Case Study-7

1.Version Control System

* In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), **version control** (also known as **revision control**, **source control**, or **source code management**) is a class of systems responsible for managing changes to [computer programs](https://en.wikipedia.org/wiki/Computer_program), documents, large web sites, or other collections of information. Version control is a component of [software configuration management](https://en.wikipedia.org/wiki/Software_configuration_management). **Version control systems** (**VCS**) are most commonly run as stand-alone applications, but revision control is also embedded in various types of software such as [word processors](https://en.wikipedia.org/wiki/Word_processor) and  [spreadsheets](https://en.wikipedia.org/wiki/Spreadsheet), collaborative  and in various [content management systems](https://en.wikipedia.org/wiki/Content_management_system). As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates. [Bugs](https://en.wikipedia.org/wiki/Computer_bug) or features of the software are often only present in certain versions (because of the fixing of some problems and the introduction of others as the program develops). Therefore, for the purposes of locating and fixing bugs, it is vitally important to be able to retrieve and run different versions of the software to determine in which version(s) the problem occurs. It may also be necessary to develop two versions of the software concurrently: for instance, where one version has bugs fixed, but no new features ([branch](https://en.wikipedia.org/wiki/Branching_(revision_control))), while the other version is where new features are worked on ([trunk](https://en.wikipedia.org/wiki/Trunk_(software))). Revision control manages changes to a set of data over time. These changes can be structured in various ways. When checking into a different repository, this is interpreted as a merge or patch.

2. Version Control System

* **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. For the examples in this book, you will use software **source** code as the files being **version** controlled, though in reality you can do this with nearly any type of file on a computer.

Why important?

. Version control is important to keep track of changes and keep every team member working off the latest version. You should use version control software for all code, files, and assets that multiple team members will collaborate on. It needs to do more than just manage and track files. It should help you develop and ship products faster. This is especially important for teams practicing DevOps.

That’s because using the right one:

1. Improves visibility.
2. Helps teams collaborate around the world.
3. Accelerates product delivery.

3. Popular version control systems include

1. Git
2. Helix VCS
3. Microsoft Team Foundation Server
4. Subversion

Git version control system is used now.

Its core functions are :

* **git config**
* **git init**
* **git clone**
* **git add**
* **git commit**
* **git status**
* **git log**
* **git push**
* **git pull**

4. Testing

* Software testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

## Why Software Testing is Important?

**Software Testing is Important** because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction. Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of one example.

* In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.

5.Testing tools

* Here are top 8 tool that are used in java project testing:
* Arquillian
* JTest
* The Grinder
* TestNG
* JUnit
* JWalk
* Mockito
* Powermock

Among them, Junit testing is the most useful tool.