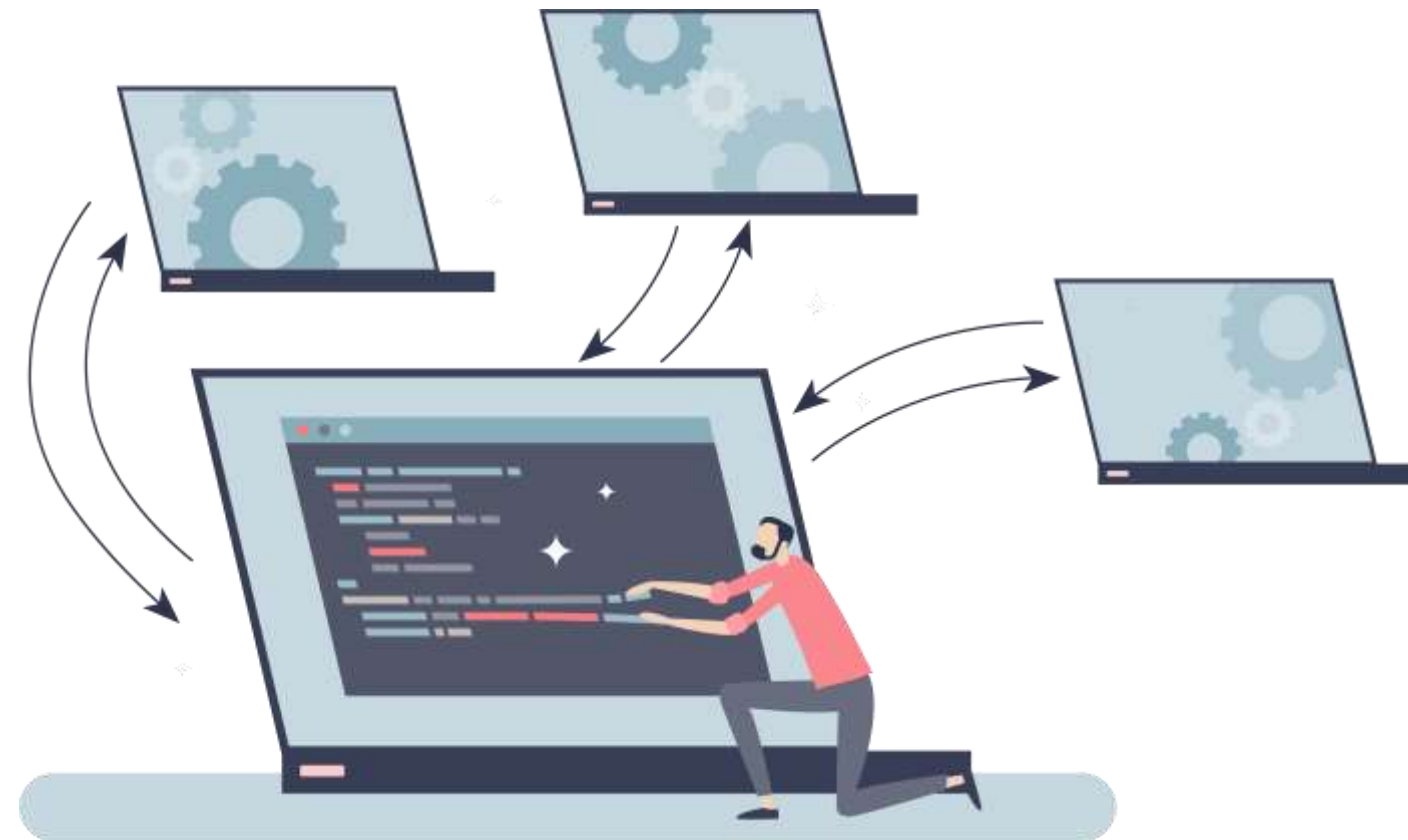
The proposed changes or updates are pushed to the central repository, where they are reviewed and merged with the main codebase.

# Distributed Version Control System: Features

Helps to create reliable backup copies

04

Does not rely on a single backup, making development more dependable

03

05

Helps in fast merging and flexible branching

Helps in online collaboration

02

06

Helps in rapid feedback and fewer merge conflicts

Allows for automatic branching and merging

01

07

Provides the flexibility to work offline

simplilearn

What Is Git and GitHub?

What Is Git?

Git is a version control system for tracking changes in computer files. It allows multiple developers to collaborate on a project efficiently.

