**[1. What is a version control system?](https://www.vogella.com/tutorials/Git/article.html" \l "versioncontrolssystems)**

A version control system (VCS) allows you to track the history of a collection of files. It supports creating different versions of this collection. Each version captures a snapshot of the files at a certain point in time and the VCS allows you to switch between these versions. These versions are stored in a specific place, typically called a repository.

A **localized version control system** keeps local copies of the files.

A **centralized version control system** provides a server software component which stores and manages the different versions of the files. A developer can copy (checkout) a certain version from the central sever onto their individual computer.

In a **distributed version control system** each user has a complete local copy of a repository on his individual computer. The user can copy an existing repository. This copying process is typically called **cloning** and the resulting repository can be referred to as a clone. Every clone contains the **full history of the collection of files** and a cloned repository has the same functionality as the original repository.

**Git**

Git is currently the most popular implementation of a distributed version control system.

### [Working tree](https://www.vogella.com/tutorials/Git/article.html#workingtree)

A local repository provides at least one collection of files which originate from a certain version of the repository. This collection of files is called the *working tree*. It corresponds to a checkout of one version of the repository with potential changes done by the user.

A file in the working tree of a Git repository can have different states. These states are the following:

**untracked**: the file is not tracked by the Git repository. This means that the file never staged nor committed.

**tracked**: committed and not staged

**staged:** staged to be included in the next commit

**dirty / modified**: the file has changed but the change is not staged

Git user prefer the usage of the index instead of staging area.

**Committing** creates a new persistent snapshot (called commitor commit object) of the staging area in the Git repository. A commit object, like all objects in Git, is immutable.

The staging area keeps track of the snapshots of the files until the staged changes are committed.

git commit command to commit the staged changes.

Users with sufficient authorization can send new version in their local repository to to remote repositories via the push operation.

Use of A branch allows the user to switch between these versions so that he can work on different changes independently from each other.