Design Basics - Version control Introduction

- Aditya Kumar
  Chief Technology Officer, edwisor.com

We will cover with the following concepts

1) Introduction to Version control systems

2) Why they are needed?

Definition is a bit tricky to understand

- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

- It allows you to revert files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

- Using a VCS also generally means that if you screw things up or lose files, you can easily recover.

A real life example may help you understand better

Suppose you are working on a project. It can be anything like a simple doc file, an image or some application you have coded.

Whenever you are working on such project, you always get a stable version. A version that works, a version that you know has greater chances to lead to complete working project.

To protect this version, you need to back it up. Store it somewhere, so that if your work after that is not along the similar line of expectation, you can revert it back to this stable version. T**his is where the need of Version Control system arises.**

Lets cross off the sub-optimal options people were used to

1) Copy paste the file in some other folder, maybe a time stamped folder.

2) Save the file with a new name, make a copy and start working on the copy

3) Give the file to someone else as backup, upload it on dropbox, drive or other online storage service.

Imagine if you are not the only person working on that project!

Suppose you are working with that project with few of your friends/colleagues or a full team. Now Imagine the same problem in this context. Your problems include -

1) Maintaining a stable version of every person’s work

2) Merge the work when everyone is done

3) Revert a particular version or part of work or work of a particular user

4) Ensure easy accessibility of your work to your team members

Distributed Version Control systems come to your rescue

1) A simple database which keeps tracks of changes to your files and folders

2) You can maintain multiple version entries in that database. Each version is “commited” to keep a track of changes in that commit.

3) These version database are synced over a distributed cloud system, So there is no single point of failure.

4) Collaboration is easy. Anybody with proper authorization can contribute to the **repository**.

Next Steps -

1) GIT - introduction and basic commands