Developers

Git Server
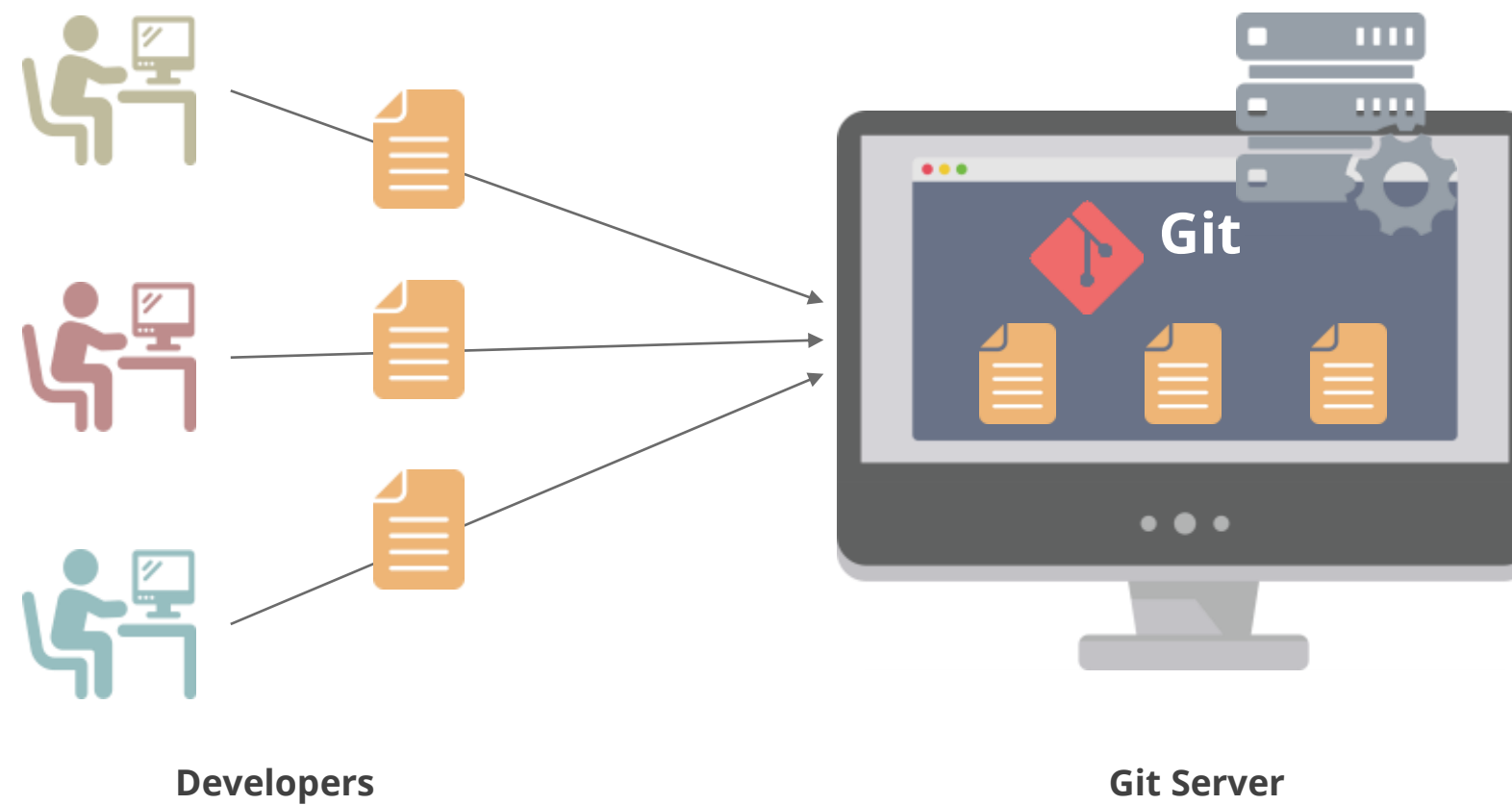
It is a well-known example of a DVCS in software development.

What Is GitHub?

GitHub provides a web-based Git repository hosting service that provides a web interface to upload files.

