System Analysis & Design

Case Study (7)

Version Control System

- Version control systems are a category of software tools that helps record changes to files by keeping a track of modifications done to the code.

Types of Version Control Systems:

- Local Version Control Systems
- Centralized Version Control Systems
- Distributed Version Control Systems

Local Version Control System

- Simplest
- Changes can be controlled by the revision control(RCS)
- RCS is one of the most common tools
- Keep patch sets in special format on disk
- After adding all patches on database, all files can be viewed every time.

Centralized Version Control System

- Include only one repository(central)
- One user can access only one copy of work
- Need to commit and update if it is changes occurred
- Benefit in many developers to collaborate the system
- Administrator can only control the system(fine-grained-control)

Distributed Version Control System

- Include multiple repositories
- Each user can access a repository and a copy of work
- Has many functions such as commit, push , pull, update and so on
- Eg . Git,Mercurial

Why Version Control System is needed in System Development Plan?

- Multiple people can work simultaneously on a single project.
- It also enables one person to use multiple computers to work on a project.
- It integrates the work that is done simultaneously by different members of the team.

- Version control provides access to the historical versions of a project.

Git

- Distributed Version Control System
- Track changes in source code during software development
- Design for coordinating work among programmers
- Track changes in any set of files
- Goals include speed, data integrity, and support for distributed, non-linear workflows

Functions of Git (commands)

- git status
- git commit
- git clone
- git add
- git init
- git branch
- git merge
- git push
- git pull

Testing

- One of the important aspects in **SDLC**
- Prove that all the software requirements are always implemented correctly or not
- Identify defects and ensuring that testing are addressed before software deployment
- Improves the quality of product and project

Importance of Testing in SDLC

- **Testing** plays an important role in **SDLC** and apart from that testing also improves the quality of the product and project by discovering bugs early in the software.
- And remember **testing** not only improves the quality of the product, but it also improves the company quality also.

Junit (Testing Tool in Java)

- It finds bugs early in the code, which makes our code more reliable.
- JUnit is useful for developers, who work in a test-driven environment.
- Unit testing forces a developer to read code more than writing.
- Develop more readable, reliable and bug-free code which builds confidence during development.

```java
import static
org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

public class MyTests {

    @Test
    public void
multiplicationOfZeroIntegersShouldReturnZero() {
        MyClass tester = new MyClass(); // MyClass is
tested

        // assert statements
        assertEquals(0, tester.multiply(10, 0), "10 x 0
must be 0");
        assertEquals(0, tester.multiply(0, 10), "0 x 10
must be 0");
        assertEquals(0, tester.multiply(0, 0), "0 x 0
must be 0");
    }
}
```

Fig . Example used of Junit in Java program