Name: Ma Phyo Zin Mar

Case Study7

Task 1

**Version control systems** are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Task2

**Version control** (also known as source control) is the management of file changes within a version control system. These systems automatically maintain character level changes for all files stored within allowing for a complete retrace of all versions of each file, the author of those versions and a complete rollback of all changes from the beginning of version control.

For developer-oriented work, it is critical to utilize version control systems for all non-binary files (read Notepad readable) to enable multiple developers or teams to work in an isolated fashion without impacting the work of others. This isolation enables features to be built, tested, integrated or even scrapped in a controllable, transparent and, maintainable manner.

**Task3**

Examples of version control system are

* Git
* Microsoft team foundation server
* SebVersion

**Git** is the best choice for most software teams today. The basic of version control are the ability to save changes made to files, whilst retaining the changes from all previous versions. All changes made within Git receive a unique version identifier alongside a user-written comment where the changes can be described.

Features of Git are

* **Free and open source**
* **Speed**
* **Scalable**
* **Reliable**
* Secure
* Economical
* **Supports non-linear development**

Task4

**Testing** plays an important role in SDLC and it is very crucial because in SDLC, testing improve reliability, performance and some other factors that may require during this process.

**Importance of testing**

1. Testing always helps to verify that complete software requirements are implemented correctly or not, means it is implemented according to the defined requirements or not.

2. Testing definitely helps in identifying defects/bugs and ensuring they are recognized/addressed before software deployment stage.

3. Testing also demonstrates that software/application appears to be working according to specifications and the performance requirements that have been defined is also met.

4. One of the main reason **why testing is important in SDLC** is, testing always helps to verify proper integration and interaction of each component in the system.

**Task5**

Tools for testing phases are

* Protractor
* Gauge
* Galen
* Night watch
* Testing
* JUnit
* Webdriver.io
* BDD frameworks like [Quantum](https://www.perfecto.io/integrations/quantum/)

**JUnit** is a Regression Testing Framework used by developers to implement unit testing in Java, and accelerate programming speed and increase the quality of code. JUnit Framework can be easily integrated with either of the following −

* Eclipse
* Ant
* Maven