

Version Control is the management of changes to documents, computer programs, large websites and other collections of information. These changes are usually termed as versions. Changes can be of any kind like adding some new file to the project or modifying some existing file in the project. Whenever any of the developers collaborating on the project, changes it in any way the Version Control System creates a snapshot of the entire project and saves it. Snapshot is the entire state of the project at a particular time. These snapshots are basically known as versions.

Let's say a developer named 'A' is developing a website and initially has added only one web page named 'index.html'. The Version Control System detects the addition of a new file and hence this will be Version 1. After a few days, he wishes to add another webpage named 'about.html' and this will be Version 2. After a few days if A adds an image to 'about.html' then this change will also be detected by the Version Control System and this will be Version 3.

At any point of time if the developer wishes to revert back to a previous version, because he does not like the changes he has made to the project currently, then the Version Control System makes this possible for the developer. The developer can move back to any of the previous versions of his project.

## **Need for Version Control Systems**

### **1. Collaboration**

Let's say there are 3 developers – P1, P2 and P3 working on a website. P1 creates the initial webpage 'abc.html' and asks P2 and P3 to create different web pages and follow the same layout as of this page. They decide to merge their work after 2 days. In the span of these 2 days developer P1 feels that he should modify the layout of the webpage 'abc.html' or that he should add an image to this webpage and he changes 'abc.html'. Now, after 2 days when they try to merge their work there will be a lot of conflicts and the work might end up in chaos, due to the other 2 developers P2 and P3 not being made aware of the new changes made to the webpage 'abc.html' on time.

But, with Version Control Systems we have a shared workspace and it continuously tells/notifies all the developers as to which developer has made what change and when has that change been made. So, the project evolves as a whole from the very start and hence saves a lot of time by avoiding any kind of chaos and conflicts. Thus, with the help of Version Control Systems we address a very common problem faced by developers around the world. Now, any developer located in any part of the world can work with other developers located in any part of the world, hence making collaboration possible.

## **2. Storing Versions & Backup**

Saving a version once changes have been made by a developer is very important. But there are certain questions that arise in our minds when we think of saving a version. The very first question is should we save the entire project or just the changes that we have made? In case we save only the changes then in that case it will be very difficult for us to view the entire project at the same time and in case we save the entire project at every time there will be a huge amount of unnecessary and redundant data lying around because you will be saving the same data that has remained unchanged again and again and hence this will cover up a lot of space. The last question is that how do we know what exactly is different between the versions? What change was made to Version 1 before it being named as Version 2? It is not possible for the developer to remember the changes that he has made and hence we need to document these changes.

The Version Control System is the answer to all the above questions. It acknowledges that there is only one project so when the developer is working on the project, there is only one version on the disk and all the changes made by the developer in the past are neatly packed inside the Version Control System. In case the developer needs to revert back to a previous version he can easily do so. The versions are named Version 1, Version 2 and so on.

In a Distributed Version Control System there is a central server where all the project files are located and apart from that every developer collaborating on the project has a

local copy of all the project files on his local machine and these are known as the local copies. So, what developers exactly do is, every time they start coding at the start of the day, they fetch all the project files from the central server and store it in the local machine and after they are done working, they transfer all the files back into the central server. At every point of time you will always have a local copy in your local machine and also files on the central server. So, at times of crisis let's say the central server crashes and the developer loses all the files then in that case, the developer does not have to worry because he has all the files backed up with him as local copies on his local machine.

In case the developer has not updated his local copy and has also lost all the files on the central server, then also he does not have to worry because at least one of the collaborators working on the project would have updated his local copy and hence can provide the developer with the required project files.

### **3. Analysis**

When a developer changes any of the project files in any way, then the Version Control System provides him with a complete description of the changes. It tells what was changed and when it was changed and hence this helps the developer to analyze how the project has evolved between different versions. The developer can easily conclude what he had done to improve the project and can compare previous versions with the new one. The Version Control System maintains a complete timeline of the changes incorporated and hence makes it easier for other developers to analyze the changes made by the author.

### **Version Control Systems**

Git (Distributed Version Control System), Apache Subversion (Centralized Version Control System) also known as SVN, Concurrent Versioning System (Centralized Version Control System) also known as CVS and Mercurial (Distributed Version

Control System) are some of the Version Control Systems. Centralized Version Control Tools do not maintain a local copy on the local machine. There is only one copy on the central server. It means that all the developers collaborating on a project are working directly with the central repository. Hence, nowadays developers prefer Distributed Version Control Systems over Centralized Version Control Systems for obvious reasons.