Source: https://github.com/
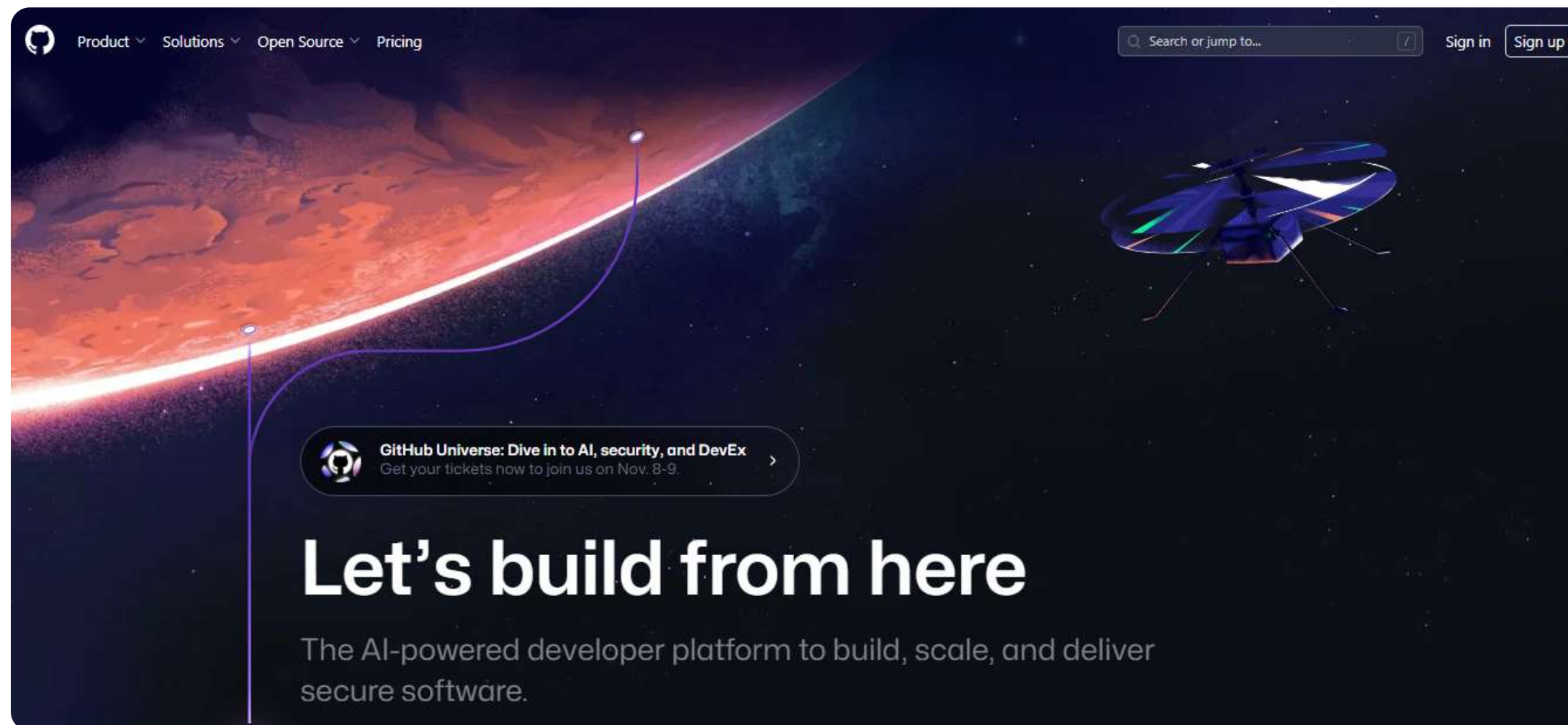
Git vs. GitHub

Git

- It is a version control system installed on the local system.

- It has a command line interface (CLI).

- It tracks changes in files locally.

- It offers no built-in user management.

- It is free and open-source.

GitHub

- It is a web-based online Git hosting service.

- It has both GUI and CLI.

- It tracks changes by hosting Git repositories.

- It offers built-in user management and access control.

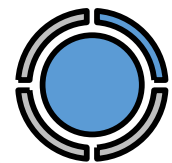- It has a free tier with paid plans for additional features.
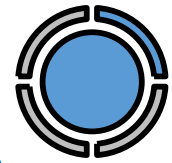
Advantages of Git

Helps the developers track the changes and updates in the project
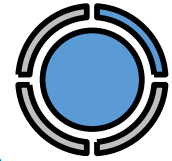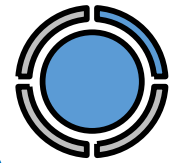
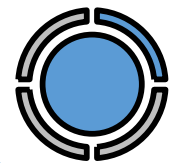Allows teams to work collaboratively
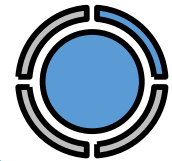
Is free and open source

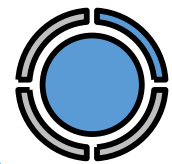Reduces the runtime costs that come with other high-level languages

Advantages of Git

Provides implicit backup for data loss

Uses a standard cryptographic hash function to identify objects

Helps in easing branch management operation

Key Takeaways

○ A version control system (VCS) is a system that records changes to a set of files over a period to recall specific versions and can be used to store every version of an image or layout.

○ A VCS offers several key functionalities including concurrency, reversibility, and reflection.

○ Reversibility in VCS refers to the ability to undo changes and revert to previous states of the project. It is achieved through a set of built-in features and commands.

○ Concurrency in VCS refers to the ability of multiple developers to work on the same project simultaneously without interfering with each other's changes.

simplilearn

Key Takeaways

- Reflection in VCS is the ability to review and analyze the repository's change history to understand the project's evolution, including when, why, and by whom changes were made.

- A centralized version control system (CVCS) is a type of version control system where a central server stores the entire history of a project.

- A distributed version control system (DVCS) is a type of version control system that allows multiple developers to work on the same codebase simultaneously.

- Git is a version control system for tracking changes in computer files. It allows multiple developers to collaborate on a project efficiently.

simplilearn

Thank You