

# SHREE L. R. TIWARI COLLEGE OF ENGINEERING

(Approved by AICTE & DTE, Maharashtra State & Affiliated to University of Mumbai)  
NAAC Accredited, NBA Accredited Program, ISO 9001:2015 Certified | DTE Code No. : 3423  
Minority Status (Hindi Linguistic)

## DEPARTMENT OF COMPUTER ENGINEERING

### EXPERIMENT NO. 10

**TITLE:** Version control of the Project

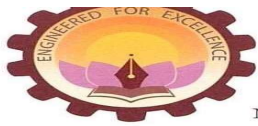
**AIM:** To Perform Version control of the Project

**OBJECTIVE:** To Study Version control of the Project.

#### **THEORY:**

Version control, which is considered to be a very important component of software development life cycle, which can also be called as revision control or source code control, is the management of changes to documents, computer programs, large websites, and other collections of information. Changes can be identified in some ways, such as "revision number", "revision level" and "revision" etc. People prefer using such VCS, which can manage teamwork with no difficulties, as that is mandatory when coming to big projects. They can either run as standalone applications or can also run by embedding them in various types of software such as word processors and spreadsheets (like google docs) and in various types and variants of CMS (like Wikipedia's Page history). Using VCS, it is also possible to perform various operations like edits tracking, mistake correction, and vandalism defending and spamming protection. In software engineering, revision control's primary aim is to provide control over changes made to source code. Its functionalities can also be enhanced, and it can be made use of to make electronic documentation too. As complex teams are involved in the modern software development, it is very frequent for many versions of the same software to be deployed in different locations or systems and for them to be working synchronously on its development. Most of the times only a specific part of the program does contain bugs or features (by the fixing of some problems and the introduction of others as the program develops). Therefore, to locate and to fix bugs, it is of absolute important to be able to retrieve and run various versions of the software to determine in which version(s) the problem occurs. It may also be necessary to develop different parts of the software in parallel (for instance, where one version which has all its bugs fixed, but no new features (branch), while the other version is where new features, which is still under testing are worked on (trunk). At the simplest level, going by the traditional way, we can retain multiple copies of the different variants of the program, and label them appropriately. It is manual version control. Formerly it was used on large software projects. While this method can work, it needs much work, and it is incredibly inefficient. It needs the humongous amount of discipline and can also create new mistakes on the part of the developer. As during all these works codebase tends to remain the same, it also involves creating an admin user who is responsible for granting privileges to who can configure which variant etc. So that the code base is not compromised no matter what, and in this way, the immense amount of complexity gets added. To overcome all these, we have got many VCS through which the version control management steps are secluded from the vision of users of the software.

Version control in project management is a tool to track changes made to project management software. It does this by saving every version of the project plan, allowing users to revert to an older or original version of the project if necessary. Also, called source control, it permits not only the tracking and saving of software code but also allows users to manage those versions. This gives project managers and their teams the ability to work faster and better. Every modification to the code or project is saved in the software database. That means if there's a mistake, the user can go back and turn back the clock to view and, if needed, revert to that version before the mistake was made.



## DEPARTMENT OF COMPUTER ENGINEERING

### Importance of Version Control in Project Management:

- 1) Version control is essential for managing projects, providing a centralized repository for files, and allowing team members to work on different aspects simultaneously. It helps ensure everyone has access to the latest version, reducing errors from outdated changes.
- 2) It enables collaboration by allowing developers to work on separate branches, experiment, and merge changes without disrupting others. Project managers can track changes, identify who made them, and monitor progress, making troubleshooting easier.
- 3) For developers, version control offers flexibility to try new ideas safely, and features like code reviews and pull requests promote collaboration and code quality. Overall, it's crucial for efficient workflows and project success.

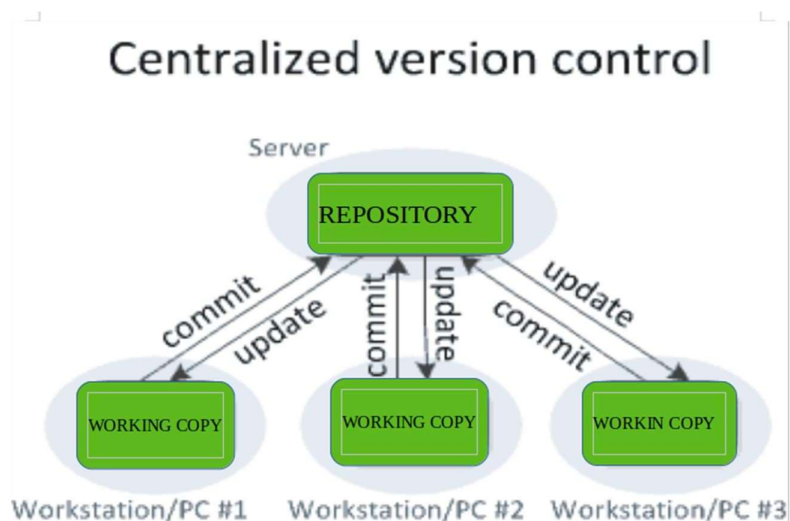
### Different Types of Version Control:

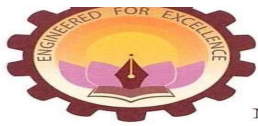
#### 1) Local Version Control Systems:

These operate on a single machine, tracking file changes within a specific project directory. They allow for experimentation and easy reversion to previous file states, but lack collaboration capabilities. Operates on a single machine, tracking changes to files in a project. It allows for easy experimentation and rollback but lacks collaboration features across multiple devices or teams. They are centralized systems that rely on a single database stored on a local machine. They are simple to use and can handle small projects effectively, but they need the scalability and collaboration features required for larger software projects. One of the most famous examples of an LVCS is RCS (Revision Control System).

