There are three type of version control systems such as local VCS,centralized VCS,distributed VCS

Firstly we’ll look at local control system.

Historically,local version control systems are oldest approachment to control versions of software projects.The projects and it’s changes stored in database on the local machine.This is basic logic of local version control systems.There is no server.Only one developer can contribute the Project efficiently.So that it is not suitable more than one developers or users.The most popular local version control system tool is Revision Control System.

The next one is centralized VCS

İt’s the versioning aproachment that acting like client-server architecture.This systems provide oppurtunity that more than one developer can contribute on a Project efficiently.In this systems,There is a server which store source of projects and contributors performs checkout and commit through this server.But here we have to be connected to internet to make some changes so it is big disadvantages for us.

The last one is distributed version control systems

In this systems(such as Git, Mercurial, Bazaar or Darcs), clients don’t just check out the latest snapshot of the files: they fully mirror the repository. In this way if any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.

* **Git** is a version system control that is widely used for software development and other version control tasks. It is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed in 2005 by Linux kernel developers (including Linus Torvalds) for Linux kernel development.
* As with most other distributed version control systems, and unlike most client-server systems, every Git working directory is a full repository with complete history and full version-tracking capabilities, independent of network access or a central server.
* Strong support for non-linear development
* Nearly every operation in local
* You need internet if and only if you want to upload your workspace to server.
* Distributed development
* Efficient handling of large projects