UNIVERSIDAD TECNOLÓGICA
DE TAMAULIPAS NORTE

PROYECTO:

“Activity 1: Introduction to Software Versioning”

REPORTE DE ACTIVIDAD PARA OBTENER EL TÍTULO DE:

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DESARROLLO Y GESTIÓN DE SOFTWARE.

PRESENTA:

MAGDIEL ENRIQUE MEJÍA MARMOLEJO

CD. REYNOSA, TAMAULIPAS SEPTIEMBRE 2024.

Introduction

In modern software development, managing changes and maintaining the integrity of a project is critical to ensuring smooth collaboration, version consistency, and high-quality releases. This is where software versioning plays a vital role. Software versioning allows developers to track the evolution of a project by assigning version numbers to different states of the software, making it easy to identify changes and address issues efficiently.

Version control systems (VCS) like Git and SVN (Subversion) are essential tools for managing these changes in a collaborative environment. They offer features like branching, merging, and version history management, enabling teams to work simultaneously on various aspects of a project without conflicts. This document explores the concept of software versioning and provides a comparison between Git and SVN, highlighting their distinct features and benefits.

Software Versioning

Software versioning is the process of assigning unique version numbers to various states of software development. It helps developers manage and track changes in software projects over time, enabling them to organize different iterations, such as bug fixes, new features, or major updates. Version numbers often consist of a sequence of numbers, such as 1.0.0, where the first digit represents major changes, the second for minor improvements, and the third for patches or bug fixes.

Characteristics

Track Progress: Clearly understand the history of changes and improvements made to the software.

Identify Issues: Pinpoint the exact version of the software where a bug or error occurred.

Collaboration: Ensure that multiple developers can work on the same project without conflicting changes, by keeping track of who made which changes and when.

How Version Control Helps: Version control systems (VCS) help manage software changes by allowing multiple team members to collaborate efficiently. VCS tools such as Git or SVN let developers:

Record Changes: Keep a history of changes, so any earlier version of the software can be restored if needed.

Collaborate: Work on the same project simultaneously without overwriting each other's work.

Branch and Merge: Enable parallel development efforts through branching and merging features, which allow independent versions of the codebase to exist and be merged when necessary.

Git and SVN

Feature	Git	SVN (Subversion)
Storage Model	Distributed: Each user has a local copy of the entire repository.	Centralized: One central repository accessed by all users.
Branching Handling	Lightweight and fast; branches are cheap and easy to create.	Heavyweight and slower; branches often cause performance issues.
Ease of Use in Collaboration	Excellent for team collaboration due to distributed nature.	Effective but dependent on central server; requires more synchronization.

	Developers can work independently.	
Merging Capabilities	Git offers robust and automatic merging features. Conflicts can be resolved easily.	SVN has less advanced merging capabilities, requiring manual conflict resolution more often.
Version History	Complete version history stored locally; developers have full access even offline.	Version history stored on the central server; access requires network connectivity.
Platform Compatibility	Compatible with multiple platforms (Windows, macOS, Linux).	Also compatible with multiple platforms, but requires server setup.
Security	Git uses cryptographic methods to ensure integrity and prevent tampering of history.	SVN offers basic access control through authentication and authorization but less emphasis on history integrity.
Popularity	Extremely popular and widely adopted in open-source and corporate environments.	Still used in some corporate environments, but declining in popularity in favor of Git.

Conclusion

Software versioning is an indispensable part of modern software development, providing a structured approach to managing project changes, fostering collaboration, and ensuring that development efforts are well-organized and traceable. Git, with its distributed architecture, flexible branching, and robust merging capabilities, has become the dominant choice for many development teams. On the other hand, SVN remains a reliable option in certain environments, offering a centralized approach that some teams prefer.