#### 2) Centralized Version Control Systems:

These use a central server for storing files and tracking changes. They facilitate team collaboration, as everyone works on files from the server. However, if the server fails, the entire team's work is affected. Uses a central server to store files and track changes. Team members can collaborate by checking out and updating files from the server, but work is disrupted if the server goes down.





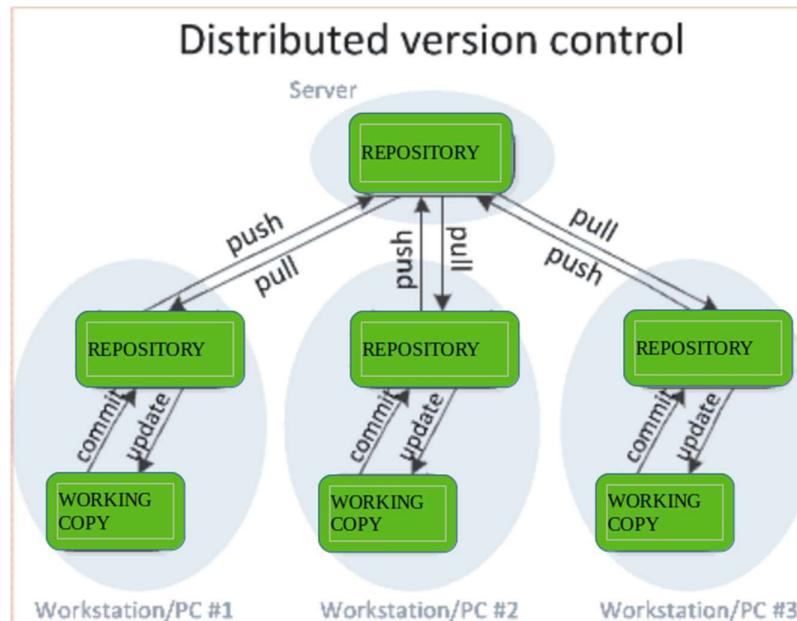
# SHREE L. R. TIWARI COLLEGE OF ENGINEERING

(Approved by AICTE & DTE, Maharashtra State & Affiliated to University of Mumbai)  
NAAC Accredited, NBA Accredited Program, ISO 9001:2015 Certified | DTE Code No. : 3423  
Minority Status (Hindi Linguistic)

## DEPARTMENT OF COMPUTER ENGINEERING

### 3) Distributed Version Control Systems:

These combine the best of both worlds. A central server stores files, but users also create local copies on their machines. This enables offline work, flexibility, and easy synchronization with the central server, enhancing collaboration and resilience. Combines local and centralized benefits. Each user has a local copy of the repository, enabling offline work and independent commits. Changes are synced to a central server for collaboration.



### How Version Control Works:

#### 1. Repository Setup

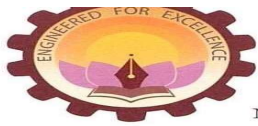
- A repository (repo) is created, either locally or on a central server, to store the project's files and version history.
- Developers can **clone** (copy) the repository to their local machine.

#### 2. Making Changes

- Developers work on their local copy of the project, making changes to the code or files.
- Staging: Before saving changes, files are added to a staging area using commands like `git add`. This prepares the changes for committing.

#### 3. Committing Changes

- A commit records changes to the repository. Each commit has a unique identifier (hash) and includes a message describing what was changed.
- Commit history tracks when changes were made and by whom, creating a timeline of revisions.



## DEPARTMENT OF COMPUTER ENGINEERING

### 4. Branches

- Developers often create branches to work on new features or bug fixes. A branch is a separate line of development that doesn't affect the main project until merged.
- This allows for independent work and experimentation without disrupting the main codebase.

### 5. Merging and Collaboration

- After completing work on a branch, changes are merged back into the main branch (often called master or main).
- If multiple developers made changes, there may be conflicts (when two people modify the same part of a file). These conflicts must be resolved manually before merging.

### 6. Pushing and Pulling

- Pulling: Developers can fetch and incorporate changes made by others from a remote repository using git pull.
- Pushing: After committing locally, developers push their changes to the central repository, sharing them with others.

### 7. Version Control Benefits

- Collaboration: Multiple users can work on the same project without overwriting each other's work.
- History: The project's complete history is tracked, making it easy to revert to previous versions or understand why changes were made.
- Branching/Experimenting: Developers can safely experiment with new ideas and merge their changes back when ready.

### Benefits of Version Control:

#### 1. Collaboration

VCS allows teams to work together on a project, with the ability to track changes made by each team member. This makes it easier to manage large projects and reduces the risk of conflicting changes.

#### 2. Error Reduction

With VCS, it's easier to catch and correct errors in code. If an error is discovered, VCS allows developers to roll back to a previous version of the code, making it easier to identify where the problem originated.

#### 3. Code Quality

VCS promotes good coding practices, such as using descriptive commit messages and documenting changes. This helps maintain the overall quality of the codebase.

#### 4. Productivity

VCS streamlines the process of making and tracking changes to the code, saving time and improving productivity.

#### 5. Backup and Disaster Recovery

VCS acts as a backup of the codebase, allowing for easy recovery in case of any risk or failure, such as a hard drive failure. VCS also keeps track of changes made to the code over time, making it easier to identify when a problem was introduced and to revert to a previous version if needed.

#### 6. Historical tracking

VCS provides a historical record of changes made to the codebase over time. This allows developers to understand the evolution of the codebase, track the progress of features, and identify who made changes and when.

**CONCLUSION:** Hence, we studied the version control of project.